

Agilidad

Perspectiva y principios del desarrollo ágil



Agilidad

Perspectiva y principios del desarrollo ágil

V.1.0

Fecha de la versión: septiembre 2020.
Imagen de la portada: Pixabay – Arek Socha

© Juan Palacio, Marta Palacio.

Derechos registrados en Safe Creative. N° de registro: 2009275454837
<https://www.safecreative.org/work/2009275454837>

Contenido

Introducción.....	8
MÓDULO 1 PERSPECTIVA.....	9
Crisis del software.....	10
Mariner I.....	10
IBM os/360.....	11
Respuesta a la crisis del software.....	13
Ingeniería del software.....	14
Desarrollo basado en procesos.....	17
Gestión de proyectos.....	32
Cuestionando las soluciones surgidas por la crisis del software. .	38
Personas, Procesos y Tecnología.....	48
MÓDULO 2 AGILIDAD.....	51
Agilidad.....	52
Desmontando la gestión de proyectos.....	64
Diferenciando las prácticas de los principios y valores scrum.....	69
Primera parte: el ciclo scrum.....	70
El ciclo scrum.....	72
Roles.....	74
Artefactos.....	78
Eventos.....	83
Medición y estimación ágil.....	92
Segunda parte: principios y valores.....	96

Principios y valores scrum.....	97
Las personas y sus roles.....	101
Artefactos.....	102
Eventos.....	103
Prácticas para flexibilizar scrum.....	105
MÓDULO 3 AGILIDAD EN LA EMPRESA.....	113
Scrum Level.....	114
Diferencias entre la gestión de proyectos ágil o evolutiva y la tradicional o predictiva.....	115
A qué nos referimos con “scrum”.....	118
Paradigmas culturales de las organizaciones.....	122
Paradigma impulsivo, o “rojo”.....	123
Paradigma conformista o “ámbar”.....	124
Paradigma logro o “naranja”.....	125
Paradigma pluralista o “verde”.....	126
Paradigma evolutivo o <i>teal</i>	127
Dimensiones de la empresa y agilidad.....	131
Por qué quiere ser ágil la empresa y cómo lograrlo.....	133
Consideraciones.....	136
Principios y valores de la agilidad.....	138
Dimensiones y facetas de la empresa.....	140
Faceta operativa: principios y prácticas.....	141
1.- Entrega de valor.....	142
2.- Mejora continua.....	143
3.- Desarrollo iterativo e incremental.....	144

4.- Ritmo de trabajo sostenible.....	145
5.- Atención continua a la excelencia.....	146
6.- Operativa visible.....	146
7.- Cadencia y sincronización global.....	147
8.- Personas sobre procesos.....	148
Faceta organizativa: valores y principios.....	149
1.- Asertividad.....	150
2.- Valoración del talento.....	150
3.- Claridad.....	151
4.- Confianza.....	152
5.- Estructura desjerarquizada.....	152
6.- Propósito común.....	154
Soporte.....	155
Criterios para evaluar la agilidad operativa, organizativa y riesgos para la mejora.....	156
Evaluación y mejora.....	157
No se haga trampas.....	158
Criterios para evaluar la agilidad operativa de la empresa.....	159
Criterio para evaluar las prácticas.....	160
Criterio para evaluar el valor aportado por las personas sobre los procesos.....	160
Cálculo de la agilidad operativa de la empresa.....	162
Criterios para evaluar la agilidad organizativa de la empresa.....	165
Dimensión cultural.....	165
Dimensión estructural.....	166
Cálculo de la agilidad organizativa de la empresa.....	168

Criterios para evaluar el soporte que proporciona la organización	172
Formulación del riesgo.....	174
Apéndices.....	176
Apéndice 1: protocolos.....	177
Homologación de protocolos.....	177
Componentes de un protocolo.....	178
Normas y restricciones para protocolos.....	179
Apéndice 2: documentar o certificar una evaluación.....	181
Documento de evaluación.....	181
Glosario.....	187
Trabajos citados.....	193
Índice de vídeos.....	195
Índice de tablas.....	196
Índice de ilustraciones.....	197

守破離

Shu Ha Ri

El aprendizaje de cualquier habilidad tiene tres etapas.

Shu: se elige una técnica y, asumiendo que es correcta, se intenta imitar.

Ha: se coleccionan más técnicas.

Ri: se experimenta e inventan nuevas técnicas mezclando, combinando y modificando.

Las técnicas de etapa shu son aplicables en general. Las técnicas de etapa ri sólo funcionan en casos concretos, y requieren de conocimiento experto para saber cuándo y cómo aplicarlas.

«No puedes ganar en una industria competitiva utilizando técnicas shu.»

Alistair Cockburn

Introducción

Este es un libro de texto que ofrece apoyo didáctico a la capacitación profesional para la gestión ágil de proyectos, equipos y organizaciones.

Antes de abordar las técnicas y principios del desarrollo ágil, el libro muestra en el bloque de introducción, el cuerpo de conocimiento profesional desarrollado a finales del siglo pasado para afrontar los retos planteados por la industria del software y por qué razón no ha dado los resultados previstos.

Este vistazo histórico de introducción revela las características propias de las industrias del conocimiento y las razones por las que la gestión ágil resulta más adecuada. Esta perspectiva ayuda a comprender el valor de las personas comprometidas sobre la rigidez de la ingeniería de procesos; el valor de la capacidad para responder al cambio frente al cumplimiento de agendas planificadas, así como las nuevas perspectivas que abre la agilidad al dirigir el foco de valor hacia el cliente, antes que hacia los propios resultados.

La introducción toma como referente la industria del software, por haber sido cuna de la agilidad. Los principios desarrollados en el libro para la gestión ágil de equipos (segundo módulo) y de organizaciones (tercer módulo) no son específicos para el software y resultan válidos para las empresas del conocimiento.

A continuación, el segundo módulo enseña las técnicas de gestión ágil para equipos, desde la perspectiva de los valores ágiles que potencian y por tanto desde la razón de su funcionamiento. Este módulo es el temario troncal 1 del marco de formación profesional Scrum Manager®.

El tercer bloque incluye completo el modelo Scrum Level© como estrategia de gestión para desarrollar la agilidad a nivel de organización.

MÓDULO 1

PERSPECTIVA

Crisis del software

Mariner I

Mariner 1 fue la primera nave espacial en el programa *American Marine*.

Fue lanzado el 22 de julio de 1962 desde Cabo Cañaveral en una misión para recopilar una variedad de datos científicos sobre Venus.

Sólo 293 segundos después del lanzamiento, un oficial de seguridad de rango ordenó un aborto destructivo, cuando se desvió del rumbo después de una maniobra no programada. El comando de destrucción se envió seis segundos antes de la separación, después de lo cual no hubiera podido ser destruido.

En una misión para volar sobre Venus en 1962, esta nave espacial apenas logró salir de Cabo Cañaveral porque un error de codificación del software hizo que el cohete se desviara peligrosamente de su curso, amenazando con estrellarse de nuevo en la tierra.

Un comité de revisión determinó más tarde que la omisión de un guión en las instrucciones codificadas de la computadora permitió la transmisión de señales de guía incorrectas a la nave espacial. El costo del cohete fue de más de 18 millones de dólares en ese momento.



Ilustración 1:
Lanzamiento del Mariner I (NASA Great Images)

IBM os/360



Ilustración 2: IBM System 360 - Computer History Museum - Wikimedia Commons.

La producción del sistema os/360 es un buen ejemplo de la crisis del software.

El os/360 iba a ser producido con el sistema 360.

Su producción comenzó en 1960 y el lanzamiento se planificó para 1966 .

El software era grande y complejo,

con más de un millón de líneas de código y una inversión inicial de 125 millones de dólares.

En la primavera de 1964 se puso en marcha el desarrollo. Había alrededor de 70 programadores trabajando en el proyecto, pero más tarde se vió que los tiempos se estaban alargando por lo que contrataron más programadores aumentando de 70 a 150.

Pero a medida que aumentaban el número de programadores, disminuía el rendimiento.

A pesar del aumento de programadores, el desarrollo se estaba retrasando aproximadamente 6 meses. Además, se hizo una prueba del sistema que determinó que el sistema era muy lento, lo que implicó reprogramación de lo ya hecho y más retraso en el progreso del trabajo.

A finales de 1965 se descubrió que había fallos fundamentales y no parecía haber una forma fácil de arreglarlos. Se replanteó el plan de desarrollo añadiendo 9 meses de retraso.

En el pico del desarrollo del sistema se emplearon 1000 personas.

Finalmente, a mediados de 1967 el sistema se produjo un año más tarde de la fecha inicial estipulada con una pérdida aproximada de 500 millones de dólares.

Este es un buen ejemplo del tipo de problemas que dieron lugar al concepto "crisis del software"

 <https://youtu.be/vOXsrIDYnyo>

Los problemas que planteaba la integración de sistemas de software en los grandes proyectos militares se trataron por primera vez en 1968, año en el que la organización NATO desarrolló la primera conferencia sobre desarrollo de software, (Nato 1969) y en la que se acuñaron los términos "crisis del software" para definir a los problemas que surgían en el desarrollo de sistemas de software, e "ingeniería del software" para describir el conjunto de conocimientos que existían en aquel estado inicial.

Algunas referencias útiles para comprender cuáles eran los conocimientos estables para el desarrollo de software en 1968 son:

- En 1962 se publicó el primer algoritmo para búsquedas binarias.
- C. Böhm y G. Jacopini publicaron en 1966 el documento que creaba una fundación para la eliminación de "GoTo" y la creación de la programación estructurada.
- En 1968 los programadores se debatían entre el uso de la sentencia GoTo, y la nueva idea de programación estructurada; ese era el caldo de cultivo en el que Edsger Dijkstra escribió su famosa carta "GoTo Statement Considered Harmful" en 1968.
- La primera publicación sobre programación estructurada no vio la luz hasta 1974, publicada por Larry Constantine, Glenford Myers y Wayne Stevens.
- El primer libro sobre métrica de software fue publicado en 1977 por Tom Gilb.
- El primero sobre análisis de requisitos apareció en 1979.

Respuesta a la crisis del software

Tres son las áreas de conocimiento que se decidieron aplicar a partir de los 70 para solucionar los problemas que generaba el desarrollo de sistemas de software fueron:

Ingeniería del software

Aplicación de un enfoque sistemático, disciplinado y cuantificable en el desarrollo, operación y mantenimiento de software.

Desarrollo basado en procesos

Desarrollo basado en procesos y mejora continua de los procesos empleados.

Aplicación del principio de calidad de Jurán empleado en la producción industrial: “La calidad del resultado depende de la calidad del proceso”. Dio lugar al desarrollo de modelos de procesos específicos para software: CMMI, ISO 15504...

Gestión de proyectos

Aplicación de procesos de planificación, ejecución y control para alcanzar un objetivo final en un plazo de tiempo determinado con el coste y nivel de calidad previstos.

Ingeniería del software

En los años 60 el desarrollo de software no disponía de un conocimiento profesionalizado, por lo que parecía necesario determinar cuál debe ser el currículo profesional de un ingeniero de software.

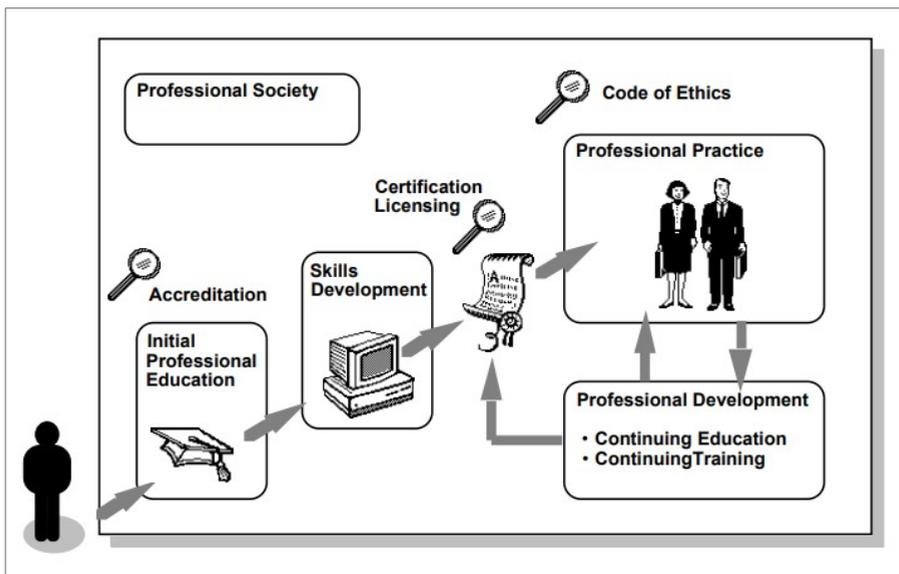


Ilustración 3: Proceso y componentes de una profesión - "A Mature Profession of Software Engineering - Gary Ford, Norman E. Gibbs - SEI, 1996).

La profesionalización exige (Ford & Gibbs 1996):

- Disponer de una base de conocimiento, desarrollada con metodología científica, y contrastada con la experiencia.
- Consenso y aceptación por la comunidad que ejerce la profesión.
- Desarrollo de currículos profesionales, que gracias al consenso resulten homogéneos sobre centros de formación, universidades y países diferentes.

- Desarrollo de las organizaciones académicas y profesionales de autorregulación.

En definitiva este es el esquema que han seguido muchas disciplinas para alcanzar un nivel de solvencia profesional socialmente reconocido.

En esta línea han estado trabajando organizaciones profesionales agrupadas en el proyecto SWEBOK (Software Engineering Body Of Knowledge <http://www.swebok.org>)

Su objetivo ha sido asentar de forma consensuada la base de conocimiento necesaria para el currículo de un ingeniero de software; primer paso para lograr un entorno profesional, socialmente reconocido.

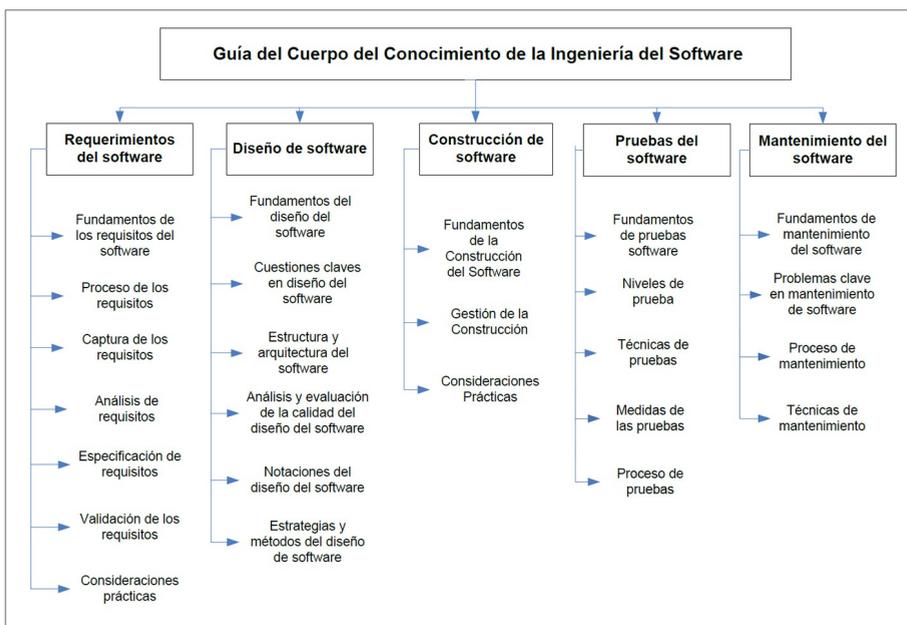


Ilustración 4: 5 primeras áreas de conocimiento de la ingeniería del software - SWBOK v.2004.

El trabajo de este proyecto comenzó en 1998, y la publicación de la primera versión, consensuada por todos los participantes, se produjo en

Febrero de 2004(swebok 2004). Esto da una idea de que se trata de una profesión con una base de conocimiento y trayectoria curricular que quizá aún no está suficientemente asentada.

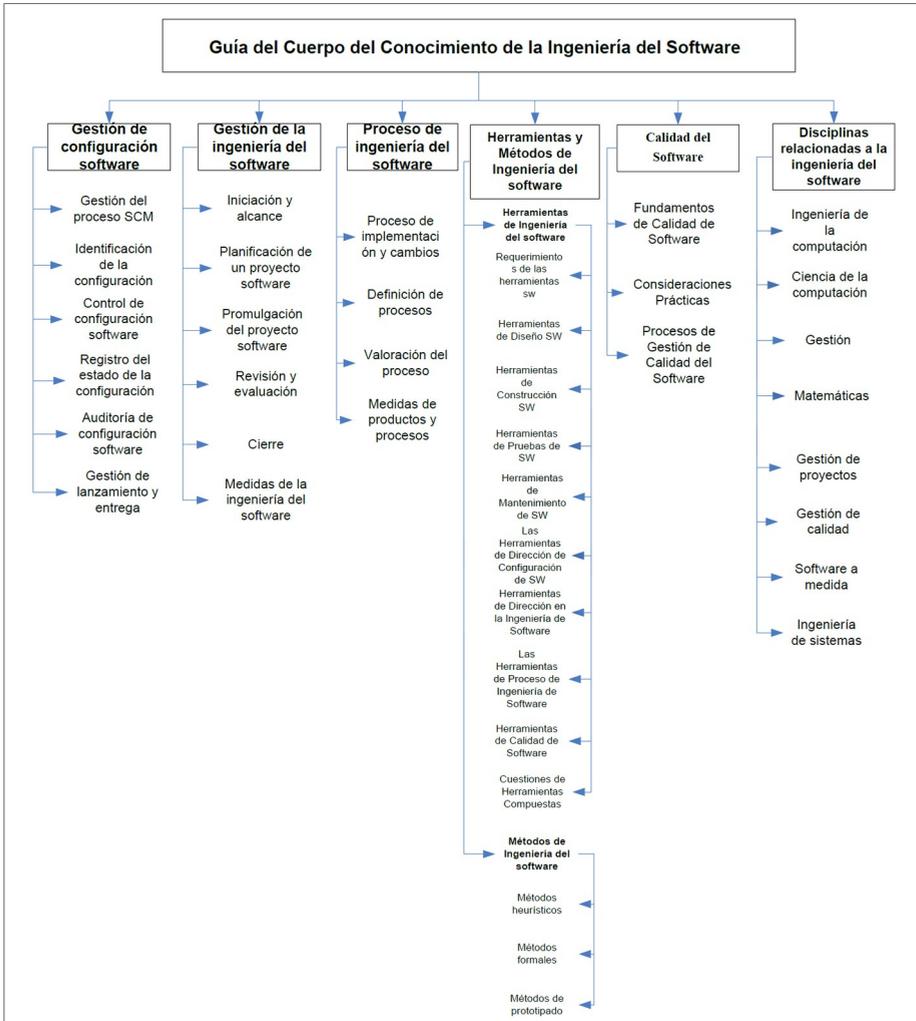


Ilustración 5: 6 segundas áreas de conocimiento de la ingeniería del software - SWBOK v.2004.

Desarrollo basado en procesos

Trabajar sin un método fue la forma empleada por los primeros programadores, en la segunda mitad del siglo pasado.

Los pioneros, atraídos por el encanto de construir artefactos útiles con las nuevas máquinas programables, fueron los primeros en sentarse frente al teclado y decir con una sonrisa: "en un par de meses esto estará funcionando".

Sin embargo la realidad demostraba que al programar sin un método la probabilidad de obtener resultados de calidad en tiempos predecibles es escasa.

SEI, (Software Engineering Institute) denomina "poco maduras" a las organizaciones que desarrollan software sin un método institucionalizado y emplea una escala de 1 a 5 para determinar el grado de madurez y en consecuencia el nivel de garantía que se ofrece en cuanto a calidad, predictibilidad y eficiencia en el desarrollo de software.

SEI establece un nivel de madurez 1 para las organizaciones que trabajan sin método, que no quiere decir necesariamente que vayan a producir mal software, de forma ineficiente o con retraso, sino que la probabilidad de éxito en estos aspectos es baja. Hay equipos que lo consiguen, pero son pocos. La razón es sencilla: los resultados en estos casos son tan buenos como las personas que los producen, y los buenos programadores escasean.

En este modo de trabajo la calidad del resultado se debe al "saber hacer" de los programadores. El éxito o fracaso de las organizaciones que trabajan sin una metodología depende del conocimiento tácito de su personal. Es el modelo "start-up": un equipo de emprendedores con talento, capaces de construir sistemas de software. El resultado será tan bueno como ellos lo sepan hacer. El cumplimiento de agendas dependerá de su capacidad de previsión y organización. Pero no hay que engañarse, en este caso no se trata de empresas que saben hacer software, sino de personas que saben hacer software.

Este es el guión habitual de las pequeñas empresas que surgen impulsadas por el empuje de sus emprendedores: pueden llegar tan lejos como la combinación de talento, capacidad relacional y marketing de sus creadores les permita: acabarán cerrando o alcanzarán el nivel de mediocridad o éxito que puedan lograr combinando esos elementos. El crecimiento más allá de ese punto supondrá un reto

importante: pasar de personas que saben hacer software a empresa que sabe hacer software, porque para ir más allá ya no tienen que ser ellos, sino la organización la que deberá saber producir con eficiencia y calidad repetible en todos los proyectos.

Las fortalezas que aporta la producción basada en procesos son:

- **Eficiencia:** La revisión y mejora continua de los procesos mejoran de forma continua su eficiencia.
- **Repetibilidad.** La calidad del resultado depende de la calidad del proceso, de forma que al aplicar el mismo proceso se obtiene el mismo resultado.
 - **Escalabilidad.** Es una consecuencia de la repetibilidad. Los resultados de calidad los obtienen todos los equipos que aplican los procesos.
- **Calidad.** La calidad del resultado depende de la calidad de los procedimientos empleados. Introduciendo ingeniería de procesos se institucionaliza la mejora continua del sistema, y por tanto de los resultados.

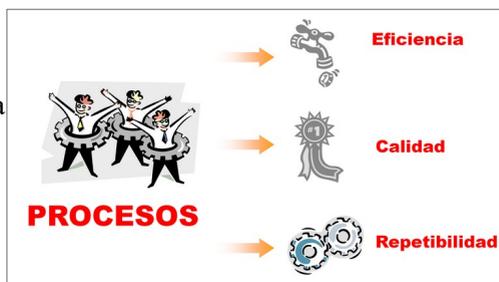


Ilustración 6: Fortalezas de la producción basada en procesos

Los procesos de la empresa contienen el "sabe hacer". El modelo de procesos es un activo valioso de la organización: la clave para hacer las cosas con eficiencia y calidad homogénea.

Procesos, personas y tecnología

Los procesos definen pautas de trabajo pero se necesitan además agentes activos: tecnología y/o personas para producir los resultados.

Un ejemplo para ilustrar las diferencias entre distintos sistemas de producción es el montaje de un mueble en kit. Es un trabajo que requiere personas y herramientas o tecnología: destornillador (manual o eléctrico), tornillos, pasadores, cola de montaje, etc.

Si realizamos el trabajo sólo con estos dos componentes, sin un proceso determinado (el manual con las instrucciones de cómo proceder) lo haremos en un sistema propio de artesanos o artífices: personas + tecnología, que no puede garantizar productividad y calidad homogéneas, porque dependen de las personas.

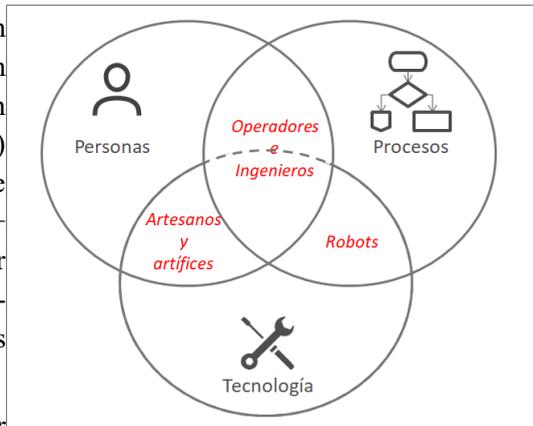


Ilustración 7: Personas, procesos y tecnología.

Hay quien es capaz de ensamblar el mueble en cuestión de minutos y quien necesita toda una tarde.

Habrá quien montará muebles sólidos y quien terminará con peor fortuna, afirmando que el montaje está bien hecho pero que las piezas venían mal cortadas de fábrica, que quien lo diseñó era un perfecto inútil o que los tornillos son de mala calidad, incluso sin percatarse de los fallos que ha cometido

"Personas + tecnología" es una combinación que produce resultados, y puede resultar más o menos apropiada para un taller de carpintería.

Una empresa de carpintería industrial debería optar por una de estas estrategias para obtener resultados de calidad:

- Introducir un tercer elemento en el sistema de producción: los procesos.
- Emplear sólo a buenos carpinteros, proporcionándoles las mejores herramientas.

Parece más apropiado optar por la primera: implementar un sistema de producción basado en procesos. Esta es la estrategia empleada en la manufactura industrial, aunque podemos adelantar que los procesos no dan resultados con la misma solvencia en todos los trabajos y esta ha sido una de las causas por las que en muchos proyectos de software no resultan tan aconsejables. Esto es así porque el desarrollo basado en procesos tiene su kryptonita: las tareas que requieren un conocimiento aprehensible por las personas pero que resulta difícil explicitar en procesos o tecnología.

Capital estructural y capital humano

El capital estructural lo componen los bienes que quedan en la empresa cuando las personas se han ido a sus casas: patentes, licencias, cartera de clientes, equipos, maquinaria, vehículos, etc.

El capital humano es el valor de la empresa, que resulta inseparable de las personas. Todas las organizaciones emplean ambos y la proporción en la que cada uno aporta a la calidad del resultado es distinta en cada caso.

Normalmente el capital estructural es el factor más relevante en las empresas industriales, pero cuál tiene mayor peso en la calidad de la producción no depende sólo del sector o tipo de industria.

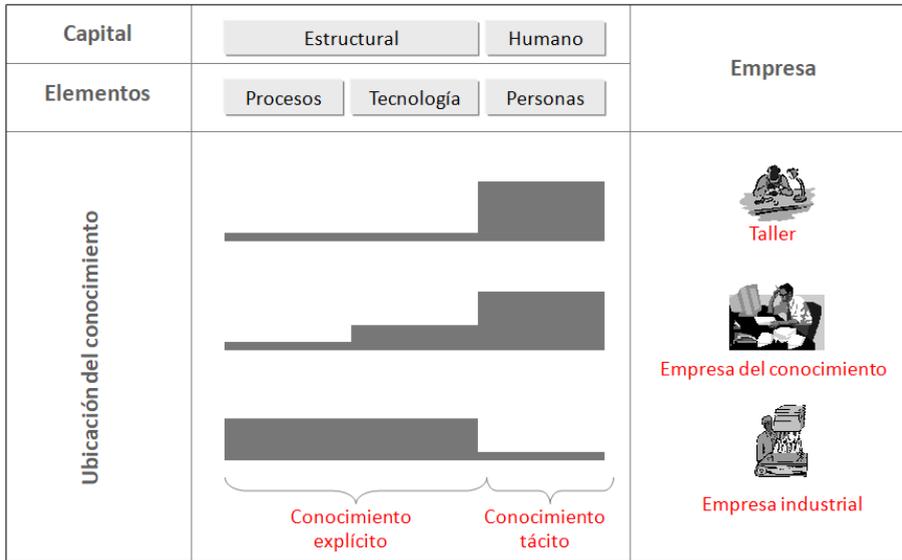


Ilustración 8: Capital estructural y capital humano.

En un mismo sector, distintas empresas pueden tener modelos de producción diferentes. Por ejemplo, dos restaurantes. Uno de ellos de alta cocina, y otro un franquiciado de "PhonoPizza". Sin entrar en consideraciones gastronómicas, ambos negocios tienen perfectamente definida su identidad, personalidad y segmento; ambos tienen también su propio patrón de calidad.

La pizzería emplea una serie de procedimientos para garantizar la calidad de su producto, de forma que depende muy poco del conocimiento tácito de los cocineros. En su mayor parte se debe a los procesos y la tecnología empleada. Los hornos regulan automáticamente el tiempo y la temperatura, los ingredientes se han producido y embolsado en las cantidades adecuadas para cada pizza y se distribuyen a todos los establecimientos en estuches, ambientes frigoríficos y medios de transporte en base a los procesos que son los principales responsables de los resultados. Por esta razón es distinto el conocimiento profesional culinario del personal de cocina de un establecimiento de PhonoPizza que el de un restaurante de alta cocina.

Optimizar el sistema de procesos, tecnología y personas no es fácil. El objetivo es conseguir que cada factor aporte valor hasta el límite de la mejor relación eficiencia / calidad. La revisión permite mejorar de forma continua el sistema e ir consiguiendo nuevos equilibrios con mejores parámetros de eficiencia y calidad en el sistema.

Procesos para la industria del software

La siguiente ilustración representa la evolución de los principales modelos de procesos de referencia para las organizaciones de software.

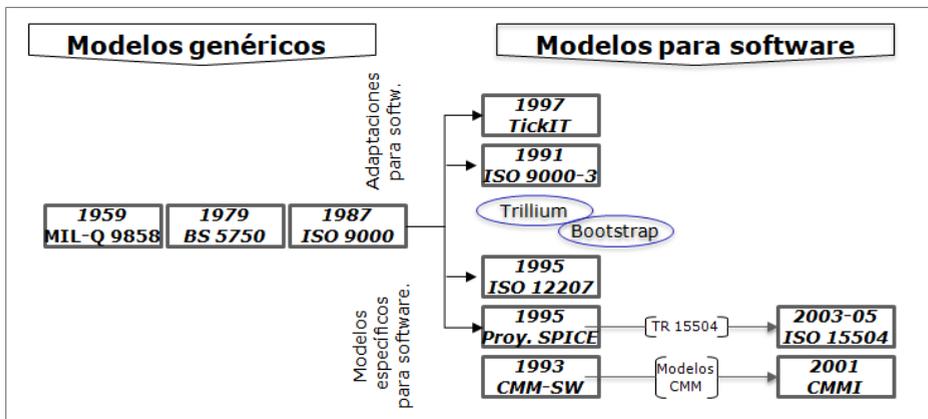


Ilustración 9: Principales modelos de procesos para desarrollo de software.

La norma considerada como punto de arranque de los estándares y modelos que se han extendido a todas las industrias es la desarrollada por EE.UU. en 1959: "Quality Program Requirements" MIL-Q-9858, diseñada para el ámbito militar que estableció un esquema auditable (a través de la norma de inspección MIL-I.45208) de los requisitos que los proveedores debían cumplir.

El uso de esta norma se fue generalizando mientras en paralelo, diferentes organizaciones comenzaron a desarrollar las suyas propias. Así por ejemplo la OTAN en 1968 adoptó las especificaciones AQAP "Allied Quality Assurance Procedures".

El estándar británico BS5750, adoptado en el Reino Unido en 1979 fue el siguiente hito relevante en el camino de las normalizaciones, al lograr gran reconocimiento en varios países. Esta normativa fue la precursora de ISO 9000. Los estándares que fueron surgiendo de los 60 a los 90 dieron respuesta a las garantías de homogeneidad, calidad y predictibilidad de los entornos de fabricación y especialmente en el desarrollo de sistemas complejos que comprendían tecnologías e ingenierías diferentes: mecánica, aeroespacial, naval, atómica etc.

Estos sistemas comenzaban a integrar una tecnología recién nacida: el software que aún no disponía de una base de conocimiento asentada y consensuada. Algunos de los fracasos que se produjeron en determinados sistemas por introducción del software sin la madurez suficiente pusieron de manifiesto la necesidad de desarrollar una ingeniería del software que pudiera ofrecer la nueva tecnología con las garantías necesarias.

En las últimas décadas la industria del software ha desarrollado:

- Modelos de procesos para garantizar la calidad del resultado.
- Técnicas de desarrollo.

En este punto es interesante diferenciar entre unos y otros.

Los primeros, los procesos, tienen como objetivo fijar "qué" cosas hay que hacer para ofrecer garantías en el resultado de los proyectos y los segundos, cómo debe hacerse esas cosas.

Un marco de procesos determinará, por ejemplo, la necesidad de gestionar adecuadamente las modificaciones de requisitos o de realizar un diseño detallado antes de comenzar la codificación.

El valor de los modelos de procesos radica en el conocimiento que aportan al señalar las actividades cuya omisión incrementa las posibilidades de fracaso en los proyectos.

Sin embargo un marco de procesos no establece si el diseño se deberá realizar diagramas IDEF o UML. Este es el campo de la técnica de la ingeniería del software.

En la línea de los modelos de procesos (qué actividades hay que hacer) a partir de finales de los 80 se comenzaron a desarrollar marcos específicos para el software, para cubrir a las particularidades de este producto, en el que las normas generales se quedaban cortas.

En esta línea ISO 9000 desarrolló una norma específica para el software: ISO 9000-3 y la BSI (British Standards Institution) hizo lo propio desarrollando TickIT.

En esta primera aparición de estándares surgieron también, aunque con menor repercusión: Trillium y Bootstrap.

El primero desarrollado por la empresa Bell Canadá, que liberó sus derechos haciéndolo de dominio público.

Bootstrap es una metodología para la evaluación, medición y mejora de los procesos de software. Su desarrollo lo llevó a cabo una comisión del ESPRIT (European Strategic Program for Research).

Sin embargo las dos líneas que surgieron a principios de los 90 y que continúan como referentes en la actualidad son. CMMI, y las normalizaciones de ISO 12207 y 15504.

CMM / CMMI

SEI (Software Engineering Institute) desarrolló un modelo de procesos sobre el concepto de la “madurez” de las organizaciones para producir software.

Por madurez se entiende el nivel de garantía de la organización para asegurar la calidad de sus proyectos (fecha, coste y funcionalidad), la homogeneidad (siempre y en todos los desarrollos) y la capacidad de aprendizaje de la propia experiencia y su aplicación en la mejora continua.

Como resultado de estas líneas de trabajo en 1990 publicó el modelo de madurez de la capacidad para el desarrollo de software (CMM-SW) que tras más de una década de existencia demostró eficacia en muchas organizaciones.

Este modelo emplea una escala de cinco niveles para determinar la madurez de una organización.

Nivel 1.- INICIAL: Pertenecen a él las empresas que no siguen procesos definidos ni aplican técnicas de gestión de proyectos. Son talleres de producción de resultados no predecibles que dependen exclusivamente de la valía de las personas.

Nivel 2.- REPETIBLE: Define a las organizaciones que aplican técnicas de gestión de proyectos, aunque no dispongan de procesos definidos.

Nivel 3.- DEFINIDO: Pertenecen a él las organizaciones que disponen de procesos definidos con precisión y ejecutados de forma regular. Las empresas con este nivel de madurez examinan la experiencia de los proyectos que realizan y emplean las lecciones aprendidas para mejorar sus procesos.

Nivel 4.- GESTIONADO: En el cuarto nivel de madurez se sitúan las organizaciones que han depurado el análisis de los proyectos realizados, hasta institucionalizarlo como procesos que miden cuantitativamente, de forma que pueden predecir cuantificablemente los resultados y evaluar las mejoras con mediciones objetivas.

Nivel 5.- OPTIMIZADO: Las organizaciones con un nivel 5 de madurez tienen definidos, y practican de forma institucionalizada, procesos de mejora continua que se nutren con la información cuantificada de los procesos del nivel 4.

