

01-053

### **OUTSOURCING PROJECT MANAGEMENT METHODOLOGY FOR INDUSTRIAL SOFTWARE DEVELOPMENT**

Vigil Berrocal, Miguel Angel <sup>(1)</sup>; Lago Collar, Antonio <sup>(1)</sup>; Villanueva Balsera, Joaquin <sup>(1)</sup>; Vergara González, Eliseo <sup>(1)</sup>

<sup>(1)</sup> Universidad de Oviedo

Currently, industry is considered to be immersed in the so-called fourth industrial revolution or industry 4.0, referring to the phenomenon of the digitalisation of production chains, the inclusion of virtual reality, artificial intelligence or the internet of things, which is essential to increase the competitiveness of products by reducing manufacturing costs, increasing production, improving reliability and the quality obtained. In this context, there is often a symbiosis between industry and software consultancy firms, which are responsible for developing and implementing the necessary solutions. In this way, the latter provide the former with up-to-date knowledge and labour flexibility. However, due to the characteristics of the outsourcing consultancy firms and the needs of the industry, there are certain problems when implementing project management methodologies, mainly due to the different nature of the two business typologies. This communication will present an industrial software outsourcing project management methodology that optimises the interface between industry and consultancy and will show its application by means of a real case.

*Keywords:* Project Management; software development outsourcing; industrial software development

### **METODOLOGÍA DE GESTIÓN DE PROYECTOS OUTSOURCING PARA DESARROLLO DE SOFTWARE INDUSTRIAL**

En la actualidad, la industria se considera inmersa en la llamada cuarta revolución industrial o industria 4.0, refiriéndose al fenómeno de la digitalización de las cadenas de producción, la inclusión de la realidad virtual, inteligencia artificial o el internet de las cosas, que se hace indispensable para poder aumentar la competitividad de los productos mediante la reducción de los costes de fabricación, el aumento de las producciones, mejorando la fiabilidad y la calidad obtenida. En este contexto resulta frecuente la simbiosis entre la industria y las consultoras software que se encargan del desarrollo e implementación de las soluciones necesarias. De este modo, estas últimas aportan un conocimiento actualizado y flexibilidad laboral a las primeras. No obstante, debido a las características de las consultoras que se dedican al outsourcing y las necesidades de la industria, existen ciertas problemáticas a la hora de implantar metodologías de gestión de proyectos, debido principalmente a la diferente naturaleza de ambas tipologías empresariales. En la presente comunicación se presentará una metodología de gestión de proyectos de outsourcing de software industrial que optimiza la interfaz entre industria y consultora y se mostrará su aplicación mediante un caso real.

*Palabras clave:* Dirección de Proyectos; externalización de desarrollo software; desarrollo de software industrial



© 2023 by the authors. Licensee AEIPRO, Spain. This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<https://creativecommons.org/licenses/by-nc-nd/4.0/>).

## 1 Introducción

Cada vez son más las empresas del mundo de la industria que deciden externalizar sus servicios informáticos. Para aumentar su productividad y competitividad, las empresas se centran en desarrollar su actividad principal y no desviar recursos a desarrollo y mantenimiento de software. La práctica de outsourcing se generalizó en la crisis económica del 2008 porque era más asequible para las empresas externalizar los servicios que contratar nuevos empleados (Selectiva, 2018).

La infraestructura IT en una empresa es vital para poder dar unos buenos ratios de productividad y calidad, pero para ello necesitan personal especializado, con experiencia en el sector, lo que resulta caro y complejo. Si la empresa no puede afrontar este gasto para poder contar con unos servicios IT de calidad deberá externalizar este servicio, esto tiene estos beneficios principales (Gestiopolis, 2018):

- Costes variables, ahorro de dinero.
- Mayor especialización tecnológica de los proveedores.
- Solución a medida y personalizada para la empresa.
- Mayor flexibilidad estructural, es más fácil cambiar de proveedor que la estructura empresarial de integración vertical.
- Responder con rapidez a los cambios del entorno.

Por contra, tiene los siguientes riesgos reconocidos (Montalván, 2009):

- Pérdida de know-how por parte de la compañía.
- Riesgo de pérdida de datos cruciales para la empresa.
- Menos agilidad, las soluciones tardan más en ser integradas
- Pérdida del control del servicio
- Riesgo de crear una relación con el proveedor con mucha dependencia
- Riesgo de externalizar actividades vitales cuyo esfuerzo extra para su mantenimiento puede no ser llevado a cabo por el proveedor al no tener la misma involucración con la compañía
- La comunicación y coordinación es más complicada.
- El costo ahorrado puede que no sea el esperado.

Un caso habitual de outsourcing es el de la provisión de servicios por parte de consultoras para grandes multinacionales industriales. En este tipo de proyectos la variable con más peso es la calidad, dado que una mala fiabilidad del software puede acarrear en el mejor de los casos unos costes enormes por mala fabricación del producto, y en el peor de los casos problemas de seguridad y de salud graves.

