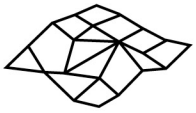


ANÁLISIS DE LAS CAUSAS DE FRACASO DE PROYECTOS EN LA INDUSTRIA FABRICANTE DE SOFTWARE

Luis Alonso

ABSTRACT. *The software industry manufacturers are really sensitive to project's failures. Missed deadlines, continuous scope changes, end-user dissatisfaction and many others, are problems that have to be solved daily in this kind of industry. Through different approaches to software industries, we have found that, to a great extent, these failures are common patterns among different companies. The way to solve this big problem is to establish a way of working according to industry best practices. It is not enough to focus only on one framework; it is necessary to work together with different frameworks to achieve the ultimate goal: complete projects in good time and in an appropriate manner.*

PROBLEMÁTICA DETECTADA

La industria de la creación de software presenta características muy particulares respecto a la gestión de proyectos, debido a la naturaleza de los productos que maneja.

Según un análisis de *Standish Group*[1] el 31% de los proyectos nunca se llegan a terminar y el 52.7% tienen importantes sobrecostes. Aún más preocupante, es el estimando de que en el 83.8% de los proyectos de software se presentan retrasos en la conclusión.

En una encuesta, de realización propia, sobre el *target* de empresas fabricantes de software de menos de veinticinco desarrolladores, se concluyó que el 76.7% de los proyectos se realizan con modificaciones sustanciales sobre los requerimientos iniciales del producto.
A la luz de estos datos, se pude concluir que la gestión de los proyectos de software, por empresas de pequeño y mediano tamaño, es claramente deficiente, presentando grandes oportunidades de mejora en cuanto a calidad y eficiencia de los procesos.

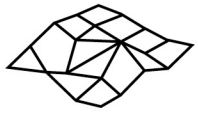
ANÁLISIS DEL PROBLEMA

El sector de la creación de software es una industria con fama de sólida, pero al tenor de los resultados de los análisis y, salvo honrosas excepciones, la calidad generalizada de sus productos y servicios dista bastante de la esperada.
Estos proyectos se clasifican en tres grandes bloques en función del resultado de los mismos:

- *Successful.* Proyectos completados, que cumplen los planes de tiempo y presupuesto, con todas las características acordadas.
- *Challenged.* Proyectos completados y operativos, en los cuales no se han respetado los límites de tiempo, presupuesto o funcionalidad acordados inicialmente.
- *Failed.* Proyectos cancelados antes de su finalización.

Para determinar las causas raíces que se esconden detrás de esos proyectos considerados deficientes (*Challenged* o *Failed*), se realizó una encuesta entre el personal que se involucra en proyectos de fabricación de software. A diferentes niveles dentro de la organización, desde programadores, hasta directivos, pasando por jefes de proyecto. El resultado de esta encuesta se muestra en la tabla 1: "factores de fracaso de los proyectos de software." Ésta busca establecer las causas objetivas para determinar los motivos por los cuales los proyectos no se llevan a buen puerto.

NOVIEMBRE 2015



CAUSA	%
Mal levantamiento de requerimientos	40,75
Deficiencia de recursos o recursos no cualificados	20,5
Mala planeación del proyecto	17,75
Falta de apoyo de dirección	11,5
Jefes de proyecto no capacitados	9,5