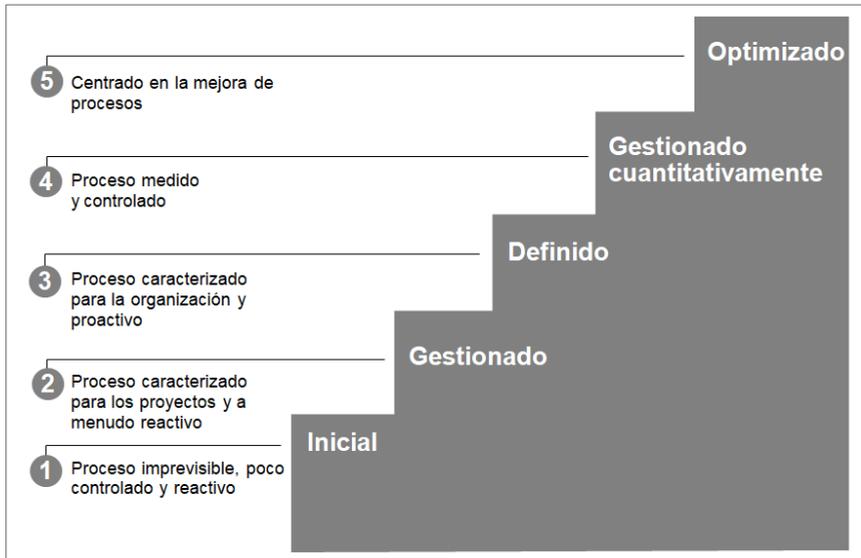


Ilustración 10: Niveles de madurez de las organizaciones.

Tras el desarrollo del modelo SW-CMM, SEI desarrolló otros para la medición y mejora de la capacidad de los procesos, todos relacionados con el foco original del SW-CMM: el desarrollo de software:

- Systems Engineering Capability Maturity Model (SE-CMM)
- Integrated Product Development Capability Maturity Model (IPD-CMM)
- People Capability Maturity Model (P-CMM)
- Software Acquisition Capability Maturity Model (SA-CMM)

En 2000 se desarrolló una versión de SW-CMM que integraba algunos de estos modelos (SE-CMM o IPD-CMM). El resultado fue un nuevo modelo que relevó al original SW-CMM: CMMI (Capability Maturity Model Integration). En la actualidad hay varios modelos CMMI, en función de las áreas que integran

- CMMI-SE/SW/IPPD/SS (Systems Engineering, Software Engineering, Integrated Product and Process Development, Supplier Sourcing).
- CMMI-SE/SW/IPPD (Systems Engineering, Software Engineering, Integrated Product and Process Development).

- CMMI-SE/SE (Systems Engineering, Software Engineering)

Además de la integración de varias disciplinas, los modelos CMMI introdujeron otra novedad que afectaba a su implantación: dos formas posibles de llevarla a cabo: escalonada o continua.

La versión escalonada prescribe a la organización en qué orden debe abordar las diferentes áreas de procesos, las prácticas que debe implantar y los objetivos que debe alcanzar para ir consiguiendo los sucesivos niveles de madurez.

La versión continua permite cierta libertad a la organización sobre las áreas de proceso que desea mejorar, y le orienta para ir mejorando el nivel de capacidad en esas áreas.

CMMI al igual que su predecesor CMM, tiene dos utilidades: puede servir como guía para la mejora en una organización, o como criterio para evaluación, pero mientras CMM centraba estas dos finalidades en la dimensión de la madurez de la organización, CMMI introduce una segunda dimensión, también válida para guiar las actividades de mejora y para evaluar a las organizaciones: la capacidad de los procesos.

CMMI establece 6 niveles para determinar la capacidad de un proceso:

Nivel 0.- INCOMPLETO: El proceso no se realiza.

Nivel 1.- REALIZADO: Se lleva a cabo el proceso, consiguiendo transformar elementos de entrada identificados, en productos de salida.

Nivel 2.- GESTIONADO: El proceso se ejecuta siempre de la misma manera, de una forma gestionada.

Nivel 3.- DEFINIDO: El proceso está definido en la organización y se ejecuta siempre.

Nivel 4.- CUANTIFICADAMENTE GESTIONADO: La ejecución del proceso tiene institucionalizado en la organización un sistema de medición objetivo y cuantificable de su capacidad.

Nivel 5.- OPTIMIZADO: El proceso, que se ejecuta siempre, está definido en la organización, se mide y está integrado en un plan también institucionalizado de

mejora continua, basada en mediciones de parámetros del mismo objetivamente cuantificables.

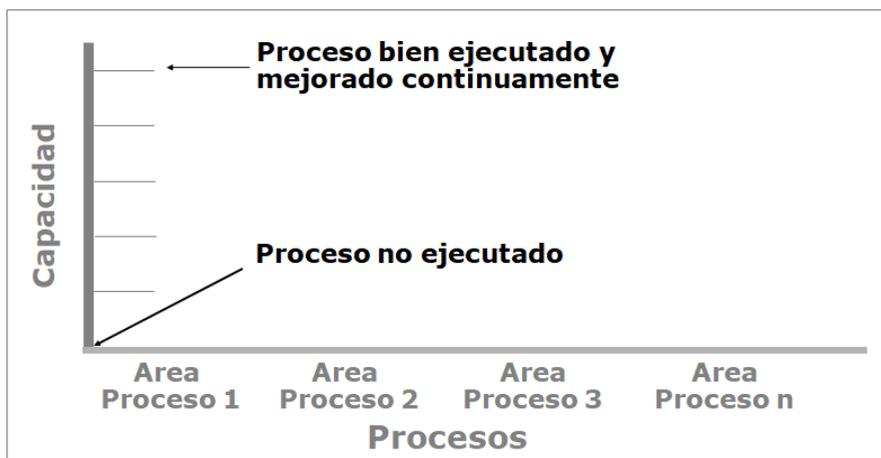


Ilustración 11: Niveles de capacidad de los procesos.

De esta forma los modelos CMMI presentan 2 versiones:

Versión escalonada. Guía a la organización acerca de las áreas de procesos que debe ir abordando, las prácticas que debe implantar y los objetivos que debe alcanzar para ir consiguiendo los sucesivos niveles de madurez.

Versión continua. Permite cierta libertad a la organización sobre las áreas de proceso que desea mejorar, y le orienta para ir elevando el nivel de capacidad de las mismas.

Consecuentemente al evaluar a una organización se puede hacer sobre la dimensión de su madurez (versión escalonada), estableciendo cuál es el nivel que ha alcanzado, o sobre la dimensión de la capacidad de cada área de proceso (versión continua) reflejando cuál es el nivel de ésta en las diferentes áreas de procesos.

Mientras SEI publicaba y comenzaba a definir su modelo CMM, ISO emprendía los otros dos proyectos que hoy forman los principales puntos de referencia en el ámbito de la calidad para nuestra industria: ISO 12207 e ISO 15504.

ISO 12207

Es un estándar internacional referente para la normalización de los procesos que constituyen el ciclo de vida de un proyecto de software, y las tareas que intervienen en cada una de sus fases.

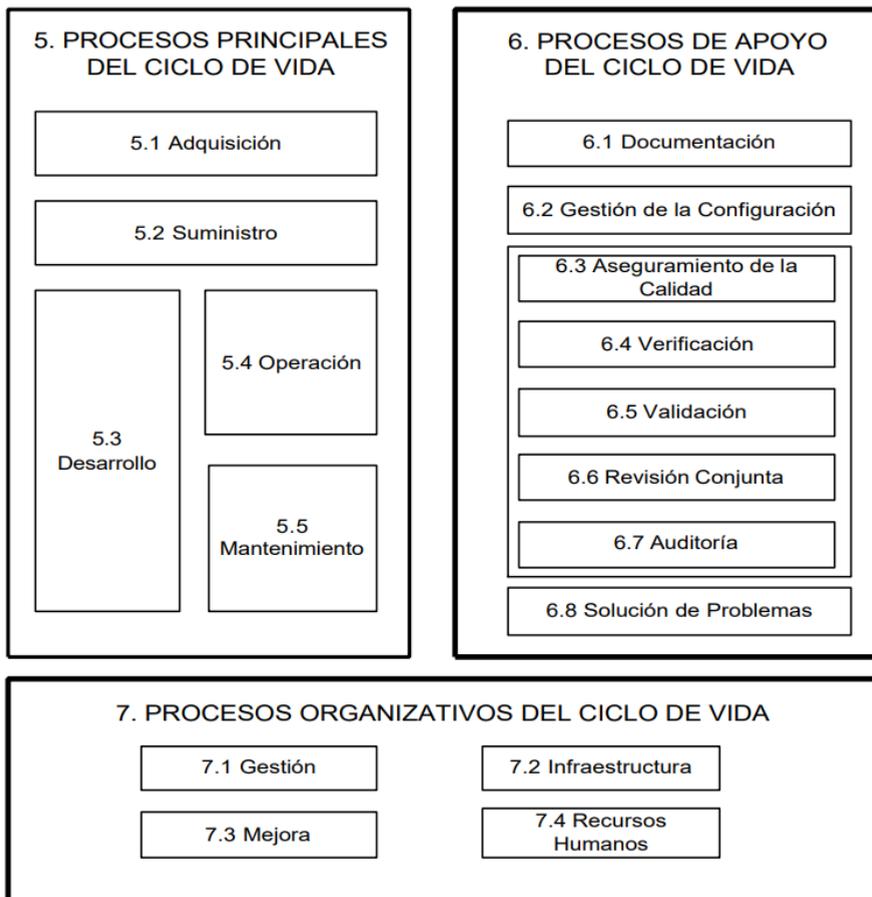


Ilustración 12: ISO 12207, v. 2005

La versión actual de este estándar ha modificado el esquema de procesos. Para la finalidad didáctica de este texto se mantiene esta versión de 2005 por ofrecer una estructura más comprensible.

El estándar 12207 considera que el ciclo de vida de un sistema de software comienza en el momento que se concibe su idea o necesidad, momento desde el que es necesario comenzar a actuar de manera normalizada para describir el ámbito del problema, las soluciones posibles, etc.

El ciclo se extiende desde ese momento, cubriendo el desarrollo, mantenimiento y operación; y no concluye hasta que el sistema deja de utilizarse y es definitivamente retirado.

SPICE – ISO/IEC STD. 15504.

En enero de 1993 la comisión ISO/IEC JTC1 aprobó un programa de trabajo para el desarrollo de un modelo que fuera la base de un futuro estándar internacional para la evaluación de los procesos del ciclo de vida del software.

Este trabajo recibió el nombre de proyecto SPICE (Software Process Improvement and Capability dEtermination), y en junio de 1995, con la publicación de su primer borrador, desde ISO fueron invitadas diferentes organizaciones para aplicarlo y valorar sus resultados. En 1998, pasada la fase de proyecto, y tras las primeras evaluaciones, el trabajo pasó a la fase de informe técnico con la denominación ISO/IEC TR 15504.

La instrucción técnica consta de 9 apartados, recogidos en volúmenes independientes, que se han ido publicando como redacción definitiva del estándar internacional ISO/IEC 15504 durante el periodo 2003-2005.

Aunque ISO comenzó con el proyecto SPICE algo más tarde que SEI con el modelo CMM, durante su evolución han ido convergiendo y ambas instituciones vienen trabajando de forma conjunta desde 1998 para lograr la compatibilidad que finalmente han garantizado entre el modelo CMMI y el estándar 15504, de forma que la conformidad con uno de ellos implica la también conformidad con el otro.

El estándar está recogido en 9 documentos:

1. (informativo) Conceptos y guía de introducción.
2. (normativo) Modelo de referencia de los procesos y de su capacidad.
3. (normativo) Ejecución de las auditorías.
4. (informativo) Guía para la realización de auditorías.
5. (informativo) Modelo de asesoría e indicadores.
6. (informativo) Guía de formación de consultores.
7. (informativo) Guía para usar en los procesos de mejora.
8. (informativo) Guía para determinar la capacidad de los procesos del proveedor.
9. (normativo) Vocabulario.

ISO/IEC 15504, al igual que CMMI es un modelo para evaluar los procesos de la organización y determinar si resultan efectivos para conseguir los objetivos.

También es un modelo para guiar las acciones de mejora de los procesos. 15504 tiene gran similitud con la representación continua de CMMI. Al igual que este último centra el foco en la capacidad de los procesos, y permite trabajar sólo con una parte de ellos en lugar de hacerlo con todos los de la organización.

ISO 15504 agrupa los procesos de las organizaciones de software en cinco categorías:

- Cliente – proveedor
- Ingeniería
- Soporte
- Gestión
- Organización

Y para la medición de la capacidad de cada proceso define una escala de 6 niveles

- 0 Proceso incompleto
- 1 Proceso realizado
- 2 Proceso gestionado
- 3 Proceso establecido
- 4 Proceso predecible
- 5 Proceso optimizado.

ISO 15504 y CMMI son los referentes de las metodologías formales.

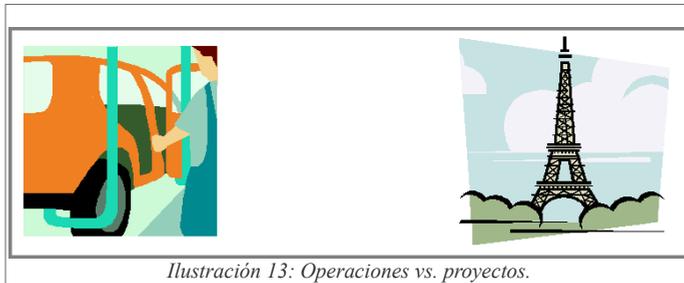
Ambos prescriben el seguimiento de procesos institucionalizados para garantizar la calidad el desarrollo, mantenimiento y operación del software.

Gestión de proyectos

Los métodos formales (CMMI, ISO15504) basan el éxito del proyecto en la planificación y el rigor normativo en los procesos de su desarrollo.

Hacen especial hincapié en la importancia de los requisitos para conocer con el mayor detalle el sistema antes de comenzar su construcción, y en las prácticas formales de la gestión de proyectos, para controlar el seguimiento de la planificación evitando desviaciones.

Diferencia entre proyectos y operaciones



Los bienes y los servicios se pueden llevar a cabo como “operaciones” o como “proyectos”.

Ambas formas comparten tres características:

- Los realizan personas.
- Se emplean recursos limitados.
- Se llevan a cabo siguiendo una estrategia de actuación.

La diferencia clave entre ellas es que las operaciones se ejecutan de forma repetitiva para obtener resultados de similares características

Los proyectos producen resultados únicos.

Se considera proyecto a la ejecución de un trabajo que además de requerir personas, recursos y ejecución controlada:

La gestión de proyectos predictiva considera que Los proyectos son desarrollos únicos que se desarrollan en un marco temporal preestablecido y los define como:

Conjunto único de actividades necesarias para producir un resultado definido en un rango de fechas determinado y con una asignación específica de recursos

Las construcciones de ingeniería civil, como puentes o edificios, son ejemplos clásicos de obras realizadas como proyectos, y en general lo es el desarrollo de cualquier sistema singular.

Un proyecto tiene objetivos y características únicas. Algunos necesitan el trabajo de una sola persona, y otros el de cientos de ellas; pueden durar unos días o varios años.

Algunos ejemplos de proyectos:

- Diseño de un nuevo producto.
- Construcción de un edificio.
- Desarrollo de un sistema de software.
- Implantación de una nueva línea de producto en una empresa.
- Diseño de una campaña de marketing.

Origen de la gestión de proyectos

Los proyectos han existido siempre. Cualquier trabajo para desarrollar algo único es un proyecto, pero la gestión de proyectos es una disciplina relativamente reciente que comenzó a forjarse en los años sesenta.

La necesidad de su profesionalización surgió en el ámbito militar. En los 50, el desarrollo de complejos sistemas militares, requería coordinar el

trabajo conjunto de equipos y disciplinas diferentes, en la construcción de sistemas únicos.

Bernard Schriever, arquitecto del desarrollo de misiles balísticos Polaris, es considerado el padre de la gestión de proyectos, por la introducción del concepto de “conurrencia”, para integrar todos los elementos del plan del proyecto en un solo programa y presupuesto.

El objetivo de la conurrencia era ejecutar las diferentes actividades de forma simultánea, y no secuencialmente, y al aplicarla en los proyectos Thor, Atlas y Minuteman se redujeron considerablemente los tiempos de ejecución.

La industria del automóvil siguió los pasos de la militar, aplicando técnicas de gestión de proyectos para la coordinación del trabajo entre áreas y equipos diferentes.

Comenzaron a surgir técnicas específicas, histogramas, cronogramas, los conceptos de ciclo de vida del proyecto o descomposición en tareas (WBS Work Breakdown Structure).

En 1960, Meter Norden, del laboratorio de investigación de IBM, en su seminario de Ingeniería de Presupuesto y Control presentado ante American Management Association, indicó:

- Es posible relacionar los nuevos proyectos con otros pasados y terminados para estimar sus costes
- Se producen regularidades en todos los proyectos
- Es absolutamente necesario descomponer los proyectos en partes de menor dimensión para realizar planificaciones.

Organizaciones referentes en la gestión de proyectos

La construcción de sistemas complejos que requerían el trabajo sincronizado de varias disciplinas hizo evidente en los 60 la necesidad de nuevos métodos de organización para evitar problemas recurrentes:

- Desbordamiento de agendas.
- Desbordamiento de costes.
- Calidad o utilidad del resultado obtenido.

Para dar respuesta a esta necesidad, desde los años 60 han surgido organizaciones que contribuyen al desarrollo del cuerpo de conocimiento de una gestión de proyectos, para ofrecer garantías de previsibilidad y calidad de los resultados.

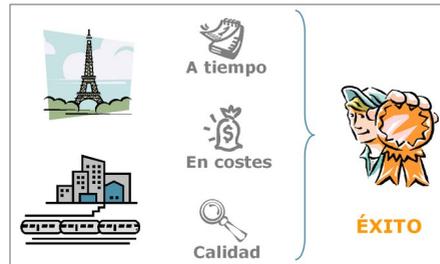


Ilustración 14: Objetivos de la gestión de proyectos predictiva.

Este conocimiento se ha configurando como el currículo de una nueva profesión: La gestión de proyectos **predictiva**.

Las organizaciones más relevantes en esta línea son:

- Internacional Project Managenet Association (IPMA), fundada en 1965
- Project Management Institute (PMI) constituido en 1965
- Más tarde surgió Prince2, que comenzó a trabajar en 1989.

IPMA y PMI surgieron como organizaciones profesionales para desarrollar metodologías y procesos para la gestión de proyectos.

Prince2 ha tenido la evolución inversa. Comenzó siendo una metodología, alrededor de la que se ha terminado creando una organización.

Modelo válido para cualquier industria.

También en este sentido el sentido de evolución ha sido diferente para Prince2.

PMI e IPMA tuvieron desde el principio como finalidad el desarrollo de un conocimiento de gestión válido para cualquier proyecto.

Sin embargo, Prince2 comenzó siendo un modelo de referencia para proyectos específicos de Tecnologías de la Información, desarrollado por la Central Computer and Telecommunications Agency (CCTA) del Gobierno Británico; y a partir de una revisión llevada a cabo en 1996 se decidió ampliar su ámbito de validez, para cualquier tipo de proyecto.

Estrategia de la gestión de proyectos predictiva

La gestión de proyectos predictiva se basa en la planificación del trabajo y en el posterior seguimiento y control de la ejecución.

La planificación se realiza sobre un análisis detallado del trabajo que se quiere realizar y su descomposición en tareas.



Ilustración 15: Claves de la gestión predictiva.

Parte por tanto de un proyecto de obra, o de unos requisitos detallados de lo que se quiere hacer.

Sobre esa información se desarrolla un plan adecuado a los recursos y tiempos disponibles.

Durante la construcción se sigue de cerca la ejecución para detectar posibles desviaciones y tomar medidas para mantener el plan, o determinar qué cambios va a experimentar.

Se trata de una gestión “predictiva”, que vaticina a través del plan inicial cuál va a ser la secuencia de operaciones de todo el proyecto, su coste y tiempos.

Su principal objetivo es conseguir que el producto final se obtenga según lo “previsto”; y basa el éxito del proyecto en los tres puntos apuntados: agendas, costes y calidad.

Es una disciplina formal de gestión, basada en la planificación, ejecución y seguimiento a través de procesos sistemáticos y repetibles.

- Establece como criterios de éxito: obtener el producto definido, en el tiempo previsto y con el coste estimado.
- Asume que el proyecto se desarrolla en un entorno estable y predecible.
- El objetivo de su esfuerzo es mantener el cronograma, el presupuesto y los recursos.
- Divide el desarrollo en fases a las que considera “ciclo de vida”, con una secuencia de tipo: Concepto, requisitos, diseño, planificación, desarrollo, cierre.

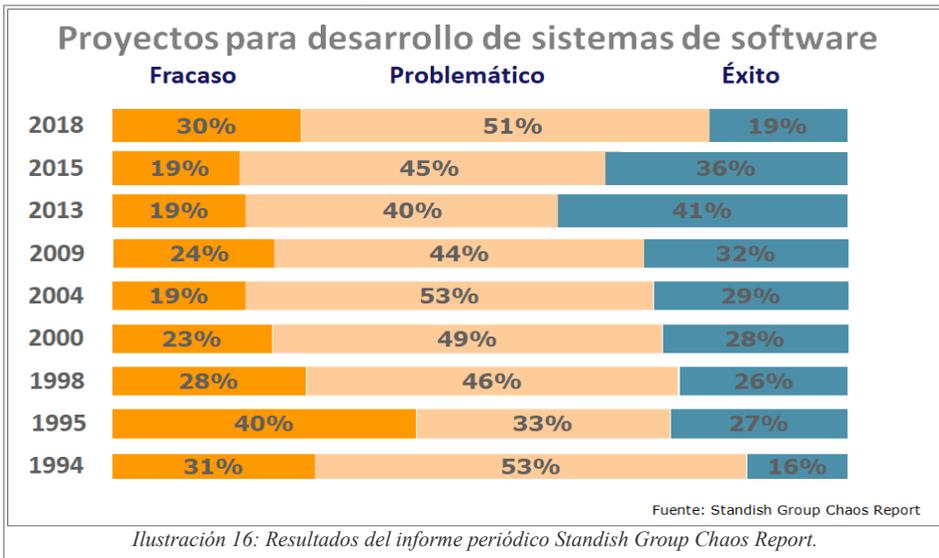
Ámbito de la gestión de proyectos

La solvencia demostrada por la gestión de proyectos en la industria militar, y en la automovilística para solucionar los problemas habituales de calidad, tiempos y costes, coincide en el tiempo con la presión que todas las industrias experimentan en mayor o menor medida para reducir la agenda de salida al mercado y los costes de producción.

Como resultado, en todos los sectores: farmacéutico, químico, servicios, tecnologías de la información, etc. se adoptan técnicas de gestión de proyectos, dándoles de facto validez para todos los ámbitos.

Cuestionando las soluciones surgidas por la crisis del software

Sin embargo, aunque en los años 70, 80 y 90 se desarrolló conocimiento específico para aplicar estas estrategias, lo cierto es que los proyectos de software continuaban siendo problemáticos.



Sólo uno de cada tres se completa reuniendo los tres requisitos que el análisis periódico de Standish Group considera necesarios para catalogarlo como exitoso en su informe:Chaos report:

- En el plazo planificado.
- Sin desbordar el presupuesto estimado.
- Satisfaciendo las necesidades del cliente.
 - Lleva a cabo las operaciones que necesita.
 - Las realiza correctamente.

Es tanta la información de estándares, modelos marcos y prácticas de desarrollo que se ha generado en las últimas décadas que no resulta fácil decidir qué referencias resultan más convenientes en cada caso.

La rectificación realizada por Tom DeMarco en 2009 ilustra muy bien el hecho de que las soluciones desarrolladas a partir de la crisis software han aportado aspectos cuestionables.

Tom DeMarco es autor de uno de los principales trabajos sobre métricas para la gestión de proyectos de software y referencia obligada en temarios como PMBoK: “Controlling Software Projects: Management, Measurement and Estimation”.

Es también “padre” del axioma de los modelos de producción basada en procesos: “Usted no puede controlar lo que no puede medir”.

Es en definitiva uno de los principales referentes en el desarrollo de la ingeniería de software y gestión predictiva.

Con motivo del 40 aniversario de la conferencia de la NATO que determinó el nacimiento de la ingeniería del software, el número de julio/agosto de IEE SOFTWARE publicó un artículo de Tom DeMarco en el que afirmaba:

“Las métricas que inicialmente expuse en mi libro Controlling Software Projects: Management, Measurement and Estimation, han definido la forma en la que muchos ingenieros construían el software y planificaban el trabajo. Con un ánimo de estado reflexivo, ahora me pregunto: ¿Fue correcto el asesoramiento en métricas? ¿Sigue siendo pertinente? y ¿creo todavía que las métricas son una necesidad para el éxito de cualquier desarrollo de software?. Mis respuestas son no, no y no.

Muchos proyectos han avanzado sin centrar la gestión en el control, sino en la creación de productos maravillosos como GoogleEarth o Wikipedia. Para entender la verdadera función del control, es necesario distinguir de manera drástica entre dos tipos diferentes de proyectos:

*Un proyecto de tipo A, con un coste estimado de un millón de dólares y un cálculo de retorno aproximado de 1,1 millones.
Un proyecto de tipo B, que con un coste estimado de un millón de dólares, produce un valor de más de 50 millones de dólares.*

Lo inmediatamente evidente es que el control resulta importante en el proyecto A, y sin embargo su importancia es mínima en el B.

Esto nos lleva a la extraña conclusión de que el control estricto es importante en los proyectos poco importantes, y viceversa.

Conocimiento en continua evolución

Los marcos de prácticas ágiles no surgen en los proyectos TIC como “tesis” sino como “antítesis” del conocimiento de Ingeniería del Software que se venía empleando.

Comenzaremos viendo qué significa esto para tomar la distancia y conseguir la perspectiva que revela las razones de la agilidad y sus diferencias con la ingeniería de procesos, no desde las prácticas concretas sino desde los principios en los que se basan, y con ello comprender las fortalezas y debilidades de la agilidad.

Esta visión desde las razones y los principios de los diferentes marcos y metodologías, más allá de la visión enfocada en las diferencias entre metodologías y prácticas concretas es clave para dar el salto de gestión técnica a gestión experta.

El patrón dialéctico

La evolución y mejora del conocimiento se produce al cuestionar el conocimiento actual, siguiendo de este modo un patrón dialéctico de tesis, antítesis y síntesis.

De manera esquemática el patrón dialéctico puede definirse como la evolución que contrapone una antítesis a una determinada concepción, entendida como tesis.

Esta antítesis muestra los problemas y contradicciones de la tesis. De la confrontación surge un tercer momento llamado síntesis, una resolución o una nueva comprensión del problema.

De esta forma la estrategia de aplicar ingeniería de procesos para atajar los retos de los proyectos de software es la tesis a cuyos problemas y contradicciones se contraponen la agilidad.

La conclusión de la conferencia de la OTAN de 1968 fue la necesidad de crear una disciplina científica que como ocurría en otras áreas, permitiera aplicar un enfoque sistemático disciplinado y cuantificable al desarrollo, operación y mantenimiento de los sistemas del software, es decir, la aplicación de la ingeniería de procesos al software. Fue el nacimiento de la Ingeniería del Software.

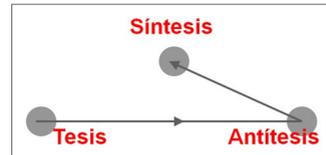


Ilustración 17: Patrón dialéctico del conocimiento.

La primera estrategia (tesis) se ha basado en tres pilares:

- Ingeniería del software
- Ingeniería de procesos
- Gestión predictiva

El primero para aplicar el principio básico de calidad contrastado con éxito en los entornos de producción industrial: “la calidad del resultado depende de la calidad de los procesos empleados”.

El segundo para garantizar el cumplimiento de agendas y presupuestos.

Sin embargo, mientras este conocimiento iba evolucionando y perfeccionándose a través de diferentes técnicas de ingeniería del software, modelos de procesos y cuerpos de conocimiento de gestión de proyectos en la industria del software surgían dudas y se cuestionaba esta estrategia.

¿La planificación predictiva es apropiada para cualquier proyecto? ¿Los criterios de éxito son siempre el cumplimiento fechas, costes y funcionalidades preestablecidas?

Empiezan a surgir proyectos que no tienen como finalidad construir un sistema que previamente se ha definido y planificado en su totalidad, y para los que no es realista trazar un plan cerrado desde su inicio. Proyectos en los que no interesa saber si el sistema final tendrá 20 o 200 funcionalidades y conocer cómo serán éstas en detalle: Su interés es poner un nuevo sistema en el mercado lo antes posible y desde ese momento evolucionar la visión y mejora el valor de forma continua.

Por otra parte se le cuestiona también a la tesis el considerar al desarrollo de software como un proceso industrial. Se empieza a aceptar que puede ser más importante para la calidad del resultado, el conocimiento tácito de la persona que lo realiza que la calidad del proceso que emplea.

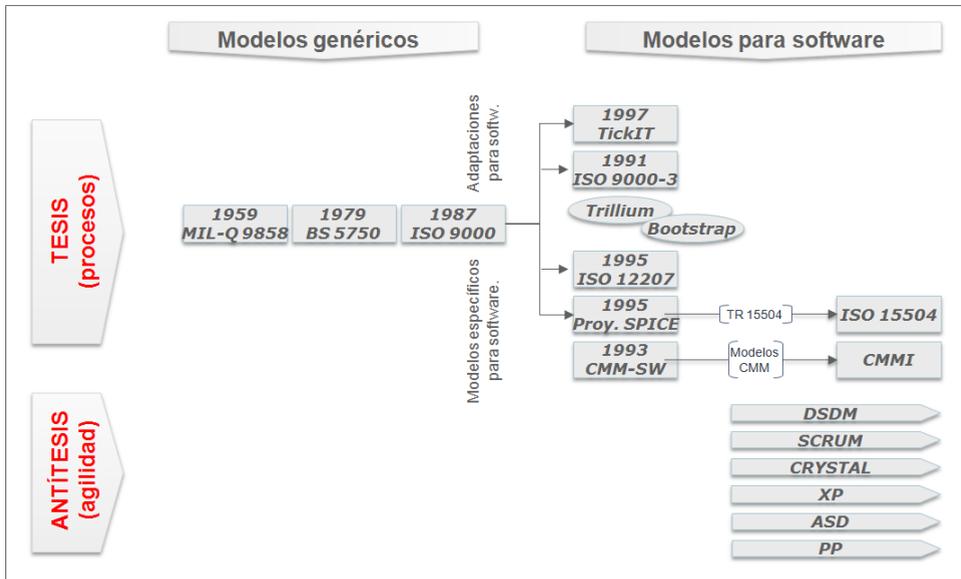


Ilustración 18: Evolución de los procesos y prácticas para desarrollo de software.

Desde los orígenes de la agilidad, mediados de los 90, hasta 2005-2010 han sido habituales las posturas radicales entre los defensores de los modelos de procesos y de los marcos ágiles, posiblemente más enfocados en descalificar al otro que en revisar y depurar los propios métodos.

Algunos ejemplos de esta tensión:

"La diferencia entre un atracador de bancos y un teórico de CMM es que con el atracador se puede negociar"...

"La evaluación en CMM depende más de una buena presentación en papel que da la calidad real del producto de software. Tiene que ver más con el seguimiento a ciegas de una metodología que con el desarrollo y puesta en producción de un sistema en el panorama tecnológico".

"Si uno pregunta a un ingeniero de software típico si cree que CMM se puede aplicar a los métodos ágiles, responderá o con una mirada de sorpresa o con una carcajada histérica".

Espiral dialéctica del conocimiento.

El conocimiento profesional evoluciona de forma continua porque la realidad en la que se aplica está en permanente movimiento, y también porque siempre es posible mejorar los métodos actuales.

La aplicación de nuevas técnicas, procesos o modelos genera sus propias antítesis que revelan las debilidades, contradicciones y puntos de mejora que conducen el avance hacia una síntesis, que pasará a ser una nueva tesis que con su uso generará su antítesis, desarrollando a través de este patrón dialéctico una espiral de evolución continua del conocimiento

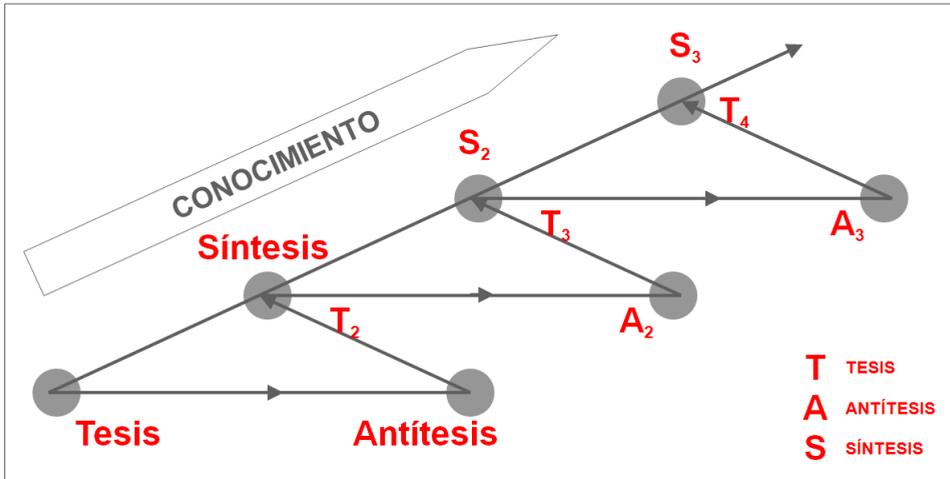


Ilustración 19: Espiral dialéctica del conocimiento

En disciplinas no técnicas y en generaciones anteriores el ritmo de avance sobre esta espiral dialéctica permitía a los profesionales desempeñarse con los conocimientos adquiridos en su licenciatura durante toda su carrera profesional. Sin embargo hoy esto no es posible, especialmente en el sector TIC

Metodologías

Coordenadas del mapa de metodologías.

Desde los 80 se han desarrollado tantos modelos de procesos, marcos y prácticas de trabajo para mejorar la calidad y eficiencia en los proyectos de software, que resulta útil trascender las etiquetas y llegar a la base de los principios que subyacen, y las estrategias con las que los desarrollan; de forma que con tres conceptos (desarrollo, trabajo y conocimiento) y dos modelos de gestión (predictiva y evolutiva) se despeja y simplifica el

aparente laberinto de modelos de procesos, marcos o prácticas de trabajo a los que nos referimos: CMM-SW, CMMI, PMBOK, DSDM, Crystal, ISO 15504, RUP, XP, Scrum, ITIL, ASD, PRINCE 2, LEAN, KANBAN, TDD, etc..

Las diferentes prácticas y metodologías responden a combinaciones de tres conceptos y dos patrones de gestión de proyectos.