Sin embargo, los continuos procesos de contratación por proyectos implican una gran incertidumbre de mantener los retornos necesarios para mantener la operativa, lo que lleva a las consultoras de outsourcing para empresas industriales a verse forzadas a proveer de servicios software a bajo coste, lo que conlleva al desarrollo de productos software con mano de obra en muchos casos, joven o con poca experiencia en el sector, por lo que los procedimientos empleados para poder dar productos con la calidad exigida en la industria son vitales.

Por otro lado, existe mucha distancia entre el usuario final, y el propio desarrollador. El desarrollador no conoce el trabajo del operador y tiene muy pocos conocimientos sobre el negocio del cliente, y por su parte el operador tiene dificultades para expresar que necesidades reales tiene y que es lo que el desarrollador de software podría hacer por él. Esta distancia es además incrementada cuando se trata de una empresa de outsourcing perteneciente a otra cultura o sociedad.

Esto hace que la gestión de proyectos sufra dificultades para entender de verdad la necesidad del cliente y poder llevar el proyecto a buen término.

Este estudio nace a partir de la experiencia de uno de sus autores que, tras años dirigiendo proyectos para el sector industrial, y aplicar diferentes metodologías para la gestión de estos, se ha percibido una carencia en las metodologías para poder hacer una gestión completa y de calidad, con los recursos del sector. Por esta razón se propone una nueva metodología de proyectos adaptada a los proyectos de outsourcing de software industrial.

## **2 Desarrollo de la metodología**

Como punto de partida para la metodología propuesta se analizan las principales metodologías de gestión de proyectos y se comprueba cuál sería su aplicación más eficiente para mejorar:

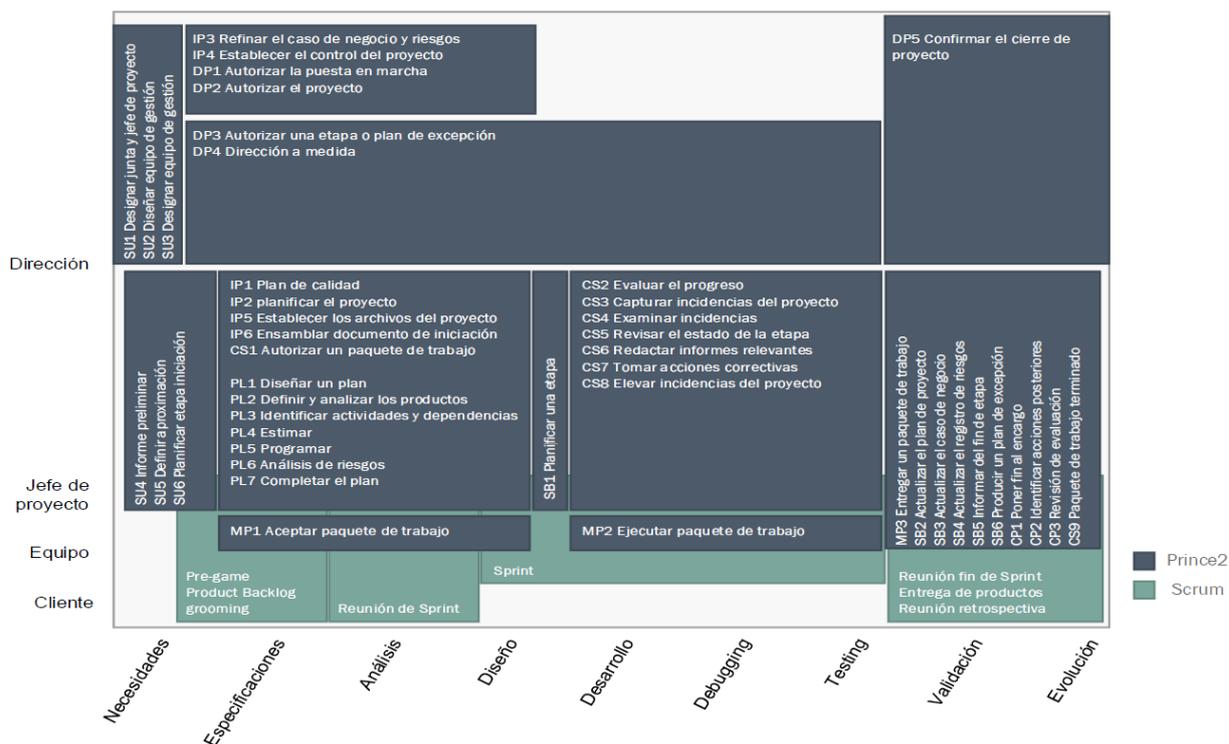
- La calidad del producto final.
- El mantenimiento del mismo y la fiabilidad.
- La agilidad con la que se responde a las necesidades del cliente.
- La gestión de equipos y su motivación.
- La capacidad para llegar a la necesidad real del cliente.
- El aporte de valor a la industria, innovación.
- La comunicación entre los implicados, especialmente el usuario final.

