

DIRECCIÓN DE PROYECTOS DE SOFTWARE  
DESDE LA METODOLOGÍA PMBOK®

DIEGO ALEJANDRO RIOS HERRERA

UNIVERSIDAD TECNOLÓGICA DE PEREIRA  
FACULTAD DE INGENIERÍAS:  
PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN  
PEREIRA - RISARALDA  
MAYO DE 2016

DIRECCIÓN DE PROYECTOS DE SOFTWARE  
DESDE LA METODOLOGÍA PMBOK®

DIEGO ALEJANDRO RIOS HERRERA

DIRECTOR: JUAN DE JESÚS VELOZA MORA

UNIVERSIDAD TECNOLÓGICA DE PEREIRA  
FACULTAD DE INGENIERÍAS:  
PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN  
PEREIRA - RISARALDA  
MAYO DE 2016

## CONTENIDO

1. INTRODUCCIÓN
2. PLANTEAMIENTO DEL PROBLEMA
3. OBJETIVO GENERAL
4. OBJETIVOS ESPECÍFICOS
5. METODOLOGÍA
6. CAPÍTULO I: Metodología PMBOK® enfocada a la Dirección de Proyectos de Software
  - 6.1. ¿Qué es un Proyecto?
    - 6.1.1. Relaciones entre Portafolio, Programas y Proyectos
  - 6.2. Dirección de Proyectos
    - 6.2.1. Porque la Dirección de Proyectos de Software es un Reto
  - 6.3. Relación entre Dirección de Portafolios, Dirección de Programas, Dirección de Proyectos y Dirección Organizacional de Proyectos
    - 6.3.1. Dirección de Portafolio
    - 6.3.2. Dirección de Programa
    - 6.3.3. Proyectos y Planificación Estratégica
    - 6.3.4. Oficina de Dirección de Proyectos
  - 6.4. Valor del Negocio
  - 6.5. Rol del Director del Proyecto
    - 6.5.1. Habilidades Interpersonales de un Director de Proyectos
  - 6.6. Gestión de la Calidad
7. CAPÍTULO II: Influencia de la Organización
  - 7.1. Influencia de la Organización en la Dirección de Proyectos
    - 7.1.1. Culturas y Estilos de la Organización
    - 7.1.2. Comunicaciones en la Organización
    - 7.1.3. Activos de los Procesos de la Organización
      - 7.1.3.1. Procesos y Procedimientos
    - 7.1.4. Factores Ambientales de la Empresa

- 7.2. Interesados y Gobierno del Proyecto
  - 7.2.1. Interesados del Proyecto
  - 7.2.2. Gobernabilidad del Proyecto
- 7.3. Equipo del Proyecto
  - 7.3.1. Composición del Equipo del Proyecto
  - 7.3.2. Equipo Colaborativo
- 8. CAPÍTULO III: Ciclo de Vida del Proyecto
  - 8.1. Características del Ciclo de Vida del Proyecto
    - 8.1.1. Fases del Proyecto
      - 8.1.1.1. Relaciones entre Fases
      - 8.1.1.2. Ciclos de Vida Predictivos
- 9. CAPÍTULO IV: Proyectos de Software desde las distintas Áreas del Conocimiento
  - 9.1. Gestión del Alcance
  - 9.2. Gestión del Tiempo
  - 9.3. Gestión del Costos
  - 9.4. Gestión de la Calidad
  - 9.5. Gestión de los Interesados
  - 9.6. Gestión del Riesgo

## INTRODUCCIÓN

La dirección de proyectos tiene como objetivo encaminar esfuerzos, técnicas y liderazgo a desarrollar con éxito ciertas actividades para lograr concluir la entrega de un servicio o un producto, según sea la naturaleza del proyecto. La aceptación del objetivo de la dirección de proyectos como profesión indica que la aplicación de conocimientos, procesos, habilidades, herramientas y técnicas puede tener un impacto considerable en el éxito de un proyecto. Desde la metodología implementada por el PMI (Project Management Institute) en su guía PMBOK® identifica ese subconjunto de fundamentos para la dirección de proyectos generalmente reconocidos como buenas prácticas. Entendiéndose como “buenas prácticas” el estar de acuerdo, en general, en que la aplicación de conocimientos, habilidades, herramientas y técnicas puede aumentar las posibilidades de éxito de una amplia variedad de proyectos.