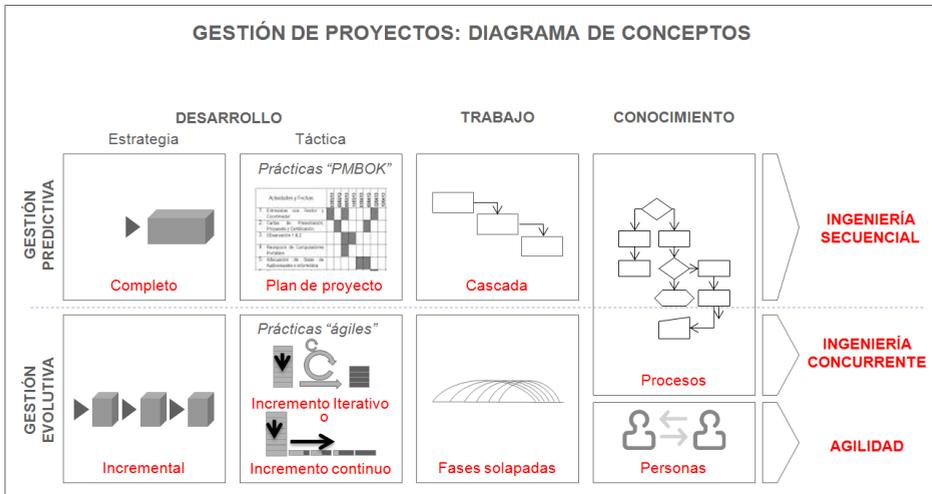


Ilustración 20: Gestión de proyectos: diagrama de conceptos.

Conceptos

1.- Desarrollo

Completo: La descripción de lo que se desea obtener está disponible al inicio del proyecto, es completa y detallada, sirve de base para estimar el plan del proyecto: tareas, recursos y agenda de trabajo. Durante la ejecución se gestiona su cumplimiento.

Incremental: La descripción de lo que se desea obtener no está disponible de forma completa y detallada al inicio: se complementa y evoluciona en paralelo al desarrollo, que genera el resultado de forma incremental y que se puede gestionar con dos tácticas diferentes:

Desarrollo incremental continuo: Empleando técnicas para lograr y mantener un flujo continuo de desarrollo de funcionalidades o partes del producto que entrega de forma continua al cliente.

Desarrollo iterativo: El marco de producción emplea técnicas de tiempo prefijado o timeboxing para mantener la producción de incrementos del producto de forma cíclica y continua. Este es el marco de producción empleado en scrum estándar, que define como sprint a cada iteración de desarrollo al final de la cual se produce un incremento del producto.

2.- Trabajo

Secuencial (cascada): Secuencia las tareas en fases, cada una de las cuales comienza al terminar la anterior y con el resultado que se ha obtenido en ella. El ejemplo más habitual es el ciclo de cascada definido en Ingeniería del software con las fases de requisitos, análisis, diseño, codificación, pruebas e implementación.

Concurrente: Solapa en el tiempo los diferentes tipos de tareas. Siguiendo con el ejemplo de ingeniería de software, la definición de requisitos, el análisis, la codificación y el despliegue del resultado se realiza y revisa de forma simultánea y continua.

3.- Conocimiento

Principal conocimiento empleado, protagonista de la calidad del resultado.

El conocimiento o know-how protagonista de la calidad del resultado se encuentra en mayor medida en los procesos y la tecnología empleada. “La calidad del resultado depende de la calidad de los procesos empleados“.

El conocimiento o know-how protagonista de la calidad del resultado se encuentra en mayor medida en el conocimiento tácito de las personas que lo construyen.

Patrones de gestión de proyectos

Gestión predictiva

Modelo de gestión de proyectos cuyo objetivo es ofrecer resultados predecibles: desarrollo del producto previsto en el tiempo previsto e invirtiendo los recursos previstos. Emplea una estrategia de desarrollo completo con prácticas de planificación tradicional los principales referentes en el desarrollo de conocimiento para este tipo de gestión son PMI e IPMA y los modelos desarrollados (CMMI, ISO 15504, SPICE entre otros) emplean ingeniería secuencial y producción basada en procesos.

Gestión evolutiva o ágil

Modelo de gestión de proyectos cuyo objetivo es la entrega en el menor tiempo posible un producto mínimo viable, e incrementar su valor de forma iterativa y continua. Emplea una estrategia de desarrollo incremental, que puede obtener con tácticas iterativas o de mantenimiento de flujo continuo, y un modelo de trabajo de fases solapadas. Puede emplearse con producción basada en procesos (ingeniería concurrente) o con producción basada en personas (agilidad).

Es importante esta distinción porque sin ella se generan situaciones confusas que llegan a considerar agilidad a la simple aplicación de un marco de desarrollo estándar de scrum (ciclo de incremento iterativo con roles y artefactos definidos), o al simple uso de técnicas de gestión visual kanban para mantener un flujo continuo de tareas.

Personas, Procesos y Tecnología

Scrum Manager reconsidera dos vértices del triángulo clásico de los factores de producción: Personas - Procesos y Tecnología. El de procesos y el de personas.

Procesos

A fin de poder diferenciar distintos tipos de ~~procesos~~ procedimientos podemos decir que en unos, las personas ayudan al proceso, y en otros son ~~los procesos~~ las prácticas las que ayudan a las personas. En el primer caso el proceso es el protagonista, el que sabe cómo hacer el trabajo, y la persona se integra en el sistema como instrumento, como operario de apoyo. En el segundo, el artífice es la persona y ~~el proceso~~ la práctica una ayuda, una herramienta que simplifica aspectos rutinarios para que pueda lograr más eficiencia y no diluir el esfuerzo en rutinas mecánicas.

La principal diferencia entre unos y otros es el tipo de conocimiento con el que trabajan.

El conocimiento pueden ser:

- *Explícito: contenido en los procesos y la tecnología*
- *Tácito: que es contenido por la persona*

Scrum Manager aporta una consideración sobre el triángulo tradicional personas-procesos-tecnología, considerando que los procedimientos de trabajo pueden ser:

- *procesos: cuando el conocimiento clave para lograr el resultado está en su mayor parte “explicitado” en el proceso y la tecnología empleada.*

- *prácticas: cuando los procedimientos ayudan a las personas, que son quienes tienen el conocimiento clave, y por tanto operan en sistemas aportando su conocimiento tácito.*

Se puede decir que en los primeros la persona ayuda al procedimiento, y en los segundos es el procedimiento el que ayuda a la persona.



Ilustración 21: Personas, procedimientos y tecnología.

Los modelos de gestión y mejora basados en procesos, centran el foco de la producción en los procesos y la tecnología, como los elementos más valiosos de la producción. Su antítesis, las prácticas ágiles, focalizan el valor de la producción en las personas.

Desde el punto de vista de Scrum Manager, ambas opciones son posibles, pero cada una para un entorno diferente. En entornos de producción industrial que centran el conocimiento en los procesos y tecnología empleada, las personas aportan trabajo para auxiliar a los procesos y la tecnología, y el valor del resultado depende principalmente de los primeros. Sin embargo para las empresas del conocimiento que trabajan en escenarios rápidos e innovadores, el principal valor es el talento de las personas.

Personas

Las organizaciones que necesitan imprimir un componente innovador importante y frecuente, o se mueven en sectores de innovación muy rápido, obtienen mejores resultados si hacen responsables de esa innovación al talento de las personas más que a la ejecución de procesos.

En este tipo de organizaciones es importante asegurar, además del nivel de creatividad del equipo, su capacidad para aprehender. El modelo de conversión del conocimiento definido por Nonaka y Takeuchi define con sus 4 fases el proceso para la adquisición de las personas del conocimiento tácito a través de compartir experiencias, comunicación directa, documentos, manuales y tradiciones, que añade conocimiento novedoso a la base colectiva de la organización.

MÓDULO 2

AGILIDAD

Agilidad

La gestión ágil surgió como antítesis a un cierto modelo de gestión al que haremos referencia con frecuencia en este manual: la gestión de proyectos predictiva. Ambas tienen sus virtudes y resultan más útiles en ciertas industrias. La predictiva se centra en planificar, en calcular un presupuesto y marcar plazos de entrega. Si el proyecto final se termina en la fecha acordada, sin exceder el coste y con todas las funcionalidades del plan inicial, se considera un éxito.

Por muy razonable que suene esta estrategia, si trabajamos en industrias que se caractericen por su constante y rápida evolución, encontraremos que tiene muchos inconvenientes. Esa definición de un proyecto exitoso sirve en un entorno estable, con productos que son resultado de una atención escrupulosa a procesos y protocolos.

La gestión predictiva es fruto de la Revolución Industrial: viene del mundo de la construcción, de la automovilística, de las fábricas. Por ejemplo: si lo que el cliente busca es una casa, ésta tendrá que construirse de forma que sea sólida, segura, cumpla con las necesidades de sus habitantes; y, en un escenario ideal, dentro del plazo previsto y sin exceder el coste.

Pero hoy en día se fabrican y venden productos que no tienen nada que ver. Primero, porque pueden ser abstractos, como una película o una *app* móvil. Se pueden probar cosas nuevas durante el desarrollo, viendo de forma empírica lo que funciona y lo que no. Es posible realizar ajustes sobre la marcha. Además, se puede partir de un primer esbozo con lo básico que se necesita e ir creciendo. El escenario puede cambiar, y una funcionalidad que parecía esencial al principio puede estar desfasada para la fecha de entrega. O puede ser que un competidor lance una novedad interesante y eso lleve a revisar las prioridades del producto. Para ser competitivo se necesita la capacidad de responder rápido en escenarios de trabajo inciertos; donde no se cuenta con requisitos estables al concebir nuevos productos o servicios; con clientes que necesitan empezar a usar el producto lo antes posible y mejorarlo de forma continua; productos en los que la innovación es un valor clave.

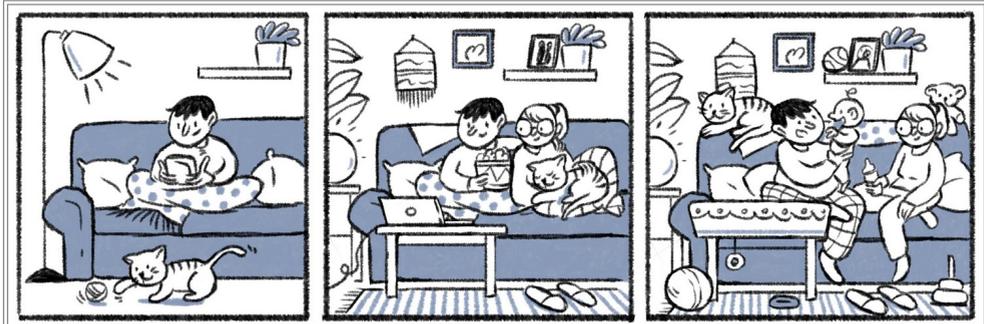


Ilustración 22: Adaptación a las circunstancias desde un mínimo viable.

Éstas y más razones que veremos son las que llevaron a cuestionar los modelos de gestión predictiva, que parecían no encajar con la realidad de lo que se necesitaba en las llamadas «empresas del conocimiento». Entendiendo como tales aquellas que desarrollan productos o servicios basándose en el conocimiento más que en las herramientas y los procesos.

El entorno de trabajo de estas empresas se parece muy poco al que originó la gestión de proyectos predictiva.

Ahora existen mercados con una evolución tan rápida que es inútil pretender iniciar proyectos con un plan cerrado. Se necesitan estrategias que entreguen resultados tangibles y pronto; que permitan responder a tiempo a los cambios. Se construye el producto al mismo tiempo que se modifican e introducen nuevos requisitos. El cliente parte de una visión más o menos clara, pero el nivel de innovación que requiere, así como la velocidad a la que se mueve el entorno de su negocio, no le permiten prever con detalle cómo será el resultado final.

Hoy hay directores de producto que no necesitan conocer cuáles van a ser las 200 funcionalidades que tendrá el producto final, ni si estará terminado en 12 o en 16 meses. Hay clientes que necesitan disponer de una primera versión con funcionalidades mínimas en cuestión de semanas, y no un producto completo dentro de uno o dos años. Su interés es poner en el mercado rápidamente un concepto nuevo y desarrollar de forma continua su valor.

De dónde venimos y por qué la agilidad se suele asociar con la informática

El conocimiento evoluciona siguiendo un patrón dialéctico de tesis, antítesis y síntesis. A cada tesis le surge una antítesis, que pone en evidencia sus problemas y contradicciones. La antítesis también resulta inadecuada de alguna manera, y de la confrontación de las dos surge un tercer momento llamado síntesis: una resolución y nueva comprensión del problema.

«Es en este momento que la tesis y antítesis anteriores se reconcilian y trascienden. No obstante, con el tiempo, incluso la síntesis resultará tendenciosa en algún aspecto. Servirá entonces como tesis para un nuevo movimiento dialéctico, y así el proceso continúa de forma zigzagueante en espiral.» (Nonaka 2004)

Los marcos de prácticas ágiles no surgieron como una tesis de conocimiento, sino como antítesis al que la ingeniería del *software* venía desarrollando.

Procesos y gestión predictiva

En 1968, durante la llamada «crisis del software», la organización OTAN celebró la primera conferencia centrada en analizar los problemas de la programación. Se puso de manifiesto la necesidad de crear una disciplina científica que permitiera aplicar un enfoque sistemático y cuantificable al desarrollo, operación y mantenimiento de sistemas informáticos. Esto se tradujo en el intento de aplicar ingeniería de procesos al software, surgiendo así la «ingeniería del software» (Bau 1969). Esta primera estrategia (tesis) se basó en dos pilares:

- Ingeniería de procesos: en los entornos de producción industrial existía un principio básico de calidad, contrastado con éxito: «la calidad del resultado depende de la calidad de los procesos empleados». Dicho de otra forma: no se necesita a gente brillante o muy cualificada; mientras los procesos empleados sean de calidad, el resultado será de calidad.
- Gestión predictiva: un tipo de gestión que se centra en garantizar el cumplimiento de agendas y presupuestos.

Mientras la disciplina evolucionaba y se perfeccionaba a través de diferentes modelos de procesos y cuerpos de conocimiento para gestión de proyectos (MIL-Q9858, ISO9000, ISO9000-3, ISO 12207, SPICE, SW-CMM...) en la industria del software surgían dudas y se cuestionaba esta estrategia.

Desde mediados de los 90 hasta 2005-2010 han sido habituales las posturas radicales entre los defensores de los modelos de procesos (tesis) y de los marcos ágiles (antítesis).

«La diferencia entre un atracador de bancos y un teórico de CMM es que con el atracador se puede negociar.» (Orr 2002)

«La evaluación en CMM depende más de una buena presentación en papel que de la calidad real del producto de software. Tiene que ver más con el seguimiento a ciegas de una metodología que con el desarrollo y puesta en producción de un sistema en el panorama tecnológico.» (Orr 2002)

«Si uno pregunta a un ingeniero de software típico si cree que CMM se puede aplicar a los métodos ágiles, responderá o con una mirada de sorpresa o con una carcajada histérica.» (Turner & Jain 2002)

No estaba claro que la planificación predictiva fuese apropiada para cualquier proyecto. En la práctica puede verse que a veces los criterios del éxito no son siempre el cumplimiento de fechas, costes y funcionalidades preestablecidas. Por otra parte, también se cuestiona si en el desarrollo de software, como en otros trabajos basados en el conocimiento, se puede producir con patrones de procesos industriales. Se empieza a aceptar que el conocimiento tácito de la persona que realiza el trabajo puede aportar más al valor del resultado que la tecnología y los procesos empleados (→ Desmontando la gestión de proyectos, Conocimiento).

El Manifiesto Ágil

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

- A los individuos y su interacción, por encima de los procesos y las herramientas.
- El software que funciona, por encima de la documentación exhaustiva.
- La colaboración con el cliente, por encima de la negociación contractual.
- La respuesta al cambio, por encima del seguimiento de un plan.

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

En marzo de 2001, 17 profesionales del software fueron convocados por Kent Beck, que había publicado un par de años antes el libro en el que explicaba la nueva metodología *Extreme Programming* (Beck 2000). Todos ellos tenían algo en común: eran críticos de los modelos de producción basados en procesos.

Se reunieron en Salt Lake City para discutir sobre los procesos empleados por los equipos de programación.

En la reunión se acuñó el término «métodos ágiles» para definir a aquellos que estaban surgiendo como alternativa a las metodologías formales, tales como CMM-SW (precursor de CMMI), PMI, SPICE (proyecto inicial de ISO 15504)... a los que consideraban excesivamente pesados y rígidos por su carácter normativo y fuerte dependencia de planificaciones detalladas previas al desarrollo.

Los integrantes de la reunión resumieron en cuatro postulados lo que ha quedado denominado como Manifiesto Ágil, que son los valores sobre los que se asientan estos métodos. Son los que abren y se desarrollan en este apartado. También establecieron 12 principios, que mencionaremos al final.

«Valoramos más a los individuos y su interacción que a los procesos y las herramientas.»

El postulado más importante. Es indudable que los procesos ayudan: sirven de guía de operación, y disponer de las herramientas adecuadas mejora la eficiencia. Pero hay tareas que requieren talento y necesitan personas motivadas que lo aporten.



En una producción basada en procesos, lo que se persigue es que la calidad del resultado sea consecuencia de éstos, más que del conocimiento aportado por las personas que los ejecutan. En el desarrollo ágil, en cambio, los procesos son sólo una ayuda; un soporte para guiar el trabajo.

La defensa a ultranza de los procesos lleva a afirmar que con ellos se pueden conseguir resultados extraordinarios con personas mediocres, pero lo cierto es que esto no es así cuando se necesita creatividad e innovación.

«Valoramos el software que funciona, por encima de la documentación exhaustiva.»

El Manifiesto Ágil no considera inútil la documentación: sólo la innecesaria. Los documentos son un soporte físico que permite registrar y comunicar información relevante para el proyecto. Además, por cuestiones legales o normativas, pueden ser obligatorios. Pero su relevancia debe ser menor que la del producto.

¿A qué nos referimos? Poder anticipar cómo funcionará el producto final observando prototipos y partes ya terminadas supone un *feedback* estimulante y enriquecedor, que genera ideas imposibles de concebir en un primer momento. Es por eso que elaborar un documento de requisitos muy detallado antes de empezar supone a menudo una pérdida de tiempo.

Se puede argumentar que la documentación detallada facilita transmitir información entre las personas implicadas en el proyecto, pero rara vez es así. Le falta la riqueza y producción de valor que se logra con la comunicación directa y la interacción con prototipos del producto. De hecho, no sólo carece de esas ventajas, sino que entorpece y crea barreras burocráticas entre departamentos e individuos.



Ilustración 24: Valoramos el software que funciona por encima de la documentación exhaustiva.

Por eso, siempre que sea posible, se debe reducir al mínimo indispensable el uso de documentación. Lo ideal es eliminar toda aquella que consuma trabajo sin aportar un valor directo al producto.

«Valoramos más la colaboración con el cliente que la negociación contractual.»

El objetivo de un proyecto ágil no es controlar la ejecución para garantizar que los planes iniciales se cumplen, sino proporcionar de forma continua el mayor valor posible al producto.

Como ya hemos comentado, al desarrollar productos en evolución continua (como una aplicación web, por ejemplo) no se puede definir en un documento de requisitos cerrado cómo debería ser el producto final. Es más eficiente tomar feedback directo a la vez que se desarrolla el producto, y en consecuencia redefinir y mejorar los requisitos de las partes que quedan.



Para que el cliente sea consciente de los cambios conforme van surgiendo, debe acompañar al equipo durante el proceso. La mejor relación entre cliente y equipo es una de implicación y colaboración directa, no una contractual, que tiende a delimitar responsabilidades al principio del proyecto y ya está.

«Valoramos más la respuesta al cambio que el seguimiento de un plan.»

Los principales valores de la gestión ágil son la anticipación y la adaptación, diferentes a los de la gestión de proyectos ortodoxa: planificación y control para garantizar el cumplimiento del plan.

Para desarrollar productos de requisitos inestables, en los que es inherente el cambio y la evolución rápida y continua, resulta mucho más valiosa la capacidad de respuesta que la de seguimiento y aseguramiento de planes.

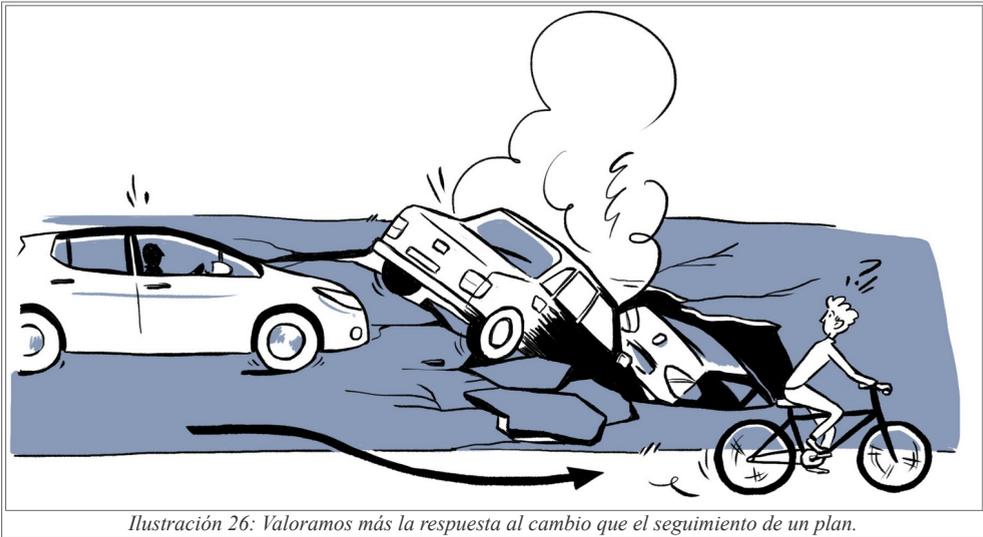


Ilustración 26: Valoramos más la respuesta al cambio que el seguimiento de un plan.

Los 12 principios del Manifiesto Ágil

Además de los cuatro postulados que acabamos de ver, el Manifiesto Ágil establece estos 12 principios:

1. Nuestra principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software de valor.
2. Son bienvenidos los requisitos cambiantes, incluso si llegan tarde al desarrollo. Los procesos ágiles se dobligan al cambio como ventaja competitiva para el cliente.
3. Entregar con frecuencia software que funcione, en periodos de un par de semanas hasta un par de meses, con preferencia en los periodos breves.
4. Las personas del negocio y los desarrolladores deben trabajar juntos de forma cotidiana a través del proyecto.
5. Construcción de proyectos en torno a individuos motivados, dándoles la oportunidad y el respaldo que necesitan y procurándoles confianza para que realicen la tarea.
6. La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
7. El software que funciona es la principal medida del progreso.
8. Los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica enaltece la agilidad.
10. La simplicidad como arte de maximizar la cantidad de trabajo que no se hace, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos que se autoorganizan.
12. En intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo y ajusta su conducta en consecuencia.

Origen de scrum

Scrum es un modelo de desarrollo ágil caracterizado por:

- Equipos autónomos y autoorganizados que comparten su conocimiento de forma abierta y aprenden juntos.
- Una estrategia de desarrollo incremental, en lugar de la planificación completa del producto.
- Basar la calidad del resultado en el conocimiento tácito de las personas y su creatividad; no en la calidad de los procesos empleados.
- Solapar las diferentes fases del desarrollo, en lugar de realizarlas una tras otra en un ciclo secuencial o «de cascada».

El origen de la palabra se encuentra en un ámbito muy alejado del de la gestión de proyectos: en el deporte. En rugby, «scrum» es el término que define a la formación en la que ambos equipos, agazapados y atenzados entre sí, empujan para obtener el balón sin tocarlo con la mano.

Ahora bien, para lo que a nosotros nos interesa, tenemos que desplazarnos al Japón de los 80, cuando los investigadores Ikujiro Nonaka y Hirotaka Takeuchi dieron una dimensión polisémica al término.

Identificaron una novedosa forma de desarrollo en las empresas de manufactura industrial que estaban obteniendo los mejores resultados de innovación y tiempo de salida al mercado: Fuji Xerox, Canon, Honda, Nec, Epson, Brother, 3M y Hewlett-Packard (Nonaka 1986). Compararon su forma de trabajo en equipos autoorganizados con el avance en formación de los jugadores de rugby, de ahí el término.

Aunque esta forma de trabajo surgió en empresas de productos tecnológicos, en la manufactura industrial, a partir de 1995 se empezaron a aplicar también a la industria del software. En este año, Ken Schwaber presentó en OOPSLA (conferencia anual Object-Oriented Programming, Systems, Languages & Applications) una metodología de desarrollo de software basada en un ambiente scrum, usando ese mismo término (Schwaber 1995). Este primer marco presentaba una serie de fases y «artefactos»: *pregame*, *game*, *postgame*, *planning*, *sprints*,

wrap... Algunos se han mantenido y los veremos, pero en general las reglas del juego han cambiado mucho desde entonces.

No existe una autoridad que determine lo que es scrum y lo que no. Ha cambiado y seguirá evolucionando con las aportaciones de la comunidad de profesionales, que define las prácticas que resultan más útiles. El espíritu original, eso sí, se mantiene: las prácticas deben ayudar a equipos a autoorganizarse y mantener un flujo de avance continuo, produciendo resultados de forma iterativa y frecuente. Es lo bonito y lo emocionante de trabajar en este tipo de empresas.

Entre los eventos y prácticas que se han ido incorporando se incluyen las reuniones retrospectivas, las reuniones de refinamiento de la pila de producto, *DoR* (*Definition of Ready*), *story maps*...

Scrum Manager® usa el término «scrum» entendido con el significado original, el que le dieron Nonaka y Takeuchi.

Desmontando la gestión de proyectos

Desde los 80 se han desarrollado tantos modelos y prácticas para mejorar la calidad y eficiencia de los proyectos¹ que puede resultar abrumador. En este apartado vamos a trascender las etiquetas y a resumir la base de estos marcos, los principios que subyacen y las estrategias con las que se desarrollan.

Usaremos como coordenadas tres conceptos y dos modelos de gestión.

- Los tres primeros son: desarrollo, trabajo y conocimiento.
- Los modelos, que ya han sido mencionados a lo largo del manual, son: gestión predictiva y gestión evolutiva.

Con estas cinco ideas se despeja y simplifica el aparente laberinto de modelos de marcos de trabajo, pues al final todos ellos se pueden entender dentro de este diagrama.

GESTIÓN DE PROYECTOS: DIAGRAMA DE CONCEPTOS

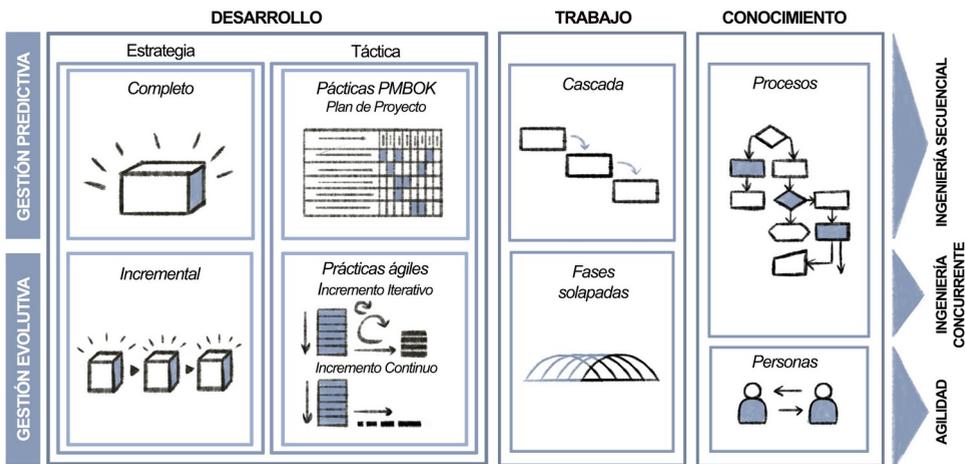


Ilustración 27: Diagrama de conceptos en la gestión de proyectos.

¹ En desarrollo de software, por ejemplo: CMMI, ISO 15504, RUP, XP, ITIL, ASD, PRINCE, LEAN, KANBAN, TDD... y scrum.

1. Desarrollo

El desarrollo del proyecto puede darse de forma completa o incremental.

- En el caso de un desarrollo completo, la descripción de lo que se desea obtener está disponible al inicio del proyecto; es completa y detallada y sirve de base para estimar. Con el plan inicial se organizan tareas, recursos y agenda de trabajo. Durante la ejecución se gestiona el cumplimiento de lo que se ha previsto.
- En los desarrollos incrementales, la descripción completa de lo que se desea obtener no está disponible al inicio. Se complementa y evoluciona durante el desarrollo, que se puede gestionar con dos tácticas diferentes:
 - Desarrollo incremental continuo: Empleando técnicas para lograr un flujo continuo de desarrollo de las funcionalidades o partes del producto, que se entregan de forma continua al cliente.
 - Desarrollo iterativo: Empleando técnicas de tiempo prefijado o «*timeboxing*» para mantener la producción de incrementos del producto a un ritmo fijo. Este es el marco de producción empleado al aplicar el marco estándar de scrum, que define como sprint (→ Eventos) a cada iteración de desarrollo, al final de la cual se produce un «incremento» del producto: una parte entregable y lista para usarse.

2. Trabajo

La forma de trabajar puede ser secuencial («en cascada») o concurrente.

- El trabajo secuencial divide el trabajo en fases. Una fase nueva comienza cuando se termina la anterior. El ejemplo más habitual es el ciclo de cascada definido en ingeniería del software, cuyas fases son: definición de requisitos, análisis, diseño, codificación, pruebas e implementación.
- Trabajar de forma concurrente significa solapar en el tiempo las diferentes fases. Siguiendo con el ejemplo de ingeniería de software, todas las fases del párrafo anterior se revisarían de forma simultánea y continua.

3. Conocimiento

Los diferentes modelos pueden ubicar el conocimiento o bien en los procesos o en las personas.

- En una producción basada en procesos: el conocimiento es explícito. La calidad del resultado se encuentra, en mayor medida, en los procesos y la tecnología empleada.
- En la producción basada en las personas: el conocimiento es tácito. La calidad del resultado depende de la experiencia de los miembros de la organización. No de seguir un proceso de manera correcta, sino de que quienes trabajen sean personas motivadas y con talento.

Un ejemplo de conocimiento explícito y tácito sería la diferencia entre la cena preparada por un robot de cocina o de forma libre. Cualquiera puede preparar la primera siguiendo las instrucciones.

El resultado siempre será el mismo sin importar la habilidad. En el segundo caso, prima el talento de la persona. Un aficionado no preparará lo mismo que un chef de alta cocina. Ambos se beneficiarán de tener buenas herramientas, como sartenes que no se peguen o cuchillos bien afilados, pero son sólo una ayuda.



Ilustración 28: Conocimiento tácito vs. explícito.

«El conocimiento tácito es personal, específico del contexto, y por tanto difícil de formalizar y comunicar. El conocimiento explícito o «codificado», por otro lado, es conocimiento que puede transmitirse con lenguaje formal y sistemático.» (Nonaka 1995)

Ingeniería secuencial

La ingeniería secuencial, también llamada gestión predictiva, tiene como objetivo ofrecer resultados predecibles. Un proyecto exitoso según estos modelos desarrollará el producto previsto sin exceder el plazo ni los recursos acordados. El tipo de desarrollo de la gestión predictiva es «completo», y se emplean prácticas de planificación tradicional.

En el mundo del software los principales referentes en el desarrollo de conocimiento para este tipo de gestión son PMI e IPMA y los modelos de procesos CMMI, ISO 15504, SPICE... Todos ellos emplean ingeniería secuencial y producción basada en procesos.

Gestión evolutiva: ingeniería concurrente y agilidad

La gestión evolutiva tiene como objetivo entregar lo antes posible un producto mínimo viable, e incrementar su valor de forma continua. Emplea una estrategia de solapamiento de las fases de trabajo y desarrollo incremental, que se puede obtener manteniendo un ritmo de iteraciones breves y cíclicas o un flujo de desarrollo constante.

Puede llevarse a cabo con producción basada en procesos (ingeniería concurrente) o con producción basada en personas (agilidad). Es importante esta distinción, porque sin ella se generan situaciones confusas que llegan a considerar agilidad a la simple aplicación de las reglas estándar de scrum (ciclo de incremento iterativo con roles y artefactos definidos), o al simple uso de técnicas de gestión visual *kanban* para mantener un flujo continuo de tareas.

Ingeniería concurrente	Agilidad
Emplea recursos propios de la gestión ágil: solapamiento de fases de desarrollo, equipos multidisciplinares e iteraciones frecuentes de mejora. Se centra en la calidad de los procesos.	Reduce o elimina tareas administrativo-burocráticas que no aportan valor al producto o al sistema de desarrollo. Propia de las empresas del conocimiento. Se centra en el conocimiento tácito de las personas, en la cultura y el talento.

Scrum

Recapitulando, volvemos a encontrarnos con las características de scrum (→ Origen de scrum) y vemos cómo encajan dentro de las características de la gestión ágil:

- Utiliza una estrategia de desarrollo incremental (que puede ser iterativo, usando «timeboxing», o continuo).

- Solapa las diferentes fases del desarrollo.
- Basa la calidad del resultado en el conocimiento tácito de las personas y su creatividad.
- Además, scrum se caracteriza también por el trabajo en equipos autónomos y autoorganizados, que comparten su conocimiento y aprenden juntos. De ahí el nombre y la metáfora de «avanzar en scrum».

Diferenciando las prácticas de los principios y valores scrum

Cuando se empieza a trabajar con scrum, como con cualquier otra herramienta, es recomendable leer el manual y seguir las instrucciones; es decir, adoptar el marco estándar: el que se explica en la primera parte de este manual, con los roles, artefactos y eventos que lo configuran.

Conviene no engañarse: si el foco está puesto en el proceso y no en el conocimiento tácito de las personas no se está haciendo agilidad, sino ingeniería concurrente. Cuando se alcanza un flujo de avance iterativo, se puede intentar ir más allá. Llega el momento de desaprender las prácticas y de apoyarse en los principios y valores de scrum, adaptando éste y otras técnicas y marcos a las características concretas del proyecto o del equipo. En la mayoría de empresas ágiles estas prácticas se pueden adaptar, y de hecho se adaptan.

La primera parte del libro desarrolla las técnicas más extendidas de scrum, el marco estándar que puede encontrarse aquí, en Internet y en otros manuales: las reglas de aplicación, roles, eventos y artefactos. En la segunda parte explicaremos cómo quitar esos «ruedines» de la bici, que vienen muy bien al principio pero pueden entorpecernos a la larga, para seguir avanzando.