Existen dos tipos de metodologías en el mundo del software, las tradicionales o en cascada y las ágiles. Dentro de estos tipos, son diversas las metodologías existentes y muchos estudios comparativos entre ellas, como por ejemplo Navarro (2018), Martínez (2013) y Canal (2016), entre otros.

Esta comunicación se centrará en las dos metodologías más utilizadas en el mundo por las principales empresas y gobiernos. Para ello se analiza Prince 2 como principal metodología en el mundo de las metodologías tradicionales, y por otro lado en Scrum la metodología más usada en metodologías ágiles.

Cada una de las citadas metodologías fue confrontada con cada punto del ciclo de vida del software, lo que servirá para deducir qué metodología es más eficiente para cubrir esta fase y responder adecuadamente a las necesidades de la industria.

El análisis generado se resume en la siguiente Figura 1:



**Figura 1: Gráfico de procesos asociados al ciclo del software**

Por otro lado, a más alto nivel, se encontró que tanto las metodologías ágiles como las predictivas presentaban ventajas e inconvenientes para su aplicación a proyectos de outsourcing de software industrial. Por estas razones, se ha optado por proponer una metodología híbrida que dé respuesta a los retos específicos del outsourcing al tiempo que se beneficia de las principales bondades de cada perspectiva. Las conclusiones del estudio se resumen en la Figura 2.

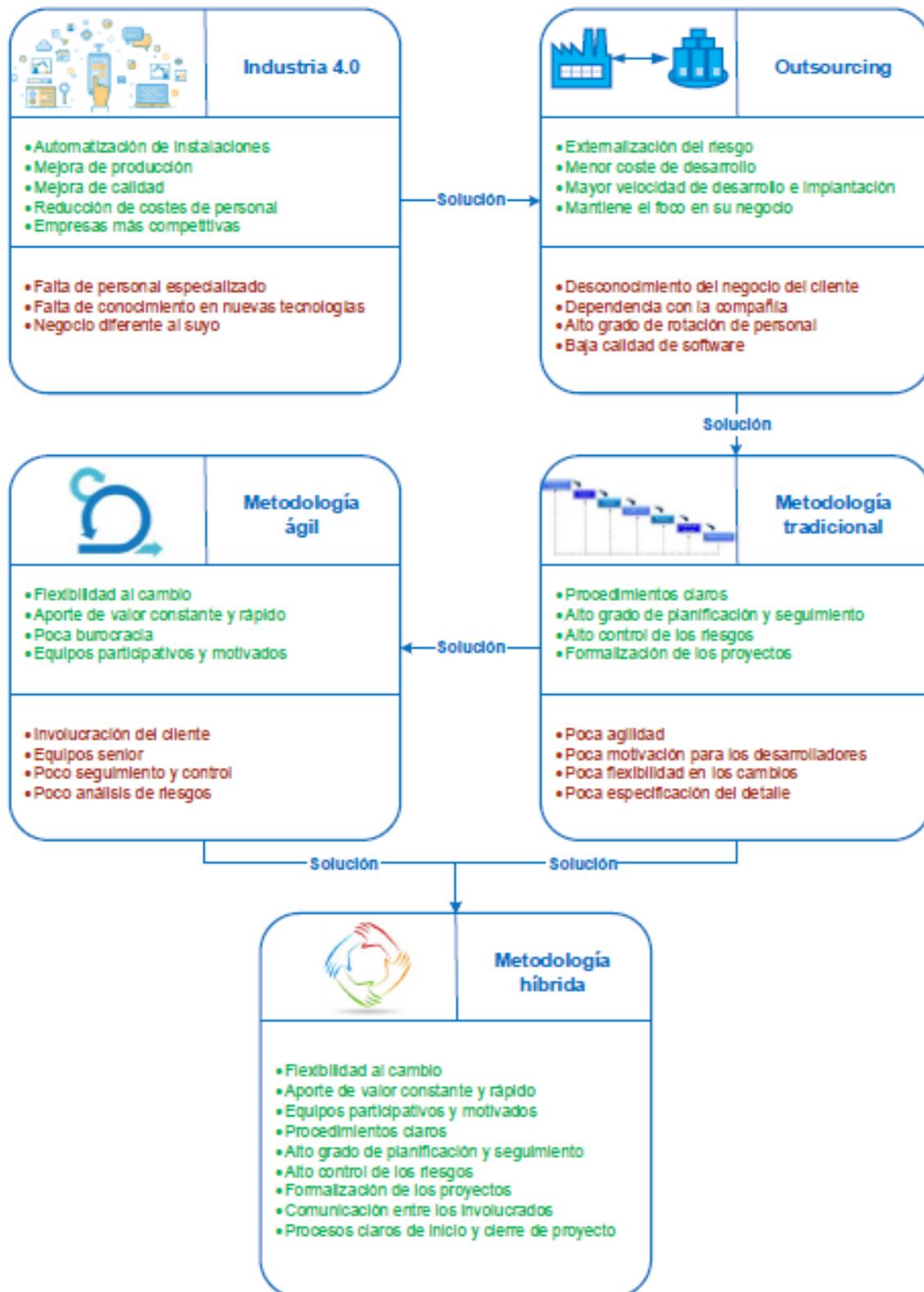
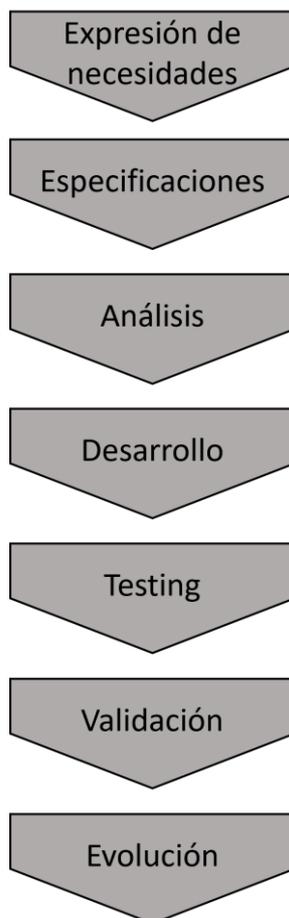


