

COMMON MISTAKES IN SOFTWARE DEVELOPMENT PROJECT MANAGEMENT

ERRORES COMUNES EN LA GESTIÓN DE PROYECTOS DE DESARROLLO DE SOFTWARE

Recibido: 14 de febrero 2013- aceptado: 29 de marzo 2013

Diana Heredia Vizcaíno.¹

Universidad Simón Bolívar.

Keywords:

Project Software
Development, Project
Management, Risk
Management.

Abstract

In all areas of human endeavor today information systems, whose implementation requires managing software development projects are used. However, many of these projects result in major headaches for organizations, is more costly in time and money than expected. This article is a personal reflection on factors that affect the failure or difficulties faced by such projects.

Palabras clave:

Proyecto de desarrollo de
software, Gestión de
proyectos, Gestión de
riesgos.

Resumen

En todos los ámbitos del quehacer humano hoy en día se utilizan sistemas de información, cuya implementación requiere de la gestión de proyectos de desarrollo de software. Sin embargo, muchos de estos proyectos resultan en grandes dolores de cabeza para las organizaciones, pues resultan más costosos en tiempo y dinero de lo que se esperaba. Este artículo es una reflexión personal sobre los factores que inciden en el fracaso o las dificultades que afrontan este tipo de proyectos.

1. Profesor de tiempo completo. Universidad Simón Bolívar. Barranquilla, Colombia. Email: dianahv@unisimonbolivar.edu.co

* diseño y desarrollo del sistema de información de historias clínicas para la unidad de salud de la Universidad Simón Bolívar

I. INTRODUCCIÓN

Conceptos de Proyectos y Gestión de Proyectos

Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. La naturaleza temporal de los proyectos indica un principio y un final definidos. El final se alcanza cuando se logran los objetivos del proyecto o cuando se termina el proyecto porque sus objetivos no se cumplirán o no pueden ser cumplidos, o cuando ya no existe la necesidad que dio origen al proyecto. (Project Management Institute, 2008)

La dirección de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con los requisitos del mismo. Se logra mediante la aplicación e integración adecuadas de los 42 procesos de la dirección de proyectos, agrupados lógicamente, que conforman los 5 grupos de procesos. Estos 5 grupos de procesos son: Iniciación, Planificación, Ejecución, Seguimiento y Control, y Cierre. Dirigir un proyecto por lo general implica: identificar requisitos, abordar las diversas necesidades, inquietudes y expectativas de los interesados según se planifica y efectúa el proyecto, equilibrar las restricciones contrapuestas del proyecto que se relacionan, entre otros aspectos, con el alcance, la calidad, el cronograma, el presupuesto, los recursos y el riesgo. (Project Management Institute, 2008).

Un Proyecto de Desarrollo de Software (Software Development Project - SDP) es una tarea compleja que emprende un equipo de trabajo dentro de unos límites de tiempo, presupuesto y personal que producen nuevo o mejorado código de computador que agrega valor significativo a un nuevo o existente proceso de negocio (Wysocki, 2006). Los proyectos de desarrollo de software son únicos en el sentido de que requieren la participación intensiva de tres grupos de interesados (stakeholders): El equipo de sistemas de información, los usuarios finales y la dirección o gerencia. Este tipo de proyectos constan de actividades organizadas y ejecutadas en equipos y, por tanto, sujetas a todas las caprichos de la dinámica de grupos, interacciones, coordinación y comunicación. Las habilidades de comunicación y coordinación se convierten en un aspecto extremadamente importante para hacer que un grupo de personas con diversos perfiles trabaje exitosamente (Ewusi-Mensah, Critical Issues in Abandoned Information Systems Development Projects, 1997).

La Gestión de Proyectos de Desarrollo de Software (Software Development Project Management - SDPM) es la disciplina que evalúa las características del software que se va a desarrollar, escoge el ciclo de vida de desarrollo que mejor se ajusta y escoge el enfoque de gestión de proyectos más apropiado para asegurar al cliente la entrega de valor agregado a su negocio de la forma más efectiva y eficiente posible (Wysocki, 2006).

La gestión de proyectos de software tiene al menos cinco componentes:

- Las características del proyecto de software en sí
- El ciclo de vida del desarrollo de software
- El ciclo de vida del proyecto
- El perfil del equipo del proyecto
- La tecnología que soporta lo anterior.

La Estrategia para la Gestión de Proyectos de Desarrollo de Software (SDPM Strategy) es una integración de un ciclo de vida de desarrollo de software y un ciclo de vida de gestión de proyectos de este tipo, dentro de un enfoque orientado al cliente que produzca el máximo valor al negocio independientemente de los obstáculos que puedan presentarse (Wysocki, 2006).