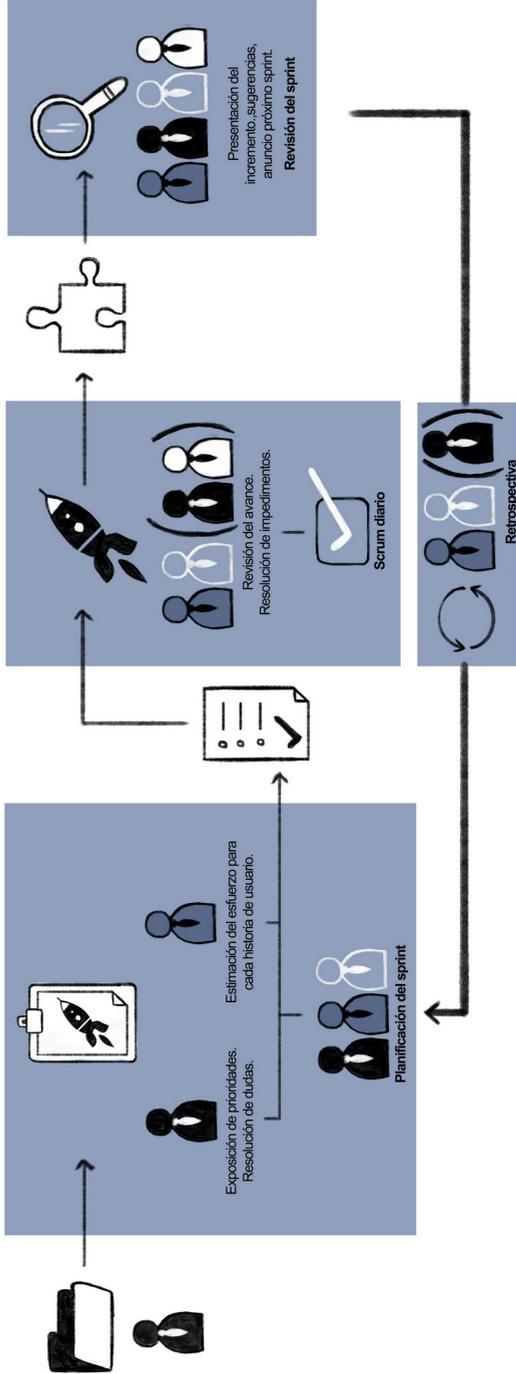


Primera parte: el ciclo scrum

APRENDIENDO LAS PRÁCTICAS ESTÁNDAR



LAS REGLAS DE SCRUM



ROLES

- PROPIETARIO DEL PRODUCTO**
Determina las prioridades. Una sola persona.
- EQUIPO DE DESARROLLO**
Constituye el producto.
- SCRUM MASTER**
Gestiona y facilita la ejecución de las reglas del Scrum.
- INTERESADOS**
Riesgo de implicados. Asesoran y observan.

ARTEFACTOS

- PILA DEL PRODUCTO**
Relación de requisitos del producto, no es necesario excesivo detalle. Priorizados. Lista en evolución y abierta a todos los roles. El propietario del producto es su responsable y quien decide.
- PILA DEL SPRINT**
Requisitos comprometidos por el equipo para el sprint con nivel de detalle suficiente para su ejecución.
- INCREMENTO**
Parte del producto desarmolada en un sprint, en condiciones de ser usada (pruebas, codificación limpia y documentada).

EVENTOS

- PLANIFICACIÓN DEL SPRINT**
1 jornada de trabajo (máx.). El propietario del producto explica las prioridades. El equipo estima el esfuerzo de los requisitos prioritarios y se elabora la pila del sprint. El equipo define en una frase el objetivo del sprint.
- SCRUM**
Ciclo de desarrollo básico en el marco estándar de scrum, de duración recomendada inferior a un mes y nunca mayor de 6 semanas.
- SCRUM DIARIO**
15 minutos máximo. Responsabilidad del equipo, cada miembro expone: Lo que hizo ayer, lo que va a hacer hoy y si tiene o prevé problemas. Se actualiza la pila del sprint.
- REVISIÓN DEL SPRINT**
Informativa, máximo 4 horas. Presentación del incremento, planteamiento de sugerencias y anuncio del próximo sprint.
- RETROSPECTIVA**
El equipo automatiza la forma de trabajo. Identificación de fortalezas y debilidades. Refuerzo de las primeras, plan de mejora de las segundas.

El ciclo scrum

A fecha de la publicación de este manual, los componentes del ciclo estándar de scrum son:

- Roles:
 - Equipo.
 - Propietario del producto.
 - Scrum master.
- Artefactos:
 - Pila del producto.
 - Pila del sprint.
 - Incremento.
- Eventos:
 - Sprint.
 - Reunión de planificación del sprint.
 - Scrum diario.
 - Revisión del sprint.
 - Retrospectiva del sprint.

Se comienza con la visión general del resultado que se desea, y a partir de ella se especifica y da detalle a las funcionalidades que se desean obtener en primer lugar.

Cada ciclo de desarrollo o iteración (sprint) finaliza con la entrega de una parte operativa del producto (incremento). La duración de cada sprint puede ser de una hasta seis semanas, aunque se recomienda que no exceda de un mes.

En scrum, el equipo monitoriza la evolución de cada sprint en reuniones breves diarias donde se revisa el trabajo realizado por cada miembro el día anterior, y el previsto para el día actual. Estas reuniones son de tiempo cerrado, de 5 a 15 minutos máximo, se realizan de pie junto a un tablero o pizarra con información de las tareas del sprint y el trabajo pendiente en cada una. Se denominan «reunión de pie» o «scrum diario» (en inglés *stand-up meeting*, *daily scrum* o *morning rollcall*).

 : <https://youtu.be/SIZfzLRGYBA>

Scrum maneja empíricamente la evolución del proyecto con las siguientes tácticas:

Revisión de las iteraciones

Al finalizar cada sprint se revisa funcionalmente el resultado, con todos los implicados en el proyecto. Por tanto, la duración del sprint es el período de tiempo máximo para descubrir planteamientos erróneos, mejorables, o malinterpretaciones en las funcionalidades del producto.

Desarrollo incremental

No se trabaja con diseños o abstracciones. El desarrollo incremental ofrece al final de cada iteración una parte de producto operativa, que se puede usar, inspeccionar y evaluar.

Solapamiento de fases

Durante la construcción se depura el diseño y la arquitectura, y no se cierran en una primera fase del proyecto. Las distintas fases que el desarrollo en cascada realiza de forma secuencial, una tras otra, en scrum se solapan y avanzan de forma simultánea.

Autoorganización

La gestión predictiva asigna al rol de gestor del proyecto la responsabilidad de su gestión y resolución. En scrum los equipos son autoorganizados, con un ámbito de decisión suficiente para adoptar las resoluciones que consideren oportunas. Esto agiliza la toma de decisiones y permite responder con rapidez ante imprevistos.

Colaboración

Todos los miembros del equipo colaboran de forma abierta con los demás, según sus capacidades y no según su rol o su puesto.

Mediante la autoorganización y la colaboración se puede gestionar con solvencia la labor que de otra forma realizaría un gestor de proyectos.

Roles

Todas las personas implicadas con el proyecto tienen, según sus roles, diferentes niveles de compromiso y responsabilidad. En función de esto se suele diferenciar entre roles «comprometidos» e «implicados».

Comprometidos:

- Propietario del producto: es la persona responsable de lograr el mayor valor de producto para los clientes, usuarios y resto de implicados.
- Equipo de desarrollo: grupo o grupos de trabajo que desarrollan el producto.

Implicados:

- Otras partes interesadas: dirección, gerencias, comerciales, *marketing*...
- Scrum master: persona responsable del funcionamiento de scrum como marco de trabajo en la organización.

En círculos de scrum es frecuente llamar a los primeros (sin ninguna connotación peyorativa) «cerdos» y a los segundos «gallinas». El origen de estos nombres está en la siguiente historia, que ilustra de forma gráfica la diferencia entre compromiso e implicación en el proyecto:



Ilustración 29: Roles: comprometidos e implicados.

Una gallina y un cerdo paseaban por la calle. La gallina preguntó al cerdo:

-¿Quieres abrir un restaurante conmigo?

El cerdo consideró la propuesta y respondió:

-Sí, me gustaría. ¿Cómo lo llamaríamos?

-Huevos con Jamón.

El cerdo se detuvo, hizo una pausa y contestó:

-Pensándolo mejor, creo que no voy a abrir un restaurante contigo. Yo estaría realmente comprometido... mientras que tu estarías sólo implicada.

Propietario del producto

El «propietario del producto» o «*product owner*» es quien toma las decisiones del cliente. Su responsabilidad es el valor del producto.

Para simplificar la comunicación y toma de decisiones es necesario que este rol recaiga en una única persona. Si el cliente es una organización grande, o con varios departamentos, puede adoptar la forma de comunicación interna que consideren oportuna, pero en el equipo de trabajo sólo se integra a una persona. Ésta representa al cliente y debe tener el conocimiento y las atribuciones necesarias para tomar las decisiones que le corresponden.

En los desarrollos internos para la propia empresa, suele asumir este rol el *product manager* o el responsable de marketing. En desarrollos para clientes externos, el responsable del proceso de adquisición del cliente. Según las circunstancias del proyecto es posible incluso que el propietario del producto delegue en el equipo, o en alguien de su confianza, pero incluso en esos casos mantiene sus responsabilidades:

- Desarrollo y administración de la pila del producto.
- Exposición de la visión e historias de usuario, y participación en la reunión de planificación de cada sprint (→ Eventos).

Es quien está a cargo de la «pila del producto» (→ Artefactos). Esto quiere decir que es quien decide en última instancia cómo será el producto final y el orden en el que se van construyendo los incrementos; qué se pone y qué se quita, así como cuál es la prioridad de las «historias de usuario» (→ Artefactos). Conoce el plan del producto, sus posibilidades, plan de inversión y el retorno esperado a la inversión realizada. Como representante del cliente, también se responsabiliza de cumplir con los plazos de entrega previstos de las versiones del producto.

Tiene conocimiento experto sobre el entorno de negocio del cliente; sabe cuáles son sus necesidades y el objetivo que se persigue con el proyecto. Esto es fundamental para poder compartir esta visión con el equipo y priorizar requisitos. Debe estar al corriente y realizar análisis constantes del entorno de negocio: evolución del mercado, competencia, alternativas... Y combinar esta información con la que surja del equipo durante el proceso de creación: sugerencias, alternativas

técnicas, pruebas y evaluación de cada incremento. Es necesario que conozca scrum, para realizar con solvencia las tareas que le corresponden. Y, en circunstancias ideales, que conozca y haya trabajado previamente con el mismo equipo. La organización debe respetar sus decisiones y no modificar prioridades ni elementos de la pila del producto.

Equipo

Lo forman el grupo de profesionales que realizan el «incremento» (→ Artefactos) de cada sprint.

Se recomienda que un equipo scrum tenga entre 3 y 9 personas. Más allá de 9 resulta difícil mantener la comunicación directa, y se manifiestan con más intensidad los roces habituales de la dinámica de grupos, que comienzan a aparecer a partir de 6 personas. En el cómputo del número de miembros del equipo no se cuentan ni al scrum master ni al propietario del producto. Es un equipo multifuncional, en el que todos los miembros trabajan de forma solidaria con responsabilidad compartida. Es posible que algunos miembros sean especialistas en áreas concretas, pero la responsabilidad, el incremento, recae sobre el equipo en conjunto. Las principales responsabilidades, más allá de la autoorganización y uso de tecnologías ágiles, son las que marcan la diferencia entre «grupo de trabajo» y «equipo»:

- En un grupo de trabajo las personas tienen una asignación específica de tareas, responsabilidades, y siguen un proceso o pautas de ejecución. Los operarios de una cadena forman un grupo de trabajo; aunque tienen un jefe común, y trabajan en la misma organización, cada uno responde de forma individual.
- Un equipo tiene espíritu de colaboración y un propósito común: conseguir el mayor valor posible para la visión del cliente. Un equipo scrum responde en su conjunto. Trabaja de forma cohesionada y autoorganizada.

No hay un gestor para delimitar, asignar y coordinar las tareas. Son los propios miembros los que se coordinan. En el equipo todos conocen y comprenden la visión del propietario del producto. Todos aportan y colaboran con el propietario del producto en el desarrollo de la pila del producto, participan en la toma de decisiones, y respetan las opiniones y

aportes de los demás. Comparten el objetivo de cada sprint y la responsabilidad del logro. Por último, todos están familiarizados con scrum.

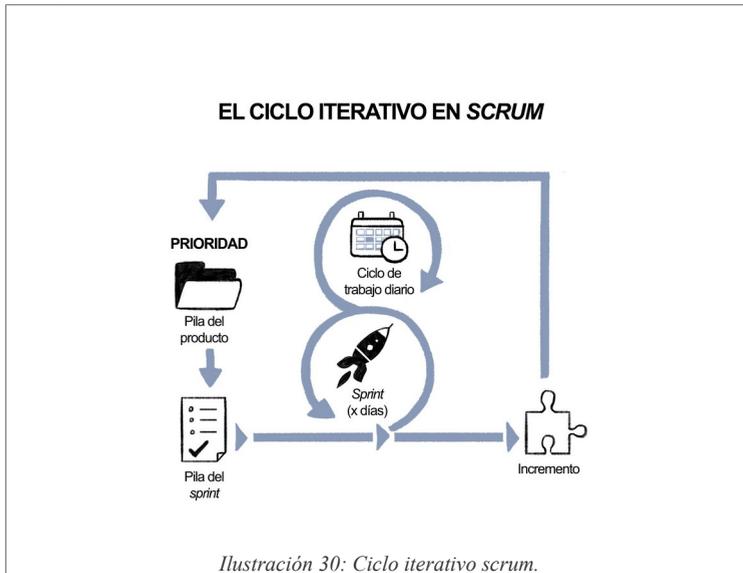
Scrum master

Es el responsable del cumplimiento de las reglas del marco de scrum. Se asegura que éstas son entendidas por la organización y de que se trabaja conforme a ellas. Asesora y da la formación necesaria al propietario del producto y al equipo, y configura, diseña y mejora de forma continua las prácticas ágiles de la organización. El fin es que equipo y cliente sean capaces de organizarse y trabajar con autonomía.

También es responsabilidad suya moderar las reuniones de scrum diarias, gestionar las dificultades de dinámica de grupo que puedan surgir en el equipo, y solucionar los impedimentos detectados durante el scrum diario para que el sprint siga avanzando.

Artefactos

Los artefactos de scrum son sus herramientas, sus bloques de construcción elementales. Ayudan a los «roles» durante los «eventos».



Artefactos más extendidos

Podemos destacar tres artefactos clave para el funcionamiento del marco estándar:

- Pila del producto / *product backlog*: Registra y prioriza los requisitos desde el punto de vista del cliente. Empieza con una visión inicial del producto y crece y evoluciona durante el desarrollo. Los requisitos suelen denominarse «historias de usuario», que se descomponen en «tarefas» de menor tamaño, normalmente de un día de trabajo como máximo.
- Pila del sprint / *sprint backlog*: Refleja los requisitos desde el punto de vista del equipo. Es una lista de los trabajos a realizar durante un sprint (→ Eventos) para generar el «incremento» previsto.
- Incremento: resultado de cada sprint.

Otros artefactos

También son habituales:

- Gráfico de avance o *burn down chart*: indica el trabajo pendiente y la velocidad a la que se están completando las tareas para deducir si se completarán todas en el tiempo estimado. El equipo lo actualiza a diario.
- Gráfico de producto o *burn up chart*: si el gráfico de avance mide lo que falta, el de producto mide cuánto se ha construido o completado.
- Definition of Ready (DoR): acuerdo que define cuándo una historia de usuario se considera «lista» para ser descompuesta en tareas, estimada e incluida en un sprint.
- *Definition of Done (DoD)*: acuerdo sobre los criterios para considerar que una parte del trabajo (tarea, historia...) está terminada.

Pila del producto: los requisitos del cliente

La «pila del producto» es el inventario de funcionalidades, mejoras, tecnología y corrección de errores que deben incorporarse al producto a través de los sucesivos sprints. Representa todo aquello que esperan cliente, usuarios y demás partes interesadas. Todo lo que implique trabajo para el equipo debe estar reflejado aquí. Lo más común es referirse a las entradas de esta pila como «historias de usuario». Algunos ejemplos:

- «Ofrecer a los usuarios la consulta de archivos publicados por un determinado miembro de la plataforma».
- «Consultar los pedidos realizados por un vendedor en un rango de fechas.»
- «Ofrecer la consulta de un archivo a través de un API web.»

La característica esencial de este artefacto es que contiene información viva, en continua evolución, y que más que un documento de requisitos es una herramienta que facilita la comunicación de información al equipo. Al comenzar el proyecto la lista contiene unos pocos requisitos, aquellos conocidos y mejor entendidos en ese momento, porque se ampliará y modificará conforme avanza el desarrollo. Este carácter dinámico permite que el producto se adapte a circunstancias cambiantes.

Se suele elaborar tras una reunión en la que el cliente comparte con el equipo una visión general del objetivo de negocio que persigue. Una vez que la pila del producto tiene historias suficientes para realizar un primer sprint es suficiente para empezar.

A partir de entonces el propietario del producto mantendrá las historias de la pila ordenadas según su prioridad. El nivel de urgencia vendrá marcado por lo necesaria y valiosa que sea cada funcionalidad.

Por otro lado, el grado de concreción de las historias de usuario deberá ser proporcional a su prioridad. Las más prioritarias deben estar lo bastante detalladas como para descomponerse en tareas y pasar al siguiente sprint.

Las tareas de priorización, detalle y preestimación de las historias, previas al sprint, se suelen llamar «preparación» o «*grooming*». El propietario del producto y el equipo pueden realizarlas en cualquier momento, de forma colaborativa, pero nunca deberían consumir más del 10% de la capacidad de trabajo del equipo. Más tarde el equipo realizará una segunda estimación más detallada, en la «reunión de planificación del sprint» (→ Eventos), al descomponer las «historias» en «tareas». La responsabilidad de estimar el esfuerzo previsible para cada elemento de la posterior lista de tareas (→ Pila del sprint) es de las personas del equipo que, en principio, harán el trabajo. (→ Métricas y estimación ágil).

Las historias de usuario de la pila del producto que pueden ser incorporados a un sprint se denominan «preparadas» o «ready». Se aplica este término (o similar) para indicar que el propietario del producto y el equipo están de acuerdo en que la historia está lista para ser seleccionada para el sprint. Es decir: está definida, preestimada, es asumible por sí misma en un único sprint, y se han establecido los criterios para considerarla terminada, así como la persona responsable de verificar que se cumplen.

Para confeccionar y mantener la pila de producto, lo ideal es emplear medios simples, conocidos y compartidos por todo el equipo. Es un radiador de información útil y una herramienta que facilita la comunicación directa. Se pueden anotar las historias de usuario, por ejemplo, en etiquetas adhesivas sobre tableros, ordenándolas según su prioridad; o emplear una herramienta de gestión con la que todo el equipo esté familiarizado, tipo Trello.

Pila del sprint

SPRINT			INICIO	DURACIÓN														
1			1-mar-07	12	J	V	L	M	X	J	V	L	M	J	V	L		
					1-mar	2-mar	5-mar	6-mar	7-mar	8-mar	9-mar	12-mar	13-mar	15-mar	16-mar	19-mar		
					23	23	19	16	16	13	9	9	9	9	9	9		
					276	246	216	190	178	158	110	110	110	110	110	110		
SPRINT BACKLOG					ESFUERZO													
Tarea	Estado	Responsal																
Descripción de la tarea 1	Terminada	Luis	16	16	16	16	16	16										
Descripción de la tarea 2	Terminada	Luis	12	8														
Descripción de la tarea 3	Terminada	Luis	4	4	4	4	4											
Descripción de la tarea 4	Terminada	Elena	8	4														
Descripción de la tarea 5	Terminada	Elena	16	16	4													
Descripción de la tarea 6	Terminada	Elena	6	6	2													
Descripción de la tarea 7	Terminada	Antonio	16	4														
Descripción de la tarea 8	Terminada	Antonio	16	16	20	12	4											
Descripción de la tarea 9	Terminada	Antonio	12	2														
Descripción de la tarea 10	En curso	Luis	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
Descripción de la tarea 11	Pendiente	Luis	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
Descripción de la tarea 12	Terminada	Luis	14	14	14	14	14	14	14									

Ilustración 31: Ejemplo de pila de sprint.

La «pila del sprint» o «*sprint backlog*» es la lista de todas las tareas necesarias para construir las historias de usuario que se van a realizar en un sprint. En ella las historias de usuario se descomponen en unidades de tamaño adecuado para monitorizar el avance a diario, así como para identificar riesgos y problemas sin procesos de gestión complejos.

Todo el equipo colabora en la confección de esta pila, durante la «reunión de planificación del sprint» (→ Eventos), indicando para cada tarea el esfuerzo previsto para realizarla. Para calcular el «esfuerzo» de cada tarea en «puntos» o «tiempo ideal» (→ Métricas y estimación ágil) es habitual emplear técnicas como la estimación de póquer (→ Prácticas para flexibilizar scrum: estimación de póquer). Las tareas de mayor tamaño se dividen en otras, de modo una sola nunca dure más de un día de trabajo.

Si la pila del producto es territorio del propietario del producto, la pila del sprint es territorio del equipo. Sus miembros son los únicos que pueden modificarla durante el sprint.

Proporciona además comunicación visual directa y sobre ella el equipo revisa a diario el avance del sprint. Lo ideal es que se encuentre en un tablero o pared en el

mismo espacio físico donde se trabaja, para que sea visible por todos. Algunos soportes habituales son tableros físicos, hojas de cálculo compartidas, y herramientas colaborativas de gestión de proyectos tales como Todoist, Flow o Trello. Lo apropiado es utilizar el formato más cómodo para todos, teniendo en cuenta los siguientes criterios:

- Debe incluir sólo la información necesaria:
 - Lista de tareas.
 - Persona responsable de cada tarea.
 - Estado en el que se encuentra y «esfuerzo» que queda para completarla.
- Debe servir de medio para registrar, en cada reunión diaria del sprint, el «esfuerzo» que le queda a cada tarea.
- Debe facilitar la consulta y la comunicación diaria y directa del equipo.

Incremento

El «incremento» es la parte de producto producida en un sprint y que se encuentra en condiciones de ser entregada al cliente; es decir: terminada, probada y operativa. No se deben considerar incrementos a prototipos, módulos ni a partes pendientes de pruebas.

Idealmente, en scrum:

- Cada elemento de la pila del producto se refiere a funcionalidades entregables, no a trabajos internos del tipo «diseño de la base de datos».
- Se produce un incremento en cada iteración / sprint.

Sin embargo, el primer sprint suele ser una excepción. Se suele denominar «sprint cero» cuando tiene objetivos como «contrastar la plataforma y el diseño», que son necesarios al comenzar algunos proyectos. Implican trabajos de diseño o desarrollo de prototipos, para contrastar las herramientas o métodos de trabajo previstos.

Si la parte desarrollada requiere documentación, o procesos de validación y verificación documentados, éstos también tienen que estar realizados para considerar al incremento «terminado» o «*done*», es decir: entregable al cliente.

Eventos

En este apartado se detallan las prácticas y actividades que constituyen la rutina de trabajo en scrum.

Sprint: es el núcleo fundamental de scrum, en torno al que se organizan todos los demás. A veces se llama también «iteración». Es el nombre que recibe cada etapa de trabajo con un objetivo concreto dentro del proyecto. La división del trabajo en sprints, que tienen una duración fija y constante (a esto se suele denominar «timeboxing») permite mantener el ritmo de avance.

Reunión de planificación del sprint: marca el inicio de cada sprint. En ella se determina cuál es el objetivo de éste y las tareas necesarias para conseguirlo.

Scrum diario: breve reunión diaria en la que el equipo hace punto de situación para confirmar que se está avanzando al ritmo adecuado, o si hay algún impedimento detectarlo y actuar en consecuencia lo antes posible.

El protocolo más frecuente es que cada miembro informe de lo realizado el día anterior, lo que tiene previsto hacer a continuación y si prevé algún impedimento.

Cada persona actualiza en la pila del sprint el tiempo o esfuerzo pendiente de sus tareas, y con esta información se actualiza a su vez el gráfico con el que el equipo monitoriza el avance del sprint: el gráfico de avance o burn down.

Revisión del sprint: análisis e inspección del «incremento» generado, y adaptación de la pila del producto si resulta necesario.

Retrospectiva del sprint: reunión al finalizar el sprint en la que el equipo analiza aspectos operativos de su forma de trabajo y crea un plan de mejoras, para aplicarlo en la siguiente iteración.

Sprint

El evento clave de scrum para mantener un ritmo de avance continuo es el sprint: un periodo de tiempo acotado, cuya duración se recomienda que no exceda de 4 semanas, durante el que se construye un incremento del producto.

El incremento, como ya se vio en Artefactos, debe estar terminado; esto es: operativo y útil para el cliente, en condiciones de ser desplegado o distribuido.

Al comenzar a trabajar con scrum es recomendable considerar el sprint como el evento contenedor de todos los demás:

- Marca el ritmo de avance diario y permite visualizarlo y compartirlo en las «reuniones de pie».
- Marca un ritmo fijo para comprobar el desarrollo del producto en las reuniones de planificación y revisión del sprint.
- A ese mismo ritmo se introducen las reuniones de retrospectiva, para reflexionar y mejorar.

En implementaciones más maduras de scrum, sin embargo, es posible considerar que el ámbito del sprint es sólo la construcción del incremento, dejando a un lado las reuniones.

Esto puede interesar al equipo, por ejemplo, para poder calcular la velocidad del sprint considerando sólo el tiempo de trabajo, sin incluir las reuniones de inicio, cierre y retrospectiva; o para tener mayor flexibilidad al realizar sprints de duraciones diferentes; o para separar la frecuencia de las retrospectivas de la de los sprints.

Reunión de planificación del sprint

Esta reunión marca el inicio de cada sprint. En ella se toman como base las prioridades y necesidades de negocio del cliente y se determinan cuáles y cómo van a ser las funcionalidades que se incorporarán al producto al terminar el sprint.

Se trata de una reunión conducida por el scrum master (o, en su ausencia, un miembro del equipo) a la que deben asistir el propietario del producto y el equipo completo, y en la que también pueden estar presentes otros implicados en el proyecto. Puede durar hasta una jornada de trabajo completa, según el volumen o complejidad de las historias de usuario que se desean incluir en el próximo incremento.

La reunión debe dar respuesta a dos cuestiones, que la articulan en dos partes de duración similar y suelen estar separadas por una pausa:

1. Qué se entregará al terminar el sprint.
2. Cómo se conseguirá el incremento, estimando el tiempo de trabajo y los requisitos necesarios.

Precondiciones	
La organización tiene determinados y asignados los recursos disponibles para llevar a cabo el sprint. Ya están «preparadas» las historias de usuario de mayor prioridad de la pila del producto, de forma que ya tienen un nivel de concreción suficiente y una estimación previa del trabajo que requieren. El equipo tiene un conocimiento de las tecnologías empleadas y del negocio del producto suficientes para realizar estimaciones, y para comprender los conceptos del negocio que expone el propietario del producto.	
Entradas	Resultados
Pila del producto. Producto desarrollado en los incrementos anteriores (excepto en el sprint 0). Velocidad o rendimiento del equipo en el último sprint, como criterio para estimar la cantidad de trabajo. Circunstancias de negocio del cliente y del escenario tecnológico empleado.	Pila del sprint. Duración del sprint y fecha de la reunión de revisión. Objetivo del sprint.

Primera mitad: ¿qué se entregará al terminar el sprint?

El propietario del producto es el responsable de la presentación en esta primera mitad de la reunión. Expone las historias de usuario de mayor prioridad de la pila del producto al equipo, explicando qué se necesita y qué prevé que se podrá desarrollar en el siguiente sprint. Si la pila ha tenido cambios significativos desde la anterior reunión, explica las causas que los han ocasionado.

El objetivo es que todo el equipo comprenda, con un nivel de detalle suficiente, el incremento que se desea obtener con el sprint. La exposición debe estar abierta a preguntas y se pueden solicitar aclaraciones. Cualquier miembro del equipo puede proponer sugerencias, modificaciones y soluciones alternativas, y modificar la pila en consecuencia.

Esta reunión es un punto caliente de scrum para favorecer la fertilización cruzada de ideas y añadir valor a la visión del producto.

Tras reordenar y replantear las historias de la pila, el equipo define el «objetivo del sprint»: una frase que sintetiza cuál es el valor que se va a entregar al cliente. Exceptuando sprints dedicados a colecciones de tareas desordenadas, la elaboración de este lema de forma conjunta en la reunión es una garantía de que todo el equipo comprende y comparte la finalidad del trabajo, y durante el sprint sirve de criterio de referencia en la toma de decisiones.

Segunda mitad: ¿cómo se conseguirá el incremento?

Esta segunda parte debe considerarse como una «reunión del equipo», en la que deben estar todos sus miembros y ser ellos quienes descompongan, estimen y asignen el trabajo. El papel del propietario del producto es atender a dudas y comprobar que el equipo comprende y comparte su objetivo.

El equipo desglosa cada funcionalidad en tareas y estima el esfuerzo para cada una de ellas, componiendo así la pila del sprint. Se establecen cuáles serán las prioritarias para los primeros días y se asignan tomando como criterios los conocimientos e intereses de cada miembro y procurando distribuir el trabajo de forma homogénea.

Funciones del scrum master durante la reunión de planificación del sprint

En caso de haber un scrum master asignado, éste será el moderador de la reunión.

Es responsable y garante de:

1. Realizar esta reunión antes de cada sprint.
2. Asegurar que se cuenta con una pila del producto preparada por el product owner.
3. Ayudar a mantener el diálogo entre el propietario del producto y el equipo.
4. Asegurar que se llegue a un acuerdo entre el propietario del producto y el equipo respecto a lo que incluirá el incremento.
5. Ayudar al equipo a comprender la visión y necesidades de negocio del cliente.
6. Asegurar que el equipo ha realizado una descomposición y estimación del trabajo realistas, y ha considerado las posibles tareas necesarias de análisis, investigación o apoyo.
7. Asegurar que al final de la reunión están objetivamente determinados:
 - Los elementos de la pila del producto que se van a ejecutar.
 - El objetivo del sprint.
 - La pila del sprint con todas las tareas estimadas.
 - La duración del sprint y la fecha de la reunión de revisión.

Scrum diario

Reunión diaria breve, de no más de 15 minutos, en la que el equipo sincroniza el trabajo y establece el plan para las 24 horas siguientes.

Entradas	Resultados
Pila del sprint y gráfico de avance con la información de la reunión anterior. Información del avance de cada miembro del equipo.	Pila del sprint y gráfico de avance actualizados. Información del avance de cada miembro del equipo.

Formato de la reunión

Se recomienda realizarla de pie junto a un tablero con la pila del sprint y el gráfico de avance o tablero kanban, para que todos puedan compartir la información y anotar.



En la reunión está presente todo el equipo, y pueden asistir también otras personas relacionadas con el proyecto o la organización, aunque éstas no intervienen.

Cada miembro del equipo de desarrollo explica lo que ha logrado desde el anterior scrum diario, lo que va a hacer hasta el siguiente, y si está teniendo algún problema o si prevé que puede encontrar algún impedimento. Se actualiza sobre la pila del sprint el esfuerzo que estima pendiente en las tareas que tiene asignadas, o se marcan como finalizadas las ya completadas. Al final de la reunión el equipo refresca el gráfico de avance del sprint (→ Prácticas para flexibilizar scrum: Gráfico burn down) con las estimaciones actualizadas.

El equipo es el responsable de esta reunión, no el scrum master; y no se trata de una reunión de inspección o control, sino de comunicación entre el equipo, para compartir el estado del trabajo, chequear el ritmo de avance y colaborar en posibles dificultades. El scrum master realizará las gestiones adecuadas para resolver estas últimas tras la reunión.

Revisión del sprint

Reunión realizada al final del sprint para comprobar el incremento. Lo habitual es que dure una o dos horas; en caso de incrementos de especial relevancia o complejidad, puede extenderse hasta 4 como máximo. Asiste todo el equipo de desarrollo, el propietario del producto, el scrum master y todas las personas implicadas en el proyecto que lo deseen.

Esta reunión marca, a intervalos regulares, el ritmo de construcción, y la trayectoria que va tomando la visión del producto. Al ver y probar el incremento, el propietario del producto y el equipo en general obtienen feedback relevante para revisar la pila del producto. Se identifican las historias de usuario que se pueden considerar «hechas» («done») y las que no.

Entradas	Resultados
Incremento terminado.	Feedback para el propietario del producto: hito de seguimiento del avance del proyecto e información para mejorar su valor. Convocatoria de la reunión del siguiente sprint.

Formato de la reunión

Se trata de una reunión informal, en la que sólo se pretende mostrar el resultado de la iteración, el incremento, en funcionamiento. Según las características del proyecto puede incluirse también documentación de usuario o técnica.

Su misión es informar, no la toma de decisiones ni la crítica del incremento. Más adelante, con la información obtenida, el propietario del producto tratará las posibles modificaciones que surjan.

Protocolo recomendado:

1. El equipo expone el objetivo del sprint, la lista de funcionalidades que se incluían y las que se han desarrollado.
2. El equipo hace una introducción general del sprint y demuestra el funcionamiento de las partes construidas. Se abre un turno de preguntas y

sugerencias. Esta parte genera información valiosa para que el propietario del producto y el equipo en general puedan mejorar la visión del producto.

3. El scrum master, de acuerdo con las agendas del propietario del producto y el equipo, cierra la fecha para la reunión de preparación del siguiente sprint.

Retrospectiva del sprint

Reunión que se realiza tras la revisión de cada sprint, antes de la reunión de planificación del siguiente. La duración recomendada es de una a tres horas. Son un ejemplo de práctica que no se encontraba en el marco original de scrum, pero que ha ido consolidándose con el tiempo.

En ella el equipo reflexiona sobre su forma de trabajar. Se identifican fortalezas y puntos débiles, para afianzar las primeras y planificar acciones de mejora sobre los segundos.

El hecho de que se realice normalmente al final de cada sprint lleva a veces a considerarla erróneamente como una reunión de «revisión de sprint», cuando es aconsejable tratarlas por separado, porque sus objetivos son diferentes. El objetivo de la revisión del sprint es analizar «qué» se está construyendo, el producto, mientras que una reunión retrospectiva se centra en el marco de trabajo, el «cómo».

En la retrospectiva del sprint participan: el equipo de desarrollo, el scrum master y el propietario de producto. Es importante que el propietario de producto se considere “equipo” más que “cliente”. Que la persona que desempeña el rol sea participativa y conocedora de los principios y valores de scrum. Si esto no fuera así, el scrum master debe actuar como facilitador para lograr su compromiso y participación o, si la situación lo requiere, no incluirlo en la retrospectiva.

Medición y estimación ágil

En este apartado vamos a explicar los conceptos básicos sobre medición y estimación ágil que conviene conocer.

Dicho esto: cuantas menos métricas, mejor. El objetivo de scrum es producir el mayor valor posible de forma continua, así que cabe preguntarse siempre cómo contribuye el uso del indicador en el valor que se proporciona al cliente. Medir es costoso y debe servir a un propósito mayor, no convertirse en un fin en sí mismo.

Los objetivos que perseguimos con las herramientas de estimación que vamos a ver ahora son: poder planificar la duración de cada sprint de forma realista, marcar el ritmo de avance (sobre todo en equipos que están empezado a trabajar de forma ágil), sincronizar equipos y cerrar fechas de entrega.

Dos conceptos clave:

- No se mide el trabajo realizado, sino el que queda.
- Se mide empleando unidades relativas.

¿Cuánto queda?

Medir el trabajo puede ser necesario por dos razones: para registrar el que ya se ha hecho, o para estimar por adelantado el que se debe realizar. En ambos casos se necesitan una unidad y un criterio objetivos de cuantificación.

Medir el trabajo ya realizado no entraña dificultad. Se puede hacer con unidades relativas al producto, como historias completadas; o a los recursos, como coste o tiempo de trabajo.



Ilustración 33: ¿Cuanto queda para terminar?.