Figura 2: Resumen de conclusiones de aplicabilidad de metodologías ágiles y tradicionales

### 3 Descripción de una nueva metodología para desarrollo de software industrial desde una empresa de outsourcing

A continuación, se procede a presentar la metodología propuesta que tiene por objeto reducir burocracia y procedimientos que no aportan valor, para así reducir tiempos en los desarrollos y abaratar en lo medida de lo posible el coste de desarrollo.

La metodología sigue el esquema mostrado en la Figura 3:



**Figura 3: Estructura de la metodología**

#### 3.1 Expresión de necesidades

Una vez que el cliente decide contratar un servicio o el desarrollo de un software, el jefe de proyecto (en la empresa externa), se reúne con el cliente y deberá recoger la necesidad del cliente.

**Reunión en cliente.** Es importante que a esta reunión asistan los implicados por parte del cliente, es decir, será interesante que al menos acuda un usuario final del requerimiento, y por parte de la empresa auxiliar, además del jefe de proyecto, en algunos casos un especialista.

Se trata de recabar información, por lo que se delimitarán a escuchar, y hacer preguntas en aquellas cosas que no están claras. Siguiendo la regla del 80/20, escuchar el 80% del tiempo y hablar sólo un 20%.

Para terminar la reunión, se deberá hacer un resumen de lo acordado, por parte del jefe de proyecto, de forma que se demuestre al cliente que se entendió su necesidad, y que se entiende lo que el cliente espera de la empresa de outsourcing, validando así, esta primera comunicación.

Más tarde se rellenará un acta de la reunión que se enviará a todos los asistentes para que estos puedan validar que toda la necesidad ha quedado suficientemente aclarada y que no se escapa alguna cosa importante o que no se mal interpreto algún aspecto.

Esta acta deberá recoger al menos:

- Fecha y lugar de la reunión
- Asistentes
- Objeto de la reunión
- Resumen acordado
- Reparto de tareas y responsabilidades
- Fecha de próxima reunión
- Diseño del equipo de gestión del proyecto

Se decide que recursos se asignarán al proyecto y se forma el equipo. No es conveniente formar equipos de más de 6 personas, en el caso de que el proyecto requiera más recursos, se recomienda subdividir el proyecto de manera que cada subproyecto este gestionado por equipos de entre 5 a 7 personas. Según Huseem (2006), un equipo debería ser de entre 4-6 personas, aspecto corroborado por la experiencia de los autores.

**Informe preliminar del proyecto.** Se debe hacer un estudio inicial de la propuesta, comprobar si ya hay trabajos semejantes realizados dentro de la propia empresa. Asimismo, se debe revisar el mercado y comprobar que tecnologías se están empleando para estos trabajos.

Valorar si el proyecto es necesario, si no existe ya algo que cubra esa necesidad, o si el proyecto no va a ser rentable para la empresa.

Tras esto realizar un breve documento donde se explique el alcance inicial del proyecto. Incluyendo un análisis preliminar de los riesgos, los principales stakeholders, el plan de comunicaciones, una estimación inicial de plazo y coste, etc.

**Registro de la documentación.** Se crea una estructura donde se pueda almacenar la documentación referida a este proyecto.

Es importante definir un estándar de nomenclatura, para que luego resulte más sencillo ordenarla y buscarla

**Planificación de la calidad.** Se deberá dejar constancia de los controles de calidad, las situaciones y las pruebas que el software debe superar para ser aceptado.

En este documento se añadirá una guía de estilos de desarrollo de código que deberán seguir los desarrolladores. Esta guía incluirá estilos de codificación, patrones de diseño si los hubiese, y procedimientos de pruebas que deberá pasar todo entregable.

Se debe dejar constancia de las herramientas de control de estilo de código usada, con las reglas y el nivel de aceptabilidad dado para el proyecto.