Cuando se enfrenta un proyecto de desarrollo de software, surgen tres preguntas:

- ¿Cuáles son las características del Software que se desarrollará?
- ¿Qué enfoque de Desarrollo de Software es el adecuado para construirlo?
- ¿Qué enfoque de Gestión de Proyectos es el adecuado para dirigir el proceso de desarrollo escogido?

¿Por qué fracasan los Proyectos de Desarrollo de Software?

Los proyectos de desarrollo de software son realizados por miles de organizaciones cada año en todo el mundo. Algunos de ellos requieren de grandes esfuerzos y recursos, implican complejos procesos de desarrollo inter-organizacionales y tienen requerimientos de tecnología sofisticada; otros proyectos de este tipo, por el contrario, son pequeños y sencillos y afectan sólo a un pequeño grupo de usuarios o un área de la organización. Los proyectos de software pueden ser desarrollados por personal de la empresa (Desarrollo "in-house") o pueden ser encomendados a terceros especializados en estas actividades (Desarrollo "outsourced") (Warkentin, Moore, Bekkering, & Johnston, 2009).

Stepanek (Stepanek, 2005) establece varias características que hacen del desarrollo de software un

proceso único: El software es complejo, es el producto más abstracto que puede ser creado en un proyecto, sus requerimientos difícilmente están completos, las tecnologías relacionadas cambian rápidamente y muchas de ellas no están maduras, el desarrollo del software debe producir el mejor software posible según las necesidades, el producto obtenido sufre cambios frecuentes. Ewusi-Mensah (Ewusi-Mensah, *Software Development Failures: Anatomy of Abandoned Projects*, 2003) expresa que el desarrollo de software, como búsqueda intelectual creativa, es más propenso a errores en la definición del problema, el diseño y la implementación de la solución que otro tipo de emprendimientos. Además, por ser una labor de equipo, requiere de colaboración y comunicación entre sus miembros y esto se constituye en otro factor de complejidad para el desarrollo.

Según una encuesta global realizada a ejecutivos de Tecnologías de Información por Standish Group International en 2004, independientemente del tamaño y alcance de un proyecto de desarrollo de software, sólo el 29% de ellos puede considerarse que son exitosos (Entregados a tiempo, dentro del presupuesto y con entregables bien definidos); el 53% experimentaron algún tipo de problemas durante su ejecución; el 18% restante fueron cancelados antes de su terminación o nunca fueron implementados (Warkentin, Moore, Bekkering, & Johnston, 2009). El reporte de la misma empresa, realizado en 2009, muestra que el 24% de los proyectos fracasaron (Domínguez, 2009). Según este mismo estudio, los proyectos de desarrollo de software fallan debido mayormente a factores de gestión (65%) que a factores de índole técnica (35%) (McManus, 2004). Entre los aspectos gerenciales o de gestión que pueden llevar al fracaso de los proyectos están:

- Estructura del proyecto inapropiada.
- Canales de comunicación no apropiados o insuficientes.
- Pobre estimación.
- Métodos de planeación inapropiados o insuficientes.
- Carencia o debilidad en la gestión de riesgos.

Y entre los aspectos técnicos:

- Requerimientos de software incompletos.
- Diseño técnico débil o inapropiado.
- Herramientas de desarrollo y pruebas no suficientes o inapropiadas.
- Carencia de documentación técnica.
- Insuficiente soporte técnico.

- Carencia de control de calidad.

Otros investigadores, como Ewusi-Mensah (Ewusi-Mensah, *Critical Issues in Abandoned Information Systems Development Projects*, 1997), han encontrado que los fracasos en los proyectos de desarrollo de software no son casos aislados y ocurren en compañías de cualquier tamaño y sector de la industria, y estos problemas ocurren a pesar de los avances significativos en las metodologías y herramientas de desarrollo de software que han surgido en los últimos años. Este autor se ha enfocado en el estudio del por qué los proyectos de desarrollo de software son abandonados o cancelados; sus resultados indican que este fracaso se debe a una mezcla de varios factores, tales como:

- No se ha acordado un conjunto bien articulado de metas y objetivos del proyecto.
- El equipo de trabajo del proyecto es débil o problemático.
- No hay una adecuada dirección del proyecto, se carece de un sistema para medir el progreso del proyecto e identificar y mitigar riesgos.
- El nivel de experiencia y pericia, junto con la aplicación de aspectos relevantes de conocimiento del dominio por parte del equipo de trabajo no son suficientes.
- La infraestructura de tecnología de información actual en la organización no fue cuidadosamente evaluados antes de emprender el proyecto.
- No hay participación activa de la alta gerencia corporativa en la evaluación de avance del proyecto y las decisiones críticas son delegadas a los expertos técnicos.
- Dilatación del cronograma y del presupuesto durante la ejecución del proyecto, lo cual es síntoma de problemas subyacentes más serios.