Pero la gestión de proyectos ágil no mide el trabajo ya hecho para calcular el avance del trabajo; es decir, restándolo del tiempo previsto. Por ejemplo: «Esto debía costar una semana. Como han pasado tres días, quedan cuatro para que esté

terminado.» Esto, que suena lógico, en la realidad no se suele cumplir. Surgen imprevistos o se encuentran atajos que hacen que el tiempo estimado al principio no sea exacto. Teniendo esto en cuenta, no se determina el grado de avance por el trabajo realizado, sino por el que queda pendiente.

Esto lo entienden muy bien los niños. Cuando se sale de viaje, ¿qué es lo que preguntan, una y otra vez? ¿Cuánto tiempo llevan en el coche? No, no, no... ¡eso no les importa! La pregunta es: ¿cuánto queda?

Es posible que otros procesos de la organización necesiten registrar el esfuerzo invertido, pero calcular el avance del proyecto es diferente.

Scrum mide el trabajo pendiente, primero: para estimar el esfuerzo y tiempo previstos para realizar determinadas tareas, historias de usuario y «*epics*» (historias de gran tamaño). Y segundo: para determinar el grado de avance del proyecto, y en especial de cada sprint.

Unidades relativas: puntos de historia

El trabajo necesario para realizar un requisito o una historia de usuario no se puede prever de forma absoluta porque rara vez son realidades de una solución única. En el caso de que se pudiera, por otra parte, la complejidad de la medición haría una métrica demasiado pesada para la gestión ágil.

No resulta posible estimar con precisión la cantidad de trabajo que hay en un requisito. En consecuencia, tampoco se puede saber con antelación cuánto tiempo exigirá, porque a la incertidumbre del trabajo se suman las inherentes al tiempo: no se puede estimar la cantidad o la calidad del trabajo que realiza una «persona media» por unidad de tiempo, porque son muy grandes las diferencias de unas personas a otras. Es más: la misma tarea realizada por la misma persona requerirá diferentes tiempos según las circunstancias.

Por todas estas razones, al estimar de forma ágil, se prefiere emplear unidades relativas. En gestión ágil se suelen emplear «puntos» como unidad de trabajo, usando denominaciones como «puntos de historia».



Ilustración 34: Unidades relativas.

Cada organización, según sus circunstancias y su criterio, institucionaliza su métrica de trabajo, su «punto». Es el tamaño relativo de tareas que se suele emplear. Es importante que el significado y la forma de aplicar la métrica sea siempre la misma en las mediciones de la organización, y que sea conocida por todos.

El tipo de «punto» dependerá de la organización. En un equipo de programación el punto puede ser equivalente a preparar una pantalla de *login*; para un equipo de diseño gráfico, la maquetación de un tríptico.

El «punto» ayuda, por un lado, a dimensionar la estimación de una tarea comparándola con una ya conocida, y por otro lado, a contrastar la dificultad que la tarea presenta para cada miembro del equipo según sus especialidades. Un ejemplo para ilustrar esto último podría ser el esfuerzo que cuesta freír un huevo. Si se estima cuántos «huevos fritos» costaría planchar una camisa, la respuesta dependerá de la persona. Alguien puede ser muy habilidoso friendo huevos, pero muy torpe para planchar camisas, y estimará que eso le costaría «8 huevos fritos»; es decir, «8 puntos». Alguien muy acostumbrado a las tareas domésticas, en cambio, podría estimar la tarea en «un punto» o «un huevo frito».

Ambos tienen razón: la cuestión es que la persona que estime sea la que va a realizar la tarea. Los «puntos de historia» suelen ser una unidad relativa o abstracta basada en algo con lo que el equipo esté muy familiarizado.

Por último, con esto se puede estimar la velocidad: en scrum, ésta es igual a la cantidad de trabajo realizado por el equipo en un sprint. Así, por ejemplo, una velocidad de 150 puntos indica que el equipo realiza 150 puntos de trabajo en cada sprint.

No obstante, al salir del marco estándar de scrum podemos encontrar sprints de diferentes duraciones. Cuando esto sucede, se puede expresar la velocidad en unidades de tiempo en lugar de por sprint. Es decir: «la velocidad media del equipo es de x puntos por semana».

Tiempo real y tiempo ideal

Cuando se calcula el calendario de un sprint tendemos a estimar el esfuerzo en «tiempo ideal»: tiempo de trabajo en condiciones ideales. Es lo que nos costaría realizar una tarea en un estado de flujo, concentrados y sin ninguna distracción o impedimento. Es importante ser conscientes de la diferencia por tanto entre el «tiempo ideal» y el «tiempo real» a la hora de estimar.

Si nos preguntan cuánto dura un partido de baloncesto, la respuesta en tiempo exacto de juego es 40 minutos, de igual forma que podemos decir que redactar un informe nos cuesta una hora. Pero el tiempo real de un partido de baloncesto suele ser de más de una hora, pues no puede terminar en empate. Se alarga por los tiempos muertos, las faltas, el entretiempo del partido y las prórrogas.



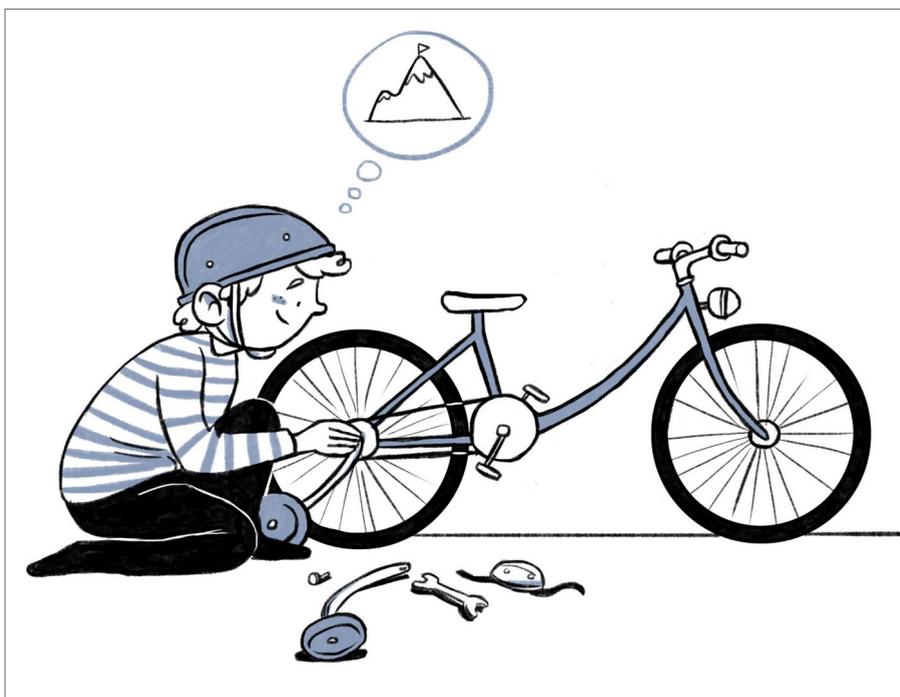
Ilustración 35: Tiempo ideal y tiempo real.

Todos sabemos que es normal que a veces un informe, que podría hacerse en una hora de «tiempo ideal», acabe ocupando media jornada de trabajo. Manejar estos dos conceptos de tiempo puede ayudarnos a organizar el trabajo de manera más objetiva y evitar el estrés de metas inalcanzables.

 <https://youtu.be/G0Ruagq7A7I>

Segunda parte: principios y valores

Interiorizando y adaptando



Principios y valores scrum

Hay algo que no se suele decir en los cursos de scrum: el marco que acabamos de estudiar, las prácticas y herramientas vistas en la primera parte de este manual, no funcionan sin unos principios y valores alineados con ellas.

¿Por qué? Porque las «prácticas» de scrum no son «procesos». No son pautas que garanticen resultados las ejecute quien las ejecute.

Scrum se asienta sobre el conocimiento tácito de las personas, así como sobre unos valores organizativos. Las prácticas son sólo las ramas del árbol y, sin unas buenas raíces, no darán frutos. Algo que conviene recordar de vez en cuando para no caer en el error de centrarnos sólo en las herramientas.

Los modelos de gestión facilitan el desarrollo de ciertos principios de trabajo y valores de la organización, sin los cuales trabajaremos en vacío.

Podemos encontrar empresas donde los principios están tan interiorizados que desarrollan su propio modelo de agilidad, con prácticas y técnicas diferentes a las que hemos visto, de otros modelos de agilidad o completamente originales, que dan como resultado productos innovadores y de calidad. Y empresas que, aplicando todas las prácticas del scrum estándar, sólo consiguen empleados alienados, desmotivados, estresados, y resultados mediocres.

Se necesita «flexibilidad» para adaptar lo aprendido a la realidad de cada empresa y proyecto. El objetivo es que la organización sea ágil en su conjunto, capaz de «avanzar en scrum», en escenarios de trabajo innovadores e inestables.

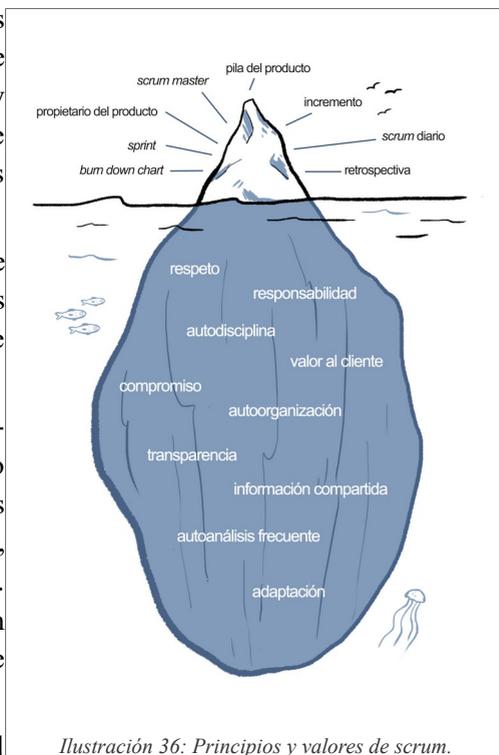


Ilustración 36: Principios y valores de scrum.

Para hacer una buena gestión, hay que estar al día de las últimas herramientas y después buscar la manera de ajustarlas al equipo, aprovechando lo que funcione y modificando o desechando lo que no; en lugar de pretender que las personas se adapten a métodos que no les convencen.

Los valores y principios pueden dividirse, aunque sea por pura semántica, en función de si están detrás de las prácticas ágiles o de la cultura de la organización. Los «principios» son el soporte de las prácticas; los «valores», de la cultura.

Principios

La finalidad de artefactos, eventos y otras técnicas ágiles es lograr que el trabajo se base en estos principios.

- **Entrega de valor.**

Entendiendo como tal a la entrega temprana y continua de valor al cliente, para lo que es necesario que éste colabore con el equipo y se comparta y comprenda su visión.

- **Mejora continua.**

En agilidad se reflexiona con frecuencia sobre los métodos de trabajo, cuestionando su efectividad y adaptándolos. El mismo esfuerzo autocrítico se aplica también a la mejora de los productos y servicios que se ofrecen.

- **Desarrollo iterativo e incremental.**

El producto final no se construye conforme a un plan inicial detallado y completo, sino que se arranca desde un «mínimo viable» sobre el que se van añadiendo incrementos.

- **Ritmo de trabajo sostenible.**

Alcanzar un ritmo de trabajo que evite la Ley de Parkinson (el trabajo se expande hasta llenar el tiempo disponible para que se termine) y la presión al descubrir los retrasos demasiado tarde.

- **Atención continua a la excelencia.**

Empleo de técnicas que garanticen la calidad de los productos y servicios y permitan detectar errores con antelación o en el momento de producirse.

- **Operativa visible.**

La información se comparte con claridad para facilitar la colaboración, identificar impedimentos de forma temprana y permitir que todo el equipo conozca el estado del producto y aporte ideas.

- **Cadencia y sincronización global.**

Este principio es más relevante cuando se intenta sincronizar a varios equipos que trabajan en productos o servicios relacionados. Se busca predecir la frecuencia de reuniones y fechas de entrega, por ejemplo sincronizando los sprints de los diferentes equipos.

- **Personas sobre procesos.**

La inteligencia colectiva del equipo, su conocimiento tácito, es responsable directa de la calidad del producto.

Valores

La cultura de la empresa es la suma de sus características organizacionales y de gobernanza, que pueden acelerar o frenar el desarrollo de la agilidad. En organizaciones cuya cultura se asienta en valores propios de trabajos industriales, basados en procesos, los principios ágiles despliegan resultados de inteligencia colectiva y valor innovador mucho más modestos que en organizaciones con valores que potencian los resultados de trabajar con principios ágiles:

- Asertividad.
- Valoración del talento.
- Claridad.
- Confianza.
- Estructura desjerarquizada.
- Propósito común.

Por último, es necesario el soporte directivo: que la gerencia de la empresa tenga una cultura afín, esté implicada y apoye a las personas con formación y recursos suficientes.

Para el propósito de esta guía, que es disponer de todas las herramientas para realizar las funciones de scrum master, vamos a conocer los «principios» asociados a los roles, artefactos y eventos de scrum. Por último, añadiremos algunas prácticas más que, aunque no forman parte del modelo de scrum estándar, se suelen utilizar.

Si se tiene interés en estudiar estos principios y valores ágiles en mayor profundidad, Scrum Manager® ha desarrollado un segundo manual que se encuentra disponible en Scrum Level: <https://scrumlevel.com>.

Las personas y sus roles

Aunque artefactos y eventos ayudan a desarrollar los principios ágiles, éstos tienen que ser asumidos primero por las personas. Sin personas con talento, comprometidas y conscientes de sus responsabilidades, las prácticas no servirán de nada. Y sin la influencia de ciertos valores culturales (→ Scrum Level), no se puede desarrollar el talento ni el compromiso.

«Si tienes un equipo de ingenieros brillantes, que usan excelentes herramientas y prácticas de ingeniería, que comprenden de arriba abajo el ámbito tecnológico y del negocio, a los que no se les interrumpe y tienen los recursos tecnológicos que necesitan, ¡entonces puedes usar scrum! Es cierto que las personas así pueden construir un incremento de software en cada iteración. ¡Eso está bien!.

Pero scrum también funciona con idiotas.

Puedes tener un grupo de idiotas que no hayan pisado una escuela, que no sepan ni de informática, ni técnicas de ingeniería de software, que se odien entre ellos, que no entiendan ni del negocio ni de las herramientas de ingeniería, y regularmente producirán mierda en cada incremento. ¡Eso está bien! Quieres saber al final de cada iteración dónde estás.» (Schwaber 2006)

Propietario del producto

El propietario del producto o product owner es el responsable de la pila del producto; quien la gestiona y prioriza los requisitos, con actitud proactiva y realizando cambios cuando lo estime necesario.

El propietario del producto es quien hace posible la entrega temprana y continua de valor. Representa al cliente y es responsable de comunicar su visión de forma clara al equipo, para que todos se alineen hacia el mismo objetivo.

Su actitud es clave para facilitar una comunicación honesta y fluida.

Equipo

Todo el equipo es responsable de producir un incremento para el cliente con cada iteración, así como de mantener un ritmo de trabajo sostenible. Comprometidos con su trabajo, son responsables de aplicar una mirada crítica para mejorar sus

resultados y sus métodos. Por último, deben ser conscientes de la importancia que tienen su talento e inteligencia colectiva, no sólo a nivel individual.

Scrum master

Es quien garantiza que el marco scrum funcione, moderando las reuniones de scrum diarias y gestionando la resolución de impedimentos identificados en éstas, para mantener el avance del equipo.

Agrupar la responsabilidad del funcionamiento de scrum en el rol de scrum master resulta aconsejable cuando el product owner o el equipo tienen poca experiencia utilizando scrum, o en organizaciones grandes, con un flujo continuo de rotación o formación de personal. En equipos pequeños, estables y con niveles de agilidad consolidados, las responsabilidades de funcionamiento y mejora del marco de scrum pueden estar ya interiorizadas y asumidas por el equipo en conjunto.

Artefactos

Los artefactos facilitan el desarrollo de los principios ágiles.

Pila del producto

Entrega de valor: permite asumir la variabilidad del entorno de negocio del cliente y centrar los esfuerzos en aquellas historias que aportan mayor valor según las circunstancias.

Desarrollo iterativo e incremental: esta pila, a diferencia de un documento de requisitos cerrado, hace posible el principio de desarrollo iterativo ya que es un documento vivo, que permite que haya cambios en las historias que contiene y en su prioridad.

Operativa visible: es un radiador de información mediante el que el propietario del producto y el equipo comparten la visión del producto en todo momento.

Pila del sprint

Desarrollo iterativo e incremental: es el artefacto que delimita el trabajo de un incremento (sprint) y sirve para marcar el pulso de avance.

Operativa visible: es una herramienta de comunicación interna para el equipo, a la que todos tienen acceso y que permite conocer el estado del sprint con un golpe de vista.

Incremento

Entrega de valor: la presentación de una parte del producto terminada y lista para usarse al final de cada sprint permite comprobar si se está generando valor.

Desarrollo iterativo e incremental: la entrega de los incrementos ayuda a marcar el ritmo de avance.

Eventos

Sprint

Mejora continua: El ritmo de avance en iteraciones breves facilita identificar hitos en los que pararse a reflexionar sobre cómo mejorar la calidad del producto y de los sistemas de trabajo.

Desarrollo iterativo e incremental: es la unidad básica de tiempo durante la que se construye cada incremento, por tanto el engranaje en torno al que gira todo el desarrollo.

Ritmo de trabajo sostenible: marca el pulso de avance.

Operativa visible: permite la identificación temprana de impedimentos.

Cadencia y sincronización global: marca la cadencia de las entregas mediante timeboxing. Permite predecir la frecuencia de reuniones y de fechas de entrega, así como sincronizar el trabajo de diferentes equipos.

Reunión de planificación del sprint

Entrega de valor: durante esta reunión equipo y el product owner colaboran de forma directa, profundizando en el conocimiento compartido de la visión.

Scrum diario

Operativa visible: en esta reunión el equipo se ubica, comparte el estado en el que se encuentra su trabajo y colabora, aportando ideas y resolviendo los impedimentos que haya.

Revisión del sprint

Entrega de valor: en este evento el equipo vuelve a colaborar directamente con el propietario del producto.

Mejora continua: el propósito de la reunión es analizar el incremento generado, para sacar conclusiones que ayuden a perfilar el siguiente sprint.

Retrospectiva del sprint

Mejora continua: se analiza no el incremento, sino los métodos de trabajo empleados por el equipo, para decidir qué mantener y qué modificar o eliminar.

 <https://youtu.be/7wd61RrJAfo>

Prácticas para flexibilizar scrum

La comunidad profesional aporta «prácticas» para dar forma a pilas de producto, historias de usuario, representar con imágenes la visión del producto, conducir eventos y reuniones, y estimar tareas. Las dos más empleadas en la formación inicial del marco scrum son el gráfico burn down y la estimación de póquer. Vamos a verlas junto con algunas otras:

- Gráfico burn down.
- Estimación de póquer.
- Kanban.
- Trabajo en pareja.
- Técnicas a prueba de errores.
- Estimación en la pared.
- Gráfico de producto.
- Diagrama de espina.
- Diagrama de árbol.

Para el propósito de este manual no nos interesa entrar a explicar la operativa de estas prácticas en profundidad, sino presentarlas como ejemplos y animar al lector a investigar y experimentar. Son sólo algunas de las técnicas de gestión ágil que pueden ayudar a personalizar el modelo de gestión de un proyecto o equipo.

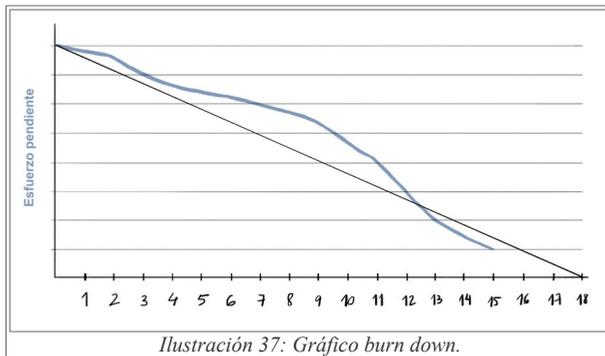
Gráfico burn down

También llamado «gráfico de avance». El equipo lo actualiza, durante el sprint, a ser posible a diario, para monitorizar el ritmo de avance. Es útil para la detección temprana de desviaciones que puedan comprometer la entrega al final del sprint.

- En el eje Y representa los puntos de trabajo que aún faltan por realizar.
- En el eje X representa los días del sprint.

Día a día, cada miembro del equipo estima en la pila del sprint el esfuerzo pendiente para cada tarea, hasta que se termina. Con esa información se actualiza el gráfico, reflejando cada día el esfuerzo pendiente total de todas las tareas que aún no se han terminado.

El avance ideal de un sprint estaría representado por la diagonal que reduce el esfuerzo pendiente de forma continua y gradual hasta el último día previsto. Como es de suponer, esto no es lo habitual. Se puede llevar un patrón de avance adecuado sin que la diagonal del gráfico sea perfecta.



Si la línea de avance se mantiene durante varios días muy por encima de la diagonal, es señal de que se ha subestimado el sprint y de que éste requerirá más tiempo. Cuando ocurre lo contrario y la línea desciende más deprisa que la diagonal, se terminará antes de lo previsto.

Estimación de póquer

Jane Grenning ideó este juego de planificación que se utiliza para conducir las reuniones en las que se estima el esfuerzo y la duración de las tareas (planificación del sprint). El modelo inicial de Grenning consta de 8 cartas, con los valores $\frac{1}{2}$, 1, 2, 3, 5, 6, 7 e infinito.

Cada participante dispone de un juego de cartas y, en la estimación de cada tarea, muestra la combinación que suma el esfuerzo estimado.

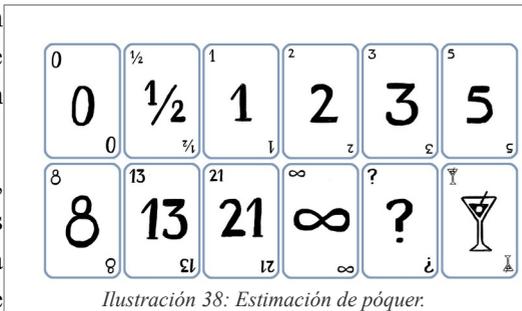
El uso más extendido, sin embargo, es empleando una baraja con números en sucesión de Fibonacci, como en la ilustración. En este caso no se

combinan números, sólo se muestra una carta para estimar cada tarea: aquella con la cifra más aproximada. Esta variante se basa en que, al aumentar el tamaño de las tareas, aumenta también el margen de error.

Así, por ejemplo, si una persona cree que el tamaño adecuado de una tarea es 6, se ve obligado a reconsiderar. Puede aceptar que parte de la incertidumbre apreciada no es tal y levantar la carta de 5, o aceptar una estimación más conservadora y levantar el 8.

Es frecuente añadir una carta con un símbolo de duda o interrogación para indicar que, por las razones que sean, no se puede precisar una estimación. También es posible incluir otra carta con alguna imagen alusiva para indicar que se necesita un descanso. El símbolo de infinito se saca cuando la tarea excede el máximo esfuerzo estimable, para indicar que debería dividirse en unidades más pequeñas.

Cuando las estimaciones son muy dispares, el responsable de la reunión puede optar por varias alternativas. Se puede preguntar a las personas con las estimaciones más extremas el por qué de su estimación y repetir el proceso. Otra opción es dejar de lado la tarea por el momento y estimarla de nuevo más tarde. O pedir al propietario del producto que descomponga la tarea y valorar cada una de



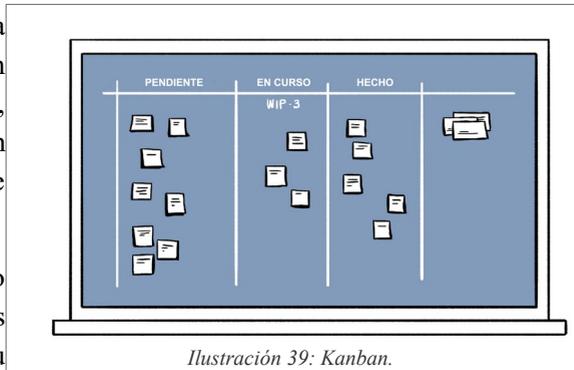
las sub-tareas resultantes. También puede tomar la decisión de optar por la estimación más optimista, más pesimista, o sacar la media. Dependerá de la tarea y del estilo de gestión de cada equipo.

El uso de la estimación de póquer, además de ser divertido y dinamizar las reuniones, evita las habituales discusiones circulares entre diversas opciones de implementación. Permite que todos los asistentes participen, ayuda a alcanzar consensos sin discusiones y a reducir el tiempo de estimación de cada funcionalidad.

Kanban

Kanban es una popular técnica para gestionar, de forma visual, un flujo continuo de avance; es decir, un flujo sin timeboxing, sin dividir el trabajo en sprints de duración predeterminada.

El equipo anota las tareas o historias de usuario y las posiciona sobre un tablero. Su ubicación determinan su estado. Los más habituales en tableros kanban son los que dicta el orden de avance: «pendiente», «en curso» y «terminado».



Se ordenan progresivamente de izquierda a derecha. El formato de cada tablero responde a las circunstancias del producto y del equipo. Puede incluir estados adicionales como «testado» o «validado», por ejemplo.

El tablero no sólo ayuda a gestionar de manera clara y visual y a mantener el ritmo, sino que también es una herramienta útil para compartir información entre el equipo. Muestra de inmediato, con cada actualización, el estado de las tareas y si se están formando cuellos de botella.

La ausencia de hitos temporales como los sprints evita la ley de Parkinson. Por el contrario, esta ausencia podría provocar retrasos, por perfeccionismo o por procrastinación. Pero gracias al WIP (work in process), la estructura y la visibilidad que ofrece el tablero kanban, se consigue paliar este efecto negativo también. El WIP es el número máximo de tareas que pueden estar a la vez en la misma fase del proceso. Un WIP=3 de la fase «en curso» indica que el equipo no puede trabajar en más de 3 tareas de forma simultánea.

La visibilidad que ofrece el tablero permite identificar a todo el equipo de forma temprana cuellos de botella y tiempos muertos, para ajustar o replantear prioridades. Con la información que se obtiene se pueden sugerir mejoras de flujo y de distribución de los miembros de los equipos.

Trabajo en pareja

Este concepto está más identificado en equipos de programación, donde se le conoce con la denominación ‘programación en pareja’, o *pair programming*.

Consiste en asignar a dos personas la ejecución de una misma tarea, de forma simultánea y conjunta, normalmente alternando entre ellos la ejecución y supervisión. Mientras uno la lleva a cabo, el otro está atento a lo que hace, realizando las observaciones que puedan ser procedentes. Esta práctica puede ser adecuada cuando la calidad del resultado depende, sobre todo, del conocimiento de la persona que lo realiza.

Por esta razón, por ejemplo, los trenes circulan con ayudante y maquinista cuando la tecnología empleada no puede evitar al 100% el fallo humano, o nos sentimos más seguros sabiendo que en la cabina del avión hay dos pilotos trabajando en pareja.

Técnicas a prueba de errores

Consiste en aplicar modos de trabajo que impiden la producción de errores. En el mundo de la programación ágil es habitual el ‘desarrollo guiado por pruebas’ o *TDD (test driven development)*, que consiste en desarrollar primero las pruebas (*tests*) que el código debe pasar y después el código.

En términos más generales, la agilidad emplea técnicas *poka-yoke* (en japonés ‘a prueba de fallos’) y dispositivos de control *andon*. Ambos conceptos provienen de los marcos de producción de manufactura *lean*.

Las técnicas ‘a prueba de fallos’ *poka-yoke* que pueden ser para:

- Hacer imposible el error humano: un ejemplo son los enchufes diseñados para que no puedan ser acoplados de forma errónea.
- Resaltar el error de forma evidente cuando se produce: como hacen los correctores ortográficos de los editores de texto, o los correctores sintácticos de programación.

Los sistemas de control *andon* son propios del proceso de producción. Suelen consistir en indicadores con luces de diferente color o representaciones gráficas

que reflejan el funcionamiento normal del sistema o posibles fallos. Están situados en el propio entorno de trabajo para alertar de forma inmediata y visual al equipo.

Estimación en la pared

Técnica empleada para estimar y priorizar una lista de historias de usuario, normalmente la pila del producto. Se basa en gestión visual, situando sobre una pared notas adhesivas con las diferentes historias. El equipo determina la posición horizontal de cada nota según el tamaño de la historia: a la izquierda las menores, a la derecha las mayores. El propietario del producto determina la posición vertical, en función de la prioridad: a más alta, mayor prioridad.

 <https://youtu.be/4O2BRg9kDU4>

Gráfico de producto

El gráfico de producto o burn up es una herramienta de planificación propia del propietario del producto. Representa la evolución previsible en función de la velocidad del equipo. Se suelen hacer tres estimaciones: pesimista, realista y optimista.

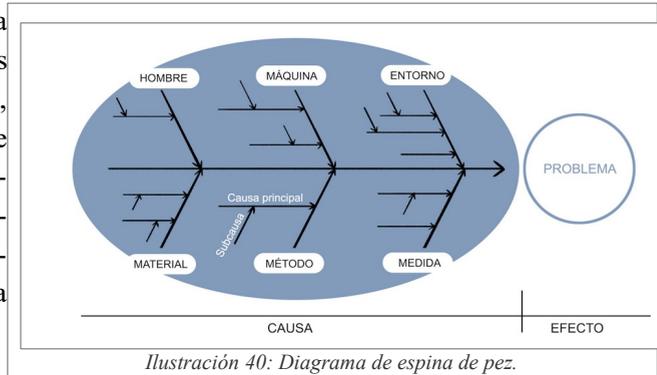
La proyección se realiza sobre un diagrama cartesiano que representa, en el eje de ordenadas, el esfuerzo estimado para construir las diferentes historias de la pila del producto, y en el de las abscisas, el tiempo medido en sprints. Para trazar la previsión se sitúa cada versión en el eje vertical en la posición correspondiente al esfuerzo estimado para construir todas las historias que incluye.

Se trata de una herramienta para desarrollo ágil, un documento vivo que no debe utilizarse para representar planes estables, sino las previsiones tras cada evolución de la pila del producto.

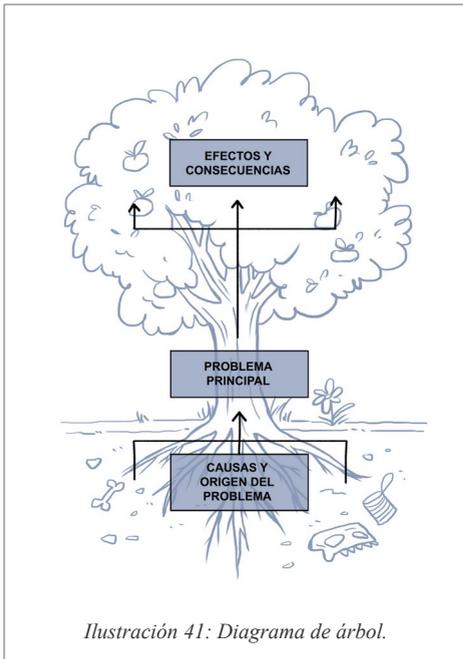
 <https://youtu.be/jdyodO-uFBk>

Diagramas para retrospectivas: espina de pez y árbol

El diagrama de espina también es llamado a veces «diagrama de Ishikawa», «de cola de pescado», «de causa-efecto», o «de Grandal». Representa las relaciones causa-efecto que actúan sobre un problema a analizar.



Tras identificar el problema, que es el eje central del diagrama (una línea horizontal central), se enumeran en torno a él posibles causas que lo expliquen, que pueden tener «sub-causas». Suele emplearse en gestión de calidad.



En un diagrama de árbol se representan los medios o recursos (en las raíces) mediante los que solucionar un problema (en el tronco) y los posibles resultados (en las ramas).

Se utiliza para alcanzar conclusiones positivas desde el análisis de una situación problemática.

MÓDULO 3
AGILIDAD EN LA EMPRESA

Scrum Level

La gestión de proyectos ágil recorta el tiempo para la entrega de valor y mejora la satisfacción de los clientes, así como la implicación de los miembros del equipo, la productividad, calidad y aprendizaje (versionone 2017).

Estos resultados atraen la atención de las empresas hacia la agilidad, deseosas de trasladar los mismos beneficios a toda su organización.

Pero “agilizar” una empresa va más allá de la implantación de prácticas ágiles en todos los departamentos, o de la aplicación de un marco de agilidad para equipos a escala XXL.

La estructura y organización de las personas resultan más complejas a nivel de empresa que a nivel de equipo, porque al aumentar la estructura, aumentan también las implicaciones de la agilidad sobre la gobernanza, organización y cultura. Por eso aplicar en toda la empresa un marco de prácticas ágiles, igual que si se tratara de un equipo, no suele producir buenos resultados.

Desde esta realidad, Scrum Level ayuda a comprender y atender de forma global y estructurada las implicaciones que tiene la agilidad en las tres dimensiones de la empresa: operativa, estructural y cultural.

El modelo está desarrollado y avalado por Scrum Manager, que desde su origen ha difundido la agilidad más allá del conocimiento operativo de las prácticas ágiles, mostrándola desde la perspectiva global y sistémica de la organización, como facilitadora de su desarrollo fructífero (Scrum Manager 2017).

Scrum Level desarrolla esta idea, integrando los aportes que en las últimas décadas han extendido las posibilidades de aplicar tácticas ágiles en ámbitos ajenos a la gestión de proyectos.

Diferencias entre la gestión de proyectos ágil o evolutiva y la tradicional o predictiva.

En el siglo XX el objetivo de la industria era la fabricación de productos estandarizados de forma eficiente.

El cambio de siglo ha traído productos y servicios que, más allá de la eficiencia lo que necesitan es innovación y mejora del valor en ciclos cortos, atendiendo al mismo tiempo a aspectos como la variedad y la personalización.

Este cambio ha introducido proyectos con requisitos inestables, que crecen y evolucionan mientras se desarrolla lo que nunca es un producto final, sino siempre una nueva versión.

Se necesitan enfoques de diseño, gestión y producción tolerantes a cambios y por tanto, diferentes del modelo de gestión tradicional, basado en planificaciones cerradas resistentes a los cambios.

La gestión predictiva tiene como objetivo entregar un producto completo y terminado, cumpliendo con los costes y las fechas estimadas.

La gestión evolutiva tiene como objetivo entregar lo antes posible un producto suficiente o “producto mínimo viable” y sobre él mantener un ritmo continuo de evolución y mejora.

El diagrama de la ilustración 1 sintetiza las principales características de cada modelo, dando perspectiva a sus objetivos y diferencias.