Por último se definirán las reglas para el control del código fuente y la estrategia a seguir para el mantenimiento de versiones y las subidas a producción.

La Figura 4 resume las entradas y salidas de esta fase de la metodología.



Figura 4: Entradas y salidas de la fase Expresión de necesidades

### 3.2 Especificaciones

Se deberá mantener una reunión en la que participe el responsable en el cliente y los miembros del equipo de desarrollo.

**Reunión inicial con el equipo de trabajo.** En una primera reunión el director del proyecto expondrá a los miembros del equipo el alcance del proyecto, y se debatirá entre todos hasta apuntalar todas las dudas al respecto, y tomar nota de las distintas alternativas. Puede ser necesario una investigación previa sobre diferentes partes del proyecto un tanto más complejas, lo que podría dar lugar a una segunda reunión de equipo.

Esta reunión debe cubrir:

- ¿Qué?, es lo que necesita el cliente, indica la necesidad que nos expresó.
- ¿Por qué?, De esta pregunta pueden surgir alternativas, porque al mejor para cubrir la necesidad que tiene el cliente, existen otras alternativas que le pueden ser más beneficiosas.
- ¿Cómo?, en esta parte se debatirá qué tecnologías emplear, incluso en que fases se podría dividir el proyecto, y se concretará una propuesta hacia el cliente.

En esta fase se recomienda emplear técnicas que fomentan la creatividad y el trabajo en equipo como por ejemplo Design Thinking. Para mayor referencia de esta técnica consultar (Dinngo, 2018).

**Reunión en cliente para presentar la propuesta de proyecto.** Se realiza una vez que se tiene clara la necesidad del cliente, y que se ha aclarado las dudas, diseñado y enfocado una posible solución. Se debe hacer una reunión con el cliente, donde al igual que en la reunión inicial, acudan los involucrados. Se presenta la solución y se explican los detalles, y como se llevaría a cabo.

Es posible que en esta reunión afloren carencias o problemas que no se hayan identificado en la reunión inicial, en ese caso, se deberá preparar una nueva reunión donde ya se aproxime a una solución satisfactoria para el cliente.

Una vez que se llega a este punto, se deben recoger todas las especificaciones y requisitos a tener en cuenta en el desarrollo a alto nivel.

Se creará un acta de la reunión de formato similar al visto en la fase anterior, donde además se incluirá:

- Los requisitos

- Un análisis del riesgo y qué medidas tomar para mitigarlo
- Una estimación de tiempo y coste

Más tarde, una vez analizado todo el proyecto en detalle, se conformarán las historias de usuario junto con los especialistas y el cliente.

La Figura 5 resume las entradas y salidas de esta fase de la metodología.



Figura 5: Entradas y salidas de la fase Especificaciones

### 3.3 Análisis

En el análisis ya se parte de un objetivo claro, y en proyectos grandes, se dispone ya la subdivisión del proyecto en fases. Por lo que se comienza por la fase acordada con el cliente.

**Creación de las historias de usuario.** Se deben reunir el cliente y el equipo de desarrollo para que el cliente explique cada historia de usuario y se vayan conformando. El equipo tomará notas siguiendo la regla del 80/20.

El resultado final de esta reunión deberá ser una pila de backlog con Historias de usuario.

**Reunión de sprint.** Se realiza la reunión de sprint a la que deberá asistir el representante del cliente. En caso de ausencia del cliente, el jefe de proyecto hará las veces de scrum máster y se pondrá en el lugar del cliente como máximo conocedor de las necesidades de este.

Con el backlog priorizado, se irán explicando las historias de usuario hasta que a los miembros del equipo de desarrollo les quede totalmente claro la funcionalidad a desarrollar.

En un Scrum puro, el equipo es multidisciplinar, y autogestionado. En empresas de outsourcing dedicadas al desarrollo a bajo coste, los miembros del equipo suelen ser gente joven con poca experiencia, por lo que será el jefe de proyecto quien deberá conocer bien la tecnología y ser consciente de las posibles dificultades que pueda haber en el desarrollo, así como las peculiaridades de los miembros de su equipo. Por esto será este quien reparta las tareas y estime los plazos, aunque velará por la formación del equipo buscando que en el futuro gane autonomía.

#### Informes a la junta del proyecto

La Figura 6 resume las entradas y salidas de esta fase de la metodología.



Figura 6: Entradas y salidas de la fase Análisis

### 3.4 Diseño y desarrollo

**Diseño.** Esta parte corre a cargo del desarrollador responsable de la tarea, aunque siempre supervisada por el jefe de proyecto.

En esta etapa se pueden generar documentos de tipo diagrama de clases, mapeo de comunicaciones, etc. Estos documentos tienen que ser correctamente registrados y almacenados en la estructura de la documentación, que quedará registrada en el control de versiones.

En diseños complejos se pueden usar técnicas como las tarjetas CRC-Clase, Responsabilidades y Colaboradores (agilesurfers.es, 2018).