El auge de la tecnología, y el ver involucrado el sistematizado de la gran mayoría de procesos en las organizaciones, se hace necesario extender la dirección de proyectos a la dirección de proyectos de software, de mano con la ingeniería de software.

## **PLANTEAMIENTO DEL PROBLEMA**

La importancia de la computación actualmente es innegable para el desarrollo de las actividades diarias de cualquier empresa a nivel mundial. El procesamiento de datos, el manejo de grandes cantidades de información y el auge de la tecnología soportada en software y su capacidad es palpable. Por esta razón se hace necesario Gestión y/o Dirección de Proyectos de Desarrollo de software, encaminado a la necesidad de incrementar su eficiencia, mejorar su expectativa de rendimiento y calidad.

## **OBJETIVO GENERAL**

Desarrollar una guía que sirva como referencia para aquellas personas u organizaciones que tienen como finalidad la construcción de nuevos productos de software, observado desde la metodología PMBOK® y su adaptación a proyectos software en lo referente al enfoque de proyectos desde ésta metodología y el ciclo de vida del software, incluido cómo influye la organización en estos proyectos.

## **OBJETIVOS ESPECÍFICOS**

- Investigar sobre la metodología PMBOK® relacionándola con la ingeniería del software en cuanto a las necesidades que tienen los proyectos que involucran la creación de nuevos productos de software.
- Observar la influencia de la organización en estos proyectos desde esta metodología PMBOK®.
- Observar lo que hace esta metodología aplicada al ciclo de vida.
- Plantear cómo se pueden atender estos proyectos de software desde las distintas áreas de conocimiento.

## **METODOLOGÍA**

- Desarrollar los criterios usados por la metodología PMBOK®.
- Desarrollar las necesidades que tiene los proyectos que involucran la creación de nuevos productos desde la perspectiva de la Ingeniería de Software.
- Plantear como pueden ser atendidos estos proyectos usando la metodología PMBOK® desde las diferentes áreas del conocimiento.



# CAPÍTULO I

## ENFOQUE METODOLÓGICO

El enfoque de la metodología PMBOK® es la dirección de proyectos, vista como una profesión. La aplicación de conocimientos en esta profesión, en los procesos, las habilidades y técnicas pueden tener un impacto considerable en el éxito de un proyecto. La Guía del PMBOK® identifica ese subconjunto de fundamentos para la dirección de proyectos generalmente reconocido como buenas prácticas. “Generalmente reconocido” significa que los conocimientos y prácticas descritos son aplicables a la mayoría de los proyectos, la mayoría de las veces, y que existe consenso sobre su valor y utilidad. “Buenas prácticas” significa que se está de acuerdo, en general, en que la aplicación de conocimientos, habilidades, herramientas y técnicas puede aumentar las posibilidades de éxito de una amplia variedad de proyectos. "Buenas prácticas" no significa que el conocimiento descrito deba aplicarse siempre de la misma manera en todos los proyectos; la organización y/o el equipo de dirección del proyecto son los responsables de establecer lo que es apropiado para cada proyecto concreto.

El estándar constituye una guía, más que una metodología específica para todos los directores de proyecto, directores de programa, directores de portafolios y otros interesados. El estándar también refiere sobre el código de ética y conducta del profesional y describe las expectativas que deberían tener respecto a sí mismos y a los demás. El código está claro en la relación con las obligaciones básicas de responsabilidad, respeto, equidad y honestidad, esto conlleva a la obligación de cumplir con leyes, regulaciones, y políticas profesionales y de la organización. En el trato con los interesados, los profesionales deben comprometerse a realizar prácticas honestas, responsables y justas, así como a mantener relaciones respetuosas.

Las pautas dadas en la metodología PMBOK® se considera un estándar y dado que los profesionales provienen de culturas y orígenes diversos, es aplicable a nivel mundial.