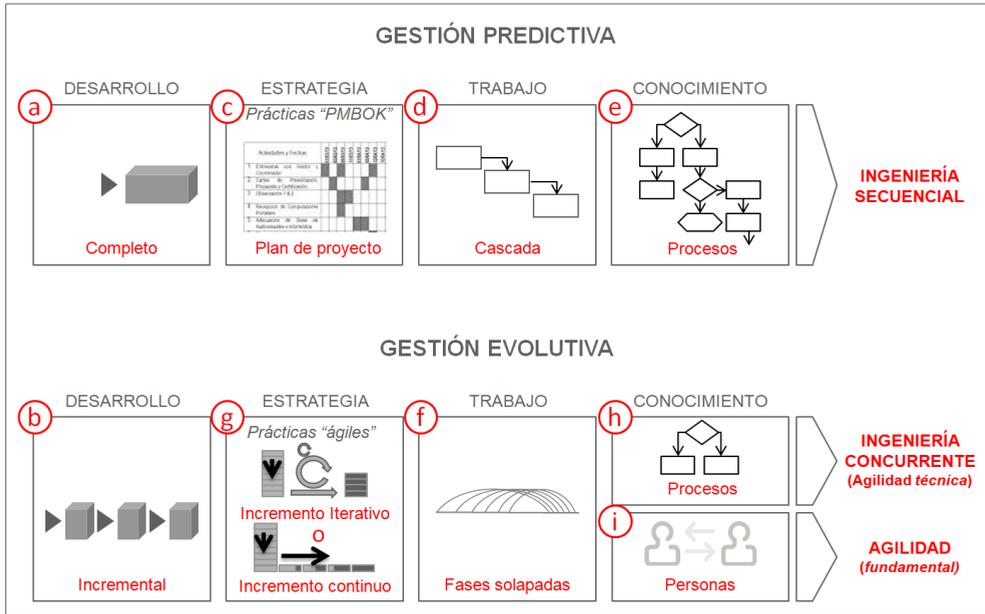


Ilustración 42: diferencias entre la gestión de proyectos predictiva y la evolutiva o ágil.

El objetivo de la gestión predictiva es entregar un producto terminado en las fechas y con los costes previstos (a).

El objetivo de la gestión evolutiva es entregar lo antes posible un producto mínimo viable e incrementarlo y mejorarlo de forma continua (b).

La gestión predictiva planifica el trabajo (c) a través de etapas secuenciales especializadas (d) como la toma de requisitos, análisis, construcción, integración y pruebas. Garantiza la calidad del resultado por la calidad de los procesos empleados (e). Este modelo de desarrollo se denomina ingeniería secuencial.

Los proyectos evolutivos no planifican fases separadas de toma de requisitos, análisis, desarrollo y pruebas. En los proyectos evolutivos se solapan estas fases, porque estas tareas se llevan a cabo de forma concurrente, atendiendo al flujo continuo de modificaciones y mejoras en las funcionalidades del producto (f).

Los ciclos de vida incrementales con fases solapadas, que emplean de técnicas ágiles (g) como pilas de producto, sprints, tableros kanban, etc. se suelen considerar "gestión ágil", pero hay un elemento que diferencia si estamos en una ingeniería concurrente que emplea prácticas ágiles o en agilidad propiamente dicha: ¿Dónde radica el conocimiento que determina la calidad del resultado?

¿En los procesos empleados (h) o en el conocimiento de las personas que lo desarrollan? (i).

En el primer caso podemos hablar de “agilidad técnica” o “agilidad operativa”, que en definitiva es aplicar ingeniería concurrente empleando técnicas propias de los equipos ágiles.

En el segundo caso sí que se trata de agilidad propiamente dicha, o “agilidad completa” para diferenciarla.

 https://youtu.be/_-BtfKHASsY

A qué nos referimos con “scrum”

En 1986, el estudio “The New New Product Development Game” (Takeuchi & Nonaka 1986) analizaba la nueva forma de organización del trabajo que estaba obteniendo los mejores resultados de innovación y tiempo de salida al mercado. Un novedoso modo de desarrollo con equipos autoorganizados al que compararon con la formación de rugby denominada *scrum*.

Durante la década siguiente algunas compañías de software empezaron a explorar prácticas en la misma línea, que surgía como antítesis a los procedimientos de ingeniería y desarrollo tradicionales que no terminaban de solucionar los problemas específicos de los proyectos de software.

Los principios de estas nuevas prácticas de trabajo se sintetizaron en 2001 en el “Manifiesto for Agile Software Development”, comúnmente denominado “Manifiesto Ágil” (Beck & Grenning & otros 2001) que define los 4 axiomas clave del nuevo modo de trabajo:

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 sobre procesos y herramientas
Software funcionando sobre documentación exhaustiva
Colaboración con el cliente sobre negociación contractual
Respuesta ante el cambio sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

Desde entonces la agilidad viene demostrando su potencial en el desarrollo de productos o servicios de forma evolutiva, con la implicación de equipos motivados y autoorganizados.

Su popularización ha trascendido el ámbito del equipo y los modelos de referencia evolucionan hacia marcos que escalan la agilidad para abarcar a organizaciones enteras, o hacia modelos orientados a otro tipo de actividades diferentes al desarrollo de software como marketing o educación.

De esta forma, el escenario de modelos y marcos de agilidad se va enriqueciendo. La finalidad de este libro es ofrecer los conceptos clave para clasificar e identificar las circunstancias en las que resulta más apropiado uno u otro modelo de gestión, así como ampliar el conocimiento para decidir estrategias particulares de implantación o mejora de agilidad.

Lo que entendemos por “*scrum*”

1.- Rugby

Scrum es el término que define a la formación del rugby, en la que ambos equipos, agazapados y atenazados entre sí, empujan para obtener el balón y sacarlo de la formación sin tocarlo con la mano.



Ilustración 43: formación de scrum en rugby.

2.- Métodos de trabajo

Ambiente de trabajo ágil autoorganizado

En 1986 los investigadores Nonaka y Takeuchi dieron dimensión polisémica al término deportivo, al emplearlo para denominar los principios de desarrollo que descubrieron en las empresas tecnológicas más innovadoras (Takeuchi & Nonaka 1986)



Ilustración 44: Ikujiro Nonaka & Hirotaka Takeuchi.

Scrum, en la concepción original de Nonaka y Takeuchi, se caracteriza por el protagonismo de equipos brillantes, autoorganizados y motivados, que abordan el desarrollo de sistemas complejos partiendo de una visión general y solapando las fases del desarrollo.

Metodología para desarrollo de software

En 1995 Ken Schwaber presentó en OOPSLA (Conferencia anual “Object-Oriented Programming, Systems, Languages & Applications”) (Schwaber 1995) una metodología de desarrollo de software, basada en un ambiente *scrum* y usó ese mismo término para definir el proceso.

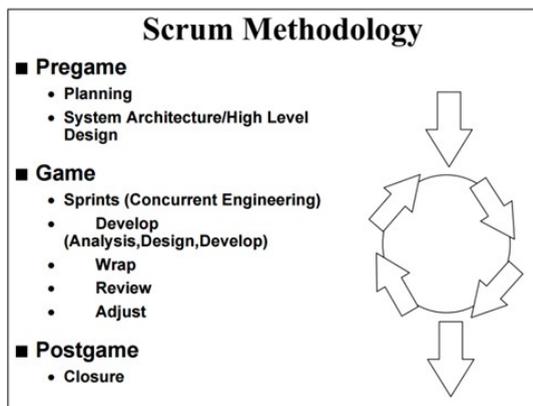
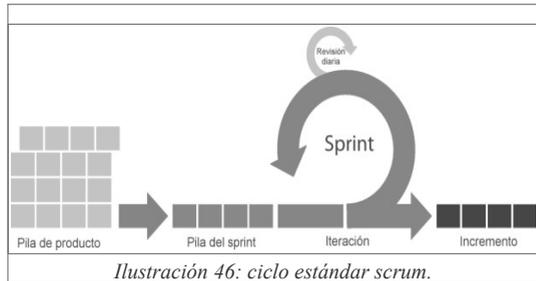


Ilustración 45: metodología scrum en OOPSLA 1995.

En 2005 Mike Cohn, Esther Derby y Ken Schwaber constituyeron la organización “Scrum Alliance” para difundir un marco de trabajo específico para el desarrollo de software, basado en esta metodología, a la que siguieron denominando *scrum*.



Scrum Level emplea el término en el significado original de Nonaka y Takeuchi, como un ambiente de trabajo caracterizado por equipos autoorganizados que trabajan de forma ágil: con autonomía y solapamiento de las fases de desarrollo y compartiendo el conocimiento y aprendizaje de forma abierta.

Paradigmas culturales de las organizaciones

El objetivo de algunas empresas es producir el máximo beneficio para sus propietarios o accionistas, pero también las hay que incluyen en su foco de interés el beneficio para grupos como clientes, empleados, para la comunidad o el medio ambiente.

Unas se organizan con estructuras verticales o piramidales y otras se despliegan en organizaciones planas. De esta forma cada empresa se desenvuelve con una “personalidad” que es un reflejo de su cultura.

Resulta aconsejable conocer la evolución, características e implicaciones de los diferentes paradigmas culturales, de forma que podremos identificar a qué paradigma pertenece nuestra organización y comprender mejor las implicaciones organizacionales de los posibles cambios, en la medida en la que la gestión ágil puede desbordar el ámbito meramente operativo e implicar cambios culturales.

Los paradigmas culturales componen el mapa de situación para comprender las tensiones que pueden generar los procesos de implantación o mejora de la agilidad cuando ésta afecta, además de a la dimensión operativa, a las dimensiones cultural y estructural.

El modelo de referencia de Scrum Level es el desarrollado por Frederic Laloux, expuesto en su libro “Reinventar las organizaciones” (Laloux 2016)² En él se describe la evolución de los valores culturales con los que la humanidad ha ido creando los diferentes modelos de organización, desde los prehistóricos “infrarrojo y magenta” hasta los 5 posteriores que centran nuestra atención, porque son los que modelan las organizaciones actuales; los denominados rojo, ámbar, naranja, verde y esmeralda o *teal*.

📺 La emergencia de un nuevo paradigma organizacional - <https://youtu.be/NrSABjCIgVc>

📺 Paradigmas culturales de las empresas https://youtu.be/_BtfKHASsY

2 El modelo de Laloux se inspira en el código de 8 estados desarrollado por Don Beck (Beck & Cowan 1996) basado a su vez en la teoría de la Dinámica Espiral (Spiral Dynamics) de Clare W. Graves (Graves 2004).

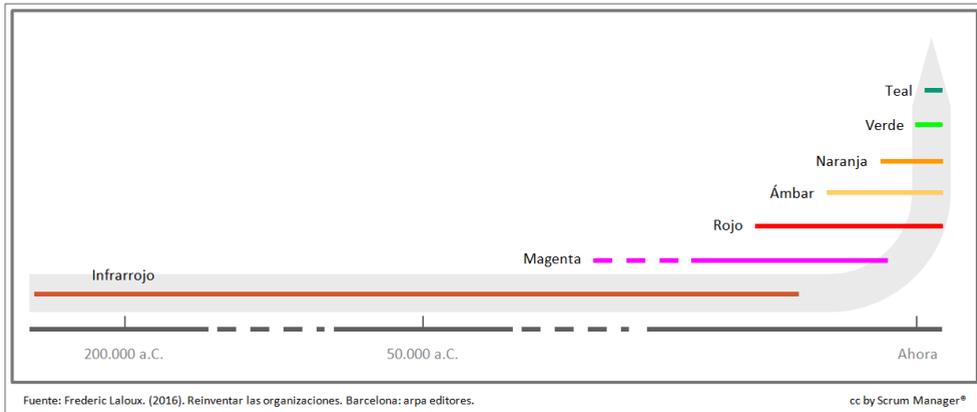


Ilustración 47: paradigmas culturales de las organizaciones a lo largo de la historia.

Paradigma impulsivo, o “rojo”

Hace unos 10.000 años surgieron las primeras formas de organización con un paradigma cultural “impulsivo” o “rojo”.

En este modelo los hombres se mueven en un mundo amenazante y peligroso en el que la fuerza es lo más valioso. Los poderosos exigen y los menos poderosos se someten a cambio de seguridad.

Este modelo generó organizaciones con jefes y soldados o siervos de a pie. Sus principios aún se encuentran presentes en sociedades tribales y también en sectores paupérrimos o marginales de la sociedad actual, en forma de pandillas callejeras o mafias. En este tipo de organizaciones el jefe mantiene su autoridad demostrando un poder absoluto y despiadado, capaz de doblegar la voluntad de cualquiera. Las relaciones emocionales entre las personas son muy toscas y la empatía escasa o nula.

Produce organizaciones muy reactivas a nuevas oportunidades y amenazas, pero al mismo tiempo muy cortoplacistas, por lo que se adaptan bien en entornos caótico-catastróficos, pero no resultan útiles para desarrollar resultados complejos en entornos estables.

Paradigma conformista o “ámbar”

Hace unos 5.000 años comenzamos a comprender la causalidad y hacer planificaciones a futuro. Surge la agricultura, que requiere anticipación y proceso: recoger y conservar las semillas para garantizar la próxima cosecha, y los cacicazgos van dando paso a una sociedad de clases sociales, rígidamente estratificada: dirigentes, funcionarios, sacerdotes, guerreros y artesanos.

Frente al oportunismo y cortoplacismo de las organizaciones rojas, las organizaciones ámbar son apropiadas para contextos estables en los que se puede planificar el futuro sobre la base de la experiencia pasada. Pueden planificar y poner en marcha proyectos a medio y largo plazo, como edificar catedrales o crear extensos y complejos circuitos comerciales.

Trabajan sobre los axiomas de que el mundo es inmutable y sólo hay una forma correcta de hacer las cosas, de forma que lo que funcionó bien en el pasado también lo hará en el futuro, por lo que son organizaciones que no aceptan el cambio.

Los valores de orden y predictibilidad que emanan de las organizaciones ámbar hacen que se perciban como refugios seguros, donde no es necesario protegerse de amenazas o imprevistos. En ellas el poder se encuentra jerarquizado en organigramas piramidales inamovibles con estratos de jefes y subordinados.

La presencia de estratos y castas con títulos formales fomenta la adopción de máscaras sociales. Las organizaciones ámbar han inventado los rangos y uniformes para institucionalizar y marcar las diferentes funciones. La imagen y vestimenta de las personas refleja su identidad funcional dentro de la organización, y en la misma línea las personas adoptan y muestran comportamientos adecuados a su casta.

Paradigma logro o “naranja”

El paradigma naranja surgió cuando empezamos a considerar que no estábamos en un universo fijo ordenado por leyes inmutables, sino en un sistema que funciona como un mecanismo complejo, cuya articulación interna es posible estudiar y comprender.

Al entender mejor la forma en la que operan las cosas, se pueden alcanzar más logros.

Este paradigma ha dado como frutos la investigación científica, la innovación y el emprendimiento. Es el más extendido actualmente entre empresarios y políticos. En los dos últimos siglos ha alargado la esperanza de vida y producido niveles de prosperidad hasta ahora desconocidos.

Las organizaciones naranjas han aportado tres avances importantes en el modelo organizativo sobre las ámbar: la innovación, la rendición de cuentas y la meritocracia.

Este paradigma también tiene asociadas facetas menos virtuosas, tales como: codicia corporativa, cortoplacismo estratégico y político, sobre-consumo, endeudamiento y sobreexplotación de recursos.

Paradigma pluralista o “verde”

El paradigma pluralista (verde) surge como antítesis a las sombras del paradigma del logro (naranja): desigualdad social, obsesión materialista y pérdida del sentido de comunidad.

Actualmente el paradigma naranja es el que predomina en la política y las empresas. El verde se está abriendo camino en organizaciones sin ánimo de lucro, entre activistas, trabajadores sociales y en general en organizaciones de personas que operan valorando las relaciones por encima de los resultados.

Aporta alternativas a los modelos organizacionales naranjas como son la descentralización, el empoderamiento, la cultura impulsada por valores y el foco en múltiples grupos de interés: además de los accionistas también proveedores, clientes, comunidades locales, medio ambiente, etcétera.

La relevancia de este paradigma, al menos en su inicio, se debe más a su carácter rompedor con las viejas estructuras que a la generación de alternativas prácticas. Frederic Laloux (Laloux 2016) afirma:

“A posteriori, sabemos que estas formas extremas de organización igualitaria no han tenido éxito ni a escala significativa ni por un lapso significativo de tiempo. Lograr el consenso entre un grupo grande de personas es intrínsecamente difícil. Muchas veces la iniciativa termina en extenuantes sesiones de conversación y un eventual punto muerto. A modo de respuesta se cuelan los juegos de poder en un intento de que las cosas vuelvan a moverse”.

Los marcos de agilidad técnica como *scrum* e de agilidad organizativa como la gobernanza dinámica o sociocracia trabajan en este sentido, para lograr eficiencia y resultados prácticos en organizaciones verdes. En algunos casos también se emplean para conducir la evolución hacia el paradigma *teal*.

Paradigma evolutivo o *teal*

La evolución a una cultura *teal* (también llamada esmeralda) implica la desidentificación de nuestro propio ego. El éxito, el reconocimiento o la riqueza son trampas para el ego. En la cultura *teal* el éxito o el reconocimiento no son el objetivo. El objetivo es una vida bien vivida y es posible que la consecuencia de ello sea éxito, reconocimiento, riqueza y amor.

Las tres aportaciones de las organizaciones evolutivas – *teal* son:

- **Autogestión:** la cultura verde aporta empoderamiento, pero el empoderamiento implica que un gestor de rango superior delega parte de su poder, mientras que para *teal* la autoridad no forma parte de un sistema de suma cero. En las estructuras de las organizaciones *teal* no hay miembros con poder de decisión y miembros ejecutores. Por naturaleza, y no por delegación, todos tienen poder de decisión y la estructura de la organización incluye procesos holocráticos para la regulación del flujo de información y decisiones.

“Cada decisión que se toma en la sede central le quita responsabilidad a la gente de otras partes de la organización y reduce el número de personas que sienten que están haciendo una contribución real a la organización.”

Dennis Bakke

- **Plenitud:** en las organizaciones las personas se muestran con una máscara profesional adecuada a las expectativas de su función y lugar de trabajo: abogado, médico, ingeniero, mecánico, oficinista, director, etc. En las organizaciones *teal* no hay ascensos por los que pelear ni jefes a los que contentar. Las personas se muestran de forma plena, siendo ellos mismos. Son organizaciones que invierten mucho tiempo en la selección de personal, informando a los candidatos de los valores y formas de trabajo para que puedan decidir si quieren ser parte de la organización o no. En este sentido es conocido el caso de Zappos.com, que ofrece a los recién incorporados un cheque de 3.000 dólares si se arrepienten durante el periodo de prueba y prefieren abandonar la empresa.

- **Propósito evolutivo:** para las organizaciones *teal*, el beneficio es un subproducto de un trabajo bien hecho. La finalidad de la organización no es el valor para los accionistas ni maximizar las ganancias, sino su propósito evolutivo. Brian Robertson, fundador de Holacracy emplea la metáfora “padre-hijo” para explicar el concepto:

¿Cuál es la identidad de la organización? ¿Y qué es lo que desea?... La metáfora es como el viaje padre-hijo: reconocemos que nuestro hijo tiene su propia identidad camino y propósito. Y si realmente me entusiasma que llegue a ser médico, no lo programo. Si lo hago, tiene lugar un proceso dañino y codependiente. ... Pero la clave tiene que ver con separar identidades y descubrir cuál es la vocación de la organización. No para qué queremos usar la organización en tanto que propietarios, sino más bien ¿cuál es el potencial creativo de esta vida, de este sistema viviente?

Paradigmas culturales de las organizaciones					
	Reactivo (rojo)	Conformista (ámbar)	Logro (naranja)	Pluralista (verde)	Evolutivo (teal / esmeralda)
Resumen	Dirección por poder. Aglutinante: miedo. Reactiva. Foco a corto plazo. Prospera en entornos caóticos.	Formalización funcional. Mando y control vertical. Se valora la estabilidad a través de procesos rigurosos.	Foco en beneficios y crecimiento vendiendo a la competencia. Valor clave: innovación. Gestión por objetivos.	Focalizado en el empoderamiento y motivación de las personas.	Desarrollo del potencial holacrático: de las personas de la organización y de la comunidad.
Modelo organizativo	Piramidal, dictadura.	Piramidal, jerarquía de castas.	Piramidal, meritocracia.	Híbridos.	Fractal holocracia.
Ejemplos	<ul style="list-style-type: none"> ▪ Mafia. ▪ Pandillas callejeras. ▪ Tribus urbanas. 	<ul style="list-style-type: none"> ▪ Iglesia. ▪ Fuerzas armadas. ▪ Gobiernos. ▪ Educación pública. 	<ul style="list-style-type: none"> ▪ Multinacionales / empresa tradicional. ▪ Educación privada. 	<ul style="list-style-type: none"> ▪ Organizaciones ágiles. 	<ul style="list-style-type: none"> ▪ Organizaciones holocráticas
Aportaciones	<ul style="list-style-type: none"> ▪ División del trabajo. ▪ Autoridad de mando. 	<ul style="list-style-type: none"> ▪ Funciones formales. ▪ Procesos. 	<ul style="list-style-type: none"> ▪ Innovación. ▪ Responsabilidad. ▪ Meritocracia. 	<ul style="list-style-type: none"> ▪ Empoderamiento. ▪ Valores culturales. ▪ Modelo de grupos de interés. 	<ul style="list-style-type: none"> ▪ Autogestión. ▪ Plenitud. ▪ Propósito.
Metáfora	Manada de lobos	Ejército	Máquina	Familia	Organismo

Fuente: Frederic Laloux. (2016). Reinventar las organizaciones. Barcelona: arpa editores.

cc by Scrum Manager®

Ilustración 48: características de las organizaciones, según su paradigma cultural.

RAZÓN DE SER DE LA EMPRESA	Beneficio económico Accionistas / propietarios	Beneficio a grupos de interés. Accionistas, proveedores, empleados, comunidad local, medio ambiente...	Propósito evolutivo. La propia empresa como organismo parte de un sistema global
ESTRATEGIA	Métodos y procesos		Personas
PARADIGMA CULTURAL	DEL LOGRO Naranja	PLURALISTA Verde	EVOLUTIVO Teal / Esmeralda
MODELO ORGANIZATIVO	Piramidal, meritocracia.	Híbridos.	Fractal, sociocracia.
EJEMPLOS	<ul style="list-style-type: none"> ■ Multinacionales/ empresa tradicional. 	<ul style="list-style-type: none"> ■ Organizaciones ágiles. 	<ul style="list-style-type: none"> ■ Organizaciones holocráticas
APORTACIONES	<ul style="list-style-type: none"> ■ Innovación. ■ Responsabilidad. ■ Meritocracia. 	<ul style="list-style-type: none"> ■ Empoderamiento. ■ Valores culturales. ■ Modelo de grupos de interés. 	<ul style="list-style-type: none"> ■ Autogestión. ■ Plenitud. ■ Propósito.
METÁFORA	Máquina.	Familia.	Organismo.

Fuente: Frédéric Laloux. (2016). Reinventar las organizaciones. Barcelona: arpa editores.

cc by Scrum Manager®

Ilustración 49: principales características de las organizaciones, según su paradigma cultural.

Dimensiones de la empresa y agilidad

Las empresas son sistemas complejos en los que se combinan factores desde dos facetas diferentes:

- Operativa, encargada de realizar los productos o servicios de la compañía.
- Organizativa que determina los principios culturales y la forma estructural de la empresa.

A menudo se aborda el escalado de agilidad de forma conjunta en las dos facetas, introduciendo modificaciones tanto en la gestión de proyectos y procesos de desarrollo, como en la cultura y modelo de gobernanza.

Al considerar que la agilidad implica cambios operativos y organizativos de forma simultánea, no se valora por separado:

- a) Si el producto, servicio o proyecto de la empresa se puede construir de forma evolutiva y si el hacerlo proporciona ventajas a los clientes o a la comercialización.
- b) Si la propiedad de la empresa desea cambios en la estructura o en el modelo de gobernanza y si es consciente de las implicaciones de esos cambios.

En las empresas que no necesitan o no pueden entregar a sus clientes resultados incrementales, la agilidad, de tener sentido, tiene su campo en en la faceta organizativa más que en la operativa. Es el caso, por ejemplo, de AES (sector energético, 40.000 empleados), Heiligenfeld (Hospitales de salud mental, 600 empleados) o Zappos.com (venta de zapatos *on line*, 1.500 empleados).

Si el objetivo de la empresa se centra en el beneficio de los accionistas y si además basa el *know-how* de sus productos en ingeniería de procesos antes que en el conocimiento tácito de las personas, el escalado de la agilidad fracasará si, más allá de la faceta operativa, intenta abordar también cambios de calado en la faceta organizativa.

Al no considerar cada faceta por separado se cierra la posibilidad de adoptar prácticas de gestión evolutiva e ingeniería concurrente sin tener que realizar cambios estructurales y culturales de calado.

Y en sentido contrario, también se hace imposible la introducción de cambios culturales o estructurales sin necesidad de adoptar también prácticas de agilidad técnica, que pueden resultar inadecuadas para el tipo de producto o servicio desarrollado.



Ilustración 50: agilidad y dimensiones de la empresa.

Por qué quiere ser ágil la empresa y cómo lograrlo

Es necesario comprender las razones por las que una empresa decide escalar la agilidad para orientar la estrategia de la forma más adecuada.

Por razones operativas

Para ofrecer respuesta ágil: entrega temprana y continua de valor.

Si este es el motivo, o uno de los motivos, se debe institucionalizar en la empresa un marco de agilidad técnica. Lo más aconsejable es empezar con uno estándar y, en base a la experiencia ir acomodándolo a las características de la organización.

El marco más apropiado para pequeñas empresas, posiblemente sea el de *scrum* técnico.

Para organizaciones más grandes los marcos de referencia más difundidos son:

- LeSS (Large-Scale Scrum).
- Nexus.
- SAFe (Scaled Agile Framework).
- Scrum of Scrums.

Para crear entornos que potencien la creación de valor basados en las personas y su motivación.

Las estrategias para llevar agilidad a la faceta organizativa de la empresa se centran en su estructura y cultura.

En el primer aspecto se trata de pasar de estructuras verticales y piramidales a estructuras planas y fractales, que hacen posible el cambio de gestión dirigida a autogestión y de dirección a autonomía.

Los modelos de referencia más difundidos en este ámbito son:

- Sociocracia o gobernanza dinámica.
- Holocracia.
- Equipos fractales.
- Sociocracia 3.0.
- Holacracy®.

Para escalar la agilidad a proyectos grandes con varios equipos implicados.

Actualmente los proyectos y productos complejos requieren varios equipos lo que implica prácticas a escala para cubrir:

- Colaboración entre equipos.
- Gestión de dependencias entre equipos y proyectos /productos.
- Roles coordinadores.
- Eventos de alineación y sincronización.

Por razones estéticas

Por moda, *marketing*, o por exigencias contractuales.

Este libro no trata de comunicación, pero en cualquier caso desaconseja la implantación simulada de modelos de gestión ágil si las razones estéticas son las únicas. Aparentar una agilidad que no se posee es una estrategia de gerencia miope, además de deshonesto.



Ilustración 51: estrategias de escalado de la agilidad.

Consideraciones

Agilidad en la faceta operativa

Se debe considerar si el producto o propósito de la organización resulta más valioso producido de forma evolutiva, con un: lanzamiento temprano y crecimiento iterativo e incremental.

Los principales riesgos al escalar la agilidad en la faceta operativa de la organización son:

- Falta de apoyo de la dirección.
- Falta de formación o conocimiento insuficiente de agilidad.
- Falta de alineación e implicación del cliente.

Agilidad en la faceta organizativa

Se debe considerar especialmente en empresas que usan el conocimiento para generar valor, frecuentemente denominadas “empresas del conocimiento”. Abarcan campos como educación, investigación y desarrollo, alta tecnología, informática, comunicaciones, robótica, nanotecnología, etc.

Es muy aconsejable que la propiedad de la organización no sea reacia a desarrollar una cultura verde o *teal*.

Los principales riesgos de introducir valores ágiles en la faceta organizativa son:

- Que la propiedad de la empresa no comparta los cambios en la estructura o en el modelo de gobernanza.
- Falta de apoyo de la dirección.
- Falta de formación acerca de los principios y operativa del modelo de gobernanza.
- Resistencia al cambio o manifiesta oposición de responsables de área hacia la nueva estructura y procesos de gobernanza.

Grado de implantación

Algunos modelos estándar, como Holacracy® o *scrum* técnico, recomiendan aplicarse de forma rígida y completa, afirmando que no es posible o recomendable usar sólo determinadas prácticas, o en combinación con procedimientos de otros modelos, o propios de la organización.

Otros modelos como Sociocracia o SAFe consideran opcionales todos sus componentes.

Lo cierto es que la flexibilidad en la construcción de la propia cultura y marco de producción es la mejor fórmula, siempre desde un criterio documentado y experto.

Compatibilidad con la cultura de la empresa

La agilidad técnica trabaja con ciclos de vida evolutivos en lugar de predictivos y no emplea ingeniería secuencial, sino ingeniería concurrente para realizar entregas evolutivas e incrementales, en lugar de predefinidas y completas.

Su ámbito de actuación por tanto se limita a prácticas de gestión de proyectos y de ingeniería, que pueden aplicarse independientemente del paradigma cultural de la organización. Sin embargo no ocurre lo mismo con la agilidad organizativa. Ésta necesita que la estructura y cultura de la empresa contemplen la autoorganización y autonomía en la toma de decisiones, de personas y equipos en un grado suficiente para no bloquear el marco de agilidad que se desea alcanzar.

PARADIGMA CULTURAL	DEL LOGRO Naranja	PLURALISTA Verde	EVOLUTIVO Teal / Esmeralda
AGILIDAD TÉCNICA	[Barra de compatibilidad: alta en Del Logro, moderada en Pluralista, baja en Evolutivo]		
AGILIDAD ORGANIZACIONAL (estructural y cultural)	[Barra de compatibilidad: moderada en Del Logro, alta en Pluralista, muy alta en Evolutivo]		
cc by Scrum Manager®	Compatibilidad		

Ilustración 52: compatibilidad entre agilidad y cultura organizacional.

Principios y valores de la agilidad

En el ámbito del desarrollo de software, los equipos ágiles gestionan los requisitos del cliente, escribiéndolos en “pilas de producto”, mientras que los equipos tradicionales usan documentos SRS (*Software Requirements Specification*).

La “pila de producto” es un formato ligero de documentación usado en gestión ágil, mientras que el “SRS” es un modelo formal para la especificación requisitos de un proyecto de software empleado en gestión predictiva. Pero la característica que diferencia a los dos modelos de gestión no es el usar pilas de producto o documentos de requisitos.

La clave no son las prácticas que emplean los modelos de gestión, sino los principios que facilitan.

En este caso, el principio ágil que hay en la base de la pila de producto es el desarrollo incremental. La agilidad produce y entrega el resultado en incrementos, siguiendo la evolución de los requisitos en paralelo. Por esa razón resultan más apropiados formatos que facilitan la adaptación continua, como las pilas de producto.

La gestión de un proyecto no es ágil por emplear prácticas ágiles, sino por aplicar principios ágiles.

Por ejemplo, la gestión de un proyecto no sería ágil por usar una pila de producto, sino por realizar el desarrollo de forma incremental.

La agilidad de una empresa en su **faceta operativa (agilidad técnica)** no se debe a las prácticas que emplea sino a los principios cuyo desarrollo facilitan esas prácticas. Cuanto más apropiadas sean las prácticas al tipo de proyecto y empresa, y cuanto mayor el conocimiento y la experiencia de las personas que las ejecutan, mayor será el grado de agilidad operativa.

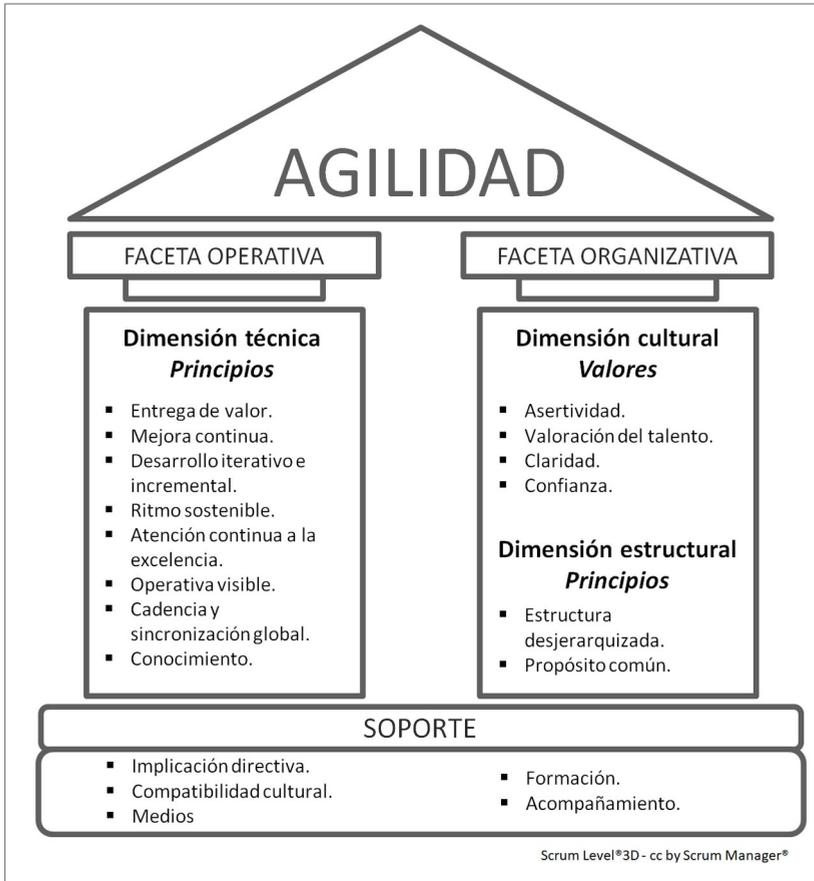


Ilustración 53: principios, valores y soportes de la agilidad en la empresa.

De forma similar, la agilidad de la empresa en su **faceta organizativa** es consecuencia de los **valores** culturales, que se manifiestan en los **comportamientos** de gestión, relación y comunicación de las personas. Asimismo también depende de los **principios** sobre los que se desarrollan su estructura y relaciones organizativas.

Y por último, lo más importante: **el desarrollo de la agilidad, tanto en la operativa como en la estructura y cultura de la organización, necesita la implicación y el apoyo de la dirección.** Si no cuenta con ellos, o peor aún, si la dirección de la empresa se conduce con patrones incompatibles, los intentos de mejora de la agilidad son inviables.

El foco de análisis al abordar la mejora de la agilidad no debe centrarse en la introducción de prácticas o marcos de agilidad, sino en los principios, valores y soportes necesarios.

Dimensiones y facetas de la empresa

En la empresa conviven dos facetas:

- La operativa, que realiza los productos o servicios de la compañía
- La organizativa, que gestiona la estructura y cultura.



Ilustración 54: dimensiones y facetas de la empresa.

Faceta operativa: principios y prácticas

La tabla 2 muestra los principios de la agilidad operativa y el tipo de prácticas habitualmente empleadas para desarrollarlos. En este capítulo se explicará cada elemento en detalle.

PRINCIPIO	PRÁCTICAS
1 ENTREGA DE VALOR	1.1 Compartir y comprender la visión del cliente.
	1.2 Colaboración del cliente con el equipo.
	1.3 Variabilidad.
2 MEJORA CONTINUA	2.1 Mejora continua de las prácticas.
3 DESARROLLO ITERATIVO E INCREMENTAL	3.1 Arranque desde un mínimo viable.
	3.2 Puntos de integración frecuentes con análisis de <i>feedback</i> para resolver y despejar el avance.
	3.3 Basar los hitos en la evaluación objetiva de partes del sistema funcionando.
4. RITMO DE TRABAJO SOSTENIBLE	4.1 Mantenimiento de un flujo de producción continuo y óptimo.
	4.2 Enfoque.
5 ATENCIÓN CONTINUA A LA EXCELENCIA	5.1 Uso de técnicas para garantizar la calidad.
6 OPERATIVA VISIBLE	6.1 Identificación temprana de impedimentos.
	6.2 Compartir información relativa al desarrollo.
7 CADENCIA Y SINCRONIZACIÓN GLOBAL	7.1 Predictibilidad de las reuniones basada en la cadencia del desarrollo.
	7.2 Predictibilidad de momentos de entregas de incrementos e integraciones.
8. PERSONAS SOBRE PROCESOS	8.1 Las personas aplican un conocimiento técnico alto en su desempeño.
	8.2 Las personas aplican las habilidades sociales necesarias para su desempeño.