Tendrá que ser muy modular, de forma que se puedan reaprovechar los diferentes módulos en otros diseños, esto hará la aplicación mucho más mantenible y robusta, permitiendo cambios en los distintos módulos sin que estos afecten al resto. Para ello se debe utilizar interfaces que abstraigan el código.

Los diseños pueden ser ágiles, no es necesario definir el diseño completo al inicio. Se puede partir de un diseño general e ir construyéndolo a medida que avanza el desarrollo, de esta manera se adaptará a cambios y evoluciones.

Separar completamente el diseño de la codificación tiene sus riesgos y siempre es mejor contar con profesionales que puedan diseñar y codificar.

Se debe cuidar el diseño estético de las interfaces de usuario ya que es la tarjeta de presentación de la empresa. Por ello, tiene que realizarse antes para que pueda además servir de maqueta, durante para gestionar los cambios que se puedan producir durante el desarrollo y al final, a la hora de integración y en producción, para recoger el feedback del cliente y la usabilidad del producto ya en producción.

**Desarrollo.** Durante la fase de desarrollo si surge alguna incidencia se comunicará al representante en cliente, y una vez replanteada, se actualizará la historia de usuario y se seguirá el ciclo normal.

De ser una incidencia, compleja se deberá tomar estas acciones en función de la situación:

- Incidencia que afecta o que genera dependencia al resto de tareas del sprint: Se volverá a convocar una reunión de Sprint, donde se volverá a analizar esa parte, y se reestructurará la lista de sprint con las nuevas tareas.
- Incidencia que no genera dependencia, se puede sustituir por otra historia de usuario y dejar el análisis de esta para la reunión de sprint siguiente.

Es necesario registrar la comunicación con el cliente donde se acuerdan los cambios en el planteamiento. Bien vía email, registrando dichos emails, y las versiones de la historia de

usuario, o de existir una reunión, un acta con dicha reunión que se enviará al cliente para que quede constancia.

**Seguimiento del sprint.** Diariamente los desarrolladores deberán rellenar una hoja para ver el gráfico de burn-down.

Se pueden dar situaciones anómalas durante el Sprint que deben ser resueltas caso a caso.

En la reunión post-sprint, se deberá analizar estas situaciones y entre todos idear soluciones para intentar que no se repitan estas situaciones anómalas o que si se repiten estén contempladas acciones para mitigarlas.

Además, en estas reuniones post-sprint se puede valorar la velocidad del equipo y discutir sobre cómo mejorarla.

Los desarrolladores rellenaran un kanban, idealmente que sea una pizarra física que esté visible por todo el equipo y de esa manera todos sean conscientes en todo momento de cómo está resultando el sprint. De esta forma el gestor del proyecto de una forma rápida puede visualizar el estado del proyecto e identificar posibles cuellos de botella o retrasos.

**Comunicación.** Se debe fomentar que los miembros del equipo estén al tanto de lo que hacen sus compañeros dado que propicia:

- Colaboraciones en forma de ayuda.
- Compartir conocimiento, lo que ayuda a la formación de equipos multidisciplinares.
- Empatías que consiguen un mayor vínculo entre los miembros del equipo
- Equipos robustos que consiguen tirar unos por otros para llegar a un buen resultado

Para ello es importante contar con herramientas tipo chat, para que, aunque trabajen juntos, no se interrumpan.

Además, se hacen reuniones diarias, a media mañana, para no interrumpir esas primeras horas de máxima productividad, de unos quince minutos aproximadamente, donde de pie y situados en círculo, los desarrolladores deberán responder a:

- ¿Qué hice ayer para ayudar al equipo a alcanzar el objetivo del sprint?
- ¿Qué voy a hacer hoy?
- ¿Estoy teniendo algún problema?

Se debe realizar en un tiempo máximo de quince minutos, porque esto anima a los equipos a empezar a trabajar rápidamente en los objetivos (Steel, 2010).

La Figura 7 resume las entradas y salidas de esta fase de la metodología:



Figura 7: Entradas y salidas de la fase Desarrollo

### 3.5 Testing

La parte diferencial de desarrollar software industrial es la alta fiabilidad y calidad que debe tener su código.

Una vez que el desarrollador termina de codificar, deberá gastar un tiempo en refactorizar bien su código, asegurándose que su desarrollo cumple las premisas del código limpio (Martin, 2012).

Para hacer el testeo del código y comprobar que se cumple con las especificaciones propuestas, con cualquier combinación de uso y se controla vía código todas las situaciones, se deben realizar tres tipos de pruebas:

1. Pruebas unitarias: Se prueba una única pieza de código.
2. Pruebas de integración: Son pruebas que se ejecutan después de las unitarias, para probar conjuntos de software más amplios.
3. Pruebas funcionales o de aceptación: Prueban el conjunto de la aplicación.

La Figura 8 resume las entradas y salidas de la fase de Testing:



**Figura 8: Entradas y salidas de Testing**

### 3.6 Validación

Esta fase se realizará en dos niveles.