Los criterios referidos en la metodología PMBOK® incluyen algunas definiciones que permiten establecer un lenguaje formal a la hora de referirnos al estándar. Durante esta primera parte de la descripción de los criterios y de la misma metodología se desglosan las definiciones pertinentes que haya lugar y que estén incluidas dentro de la guía.

## **¿Qué es un Proyecto?**

Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. Los proyectos de software, como todos los proyectos, se llevan a cabo para lograr un objetivo específico. Además de la creación de nuevos productos, los proyectos de software se llevan a cabo a menudo para modificar un producto de software existente, la integración de un conjunto de componentes de software existentes, para ampliar las posibilidades de los productos de software, o para modificar la infraestructura de software de una organización.

Los proyectos de software también pueden llevarse a cabo para satisfacer las solicitudes de servicios, necesidad de mantenimiento, o para proporcionar soportes de operaciones. Estas actividades pueden ocurrir como las actividades a nivel de esfuerzo (LOE, level-of-effort). Se consideran proyectos cuando se especifican como temporales y posibles para presentar sus prestaciones y resultados. El ciclo de vida de los productos de software, en contraste a los ciclos de vida de los proyectos, por lo general implica actividades de mantenimiento y de soporte que incluyen tanto los proyectos como las actividades a nivel de esfuerzo. Los proyectos de TI, tales como el diseño de un sistema de información empresarial, transición del servicio TI a otro proveedor, o el despliegue de una solución a los usuarios finales no son proyectos de software en el sentido tradicional, pero muchos de los conceptos y prácticas descritas en el estándar puede resultar útil en las organizaciones de TI.

## **Relaciones entre Portafolio, Programas y Proyectos**

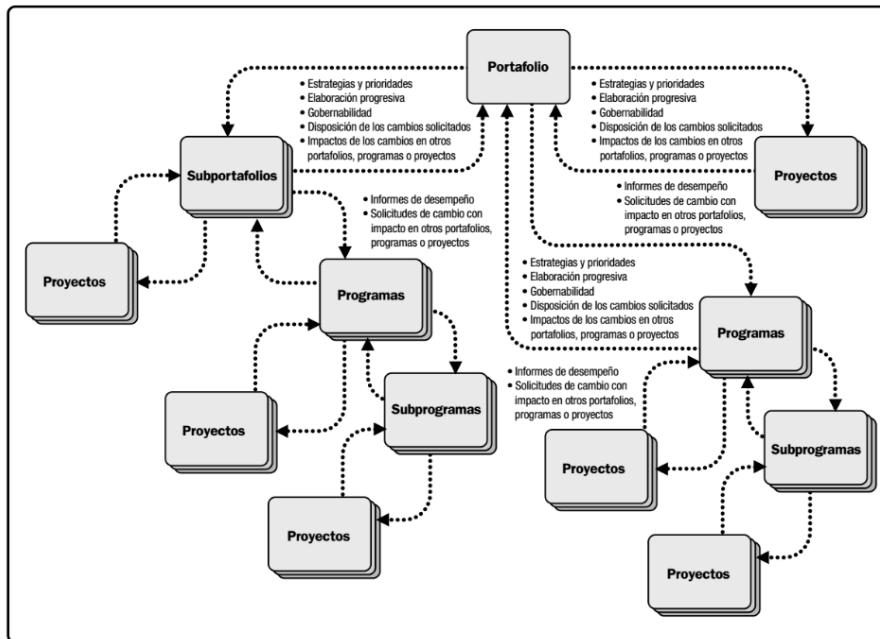
El **Software Extension to the PMBOK®**, para la especificación de muchos de sus términos y de su metodología se basa en la Guía PMBOK®, desde la cual se extrae la definición de portafolio, programa y proyecto y sus relaciones.

La relación entre portafolios, programas y proyectos es tal que un portafolio se refiere a un conjunto de proyectos, programas, subconjuntos de portafolios y operaciones que se gestionan como un grupo para alcanzar determinados objetivos estratégicos. Los programas se agrupan en un portafolio y comprenden subprogramas, proyectos o cualesquiera otros trabajos que se gestionan de manera coordinada para contribuir al portafolio. Los proyectos individuales, estén o no incluidos en el ámbito de un programa, siempre se consideran parte de un portafolio. Aunque los proyectos o programas del portafolio no son necesariamente interdependientes ni están necesariamente relacionados de manera directa, están vinculados al plan estratégico de la organización mediante el portafolio de la misma.