Tabla 1: principios de la agilidad operativa y tipos de prácticas que los desarrollan.

1.- Entrega de valor

Es importante tener clara en primer lugar la diferencia entre valor para el negocio y valor para el cliente. Steve Jobs (Isaacson 2011) la describe así:

“Tengo mi propia teoría sobre la razón del declive en compañías como IBM o Microsoft. La compañía hace un gran trabajo, innova y se convierte en un monopolio o casi, en determinado campo y luego la calidad del producto se vuelve menos importante. La compañía comienza a valorar a los grandes vendedores, porque son ellos quienes pueden mover los indicadores de los ingresos, no los ingenieros y diseñadores de productos. De esta forma los vendedores terminan dirigiendo la compañía. John Akers en IBM era un vendedor inteligente, elocuente y fantástico, pero no sabía nada sobre el producto. Lo mismo pasó en Xerox. Cuando son los vendedores los que dirigen la empresa, los productos no importan tanto y muchos simplemente se apagan.”

El objetivo de la empresa ágil no es el valor de negocio que produce con su trabajo, sino el valor que proporciona a sus clientes, de forma que los ingresos sean la consecuencia, la medida del éxito y la confirmación de que se están haciendo las cosas correctas.

Para facilitar la institucionalización de este principio, la operativa de la empresa debe proporcionar productos o servicios empleando prácticas orientadas a:

- **Compartir y comprender la visión del cliente.**
Facilitar al equipo conocer, comprender y compartir la visión de lo que el cliente quiere conseguir.
Algunos ejemplos: *Product vision board*, pila de producto, *product vision box*, *product canvas*, *PI Planning*, *product owners board*...
- **Conexión y colaboración del cliente con el equipo.**
Facilitar que el responsable del producto o servicio, directamente o a través de un propietario de producto, colabore con el equipo en los eventos de comunicación y sincronización del desarrollo, tales como reuniones de planificación de *sprint*, de revisión, etc.
- **Gestionar la variabilidad.**
Adaptar el ritmo de desarrollo a evolución o cambios en los requisitos.
Algunos ejemplos: Especificación de requisitos en formato de historias de usuario o tareas, gestionados en pilas de producto o tableros *kanban*.

2.- Mejora continua

La finalidad de este principio es perfeccionar continuamente la eficiencia de las prácticas y métodos de trabajo.

La agilidad en la dimensión operativa implica comportamientos de autoorganización. Son los propios equipos, no departamentos de calidad o de procesos, los que llevan a cabo regularmente prácticas de análisis y mejora de sus métodos de trabajo.

La definición del principio de mejora continua en el “Manifiesto por el Desarrollo Ágil de Software” refleja muy bien su espíritu a nivel de equipo en organizaciones con agilidad organizativa:

“A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.”



Ilustración 55: mejora continua.

No hay métodos, prácticas o modelos de trabajo capaces de responder con solvencia durante mucho tiempo. El conocimiento profesional evoluciona de forma continua primero porque la realidad en la que se aplica está en evolución permanente y porque la mejora siempre es posible.

Por eso la agilidad utiliza la inspección y adaptación a través del ciclo de mejora continua y lo hace en dos niveles: primero en la forma de trabajar y segundo en el producto.

En definitiva, la empresa ágil es una organización en aprendizaje continuo. Para facilitar la institucionalización de este principio, los equipos deben realizar periódicamente prácticas para:

■ **Revisar las técnicas y métodos de trabajo.**

Localizar puntos de mejora en la forma de trabajar.

Algunos ejemplos: Reuniones retrospectivas (a nivel de equipo, departamental y organizacional). Talleres de resolución colaborativa de impedimentos organizacionales.

■ **Mejora continua del producto o servicio.**

Algunos ejemplos:

Ciclo de desarrollo incremental con periodos de integración breves.

Reuniones de revisión de producto en los puntos de integración.

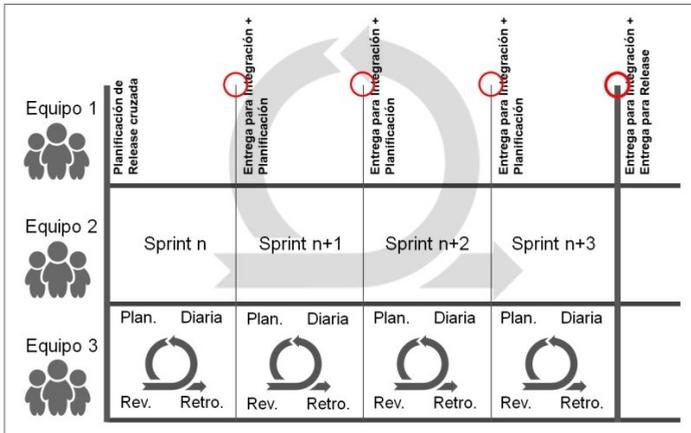


Ilustración 56: sincronización y mejora continua.

3.- Desarrollo iterativo e incremental

Los proyectos ágiles comienzan lanzando una parte mínima, útil para el cliente, que suele denominarse *producto (o servicio) mínimo viable* y sobre la que llevan a cabo incrementos continuos y frecuentes.

El desarrollo incremental no necesita contar con el mejor diseño por adelantado; la predisposición a la variabilidad permite evolucionar y mejorar durante el desarrollo.

A cada paso se obtiene retroinformación real y objetiva de los usuarios, que aportan mejoras continuas de diseño y funcionalidad.

Los hitos de avance del proyecto deben basarse en logros significativos.

La institucionalización de este principio en el ciclo de vida de los productos o servicios de la empresa se puede evaluar por la presencia y capacidad de las prácticas empleadas para:

- **Arrancar desde un mínimo viable.**
El modelo o marco empleado en la organización para la gestión de proyectos incluye el concepto de mínimo viable como hito de lanzamiento.
- **Puntos de integración frecuentes con análisis del feedback para resolver y despejar el avance.**
Durante la construcción del producto son continuos los puntos de integración tanto a nivel de prueba de subsistemas como de demostración de la solución completa.
- **Basar los hitos en la evaluación objetiva de partes terminadas.**
La gestión de proyectos no establece hitos basados en fases de desarrollo proyectados desde una planificación inicial: requisitos, diseño, desarrollo, integración, sino que los éstos están relacionados con los ciclos de incremento propios del desarrollo ágil.

4.- Ritmo de trabajo sostenible

En las empresas del conocimiento, los proyectos suelen avanzar con un ritmo más o menos distendido que se va tensando al acercarse las fechas de entrega. Esto supone un problema, ya que la productividad y la calidad del trabajo creativo disminuyen en situaciones de presión o ansiedad.

La evaluación de este principio se realiza observando la presencia de prácticas para

- **Mantener un flujo de trabajo continuo y óptimo.**
El ciclo de desarrollo de la organización emplea ciclos breves de producción de incrementos (*sprints*) o técnicas de gestión ágil, que como *kanban* facilitan el mantenimiento de un flujo de producción continuo.
- **Enfoque**
La productividad se degrada al mantener la atención del trabajo en múltiples asuntos de forma simultánea. Mantener el foco del trabajo en un único objetivo facilita una “velocidad de crucero” en el avance de las tareas y es una estrategia adecuada para el desarrollo de soluciones complejas.

5.- Atención continua a la excelencia.

El Manifiesto Ágil afirma en uno de sus principios que: “La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.”

La calidad es imprescindible tanto en el plano funcional o externo, para satisfacer las expectativas de los usuarios, como en el plano técnico o interno para permitir la evolución en iteraciones continuas. Esto es debido a que la facilidad con la que un producto se adapta al cambio es directamente proporcional a la simplicidad y calidad de su diseño e inversamente proporcional a la deuda técnica adquirida.

Para evaluar la mayor o menor institucionalización de este principio en la empresa se deben identificar herramientas y métodos de trabajo capaces de filtrar y detectar errores en el momento de producirse. Éstos suelen ser específicos para cada industria.

En el caso del software son habituales: *TDD* o “desarrollo guiado por pruebas,” (*Test-Driven Development* en inglés) integración continua, programación en pareja, refactorización, propiedad colectiva del código o simplicidad en el diseño.

6.- Operativa visible

Este principio facilita la detección temprana de problemas e impedimentos y consiste en que toda la información relativa a la funcionalidad y desarrollo del producto se encuentre accesible a todo el equipo de forma clara y directa.

El grado de transparencia se observa y se puede mejorar a través de prácticas para:

- **Identificación temprana de impedimentos**
Mediante revisiones diarias o *stand-up*, el uso de pizarras o tableros *kanban* para compartir en equipo la información relativa al desarrollo, unido con los ciclos cortos de *feedback* propios de la gestión ágil son ejemplos de este tipo de prácticas.
- **Compartir información y estado de desarrollo**
A través del uso de tableros *kanban*, reuniones de producto y protocolos de participación asertiva y abierta.

7.- Cadencia y sincronización global

La agilidad escalada a nivel de empresa debe lograr que los equipos trabajen como un único organismo. Deben compartir la visión, la hoja de ruta y estar alineados en la estrategia de trabajo.

Cuando varios equipos trabajan en productos o servicios relacionados, cada uno debe participar en una planificación cruzada con el resto, para producir de forma sincronizada. Técnicas como la sincronización de *sprints* permiten integrar los incrementos de cada equipo, asegurando que lo que se ha construido entre todos ensambla y funciona conjuntamente.

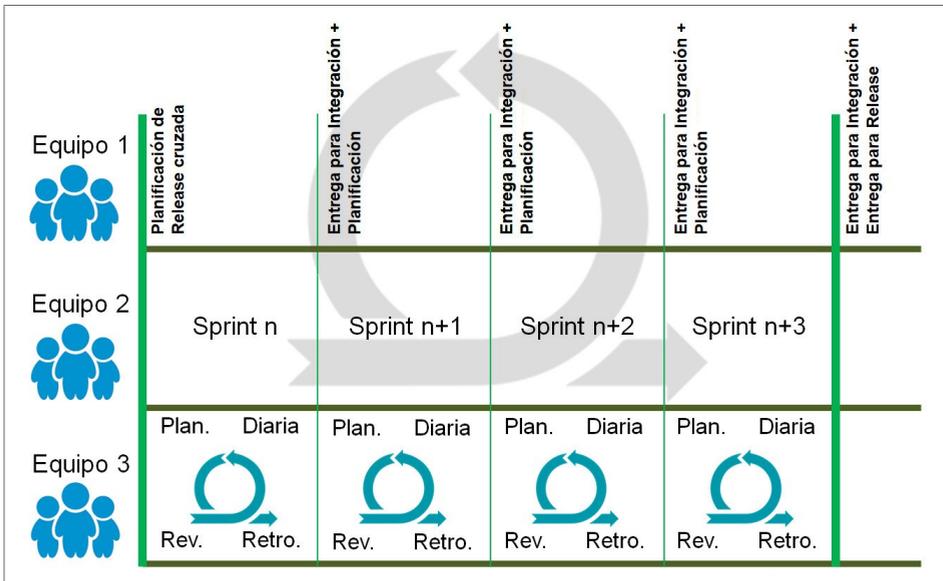


Ilustración 57: cadencia y sincronización de equipos.

Se debe evaluar la implantación de este principio en la empresa, así como su capacidad y posibilidades de mejora, analizando si las prácticas empleadas en el ciclo de vida de los productos y servicios cubren los siguientes puntos:

- **Cadencia y sincronización a través de planificaciones cruzadas.** La cadencia hace que las reuniones sean previsibles y sea posible coordinar agendas. Por otro lado reduce la variabilidad ya que la limita el tamaño de las iteraciones, proporcionando ritmo al desarrollo.

- **Predictibilidad de momentos de entregas de incrementos e integraciones.**

La combinación de cadencia, sincronización y planificación, es necesaria para construir de forma efectiva dentro de un marco de producto cambiante.

8.- Personas sobre procesos

El Manifiesto Ágil en su primer principio, afirma que valora más a los individuos y sus interacciones que a los procesos y herramientas. Esto es: la agilidad confía el valor del resultado más al conocimiento tácito de las personas que lo producen y la inteligencia colectiva que generan en equipo, que a los procesos y herramientas que utilizan.

La medida en la que las personas y su interacción aportan valor al resultado, se evalúa de forma indirecta, observando el nivel de conocimiento profesional y de habilidades sociales (*hard skills* y *soft skills*).

Faceta organizativa: valores y principios

Una empresa puede mejorar su agilidad operativa, adoptando prácticas de gestión evolutiva e ingeniería concurrente, sin realizar cambios significativos estructurales o culturales.

Al contrario, también es posible desarrollar la agilidad en la faceta organizativa, sin necesidad de adoptar prácticas de producción incrementales, si el tipo de producto o servicio que produce la empresa no lo requiere.

La faceta organizativa forma la “personalidad” de la organización y se compone de dos dimensiones: la cultural y la estructural.

Las siguientes tablas muestran los valores culturales y principios organizativos que son la esencia de una organización ágil y que se desarrollarán a continuación

DIMENSIÓN CULTURAL

VALOR	COMPORTAMIENTO
1. Asertividad	1.1 Coraje. 1.2 Respeto.
2. Valoración del talento	2.1 Incorporación y desarrollo. 2.2 Retención. 2.3 Desarrollo profesional.
3. Claridad	3.1 Transparencia. 3.2 Honestidad.
4. Confianza	4.1 Entorno seguro. 4.2 Confianza.

Tabla 2: valores culturales y comportamientos con los que se manifiestan.

DIMENSIÓN ESTRUCTURAL

PRINCIPIO	COMPORTAMIENTO
5. Estructura desjerarquizada	5.1 Autoorganización. 5.2 Horizontalidad.
6. Propósito común	6.1 Propósito conocido y compartido. 6.2 Decisiones guiadas por valores.

Tabla 3: principios estructurales y comportamientos con los que se manifiestan.

1.- Asertividad

En una organización ágil, las relaciones se conducen con asertividad. Las personas expresan de forma natural sus opiniones respetando a las de los demás.

La presencia de la asertividad en la organización se puede evaluar a través de los comportamientos habituales en la organización:

- **Coraje**
Las personas participan activamente y con confianza en los retos y situaciones diarias, manifestando objetiva y abiertamente sus criterios y opiniones.
- **Respeto**
Las personas manifiestan sus criterios y opiniones sin agresividad, considerando y respetando la dignidad de los demás.

2.- Valoración del talento

En la medida en la que el valor del producto es consecuencia del conocimiento tácito de las personas, más que de los procesos y la tecnología de la empresa, las organizaciones ágiles no gestionan personas, sino talento.

La evaluación de este valor organizativo se realiza analizando la mayor o menor presencia de tres comportamientos organizacionales:

■ **Incorporación y desarrollo del talento**

En qué medida la captación de talento es el principal objetivo de las actividades de incorporación de personas a la organización y de su desarrollo.

■ **Retención del talento**

En qué medida el ambiente laboral de la organización es profesionalmente enriquecedor y socialmente sano (“no tóxico”) con políticas orientadas a la retención del talento tales como:

- Motivación intrínseca.
- Empoderamiento.
- Conocimiento y participación en la visión de los proyectos.
- Ritmo de trabajo sostenible.

■ **Desarrollo profesional**

En qué medida es un objetivo estratégico de la organización, potenciar y mejorar el desarrollo profesional de las personas.

3.- Claridad

En las organizaciones ágiles, la disponibilidad y el intercambio de información entre las personas y los equipos se realiza honestamente y sin reservas, siendo los siguientes los comportamientos que reflejan la mayor o menor implantación de este valor.

■ **Claridad**

La información y comunicación es transparente entre las personas y equipos de la empresa.

■ **Honestidad**

Las personas se sienten libres de comunicarse de forma sincera.

4.- Confianza

Las organizaciones ágiles permiten un diálogo abierto que propicia la implicación de las personas, que sienten que se encuentran en un entorno seguro en el que pueden exponer sus preocupaciones o sugerencias y proponer soluciones.

Estos son los comportamientos que reflejan el mayor o menor enraizamiento de este valor.

- **Entorno seguro**

Las personas saben que pueden expresarse con libertad y asertividad, mostrándose tal como son, sin sentirse cuestionadas por ello.

- **Confianza**

Las personas saben que la organización confía en ellas y viceversa.

5.- Estructura desjerarquizada

Las organizaciones ágiles desarrollan estructuras organizativas descentralizadas, empoderando a las personas, que trabajan de forma autoorganizada.

Los comportamientos que reflejan la mayor o menor implantación de este principio son:

- **Autoorganización**

“Los trabajadores del conocimiento tienen que autogestionarse. Deben tener autonomía.”(Druker 1999)

La autonomía de un equipo puede producirse en distinto grado dependiendo del ámbito de responsabilidades que gestionan sus miembros; desde el mínimo, en el que cada persona sólo asume la responsabilidad de las tareas que tiene encomendadas, hasta el máximo en el que todas las funciones de desarrollo, gestión de tareas, diseño del marco de trabajo y participación en la estrategia global son asumidas por los miembros del equipo.

Se pueden establecer 4 niveles de autogestión para los equipos, desde el mínimo, propio de los equipos dirigidos y que no se pueden considerar autogestionados, hasta el máximo, que podemos denominar equipos autogobernados:

1.- Equipos dirigidos.

En ellos los miembros del equipo sólo tienen autoridad en la ejecución de las tareas. Son los gestores quienes administran el proyecto, monitorizan el avance, diseñan el marco de trabajo y participan en las decisiones estratégicas para la organización.

2.- Equipos autogestionados.

Los miembros del equipo tienen autoridad para ejecutar las tareas y también para gestionarlas en el marco del proyecto. Es el caso de los equipos que emplean marcos técnicos de *scrum* o *kanban* para gestionar el desarrollo.

3.- Equipos autodiseñados.

Los miembros del equipo, además de ser responsables de la ejecución y gestión de las tareas en el ámbito del proyecto, también diseñan el modelo organizativo interno del equipo y el marco de trabajo que emplean.

4.- Equipos autogobernados.

Además de asumir las responsabilidades de los equipos autodiseñados, en los equipos autogobernados los miembros tienen también capacidad de decisión en ciertas áreas de la estrategia global de la organización.

Al analizar la autoorganización se debe considerar que el nivel 2 (equipos autogestionados) puede resultar suficiente para facilitar la agilidad en la faceta operativa, pero para facilitarla también en la organizativa se requiere un nivel 3 para organizaciones con un patrón cultural verde y 4 en organizaciones con un patrón cultural *teal*.

PARADIGMA CULTURAL	DEL LOGRO Naranja	PLURALISTA Verde	EVOLUTIVO Teal/ Esmeralda
AUTONOMÍA DE LOS EQUIPOS	Dirigidos	Autogestionados	Autodiseñados

Ilustración 58: paradigmas culturales y autonomía de los equipos.

■ Horizontalidad.

Las estructuras jerárquicas desarrollan modelos de dirección y control para gestionar trabajadores que realizan su cometido de forma dirigida.

La jerarquía diseña en cada nivel las instrucciones y el control del nivel inferior.

“Sólo a través de la imposición de métodos estándar, de la imposición de las mejores implementaciones y condiciones para el trabajo es como se puede garantizar la ejecución más rápida. Y la obligación de hacer cumplir los estándares y forzar la cooperación necesaria recae únicamente en la gerencia.” (Taylor 1911)

El control jerárquico funciona en entornos estables y prototipos culturales naranjas, pero no favorece la creatividad y no permite a las organizaciones grandes la adaptación continua y rápida que exigen los entornos complejos.

Por otra parte cuando son las personas y no los procesos los responsables del resultado, necesitan margen de autonomía y autoorganización, lo que implica estructuras organizativas planas.

“No tiene sentido contratar a personas inteligentes y luego decirles qué hacer; contratamos gente inteligente para que nos digan qué hacer”. (Steve Jobs).

Los procesos de evaluación y mejora de la agilidad organizativa deben considerar la presencia de niveles jerárquicos reguladores de la información, comunicación, gestión y control del trabajo de niveles inferiores.

6.- Propósito común

Los comportamientos que reflejan la mayor o menor implantación de este principio son:

- **Propósito conocido y compartido.**
Las organizaciones ágiles tienen un propósito definido que conocen todos sus miembros y lo comparten como el objetivo último de su trabajo, más allá de los posibles objetivos de su equipo grupo o departamento.
- **Decisiones guiadas por valores.**
Las acciones que se llevan a cabo para alcanzar el propósito son conformes a los valores de la organización.

Soporte

La mejora de la agilidad requiere el soporte de la organización en 5 puntos.

Las deficiencias en estos puntos son la principal causa, si no la única, por la que fracasan los procesos de agilización o cambio cultural de las empresas.

- **Implicación directiva.**
Grado en el que los directivos de la empresa conocen los principios e implicaciones de la gestión ágil y promueven, con la implicación precisa, el modelo de agilidad que se desea para la organización.
- **Compatibilidad cultural.**
Compatibilidad entre el patrón cultural de la propiedad de la empresa y el objetivo de agilidad planificado en las diferentes dimensiones de la organización (operativa, cultural y estructural)
- **Medios.**
Recursos proporcionados por la organización para el desarrollo y mejora de la agilidad.
- **Formación.**
Formación facilitada por la organización para el funcionamiento de la agilidad en las dimensiones, ámbitos y grado que se han planteado como objetivos.
- **Acompañamiento (*coaching*).**
Acompañamiento proporcionado a los profesionales de la organización para abordar con éxito la gestión del cambio necesaria.

Criterios para evaluar la agilidad operativa, organizativa y riesgos para la mejora

Evaluación y mejora

Evaluar el estado de los diferentes principios y valores ágiles permite elaborar una “pila de acciones” priorizada. En la evaluación se ha de tener en cuenta el impacto previsto en la mejora general de la organización y su dificultad o nivel de riesgo de cada acción.

Con este fin Scrum Level propone unos criterios simples, que se desarrollarán a continuación, para cuantificar el grado de implantación de los principios, valores y apoyos en la empresa. Éstos permiten asignar un valor numérico a los diferentes principios y valores ágiles así como a las dificultades y riesgos previsibles.

Pero no se engañe: La razón de cuantificar el nivel de los principios y valores ágiles de la empresa no es medir sobre ningún estándar de agilidad. La finalidad es establecer una estrategia y mecánica de análisis que permitan:

- Recoger información de la organización.
- Estructurar y enfocar el análisis de mejora.
- Determinar y priorizar las actividades de mejora para cada ciclo.
- Disponer de series de datos relativos para determinar el avance.

No se haga trampas

Con Scrum Level se pueden identificar y cuantificar las fortalezas y debilidades de la gestión ágil en las empresas, pero si lo que desea no es hacer un análisis objetivo sino obtener unas altas y aparentes puntuaciones en “agilidad”, puede intentar amañar los datos. Según lo bien que lo haga y a quién se lo presente, colará. De hecho es una razón frecuente por la que se usan las certificaciones de todo tipo. Interesa más la fotografía, lo superficial, que la radiografía.

El modelo de evaluación de Scrum Level permite realizar evaluaciones denominadas “expertas”. En ellas el responsable determina los criterios con los que muestrea y pondera los datos en base a su juicio, experiencia profesional y conocimiento de las características de la organización.

En esta tarea nada le impide actuar de manera tendenciosa y deshonesto para favorecer un resultado predeterminado.

Scrum Level también permite realizar evaluaciones “técnicas” basadas en un protocolo³ propio que determina un procedimiento para realizar el muestreo, formulación y ponderación de los resultados.

Por supuesto, nada le impide formular un protocolo que no se ajuste a los criterios de Scrum Level.

Scrum Manager desarrolla este modelo para facilitar un plan de acción estructurado y útil de análisis y mejora de la gestión ágil.

La finalidad de los modelos, estándares y certificaciones es servir de guía en los procesos de mejora y de herramienta para medir el avance. Usarlos con una praxis deshonesto, con la finalidad no de mejorar, sino sólo de aparentarlo, es contrario a los principios de Scrum Manager y a la finalidad de Scrum Level.

3 . V. Apéndice I

Criterios para evaluar la agilidad operativa de la empresa

La evaluación del grado de agilidad en la faceta operativa de la empresa se determina analizando las prácticas de trabajo para valorar la implantación de los principios ágiles. La siguiente tabla muestra los principios de una operativa ágil, junto con el tipo de prácticas que hacen posible su implementación.

PRINCIPIO	PRÁCTICAS
1 ENTREGA DE VALOR	1.1 Compartir y comprender la visión del cliente.
	1.2 Colaboración del cliente con el equipo.
	1.3 Variabilidad.
2 MEJORA CONTINUA	2.1 Mejora continua de las prácticas.
3 DESARROLLO ITERATIVO E INCREMENTAL	3.1 Arranque desde un mínimo viable.
	3.2 Puntos de integración frecuentes con análisis de feedback para resolver y despejar el avance.
	3.3 Basar los hitos en la evaluación objetiva de partes del sistema funcionando.
4. RITMO DE TRABAJO SOSTENIBLE	4.1 Mantenimiento de un flujo de producción continuo y óptimo.
	4.2 Enfoque.
5 ATENCIÓN CONTINUA A LA EXCELENCIA	5.1 Uso de técnicas para garantizar la calidad.
6 OPERATIVA VISIBLE	6.1 Identificación temprana de impedimentos.
	6.2 Compartir información relativa al desarrollo.
7 CADENCIA Y SINCRONIZACIÓN GLOBAL	7.1 Predictibilidad de las reuniones basada en la cadencia del desarrollo.
	7.2 Predictibilidad de momentos de entregas de incrementos e integraciones.
8. PERSONAS SOBRE PROCESOS	8.1 Las personas aplican un conocimiento técnico alto en su desempeño.
	8.2 Las personas aplican las habilidades sociales necesarias para su desempeño.

Tabla 4: principios de la agilidad operativa y tipos de prácticas que los desarrollan.

Criterio para evaluar las prácticas

Los parámetros para cuantificar la implantación de las prácticas de los principios 1 a 7 son:

Valor	Criterio
0	No implementado. Poca o ninguna evidencia de prácticas para la finalidad analizada.
1	Parcialmente implementado. En ocasiones se emplean prácticas adecuadas, pero no es frecuente en los proyectos del <i>ámbito de la evaluación</i> , ni de forma homogénea.
2	Ampliamente implementado. Frecuentemente se emplean prácticas adecuadas, aunque no de una forma institucionalizada (sin garantía de homogeneidad y mejora continua).
3	Completamente implementado. Las prácticas se emplean siempre y su aplicación está institucionalizada (con formación y en mejora continua).

Tabla 5: criterio de evaluación de los principios de la agilidad operativa.

Criterio para evaluar el valor aportado por las personas sobre los procesos

El conocimiento técnico de las personas, sus habilidades de comunicación y relación son un factor importante en entornos ágiles, por lo que se deben considerar en los análisis de mejora. Pero evaluar individualmente el nivel profesional de las personas queda fuera del ámbito de Scrum Level, y por la complejidad y sensibilidad que encierra se recomienda no incluirla.

La evaluación en este punto puede incluir exclusivamente el grado de conocimiento de las técnicas ágiles con los criterios que se desarrollan en el siguiente apartado.

Criterio para valorar el conocimiento de prácticas ágiles del grupo evaluado

Valor	Criterio
0	<p>Insuficiente: La persona no tiene un conocimiento suficiente para desempeñar las tareas propias de su trabajo empleando prácticas ágiles.</p> <p>Referencia para la evaluación:</p> <ul style="list-style-type: none"> El nivel de conocimiento es insuficiente para superar un examen de acreditación Scrum Manager de nivel técnico, o un examen estándar de <i>Scrum Master</i> (scrumstudy.org, scrum.org, Scrum Alliance).
1	<p>Nivel técnico: La persona tiene un conocimiento suficiente para desempeñar las tareas propias de su trabajo empleando prácticas ágiles.</p> <p>Referencia para la evaluación:</p> <ul style="list-style-type: none"> El nivel de conocimiento es suficiente para superar un examen de acreditación Scrum Manager de nivel técnico, o un examen estándar de <i>Scrum Master</i> (scrumstudy.org, scrum.org, Scrum Alliance).
2	<p>Nivel experto: La persona tiene conocimiento holgado para desempeñar las tareas propias de su trabajo empleando prácticas ágiles.</p> <p>Referencias para la evaluación:</p> <ul style="list-style-type: none"> El nivel de conocimiento es suficiente para superar un examen de acreditación Scrum Manager de nivel experto. El nivel de conocimiento es suficiente para superar un examen de acreditación Scrum Manager de nivel técnico, o un examen estándar de <i>Scrum Master</i> (scrumstudy.org, scrum.org, Scrum Alliance) y tiene al menos 12 meses de experiencia de trabajo en equipos empleando prácticas ágiles.
3	<p>Nivel autoridad: La persona es conocedora experta de la gestión ágil de proyectos y organizaciones, así como de scrum.</p> <p>Referencias de contrastación:</p> <ul style="list-style-type: none"> La persona tiene certificado un nivel de autoridad en Scrum Manager de más de 200 PDAs. El nivel de conocimiento es suficiente para superar un examen de acreditación Scrum Manager de nivel técnico, o un examen estándar de <i>Scrum Master</i> (scrumstudy.org, scrum.org, Scrum Alliance) y tiene al menos 24 meses de experiencia trabajando en equipos ágiles, empleando prácticas ágiles.

Tabla 6: criterio para evaluar el conocimiento de prácticas ágiles.

Cálculo de la agilidad operativa de la empresa

Con una valoración de las prácticas de 0 a 3 se determina el grado de agilidad de la organización en su faceta operativa, teniendo en cuenta las siguientes directrices.

Directrices

1.- Valores medios

El valor de agilidad operativa se obtiene promediando los valores de los componentes que lo forman: prácticas ágiles y conocimiento de las personas.

Se debe considerar la media aritmética y no la moda o la mediana.

2.- Homogeneidad

La falta de homogeneidad entre los valores parciales es síntoma de tensiones o fricciones en la organización y disminuye o incluso neutraliza las aportaciones de agilidad que serían esperables considerando la media aritmética.

Cuando esto se produce, se deben analizar las causas y ponderar si afecta y debe minorarse la media aritmética.

3.- Ponderación

El análisis puede ponderar los valores medios obtenidos considerando:

En las evaluaciones técnicas (v. Glosario) se recomienda incluir criterios de formulación que consideren el impacto atenuador cuando se produce falta de homogeneidad.

Las evaluaciones expertas, a diferencia de las técnicas, son actuaciones ágiles y en ellas prevalece el criterio experto de la persona que realiza la evaluación, por encima del criterio técnico de la formulación.

La ponderación de las evaluaciones expertas debe incluir el razonamiento y argumentación de los ajustes realizados a los valores medios de la formulación.

Formulación

El valor de la agilidad operativa se calcula promediando los valores de los principios, a la vez que el valor de cada principio es la media aritmética de los valores obtenidos en las prácticas.

Ejemplo de una posible valoración de las prácticas del principio “Entrega de valor”:

PRÁCTICAS ORIENTADAS A	0	1	2	3
1.1 Compartir y comprender la visión del cliente	■	■		
1.2 Colaboración del cliente con el equipo	■	■	■	
1.3 Variabilidad.	■	■		

Entrega de valor: 1.3.

Observaciones para la valoración del principio “Personas sobre procesos”

Los protocolos o evaluaciones expertas que estimen oportuno ponderar el resultado con información de las habilidades técnicas y sociales de las personas, deben evaluarlo a nivel de grupo, no individual.

Por ejemplo, si el protocolo empleado usa cuestionarios de evaluación, puede emplear preguntas con carácter global del tipo:

- En general, el nivel profesional técnico de las personas es: 0.- Deficiente / 1.- Adecuado / 2.- Bueno / 3.- Excelente.

El conocimiento específico sobre agilidad se valora aparte de estas habilidades.

Conocimiento de las prácticas ágiles

Con los valores verificados se obtiene su media aritmética.

El análisis debe considerar aplicar criterios de ponderación en la media si más de un 20% de las personas del ámbito de la evaluación tienen valoración 0.

Ejemplo de un posible resultado de la verificación del conocimiento técnico:

CONOCIMIENTO TÉCNICO	0	1	2	3
Nombre de la persona	■	■	■	
Nombre de la persona	■	■		
Nombre de la persona	■	■	■	
Nombre de la persona	■	■		
Nombre de la persona	■	■	■	■
Nombre de la persona	■	■	■	
Nombre de la persona	■			
Nombre de la persona	■	■		

Resultado

Con la media de los resultados de los principios se obtiene el grado de agilidad en esta dimensión.

Ejemplo de un posible resultado:

Principio	Resultado
Entrega de valor	1,3
Mejora continua	2
Desarrollo iterativo e incremental	1,5
Ritmo de trabajo sostenible	2
Atención continua a la excelencia	1,1
Operativa visible	1,3
Cadencia y sincronización global	1
Personas sobre procesos	2

Grado de agilidad de la organización en su dimensión operativa: 1,52

Criterios para evaluar la agilidad organizativa de la empresa

La agilidad en la faceta organizativa de la empresa es el resultado de la agilidad en las dos dimensiones que la componen:

1. Dimensión cultural.
2. Dimensión estructural.

Dimensión cultural

Valores culturales ágiles y comportamientos que éstos producen.

DIMENSIÓN CULTURAL

VALOR	COMPORTAMIENTO
1. Asertividad	1.1 Coraje. 1.2 Respeto.
2. Valoración del talento	2.1 Incorporación y desarrollo. 2.2 Retención. 2.3 Desarrollo profesional.
3. Claridad	3.1 Transparencia. 3.2 Honestidad.
4. Confianza	4.1 Entorno seguro. 4.2 Confianza.

Tabla 7: valores culturales y comportamientos para evaluarlos.

Criterio para evaluar los comportamientos

El criterio para evaluar la mayor o menor presencia de un comportamiento es:

Valor	Criterio
0	Comportamiento insólito Es extraño que se produzca.
1	Comportamiento posible Comportamiento poco habitual .
2	Comportamiento habitual Se presenta con frecuencia.
3	Comportamiento institucionalizado Se produce siempre y está formalmente institucionalizado.

Tabla 8: criterio para evaluar los comportamientos de los valores culturales.

Dimensión estructural

Principios que generan estructuras organizativas ágiles y comportamientos organizativos ligados a ellos.

DIMENSIÓN ESTRUCTURAL

PRINCIPIO	COMPORTAMIENTO
5. Estructura desjerarquizada	5.1 Autoorganización. 5.2 Horizontalidad.
6. Propósito común	6.1 Propósito conocido y compartido. 6.2 Decisiones guiadas por valores.

Tabla 9: principios y comportamientos estructurales ágiles.

Criterio para evaluar los comportamientos

Estructura desjerarquizada

Autoorganización.

El grado de agilidad de este comportamiento se determina en función del modelo de equipo⁴ empleado en el ámbito de la evaluación⁵ con el siguiente criterio.

Valor	Criterio
0	Equipos dirigidos.
1	Equipos autogestionados.
2	Equipos autodiseñados.
3	Equipos autogobernados.