**Validación por el equipo de desarrollo.** Una vez que el desarrollador da por finalizada la historia de usuario, otro miembro del equipo o incluso de otro equipo, deberá probar que cumple con todos los requisitos, además de las reglas de estilo que tenga la aplicación y las normas de usabilidad, dejando el debido registro.

En esta plantilla se cubre si la funcionalidad indicada, más todas las normas de la empresa o las fijadas por el cliente que sean comunes a todo el proyecto cumplen o no cumplen.

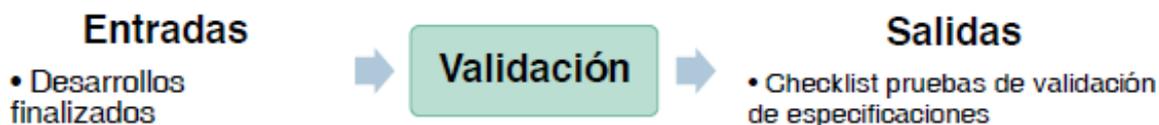
**Validación en cliente.** El jefe de proyecto informará al responsable en cliente, de que el producto está ya en integración listo para subir a producción.

El cliente deberá verificar que el producto cumple con los requisitos solicitados, y podrá mostrarlo al resto de implicados.

Será probable que se necesiten realizar algunos cambios o mejoras, se tratan como nuevas historias de usuario que pasarán a la pila de backlog priorizadas según el criterio del cliente y que comenzarán el ciclo de desarrollo como si de una historia nueva se tratase.

Es por esto que las tareas del sprint terminan una vez que se ponen las historias de usuario en integración y no en producción como marca la teoría, esto se debe a que esta puesta en producción no depende del equipo de desarrollo sino del cliente.

La Figura 9 resume las entradas y salidas de la etapa de Validación:



**Figura 9: Entradas y salidas de Validación**

### 3.7 Evolución

Una vez que se termina la fase anterior, se volvería a realizar una nueva interacción del proyecto, con una nueva reunión de Sprint, donde se volverían a decidir qué historias de usuario abordar para seguir avanzando en el proyecto.

Se puede dividir esta reunión en dos fases y en la primera fase dedicarla a hacer retrospectiva, de forma que se repasen todas las historias de usuario realizadas en el sprint anterior contemplando lo siguiente:

- Si se terminó según lo esperado y sin problemas.
- Si no se terminó, porque fue, ¿mala planificación de tiempo?, ¿problemas inesperados?, ¿problemas del miembro del equipo asignado?
- ¿Qué problemas surgieron que no estaban planificados?
- ¿Cómo abordar esos problemas en el futuro?
- ¿Cómo mejorar el rendimiento del grupo?
- ¿Qué riesgos empiezan a aparecer? ¿Cómo afrontarlos?
- Perspectiva general del proyecto

Es importante recoger la opinión de todos los miembros del equipo para que se mantengan involucrados en el proyecto y se sientan participes de las decisiones que adopta el grupo.

Intentar recoger el feedback del usuario final como parte de la mejora de calidad, con encuestas adaptadas al tipo de proyecto que se entregó para lograr ver donde se puede mejorar en los siguientes desarrollos o atajar posibles fallos de diseño y funcionalidad.

Estos proyectos cuentan con una garantía de unos meses, en los que se deben resolver posibles incidencias que surjan durante su funcionamiento. Esta garantía debe quedar bien definida, para que ambas partes tengan claro qué tipo de incidencias se cubren y que siempre sean de tipo bug, falla en el funcionamiento, un requerimiento mal entendido, por parte de la empresa de outsourcing a pesar de estar bien contemplado en la especificación, etc. Pero nunca una modificación en las especificaciones, esto no lo debería de cubrir la garantía del proyecto, porque en ese caso sería difícil dejar cerrado un proyecto, ya que estos siempre son susceptibles de cambios y mejoras.

La Figura 10 resume las entradas y salidas de esta fase de la metodología:



Figura 10: Entradas y salidas de la fase de Evolución

### 3.8 Seguimiento del proyecto

Tal y como se muestra en la Figura 11, esta fase es transversal a varias fases del proyecto.