TABLA 1. FACTORES DE FRACASO EN LOS PROYECTOS SW

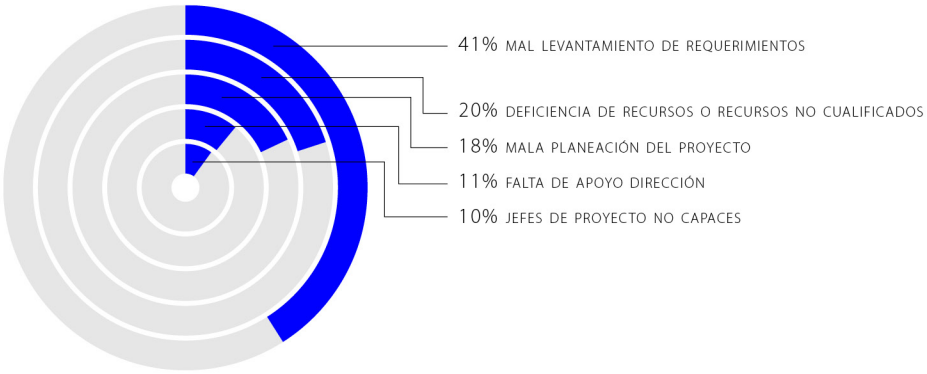


GRÁFICO 1. FACTORES DE FRACASO EN LOS PROYECTOS SW

Como se observa en el gráfico 1, se puede establecer una serie de causas o factores que potencian los malos resultados a la hora de desarrollar software. Por tanto, se puede afirmar que si se minimizan estos factores se conseguirá que los proyectos sean más eficientes y eficaces.

Analizando los 5 factores claves para el fracaso en los proyectos software, se pude establecer un patrón común: *Todos son abordables en la fase de planeación del proyecto.*

BÚSQUEDA DE LA CAUSA RAÍZ

La mayoría de los proyectos fallidos en la industria de desarrollo de productos software se podrían resolver aplicando correctamente los marcos de buenas prácticas, que están presentes en la actualidad para el global de la industria tecnológica.

Si bien es cierto, que esos marcos suelen ser muy generales una buena interpretación, acorde a la realidad de la industria fabricante de software, es vital para poder cortar de raíz los problemas que son la causa de los fallos en los proyectos.

Analizando las causas subyacentes de los factores de fracaso, detectados en el grafico 1, se pueden concluir las siguientes afirmaciones:

- El problema principal, identificado por la gran mayoría de profesionales de la industria de desarrollo de software es un *mal levantamiento de los requerimientos (41%)*. Esto es una clara evidencia de un problema mayor que está presente en todas las industrias tecnológicas: la desconexión y el desconocimiento de las necesidades de los clientes. Conseguir que los departamentos más técnicos de la organización (desarrollo, soporte, etcétera) conozcan y empaticen con las necesidades reales de los clientes, es crítico para obtener el nivel de calidad de productos que demanda el mercado.
- La *deficiencia de los recursos* en cuanto a cualificación, ya sean los jefes de proyecto o los propios desarrollados, es el segundo gran bloque de factores de fracaso *(30%)*. Los recursos son las piezas básicas que van a permitir que un proyecto se concluya correctamente. Poder contar con personal adecuado a la naturaleza de cada proyecto, es básico para tener garantía de éxito.
- La *mala planeación (18%)* es la tercera causa de fracaso de proyectos. En este bloque se podría hablar tanto de mala planeación por desconocimiento de las fases necesarias para llevar a cabo un proyecto de desarrollo de software, como los *deadlines* no alineados

NOVIEMBRE 2015

con la realidad del mismo. Uno de los grandes problemas presentes en todos los proyectos de tecnología es la falta de realismo a la hora de marcar plazos. En ese sentido, es necesario tener un conocimiento muy amplio de la capacidad de la organización y del calendario de otros proyectos para poder programarlos de forma correcta.

- Por último, aunque no menos importante, *la falta de apoyo de dirección (11%)* también es un factor a tener en cuenta. Se considera esta falta de apoyo como la carencia de poder de los departamentos tecnológicos a la hora de tomar decisiones, frente a otros departamentos con mayor presencia, como podría ser el de ventas. Es importante que los departamentos tecnológicos tengan el suficiente apoyo de la alta dirección, para trasmitir al resto de áreas la urgencia que tienen en la consecución de los objetivos globales de la organización.

La solución, como ya se ha comentado, pasa por aplicar, de forma correcta y adecuada a las necesidades de la industria del desarrollo de software, las buenas prácticas del sector, pero ¿Cuáles son esas buenas prácticas?

- Buenas prácticas relacionadas con la gestión de servicios *(ITSM – IT Service Management)* más en concreto el marco de referencia ITIL®.
- Normas y estándares de la industria: ISO/IEC 29110 Ingeniería de Software –Perfiles de Ciclo de Vida en Pequeñas Entidades.
- Buenas prácticas relacionadas con la integración de las áreas de desarrollo y operación: DevOps.

SOLUCIÓN

Las empresas desarrolladoras de software no se dedican única y exclusivamente al desarrollo de software. Dentro de sus cometidos también está el gestionar por completo el ciclo de vida de sus productos. Esto es, dar soporte a las soluciones, detectar y corregir bugs o gestionar las solicitudes de nuevas funcionalidades.