Es recurrente en varios estudios incluir la carencia o debilidad en la identificación y mitigación de riesgos asociados a los proyectos de este tipo. Un riesgo es un evento o condición incierta que, si sucede, tiene un efecto en por lo menos uno de los objetivos del proyecto (Project Management Institute, 2008). La mayoría de organizaciones de desarrollo de software perciben el riesgo en diferentes y no sistemáticas formas, lo cual contribuye a incrementar la inestabilidad e ineffectividad de lo proyectos de este tipo (Sarigiannidis&Chatzoglou, 2011). Wallace y Keil (Wallace & Keil, 2004) citan un estudio realizado entre directores de proyectos de software, en el cual se agrupan los factores de riesgo en cuatro categorías:

- Factores asociados con los usuarios. Incluyen la falta de compromiso de la alta gerencia de la organización y la insuficiente participación de los usuarios. Estos factores no siempre están al alcance del director del proyecto.
- Factores de riesgo asociados con la incapacidad de la dirección del proyecto para juzgar el alcance del sistema y con la pobre identificación de la funcionalidad requerida. Estos factores si deben ser controlados por la dirección del proyecto.
- Factores asociados a la ejecución del proyecto, tales como personal inadecuado, metodología de desarrollo inapropiada, fallas en la definición de roles y responsabilidades, pobre planeación y control del proyecto. Estos factores generalmente si son considerados y controlados por la dirección del proyecto.
- Factores de riesgo asociados al entorno, que se enfoca principalmente en los cambios en la gerencia organizacional y que podrían afectar al proyecto.

Según estos mismos autores, los factores de las categorías 2 y 3 son los que afectan directamente que los proyectos sean culminados en el tiempo y con el presupuesto previsto. De esta forma, los proyectos pobremente ejecutados o con un alcance inestable y requerimientos vagos normalmente no son culminados exitosamente. Los factores de las dos categorías restantes no siempre afectan el éxito en los procesos de un proyecto.

Los factores de las categorías 1, 2 y 3 tienen una importancia significativa en el producto resultante del proyecto. Si los riesgos de ejecución (Categoría 3) son minimizados, el efecto de los factores de los grupos 1 y 2 no será significativo. Alta probabilidad de ocurrencia en los factores de estas tres categorías, aumentan significativamente la complejidad y dificultad del proyecto. Los factores de riesgo asociados al entorno se presentan con menos frecuencia, pero sus efectos son impredecibles. La Tabla 1 muestra en detalle los factores de riesgo que hacen parte de cada categoría.

La Gestión del Riesgo no ha recibido la suficiente atención por parte de la comunidad de Ingeniería del Software y generalmente se deja esta labor a la iniciativa e intuición de las personas, donde los riesgos sólo se identifican cuando aparecen los primeros síntomas de sus efectos y las soluciones se aplican de manera informal y no se documentan (Sarigiannidis&Chatzoglou, 2011). A primera vista, la Gestión del Riesgo puede parecer que agrega complejidad a un proyecto de por si complejo. Sin embargo, en la realidad es todo lo contrario. Por ejemplo, La identificación y priorización de riesgos permite a la dirección y al personal del proyecto enfocarse en los aspectos que tienen mayor impacto en él; las acciones de mitigación de riesgos reducen el riesgo global y pueden ayudar a acelerar la culminación del proyecto y sus costos (McManus, 2004).

Tabla 1. Factores de Riesgo por Categoría (Adaptado de Wallace y Keil) (Wallace &Keil, 2004)

CATEGORÍA	FACTOR DE RIESGO
1	Participación del usuario e involucrados faltante o insuficiente
	Resistencia al cambio
	Conflictos entre usuarios
	Falta de soporte de la alta gerencia corporativa
	Falta de acuerdos organizacionales sobre el proyecto
2	Criterios de éxito del proyecto no definidos
	Requerimientos del sistema no están claramente definidos, son confusos o erróneos
	Continuos cambios en los requerimientos y/o en el alcance del proyecto
	Usuarios no comprenden las capacidades del sistema y sus limitaciones
3	Dificultad para definir las entradas y salidas del sistema
	Miembros del equipo no han sido entrenados adecuada y suficientemente
	Falta de acuerdo sobre el proyecto entre los miembros del equipo
	Personal del proyecto sin experiencia
	Conflictos frecuentes entre los integrantes del equipo
Frecuente rotación del personal	
Personal de desarrollo poco familiarizado con las herramientas	