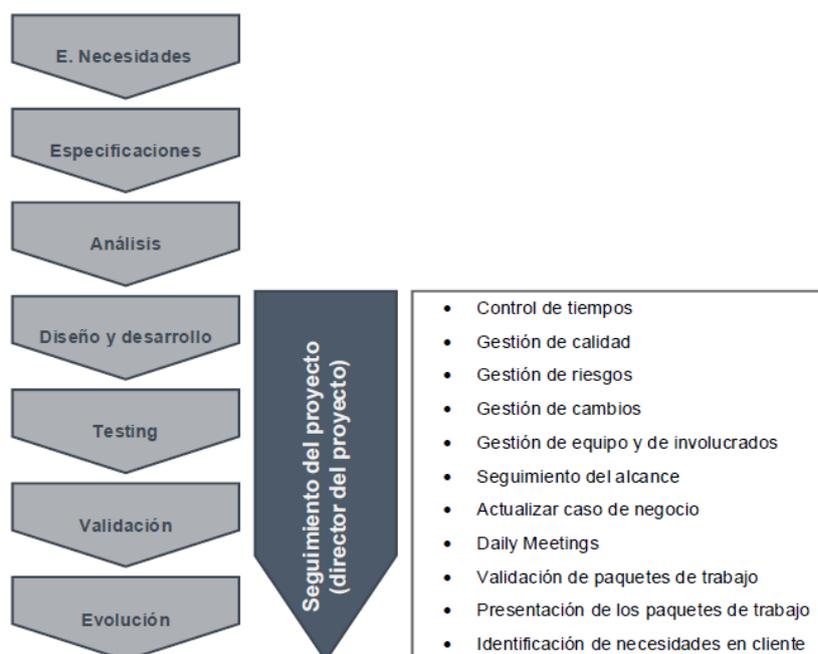


Figura 11: Contextualización de la fase Seguimiento del proyecto

Lo ideal es que el jefe de proyecto no participe del diseño y desarrollo en lo que se refiere a la codificación, sólo asesore al equipo, resuelva dudas, mantenga comunicación activa con el cliente y con la dirección de la empresa. Además, es probable que el director de proyecto esté gestionando varios proyectos a la vez, por lo que en estas fases sus labores deberían ser:

- Control de tiempos, que puede contar con la asistencia de herramientas tipo Microsoft Project, Jira, etc.
- Gestión de calidad. Al dividir el proyecto en historias de usuario, es fácil perder la perspectiva, es labor del director de proyecto que tiene una visión global del producto, comprobar que no se está excediendo en las funcionalidades ni por supuesto quedándose cortos.
- Gestión de riesgos
- Gestión de cambios

- Gestión de equipo
- Gestión de involucrados
- Seguimiento del alcance del proyecto. Poniendo limites en las fases del proyecto si es necesario, es importante hacer un buen seguimiento del proyecto de forma que no se exceda el alcance y no se incurran en sobrecostos para la empresa.
- Actualizar el caso de negocio. Alertando en el caso de que se detecte de que el producto deja de tener sentido para el cliente o para la empresa.
- Gestión documental
- Identificar necesidades en el cliente

La Figura 12 resume las entradas y salidas de esta fase de la metodología:



**Figura 12: Entradas y salidas de la fase de Seguimiento del proyecto**

#### 4 Conclusiones

En plena revolución industrial 4.0, la industria necesita seguir modernizando sus instalaciones y automatizando sus procesos para poder seguir siendo competitiva, mejorar su producción, su fiabilidad y su calidad.

Para ello sus departamentos de IT, necesitan más recursos humanos para poder abastecer de esa tecnología a las instalaciones, lo que supone un alto coste para la empresa que muchas veces no está dispuesta a asumir, de ahí que se produzca la subcontratación de estos servicios y crezcan con fuerza las empresas de outsourcing en el sector industrial.

Este tipo de proyectos tiene una casuística muy concreta que hace que no encajen totalmente ni en las metodologías ágiles de gestión de proyectos, ni en las tradicionales.

A lo largo de esta comunicación se ha presentado una nueva metodología híbrida adaptada específicamente a los proyectos de outsourcing de desarrollo software industrial que ha sido probada con éxito en un caso real.

#### 5 Referencias

Canal, J. M. (2016). *Identificación de procesos para la mejora de los proyectos multidisciplinares y de corta duración. Trabajo Fin de Máster*. Universidad de Oviedo

Gestiopolis. (2018). Obtenido 15 de marzo de 2023 de: <http://www.gestiopolis.com/que-es-outsourcing-ventajas-y-desventajas/>

Jerry Useem (2006, 1 de junio). How to Build a Great Team How to Build a Great Team. *Fortune*

Martin, R. C. (2012). *Código Limpio*. Madrid: Anaya Multimedia

Martínez, P. S. (2013). *Identificación y valoración de técnicas ágiles de gestión de proyectos software*. Trabajo Fin de Máster. Universidad de Oviedo

Montalván, J. (2009, 21 de marzo). *Blog de Julio Montalván*. Obtenido en 10 de marzo de 2023 de <https://juliomontalvan.wordpress.com/2009/03/21/el-outsourcing-en-tecnologias-de-la-informacion-ti/>

Navarro, J. G. (2018). *Estudio comparativo de metodologías, herramientas y wiki de soporte para la gestión de proyectos de desarrollo software*. Trabajo Fin de Grado Universidad Pública de Cataluña.

Selectiva. (2018). Obtenido 15 de marzo de 2023 de <https://selectiva.es/que-es-outsourcing-beneficios-empresas/>

Steel, P. (2010). *The Procrastination Equation: How to Stop Putting Things Off and Start Getting Stuff Done*. Calgary: HarperCollins

**Comunicación alineada con los Objetivos de Desarrollo Sostenible**