Entender que el ciclo de vida de un producto va más allá de su mero diseño y creación, acerca el enfoque de la gestión de productos software a una visión de servicios como la que se realiza en ITIL®

Entendiendo un servicio desde una perspectiva VOCR – Generar valor *(Value)* a los clientes, facilitando la opción de resultados *(Outcomes)*, sin que estos asuman los costes *(Costs)* y riesgos *(Risks)* específicos de tenerlo en propiedad. ITIL® establece una ciclo de vida completo para gestionar estos servicios.

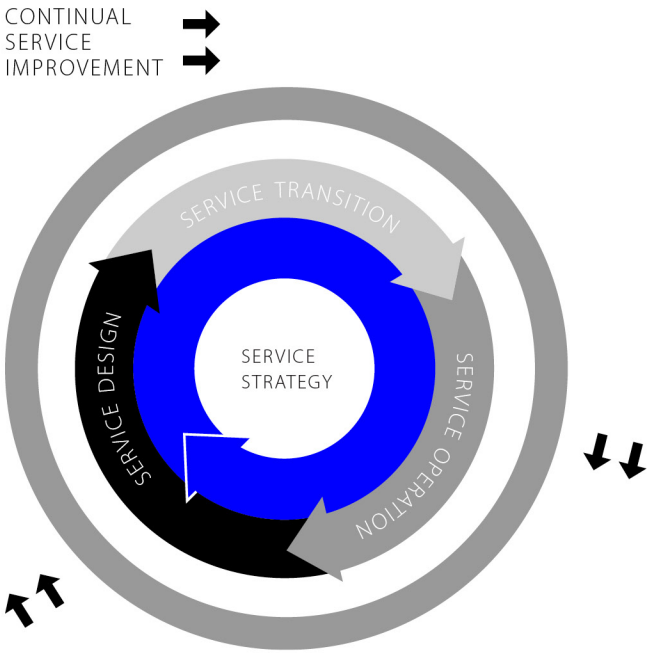


GRÁFICO 2. CICLO DE VIDA DEL SERVICIO EN ITIL®

Una de las tónicas que se repite durante los libros de ITIL®, es la necesidad de que los servicios de TI estén alineados a las necesidades del negocio. Tratan-

do de inculcar un cambio de mentalidad, en el que la tecnología se deje de ver como un gasto y se empiece a considerar una inversión. La tecnología en general y el desarrollo de software en particular, son piezas clave para obtener los resultados que la organización tiene proyectados.
El objetivo final de ITIL® es generar valor a la organización con base en el correcto uso de los servicios de tecnología. Así, se establecen cinco fases a lo largo del ciclo de vida del servicio:

- Estrategia
- Diseño
- Transición
- Operación
- Mejora continua

Estas fases permiten establecer un control total sobre el servicio que se presta, garantizando la creación de valor. Hay que entender valor como la unión de la funcionalidad y la garantía de un producto, tal y como se muestra en el gráfico 3.

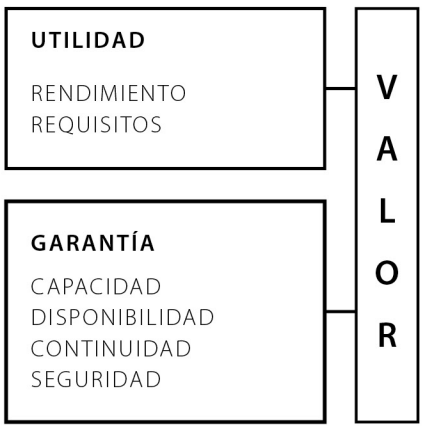


GRÁFICO 3. GENERAR VALOR

Partiendo de estas premisas, aplicar las buenas prácticas en la gestión de servicios a la industria fabricante de software, permitirá minimizar el impacto de los factores comentados en la tabla 1. Conocer las necesidades de los clientes y del negocio, realizar un buen levantamiento de requerimientos, calcular plazos de forma realista y gestionar las expectativas, son, entre otras, las ventajas que acarrearía el alineamiento de las empresas fabricantes de software a las buena prácticas de ITIL®.