	Carencia o insuficiencia de habilidades especiales requeridas por el proyecto
	Proyecto involucra el uso de nuevas o inmaduras tecnologías
	Alta complejidad técnica en el proyecto
	Falta de una metodología de dirección de proyectos
	Estimación del cronograma y agenda del proyecto inadecuada
	Falta de habilidades en el liderazgo de proyectos
	Monitoreo insuficiente al progreso del proyecto
	Estimación de recursos requeridos es inadecuada
	Pobre planeación del proyecto
	Hitos del proyecto no están claramente definidos
	Estimación del presupuesto del proyecto es inadecuada
	Dirección del proyecto no efectiva
	Director del proyecto sin experiencia
	Comunicaciones inefectivas
4	Cambio en la dirección organizacional durante el proyecto
	Reestructuración organizacional durante el proyecto
	Recursos dados al proyecto llevan a cambios en las prioridades organizacionales
	Políticas corporativas tienen efectos negativos en el proyecto
	Entorno organizacional inestable
	Dependencia de proveedores externos

También es frecuente encontrar en la literatura disponible sobre el tema de gestión de proyectos, que muchos de sus problemas se deben al deficiente liderazgo del mismo. Un punto de vista interesante lo presenta Hass (Hass, 2009), quien plantea que tradicionalmente los proyectos se centran más en la gerencia que en el liderazgo; un gerente se ocupa de planear, organizar, establecer el equipo de trabajo y los presupuestos y monitorear y controlar la ejecución del proyecto, mientras que los proyectos complejos requieren más bien del trabajo coordinado y colaborativo de diversos grupos y de liderazgo más que control.

Errores comunes en Gestión de Proyectos de Desarrollo de Software

A pesar de que las metodologías de gestión de proyectos contemplan herramientas para facilitar la exitosa culminación de proyectos de desarrollo de software, aún persisten los fracasos o problemas de sobrecostos y tiempos excedidos. Los errores de la gestión son tan comunes que sus consecuencias son predecibles y, aun así, se siguen cometiendo. Varios autores han realizado recopilaciones de estos errores, basados en su propia

experiencia y en apreciaciones algo empíricas; Varas, M (Varas, 2000) hace la siguiente clasificación:

1. Relacionados con las Personas:
 - Motivación débil.
 - Personal con poca capacidad individual o suficientes competencias para el proyecto.
 - Empleados problemáticos, sin control.
 - Desarrolladores que realizan “hazañas” (trabajos adicionales o esfuerzos mayores en ciertos momentos para des-atrasar o resolver problemas en la ejecución del proyecto), en lugar de realizar un trabajo constante y disciplinado.
 - Añadir más personal a un proyecto retrasado.
 - Lugares de trabajo con mucha gente y ruidosas.
 - Fricciones entre los clientes y los desarrolladores del proyecto.
 - Expectativas poco realistas.
 - Falta de participación de los implicados.
 - Falta de un promotor efectivo del proyecto.
2. Relacionados con el Proceso:
 - Planificación excesivamente optimista.
 - Planificación insuficiente.
 - Abandono de la planificación bajo presión.
 - Gestión del riesgo insuficiente.

- Escatimar en las actividades iniciales del proyecto.
 - Diseño inadecuado.
 - Escatimar en el control de la calidad.
 - Control insuficiente de las directivas.
Relacionados con el Producto:
 - Exceso de requerimientos.
 - Cambio en los requerimientos.
 - Desarrollo orientado a la investigación (Necesidad de crear/usar nuevas tecnologías).
3. Relacionados con la Tecnología:
- Exceso de requerimientos.
 - Cambio en los requerimientos.
 - Desarrollo orientado a la investigación (Necesidad de crear/usar nuevas tecnologías).
4. Relacionados con la Tecnología:
- Confiar demasiado en tecnologías no usadas previamente.
 - Sobreestimar de las ventajas del uso de nuevas herramientas o tecnologías.
 - Cambio de herramientas durante la ejecución del proyecto.