Tabla 10: criterio para evaluar el grado de autoorganización.

Horizontalidad

El grado de agilidad de este comportamiento se determina en función de los niveles jerárquicos en los que se estructura la organización de la empresa:

Valor	Criterio
0	Jerarquía de más de tres niveles (ej: consejeros, directores, gestores, técnicos).
1	Jerarquía de tres niveles (ej: estratégico, táctico y operativo).
2	Jerarquía de dos niveles (ej: estratégico y operativo).
3	Organización sin jerarquía.

Tabla 11: criterio para evaluar la horizontalidad de la organización.

4 . V. Principios de la dimensión estructural: Estructura desjerarquizada: tipos de equipos.

5 . V. Glosario.

Propósito común

El criterio para establecer una gradación de arraigo de los valores ágiles en la cultura de la empresa es el siguiente:

Valor	Criterio
0	Comportamiento insólito . Es extraño que se produzca.
1	Comportamiento posible . Comportamiento poco habitual.
2	Comportamiento habitual . Se presenta con frecuencia.
3	Comportamiento institucionalizado . Se produce siempre y se encuentra institucionalizado.

Tabla 12: criterio para evaluar la institucionalización de un propósito común.

Cálculo de la agilidad organizativa de la empresa

Los valores de los comportamientos en las dimensiones cultural y estructural determinan el grado de agilidad organizativa de la empresa, con las siguientes directrices y formulación:

Directrices

1.- Valores medios

El valor final de cada dimensión (cultural y estructural) se obtiene promediando los valores de los comportamientos. Se debe considerar la media aritmética y no la moda o la mediana.

El valor final de la faceta organizativa se obtiene promediando los valores de las dimensiones cultural y estructural.

2.- Homogeneidad

La falta de homogeneidad entre los valores parciales es síntoma de tensiones o fricciones en la organización y disminuye, o incluso neutraliza las aportaciones de agilidad que serían esperables considerando la media aritmética.

Se deben analizar las causas de estas tensiones y ponderar si debe minorarse la media aritmética.

3.- Ponderación

El análisis puede ponderar los valores medios obtenidos considerando:

Las evaluaciones técnicas⁶, cuando se produce falta de homogeneidad, deben incluir criterios de formulación que consideren el impacto atenuador.

Las evaluaciones expertas*, a diferencia de las técnicas, son actuaciones ágiles. En ellas prevalece el criterio experto de la persona que realiza la evaluación, por encima del criterio técnico de la formulación.

La ponderación de las evaluaciones expertas debe incluir el razonamiento y argumentación de los ajustes realizados a los valores medios.

Formulación

Cultura ágil

El grado de agilidad de cada valor cultural es la media de los resultados obtenidos en sus comportamientos.

Ejemplo de un resultado para el valor “Asertividad”:

COMPORTAMIENTOS	0	1	2	3
Coraje	■	■	■	
Respeto	■	■		

Asertividad: 1,5

El grado de agilidad en la dimensión cultural es la media de los resultados obtenidos en los valores ágiles culturales.

Ejemplo:

Valor	Resultado
Asertividad	1,5
Valoración del talento	2
Claridad	1,5
Confianza	2

Grado de agilidad en la cultura de la organización: 1,75

6 . V. Glosario.

Estructura ágil.

Con la media de los resultados obtenidos en los diferentes comportamientos se obtiene el grado de agilidad estructural.

Ejemplo de un resultado del principio “Estructura desjerarquizada”.

COMPORTAMIENTOS	0	1	2	3
Estructura desjerarquizada	■	■		
Propósito común	■	■		

Estructura desjerarquizada: 1

Con la media de los resultados de los valores estructurales se obtiene el grado de agilidad en esta dimensión.

Ejemplo de un posible resultado.

Valor	Resultado
Estructura desjerarquizada	1
Propósito común	1

Grado de agilidad en la estructura de la organización: 1

Resultado

El valor de agilidad organizativa es la media de las dos dimensiones que la componen:

Ejemplo:

Valor	Resultado
Dimensión cultural	1,2
Dimensión estructural	1

Grado de agilidad organizativa: 1,1

Criterios para evaluar el soporte que proporciona la organización

El éxito o fracaso de un plan de mejora depende en gran medida del soporte que le brinda la organización. En la implicación de sus directivos con la agilidad, la compatibilidad con la cultura de la empresa y los medios consignados para acometer las mejoras. Entre estos medios se incluiría la formación necesaria en técnicas ágiles y entrenamiento profesional (*coaching*).

En resumen, depende de estos 5 factores:

- Implicación directiva.
- Compatibilidad de la dimensión y grado de agilidad deseado con la cultura de la empresa.
- Medios disponibles.
- Formación.
- Entrenamiento profesional.

Los dos primeros son decisivos. Si la dirección de la empresa no cree en la agilidad, o si la cultura de la organización no es compatible con los resultados que se buscan y no parece probable que vaya a cambiar, es mejor no comenzar y reconsiderar la situación.

Si la dirección está convencida e implicada y el cambio cultural previsto es asumible, el proceso de mejora funcionará si se dispone de los medios necesarios y se proporciona ayuda a las personas para llevar a cabo el cambio, a través de formación y entrenamiento o *coaching* profesional.

Las deficiencias en estos soportes pueden entorpecer o incluso desbaratar la mejora, por lo que conviene estar al tanto desde el principio. Hay que analizar la situación en la que se encuentra la organización, la voluntad de cambio y, en consecuencia, el nivel de riesgo que puede aportar cada factor.

Con un rango de cuatro valores para determinar la situación actual y la predisposición al cambio de cada soporte, se puede componer un mapa de los niveles de riesgo (alto, medio o bajo).

Los criterios para determinar los niveles de soporte y la predisposición al cambio son:

Situación actual

Implicación directiva

Valor	Criterio
0	Los directivos de la empresa no conocen o no promueven los principios y valores ágiles necesarios para el cambio.
1	Los directivos de la empresa tienen un conocimiento muy somero de los principios y valores ágiles y una actitud de apoyo cuestionable.
2	El conocimiento y apoyo de los directivos de la empresa a los principios y valores ágiles necesarios para el cambio puede no ser suficiente.
3	Los directivos de la empresa conocen y apoyan comprometidamente los principios y valores necesarios.

Tabla 13: criterio para evaluar la implicación directiva.

Compatibilidad cultural

Valor	Criterio
0	El modelo cultural de la empresa es manifiestamente incompatible con las acciones de cambio necesarias.
1	El modelo cultural de la empresa necesita una evolución importante para resultar compatible con las acciones de cambio necesarias.
2	El modelo cultural de la empresa es relativamente compatible con las acciones de cambio necesarias y necesita algunos cambios.
3	El modelo cultural de la empresa es completamente compatible y está alineado con las acciones de cambio necesarias.

Tabla 14: criterio para evaluar la compatibilidad cultural los principios y valores ágiles.

Medios, formación y coaching

¿La empresa facilita los medios, formación o *coaching* a las personas y en la medida necesaria.?

Valor	Criterio
0	No
1	Sí, pero escasamente.
2	Sí, pero no siempre o insuficientemente.
3	Sí y de forma suficiente.

Tabla 15: criterio para evaluar la provisión de medios, formación y coaching.

Predisposición al cambio

Valor	Criterio
0	La situación actual está asentada y no desea o no encuentra razón para cambiar.
1	La situación actual se puede cuestionar para aplicar los cambios que se consideren procedentes.
2	La situación actual está abierta y es propensa a los cambios para facilitar la mejora.
1	Se desconoce cuál es la voluntad de cambio.

Tabla 16: criterio para evaluar el grado de predisposición al cambio.

Formulación del riesgo

RIESGOS PARA LA AGILIDAD ORGANIZACIONAL							
	GRADO DE SOPORTE						
	0	1	2	3	4	5	6
Implicación directiva							
Compatibilidad cultural							
Medios							
Formación							
Acompañamiento, <i>coaching</i>							

cc by ScrumLevel®

Ilustración 59: mapa de riesgos según el grado de soporte para la mejora.

Considerando “grado de soporte” a la suma del valor obtenido como **situación actual** más el de **predisposición al cambio**, la ilustración 18 muestra los niveles de riesgo probables.

Ejemplo:

Valores determinados para cada línea de soporte.

Factores	Situación	Predisposición	Grado de soporte
Implicación directiva	1	2	3
Compatibilidad cultural	1	2	3
Medios	2	2	4
Formación	2	2	4
Acompañamiento, <i>coaching</i>	2	2	4

Mapa de riesgos que resulta del ejemplo:

RIESGOS PARA LA AGILIDAD ORGANIZACIONAL

		GRADO DE SOPORTE						
		0	1	2	3	4	5	6
Implicación directiva					☹️			
Compatibilidad cultural					☹️			
Medios						☹️		
Formación						☹️		
Acompañamiento, <i>coaching</i>						☹️		

cc by ScrumLevel®

Nota: El valor máximo de predisposición al cambio es 2 en evaluaciones técnicas. En evaluaciones expertas puede llegar a 3, si el evaluador responsable considera justificado ponderar positivamente un factor valorado a 2, para reflejar ausencia de riesgo.

Apéndice

Apéndice 1: protocolos

Scrum Level define **qué** factores se deben considerar en los procesos de agilidad y cambio cultural de las empresas: qué **valores, principios y soportes** se deben analizar, junto con los **criterios** para determinar el nivel de agilidad de cada uno.

Establece por tanto directrices, sin marcar un guion o dinámica concreta. Eso es en definitiva un protocolo: el **cómo**, el guion, las dinámicas y artefactos para analizar o evaluar la agilidad de una organización, con una implementación adecuada a sus circunstancias.

Los derechos de *copyright* para el uso de Scrum Level como marco de referencia se encuentran liberados. Por tanto otras personas pueden desarrollar protocolos propios, reconociendo basarse en Scrum Level.

El protocolo OKs, desarrollado por Scrum Manager para se encuentra disponible en la página de descargas de scrumlevel.com.

Homologación de protocolos

Estos protocolos independientes pueden homologarse.

La homologación garantiza que el protocolo ha sido revisado por Scrum Level y se considera conforme a las pautas de evaluación y mejora definidas en esta guía.

Componentes de un protocolo

Los componentes de un protocolo son:

Componente	Obligatorio
Guía del protocolo	Siempre
Cuestionarios o listas de verificación	Opcional
Herramientas de formulación	Opcional
Formatos de documentación	Opcional

Guía del protocolo

Incluye:

- Ficha técnica con:
 - Nombre del protocolo.
 - Versión de Scrum Level para la que se ha desarrollado.
 - Versión del protocolo.
 - *Validez*⁷.
 - Derechos y / o licencia.
 - Categoría o categorías^(*) de evaluación para las que es apto.
 - Tipo o tipos de evaluación para las que es apto.
- Exposición del método con explicación suficiente de su operativa y formulación.

7 . V. Glosario.

Cuestionarios o listas de verificación.

Son componentes necesarios si el protocolo los emplea en evaluaciones técnicas.

Incluyen instrucciones de usuario para su ejecución, e instrucciones de administración y formulación, para el evaluador.

Herramientas de formulación

Los protocolos que incluyen verificaciones técnicas, pueden facilitar herramientas para el cálculo de la formulación, tales como hojas de cálculo o aplicaciones de software.

Formatos de documentación

Formatos o modelos para la hoja o resumen diploma y para el informe de la evaluación.

Normas y restricciones para protocolos

El modelo Scrum Level es de uso libre como guía en procesos de asesoría y evaluación.

El desarrollo de protocolos específicos para evaluaciones Scrum Level está también liberado, si bien deben cumplir las siguientes normas y restricciones.

Validez

Debe incluirse la validez del protocolo:

- **Autocomprobada:** la conformidad formal y funcional con el modelo Scrum Level ha sido comprobada por la organización o autor que ha desarrollado el protocolo.
- **Homologada:** la conformidad formal y funcional ha sido homologada por Scrum Level.

La indicación de validez debe incluir la versión de Scrum Level para la que ha sido comprobada u homologada.

Para obtener el permiso para indicar “homologado por Scrum Level” es necesario solicitar la evaluación del protocolo a Scrum Level.

Número de proyectos

El procedimiento definido por el protocolo debe exigir que el número de proyectos incluido en las evaluaciones sea al menos el 33% de los que formen parte del ámbito⁸ de la misma.

Número de personas

El número de personas incluidas en el equipo de evaluación^(*) debe ser al menos el 20% de los implicados en el grupo evaluado^(*) e incluir todas las áreas de responsabilidad de la organización.

Selección de proyectos y personas

La selección de los proyectos que conforman el ámbito^(*) y de las personas que forman parte del equipo de evaluación^(*), debe realizarse de forma aleatoria, o determinada por el criterio de un evaluador (responsable o auxiliar) pero no son admisibles selecciones realizadas por la organización evaluada.

8 . V. Glosario.

Apéndice 2: documentar o certificar una evaluación

Documento de evaluación

El uso de Scrum Level como guía para evaluar la agilidad de una organización es libre. Si bien su uso para la elaboración de documentos de evaluación con la marca registrada Scrum Level® está sujeto a las siguientes condiciones.

La documentación de la evaluación debe incluir un resumen o diploma, así como un informe de evaluación, conteniendo la información que se detalla en este apéndice.

Información	En el resumen o diploma	En el informe
Rango de la evaluación	X	X
Tipo de evaluación	X	X
Organización evaluadora (si procede)	(X)	X*
Evaluador responsable (si procede)	X	X
Nivel de independencia (si procede)	X	X
Protocolo empleado		X
Organización evaluada	X	X*
Categoría de la evaluación	X	X
Ámbito (si procede)		X
Dimensión o dimensiones	X	X
Resultado de la evaluación	X	X*
Resultados parciales		X
Puntos de atención		X

Tabla 17: Elementos que deben incluirse en los informes de evaluación.

(X) Opcional | X* Información completa, además del nombre.

Rango de la evaluación

Con la indicación “Oficial”, “Independiente” o “Autoevaluación” según sea el caso.

Evaluador responsable	Debe indicar
No certificado por Scrum Manager	Independiente
Certificado por Scrum Manager	Oficial
Sin evaluador (Evaluación técnica)	Autoevaluación

Las evaluaciones oficiales son emitidas por Scrum Manager.

Tipo de evaluación

Con la indicación “Experta” o “Técnica” según las valoraciones estén o no ponderadas por el criterio experto de un evaluador responsable.

Organización evaluadora

No debe indicarse en las autoevaluaciones.

Las evaluaciones oficiales las debe realizar una organización evaluadora acreditada por Scrum Manager.

En el resumen o diploma:

- Nombre comercial de la organización evaluadora y razón social (en el caso de que sean diferentes).

En el informe de evaluación:

- Nombre comercial de la organización evaluadora y razón social (en el caso de que sean diferentes).
- Dirección completa: vía, ciudad, país y código postal.
- Teléfono y o dirección de correo electrónico para contacto relacionado con las evaluaciones Scrum Level.
- Persona de contacto para cuestiones relacionadas con la evaluación realizada.

No debe indicarse en las autoevaluaciones.

Nombre completo del evaluador en el caso de evaluaciones oficiales e independientes.

Nivel de independencia

Debe indicarse si la evaluación forma parte de un proceso de asesoría o mejora, o si se ha realizado en la organización evaluada un proceso de asesoría en los 6 meses previos a la evaluación.

Nivel 1.- Algún evaluador (responsable o auxiliar) ha participado también en el proceso de asesoría.

Nivel 2.- Ni el evaluador responsable⁹, ni los posibles evaluadores auxiliares^(*) han participado en el proceso previo de asesoría.

Nivel 3.- Evaluación con nivel de independencia 2, realizada por una organización distinta e independiente de la responsable de la asesoría previa.

Protocolo empleado

Si se ha empleado un protocolo de uso público: Nombre y versión del protocolo empleado para realizar la evaluación.

En caso contrario debe indicarse: “protocolo propio homologado” o “protocolo propio no homologado” según sea el caso.

Organización evaluada

En el resumen o diploma:

- Nombre comercial de la organización evaluada y razón social (en el caso de que sean diferentes)

9 . V. Glosario.

En el informe de evaluación:

- Nombre comercial de la organización evaluada y razón social (en el caso de que sean diferentes)
- Dirección completa: vía, ciudad, país y código postal.
- Teléfono y o dirección de correo electrónico para contacto relacionado con las evaluaciones Scrum Level.
- Persona de contacto para cuestiones relacionadas con la evaluación realizada.

Categoría de la evaluación

Indicación de la categoría de la evaluación: 1, 2 o 3.

Categoría 1: El sistema está compuesto por un único equipo de personas y puede comprender un único proyecto o varios.

Categoría 2: El sistema está compuesto por más de un equipo de personas y más de un proyecto. Todos los equipos y proyectos pertenecen a la misma unidad organizativa (área, departamento o sucursal).

Categoría 3: El sistema incluye a más de una unidad organizativa (área, departamento o sucursal) y cada una de ellas puede incluir un único o varios equipos y / o proyectos.

Ámbito de la evaluación

Opcional en evaluaciones de categoría 1.

En evaluaciones de categoría 2: relación de proyectos y equipos objeto de evaluación.

En evaluaciones de categoría 3: relación de secciones o departamentos de la organización objeto de evaluación, con indicación en cada uno de ellos de los proyectos y equipos incluidos.

Faceta

Indicación de una de las opciones de evaluación posibles:

- **Operativa:** La evaluación abarca la faceta operativa.
- **Operativa y organizativa:** La evaluación abarca las facetas operativa y organizativa.
- **Organizativa:** La evaluación abarca la faceta organizativa.

Resultado de la evaluación

En el resumen o diploma:

- Indicación del nivel alcanzado en la dimensión o dimensiones evaluadas: insuficiente, apto o alto según el criterio de valoración.

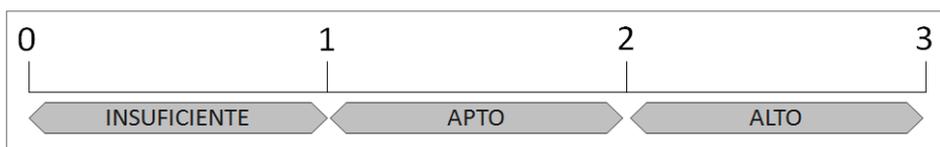


Ilustración 60: niveles de agilidad.

En el informe de evaluación:

- Indicación del nivel alcanzado en la faceta, o facetas evaluadas: Bajo o insuficiente, medio o alto, según el criterio de valoración.
- Valor numérico del nivel alcanzado en la faceta, o facetas evaluadas, con precisión de 2 decimales.

Si se evalúan las dos facetas (operativa y organizativa):

- Posición del valor en ambas facetas, representado sobre un gráfico de coordenadas.

Resultados parciales

El informe de evaluación incluye los resultados parciales de las prácticas y en su caso valores que se han analizado.

Ejemplo:

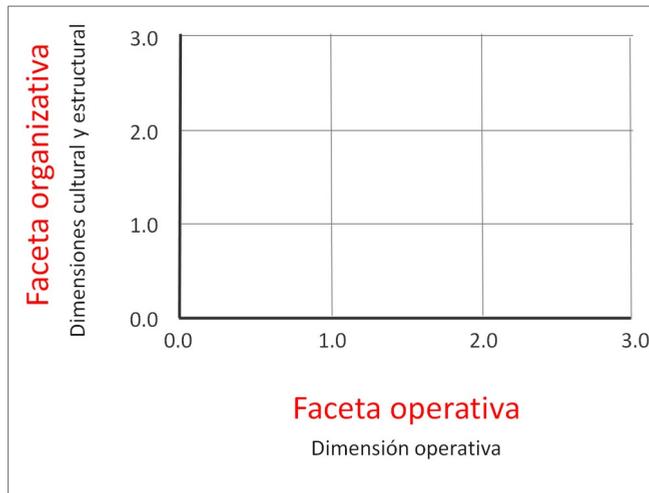


Ilustración 61: diagrama de evaluación.

Puntos de atención

El informe de evaluación incluirá un apartado denominado “Puntos de atención” si se detecta al menos una de las circunstancias siguientes, exponiendo éstas y las medidas de ponderación que hayan requerido:

- Falta de homogeneidad (desviación estándar poblacional > 1) entre los valores de flexibilidad de las áreas de prácticas.
- Falta de homogeneidad (desviación estándar poblacional > 1) entre los valores de fluidez de los comportamientos organizacionales.

Glosario

Ágil

Dicho de una técnica, procedimiento o método de trabajo o de gestión, indica que es compatible y desarrolla principios o valores ágiles en la operativa, estructura o cultura de la organización.

Agilidad

Característica que facilita en los modos de gestión de proyectos, personas u organizaciones la adaptación a los cambios y la entrega de resultados continua.

Agilidad fundamental

Agilidad en la operativa, estructura y cultura de la organización.

Agilidad técnica

Agilidad en la operativa de la organización.

Agilidad operativa

V. agilidad técnica.

Agilidad organizativa

V. agilidad en la estructura y cultura de la organización.

Ámbito de la evaluación

En evaluaciones de categoría 2: relación de proyectos y equipos objeto de evaluación.

En evaluaciones de categoría 3: relación de secciones o departamentos de la organización objeto de evaluación, indicando en cada uno los proyectos y equipos incluidos.

Categoría (de evaluación)

Término que se aplica a un análisis o evaluación Scrum Level de una organización para determinar su amplitud, según el tamaño del sistema de personas y proyectos que comprende.

Las categorías son:

Categoría 1 o “evaluación de equipo”: El sistema está compuesto por un único equipo de personas y puede comprender un único proyecto o varios.

Categoría 2 o “evaluación de departamento”: El sistema está compuesto por más de un equipo de personas y más de un proyecto. Todos los equipos y proyectos pertenecen a la misma unidad organizativa (área, departamento o sucursal).

Categoría 3 o “evaluación de empresa o corporación”: El sistema incluye a más de una unidad organizativa (área, departamento o sucursal) y cada una de ellas puede incluir un único o varios equipos y / o proyectos.

Equipo de evaluación

Grupo de personas pertenecientes a la organización evaluada e implicadas en los proyectos que forman parte del ámbito de una evaluación. Se refiere sólo a los miembros del grupo evaluado que participan en la evaluación. V. Grupo evaluado para diferenciar.

Evaluación experta

Evaluación dirigida por una persona que desempeña el rol y las responsabilidades de evaluador.

Evaluación independiente

V. Rango.

Evaluación oficial

V. Rango.

Evaluación técnica

Evaluación sin rol de evaluador, procedimentada y verificada a través de formularios y / o aplicativos de evaluación. V. Tipo.

Evaluador auxiliar

Persona que auxilia al evaluador responsable.

Evaluador responsable

Persona que dirige una evaluación. En función del ámbito de la evaluación puede requerir el apoyo de evaluadores auxiliares.

Flexibilidad

Índice del modelo Scrum Level para expresar el grado de agilidad técnica, resultado de una evaluación.

Fluidez

Índice del modelo Scrum Level para expresar el grado de agilidad organizativa, resultado de una evaluación.

Grupo evaluado

Todas las personas pertenecientes a la organización evaluada e implicadas en la gestión o desarrollo de los proyectos que forman parte del ámbito de una evaluación. A diferencia del equipo de evaluación, abarca a todos, participen en la evaluación o no.

Nivel de independencia

Parámetro aplicable a las evaluaciones en organizaciones que con carácter previo han realizado actividades de asesoría o formación relacionadas con la evaluación.

Nivel 1.- Algún evaluador (responsable o auxiliar) ha participado en las actividades previas de asesoría o formación.

Nivel 2.- Ni el evaluador responsable, ni los posibles evaluadores auxiliares han participado en las actividades previas de asesoría o formación.

Nivel 3.- Además de cumplirse el nivel de independencia 2, no se han realizado actividades previas de asesoría o formación para mejora de la agilidad, o las personas que han participado en las actividades previas y los evaluadores pertenecen a organizaciones distintas, sin relaciones societarias o de colaboración.

Producto mínimo viable

Del inglés “minimum viable product”(MVP). Término frecuentemente empleado en el ámbito del desarrollo ágil de productos para definir al producto desarrollado con las características suficientes para satisfacer a los clientes iniciales proporcionando retroalimentación para la evolución de su desarrollo.

Protocolo

Conjunto de instrucciones específicas de procedimiento y recursos diseñados para guiar evaluaciones Scrum Level en un ámbito o ámbitos concretos, determinados por el tamaño de la organización que se va a evaluar y el tipo de evaluación empleado.

Rango

Caracteriza a una evaluación y puede ser:

Oficial: Si ha sido emitida por Scrum Level y en ella ha intervenido como evaluador responsable un evaluador certificado.

Independiente: En caso contrario.

Retrospectiva

Nombre de la reunión en la que un equipo analiza su forma de trabajo para mejorarla de forma continua.

Scrum

Scrum en su acepción más amplia abarca el conjunto de principios y valores de autoorganización y desarrollo ágil.

Subjetividad en la evaluación

Indicador que detecta si la información facilitada por una persona a través de un formulario o entrevista resulta poco fiable o parece distorsionada. La distorsión se puede producir de forma intencionada por el entrevistado para aparentar o simular una situación que no es real, o no intencionada por no comprender correctamente o tergiversar las preguntas.

Tipo (de evaluación)

Término que se aplica a un análisis o evaluación Scrum Level de una organización para determinar si está dirigida por un evaluador o no. Los tipos posibles son: experta (v. Evaluación experta) y técnica (v. Evaluación técnica).

Validez (de un protocolo)

Característica de un protocolo, que puede ser:

- 1.- **Autocomprobada:** La validez formal y compatibilidad funcional con el modelo no han sido comprobadas por Scrum Level.
- 2.- **Homologada:** la validez formal y compatibilidad funcional del modelo han sido revisadas y aprobadas por Scrum Level.

Verificación

Comprobación del valor de un determinado parámetro.

Verificación mixta

Verificación que basa sus resultados en la ponderación que el asesor o evaluador realizar sobre los obtenidos con una verificación técnica previa.

Verificación pericial

Verificación realizada por el asesor o evaluador basada en su criterio experto.

Verificación técnica

Verificación basada en los resultados de un cuestionario o lista de comprobación.

Trabajos citados

Bibliografía

- Nato 1969: Bauer F.L., Bolliet L. & Helms H.J., Software Engineering, 1969
- Bau 1969: Bauer, F., Bolliet, L., & Helms, H., Software Engineering. Reporton a conference sponsored by the NATO, 1969
- Beck & Cowan 1996: Beck, D. & Cowan C., Spiral Dynamics: Mastering Values, Leadership and Change, 1996
- Beck & Grenning & otros 2001: Beck, K., Grenning, J., Martin, R.C., Beedle, M., Highsmith., Mellor, S., y, Manifiesto for Agile Software Development, 2001
- Druker 1999: Druker, P.F., Knowledge-Worker Productivity: The Biggest Challenge.,
- Ford & Gibbs 1996: Ford, Gary & Gibbs, Norman E., A Mature Profession of Software Engineering, 1996
- Graves 2004: Graves, C. W., Levels of Human Existence, 2004
- Isaacson 2011: Isaacson, W., Steve Jobs, 2011
- Laloux 2016: Laloux, F., Reinventar las organizaciones, 2016
- Nonaka 2004: Nonaka, I., & Takeuchi I., Hitotsubashi on Knowledge Management, 2004
- Nonaka 1986: Nonaka, I., & Takeuchi, H., The New New Product Development Game, 1986
- Nonaka 1995: Nonaka, I., & Takeuchi, I., The Knowledge-Creating Company, 1995
- Orr 2002: Orr, K., CMM versus Agile Development: Religious wars and software development, 2002
- Schwaber 1995: Schwaber, K., SCRUM Development Process - OOPSLA 95, 1995
- Scrum Manager 2017: Scrum Manager, Scrum Manager: identidad y aportaciones a la gestión ágil, 2017
- Takeuchi & Nonaka 1986: Takeuchi, H. & Nonaka, I., The New New Product Development Game, 1986

Agilidad: perspectiva y principios del desarrollo ágil

Taylor 1911: Taylor, F.W., The Principles of Scientific Management, 1911
versionone 2017: Versionone.com, The 11th annual State of Agile Report, 2017
Schwaber 2006: , Scrum Et Al, 2006
Beck 2000: , Extreme Programming Explained, 2000
Turner & Jain 2002: , Agile Meets CMMI: Culture Clash or Common Cause?, 2002
swebok 2004: , SWEBOK, 2004

Índice de vídeos

- Vídeo 1: Crisis del software: IBM 360 - <https://youtu.be/vOXsrIDYnyo>
Vídeo 2: Crisis del software: Mariner 1 - <https://youtu.be/rtSIKgxW-D4>
Vídeo 3: Objetivo de la gestión ágil vs. gestión predictiva - <https://youtu.be/BoBDosdCsps>
Vídeo 4: Scrum - <https://youtu.be/SIZfzLRGYBA>
Vídeo 5: Planificación y avance con scrum - <https://youtu.be/G0Ruagq7A7I>
Vídeo 6: Retrospectiva - <https://youtu.be/7wd61RrJAfo>
Vídeo 7: Gráfico Burn Down - <https://youtu.be/hveuhx7rZgw>
Vídeo 8: Estimación en la pared - <https://youtu.be/4O2BRg9kDU4>
Vídeo 9: Gráfico de producto - <https://youtu.be/jdyodO-uFBk>
Vídeo 10: Gestión predictiva y evolutiva - https://youtu.be/_-BtfKHASsY
Vídeo 11: La emergencia de un nuevo modelo organizacional: <https://youtu.be/NrSABjCIGVc>
Vídeo 12: Paradigmas culturales de las empresas https://youtu.be/_-BtfKHASsY

Índice de tablas

Tabla 1: principios de la agilidad operativa y tipos de prácticas que los desarrollan.	141
Tabla 2: valores culturales y comportamientos con los que se manifiestan.....	149
Tabla 3: principios estructurales y comportamientos con los que se manifiestan.	150
Tabla 4: principios de la agilidad operativa y tipos de prácticas que los desarrollan.	159
Tabla 5: criterio de evaluación de los principios de la agilidad operativa.....	160
Tabla 6: criterio para evaluar el conocimiento de prácticas ágiles.....	161
Tabla 7: valores culturales y comportamientos para evaluarlos.....	165
Tabla 8: criterio para evaluar los comportamientos de los valores culturales.....	166
Tabla 9: principios y comportamientos estructurales ágiles.....	166
Tabla 10: criterio para evaluar el grado de autoorganización.....	167
Tabla 11: criterio para evaluar la horizontalidad de la organización.....	167
Tabla 12: criterio para evaluar la institucionalización de un propósito común.....	168
Tabla 13: criterio para evaluar la implicación directiva.....	173
Tabla 14: criterio para evaluar la compatibilidad cultural los principios y valores ágiles.....	173
Tabla 15: criterio para evaluar la provisión de medios, formación y coaching.....	174
Tabla 16: criterio para evaluar el grado de predisposición al cambio.....	174
Tabla 17: Elementos que deben incluirse en los informes de evaluación.....	181

Índice de ilustraciones

Ilustración 1: Lanzamiento del Mariner I (NASA Great Images).....	10
Ilustración 2: IBM System 360 - Computer History Museum - Wikimedia Commons.....	11
Ilustración 3: Proceso y componentes de una profesión - “A Mature Profession of Software Engineering – Gary Ford, Norman E. Gibbs – SEI, 1996).....	14
Ilustración 4: 5 primeras áreas de conocimiento de la ingeniería del software - SWBOK v.2004.....	15
Ilustración 5: 6 segundas áreas de conocimiento de la ingeniería del software - SWBOK v.2004.....	16
Ilustración 6: Fortalezas de la producción basada en procesos.....	18
Ilustración 7: Personas, procesos y tecnología.....	19
Ilustración 8: Capital estructural y capital humano.....	20
Ilustración 9: Principales modelos de procesos para desarrollo de software.....	21
Ilustración 10: Niveles de madurez de las organizaciones.....	25
Ilustración 11: Niveles de capacidad de los procesos.....	27
Ilustración 12: ISO 12207, v. 2005.....	28
Ilustración 13: Operaciones vs. proyectos.....	32
Ilustración 14: Objetivos de la gestión de proyectos predictiva.....	35
Ilustración 15: Claves de la gestión predictiva.....	36
Ilustración 16: Resultados del informe periódico Standish Group Chaos Report..	38
Ilustración 17: Patrón dialéctico del conocimiento.....	41
Ilustración 18: Evolución de los procesos y prácticas para desarrollo de software.	42
Ilustración 19: Espiral dialéctica del conocimiento.....	44
Ilustración 20: Gestión de proyectos: diagrama de conceptos.....	45
Ilustración 21: Personas, procedimientos y tecnología.....	49
Ilustración 22: Adaptación a las circunstancias desde un mínimo viable.....	53
Ilustración 23: Valoramos más a los individuos y su interacción que a los procesos y las herramientas.....	57
Ilustración 24: Valoramos el software que funciona por encima de la documentación exhaustiva.....	58
Ilustración 25: Valoramos más la colaboración con el cliente que la negociación contractual.....	59

Ilustración 26: Valoramos más la respuesta al cambio que el seguimiento de un plan.....	60
Ilustración 27: Diagrama de conceptos en la gestión de proyectos.....	64
Ilustración 28: Conocimiento tácito vs. explícito.....	66
Ilustración 29: Roles: comprometidos e implicados.....	74
Ilustración 30: Ciclo iterativo scrum.....	78
Ilustración 31: Ejemplo de pila de sprint.....	81
Ilustración 32: Scrum diario.....	88
Ilustración 33: ¿Cuanto queda para terminar?.....	92
Ilustración 34: Unidades relativas.....	93
Ilustración 35: Tiempo ideal y tiempo real.....	95
Ilustración 36: Principios y valores de scrum.....	97
Ilustración 37: Gráfico burn down.....	106
Ilustración 38: Estimación de póquer.....	107
Ilustración 39: Kanban.....	109
Ilustración 40: Diagrama de espina de pez.....	112
Ilustración 41: Diagrama de árbol.....	112
Ilustración 42: diferencias entre la gestión de proyectos predictiva y la evolutiva o ágil.....	116
Ilustración 43: formación de scrum en rugby.....	119
Ilustración 44: Ikujiro Nonaka & Hirotaka Takeuchi.....	120
Ilustración 45: metodología scrum en OOPSLA 1995.....	120
Ilustración 46: ciclo estándar scrum.....	121
Ilustración 47: paradigmas culturales de las organizaciones a lo largo de la historia.....	123
Ilustración 48: características de las organizaciones, según su paradigma cultural.....	129
Ilustración 49: principales características de las organizaciones, según su paradigma cultural.....	130
Ilustración 50: agilidad y dimensiones de la empresa.....	132
Ilustración 51: estrategias de escalado de la agilidad.....	135
Ilustración 52: compatibilidad entre agilidad y cultura organizacional.....	137
Ilustración 53: principios, valores y soportes de la agilidad en la empresa.....	139
Ilustración 54: dimensiones y facetas de la empresa.....	140
Ilustración 55: mejora continua.....	143
Ilustración 56: sincronización y mejora continua.....	144
Ilustración 57: cadencia y sincronización de equipos.....	147
Ilustración 58: paradigmas culturales y autonomía de los equipos.....	153

Agilidad: perspectiva y principios del desarrollo ágil

Ilustración 59: mapa de riesgos según el grado de soporte para la mejora.....	174
Ilustración 60: niveles de agilidad.....	185
Ilustración 61: diagrama de evaluación.....	186