De forma paralela, la norma ISO/IEC 29110 busca regular y mejorar la calidad de los productos y/o servicios de las empresas desarrolladores de software. El enfoque de esta norma está en el correcto desempeño de las pequeñas y medianas empresas del sector. Estas empresas fueron, como ya se ha comentado al principio de este artículo, el foco de la encuesta para detectar los factores críticos de fracaso de los proyectos de desarrollo software.

La norma ISO/IEC 29110 describe una serie de actividades y tareas para los procesos de "gestión de proyectos" y de "implantación de software". Aunque ambos trabajan en partes diferentes de la cadena de elaboración de productos software, presentan relaciones entre sí, y es necesaria la colaboración de ambos procesos para obtener los resultados esperados.

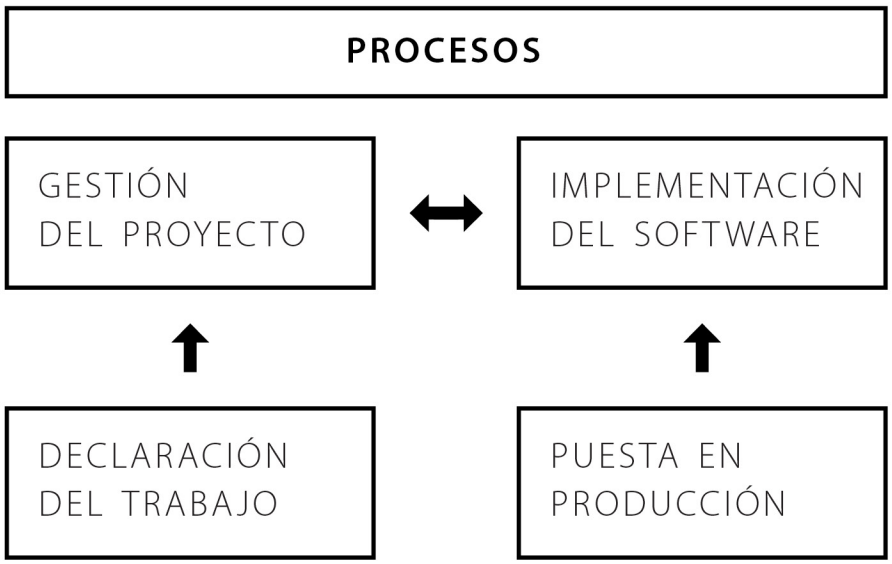


GRÁFICO 4. PROCESOS ISO/IEC 29110

Los objetivos específicos de cada uno de estos procesos son diferentes. La colaboración entre éstos va a permitir conseguir el objetivo final, que no es otro que garantizar la calidad de los productos software.

- El proceso de *Gestión de Proyectos* tiene por objeto, establecer y llevar a cabo las tareas definidas y necesarias para la correcta implementación de software. Este proceso, es el que permite que se

cumplan los objetivos del proyecto en términos de calidad esperada, tiempo y costo.

- El proceso de *Implementación del Software*, tiene como objetivo que la construcción, las pruebas y la entrega de productos para el software se hagan de acuerdo a los requisitos especificados.

Dentro de la norma ISO/IEC 29110 se establecen como elemento básico los paquetes de puesta en operación (PPO). Estos paquetes, son un conjunto de elementos desarrollados para facilitar la implementación. En el gráfico 5 se observan los paquetes de puesta en operación que han sido desarrollados para ayudar a implementar los procesos amparados por la norma ISO/IEC 29110.