Reflexión

Después de revisar la literatura disponible sobre fracaso en los proyectos de desarrollo de software, así como de la propia experiencia, se puede observar que los errores en que se incurre cuando se emprenden esfuerzos de este tipo son:

- Muchas veces el cliente del sistema a desarrollar impone el tiempo de ejecución y presiona para que la planeación se realice a partir de éste. Este tiempo, en la mayoría de casos, se fija aun antes de tener claros los requerimientos y el alcance del proyecto.
- Se considera que las actividades de preparación del proyecto no aportan y consumen tiempo y recursos, por lo cual se omiten.
- Los requerimientos no se definen claramente antes de comenzar el diseño; éstos cambian frecuentemente, aun durante el desarrollo.
- La arquitectura tecnológica (hardware, comunicaciones, herramientas de desarrollo) se define sobre la marcha; esto conlleva una estimación insuficiente de este tipo de recursos.
- Cuando se quiere llevar a cabo un proyecto con algún componente innovador, se tiende a

experimentar con tecnologías o metodologías nuevas y poco probadas. En estos casos, el tiempo de aprendizaje del personal es mayor y no se cuenta con soporte suficiente para su implementación.

- Generalmente, se contrata personal de desarrollo con poca o ninguna experiencia, y con deficientes condiciones contractuales y salariales.
- Cuando se detectan atrasos en los cronogramas, la solución más fácil es agregar personal de desarrollo, el cual deberá pasar por la etapa de aprendizaje antes de ser realmente productivo para el proyecto.
- Es una práctica común no considerar un sitio de trabajo adecuado, amplio, cómodo y con las facilidades técnicas suficientes, para el desarrollo del proyecto.
- No se adoptan sistemas de medición de progreso suficientes; el principal indicador usado en muchos casos es la ejecución del presupuesto.
- Se da mayor importancia a la etapa de codificación, subestimando otras etapas importantes como el diseño y las pruebas, y por tanto, no se planifican adecuadamente.
- El director del proyecto debe tener capacidades tanto técnicas como administrativas, así como experiencia suficiente en proyectos similares al que pretende dirigir. En muchos casos, el director de proyecto es un especialista técnico.
- Las empresas cuyo núcleo de negocio no es el desarrollo de software, emprenden este tipo de proyectos sin tener experiencia en ello.

II. CONCLUSIONES

Como se puede observar, la mayor parte de errores comunes cometidos en la gestión de este tipo de proyectos son considerados por varios autores como factores de riesgo y por lo tanto son susceptibles de ser gestionados para evitar su ocurrencia o reducir su impacto negativo. Generalmente, los riesgos en los proyectos de desarrollo de software son más de índole gerencial que técnicos. Se puede concluir que, en general, los directores de proyecto desconocen las metodologías de gestión o por presiones del cliente o por intentar reducir costos y tiempo de ejecución, no las aplican. Tampoco se aprende de la experiencia y los emprendimientos de desarrollo de software seguirán condenados a los mismos problemas o a su fracaso.

Trabajos citados

- [1] Project Management Institute, Guía de los Fundamentos para la Dirección de Proyectos (Guía del PMBOK). Pennsylvania, 2008.

- [2] R. Wysocki, Effective Software Project Management.: John Wiley & sons, 2006.
- [3] K. Ewusi-Mensah, "Critical Issues in Abandoned Information Systems Development Projects," Communications of the ACM, vol. 40, no. 9, pp. 74-80, 1997.
- [4] M. Warkentin, R. Moore, E. Bekkering, and A. Johnston, "Analysis of Systems Development Project Risks: An Integrative Framework," The DATA BASE for Advances in Information Systems, pp. 8-27, 2009.
- [5] G. Stepanek, Software Project Secrets. Why software projects fail. USA: Apress, 2005.
- [6] K. Ewusi-Mensah, Software Development Failures: Anatomy of Abandoned Projects. USA: Massachusetts Institute of Technology, 2003.
- [7] J. Dominguez. The Curious Case of the CHAOS Report 2009. [Online]. HYPERLINK "<http://www.projectsmart.co.uk/the-curious-case-of-the-chaos-report-2009.html>"<http://www.projectsmart.co.uk/the-curious-case-of-the-chaos-report-2009.html>. Julio, 2009.
- [8] J. McManus, Risk Management in Software Development Projects.: Taylor and Francis, 2004.
- [9] L. Sarigiannidis and P. Chatzoglou, "Software Development Project Risk Management: A New Conceptual Framework," Journal of Software Engineering and Applications, vol. 4, pp. 293-305, Mayo 2011.
- [10] L. Wallace and M. Keil, "Software Projects Risks and Their Effects on Outcomes," Communications on ACM, vol. 47, no. 4, pp. 68-73, Abril 2004.
- [11] K. Hass, Managing Complex Projects: A New Model. USA: Management Concepts, Inc., 2009.
- [12] M. Varas, Gestión de Proyectos de Desarrollo de Software, Departamento de Ingeniería Informática y Ciencias de la Computación - Universidad de Concepción. 2000.