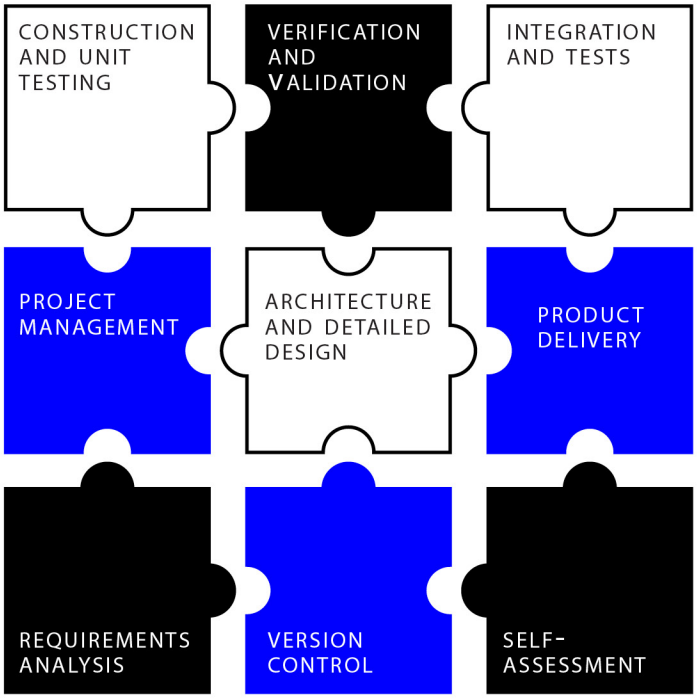
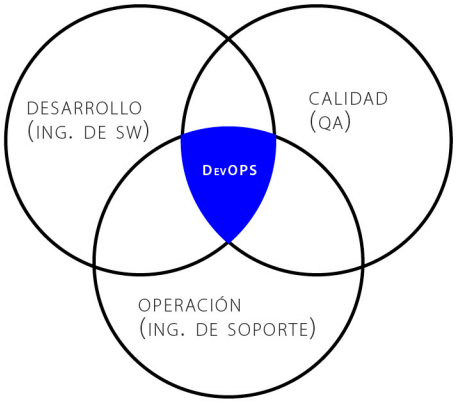


GRÁFICO 5. PAQUETES DE PUESTA EN OPERACIÓN

La implementación de la norma ISO/IEC 29110 permitirá a las empresas de desarrollo de software controlar que sus productos estén alineados a las necesidades de los clientes en términos de funcionalidad, calidad, costo y tiempo de ejecución. Lo que derivará en que los productos software puedan ser desarrollados en tiempo y forma, mejorando la percepción de los clientes y, en definitiva, la satisfacción de los mismos.

Un gran objetivo de la industria tecnológica en general, y de los desarrolladores de software en particular, es eliminar la independencia entre las áreas de desarrollo de software y las áreas operativas. Es necesario inculcar a la organización un cambio de mentalidad que potencie un acercamiento entre áreas, encaminado a fomentar la transferencia de conocimiento. Este cambio de mentalidad viene impulsado por la cultura *DevOps*, que no deja de ser un movimiento que busca la colaboración entre los desarrolladores de software y los

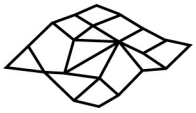


profesionales de operaciones de tecnologías de la información.

GRÁFICO 6. DEVOPS

Conseguir unir o acercar áreas tan diferentes, como el área de desarrollo y el área de operación, va a fomentar que el conocimiento y las lecciones aprendidas se transfieran de forma transversal a la organización. Integrar en fases tempranas de desarrollo al personal que va a operar el producto software, va a conseguir que se detecten desviaciones con las necesidades de los clientes, antes de lanzar el producto.

Aplicando la cultura *DevOps* a las empresas de desarrollo de software, se conseguirá que el producto terminado sea de más calidad y que la apreciación final de los usuarios y clientes sea positiva.



CONCLUSIONES

Ante la problemática detectada, no basta con aplicar un único marco de referencia o de buenas prácticas. Si se desea terminar con un mal endémico de la industria desarrolladora de software, es necesario utilizar las bondades que cada *framework* puede aportar:

- *ITIL®* aporta un cambio de mentalidad y la cultura de que los servicios de TI (en este caso los productos software) son mecanismos para generar valor. Es necesario conocer lo que demanda el negocio para poder ofrecer lo que la organización espera.
- *ISO/IEC 29110* aporta un marco de trabajo para las pequeñas empresas de software, mismo que se adapta a las necesidades de estas empresas y permite desarrollar sus productos software, cumpliendo los estándares de calidad, coste y plazos demandados por los clientes.
- *DevOps* aporta una cultura de colaboración entre diferentes áreas, permitiendo crear sinergias que potenciarán las bondades de la organización y permitirán crear productos software de mayor calidad y más robustos.

Por tanto, se puede concluir que la aplicación de estos tres *frameworks* es una forma de disminuir los proyectos de desarrollo software no exitosos, al atacar la raíz de los factores críticos de fracaso identificados en el gráfico 1.

NOVIEMBRE 2015