Las estrategias y prioridades de una organización se vinculan, y se establecen relaciones entre portafolios y programas, y entre programas y proyectos individuales. La planificación de la organización ejerce un impacto en los proyectos a través del establecimiento de prioridades entre los mismos teniendo en cuenta los riesgos, el financiamiento y otras consideraciones relativas al plan estratégico de la organización.

La planificación de la organización puede guiar la gestión de los recursos y el apoyo a los proyectos que componen el portafolio basándose en categorías de riesgo, líneas de negocio específicas o tipos de proyectos generales, como infraestructura y mejora de los procesos.

El siguiente gráfico ilustra la vinculación y el establecimiento de las relaciones entre portafolio, programa y proyectos.



## Interacciones entre la Dirección de Proyectos, la Dirección de Programas y la Dirección de Portafolios

### Dirección de Proyectos

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 47 procesos de la dirección de proyectos, agrupados de manera lógica, categorizados en cinco Grupos de Procesos. Estos cinco Grupos de Procesos son:

- Inicio,
- Planificación,
- Ejecución,
- Monitoreo y Control, y
- Cierre.

La naturaleza única de software, permite que los 47 procesos dentro de los 5 grupos de procesos de la Guía PMBOK® se solapen, intercalan, y se repitan en varias formas, lo que

se traduce en modificaciones y extensión a los métodos, herramientas y técnicas de la Guía PMBOK® que se utiliza para gestionar los proyectos de software.

Equilibrar las restricciones contrapuestas del proyecto incluyen, entre otras:

- El alcance,
- La calidad,
- El cronograma,
- El presupuesto,
- Los recursos y
- Los riesgos.

Las características específicas del proyecto y las circunstancias pueden influir sobre las restricciones en las que el equipo de dirección del proyecto necesita concentrarse.

Los factores tecnológicos que pueden poner limitaciones en proyectos de software y productos de software incluyen:

- Estado de hardware y tecnología de software
- Plataforma del hardware, plataforma del software, sistemas operativos, y protocolos de comunicación,
- Integridad de la arquitectura de TI, limitaciones, y protocolos,
- Herramientas de desarrollo de software,
- Arquitectura de Software,
- Requisitos de compatibilidad de backward y forward,
- Reutilización de componentes de una librería de software,
- Uso de código abierto versus componentes de software de código cerrado,
- Uso de componentes de software suministrados por el cliente,
- Las interfaces con el hardware y otro software, y
- Creación y uso de la propiedad intelectual.

Otros factores que pueden poner limitación de los proyectos de software incluyen pero no están limitados a los requisitos para la seguridad del sistema son, el cumplimiento de la seguridad, la fiabilidad, la disponibilidad, la escalabilidad, el rendimiento, la capacidad de prueba, garantía de la información, la localización, el mantenimiento, la compatibilidad, los

reglamentos, las políticas del cliente, la infraestructura de apoyo, la disponibilidad de los miembros del equipo y sus habilidades, el entorno de desarrollo de software y sus métodos, y la madurez de la organización y la capacidad.

### **Porque la Dirección de Proyectos de Software es un Reto**

Cada disciplina tiene aspectos únicos que lo diferencian de otras disciplinas. Dentro de estas disciplinas, los principios generales de la dirección de proyectos están adaptados para tener en cuenta las particularidades de los proyectos en esas disciplinas. Hay muchos factores que hacen que los proyectos de software y la dirección de proyectos de software sean un reto. Algunos de estos factores son:

- Los proyectos de software son un reto ya que el software es un producto intangible y maleable; el código fuente para el software está escrito en texto. En la mayoría de los casos, los equipos de desarrolladores de software generan y revisan los documentos compartidos (por ejemplo., Requisitos, especificaciones de diseño, planes de código y pruebas). El desarrollo de software se caracteriza a menudo como un proceso de aprendizaje en el que se adquiere el conocimiento y la información que se genera durante el proyecto.
- Los atributos clave que hacen que los proyectos de software sean un reto son la complejidad del proyecto y el producto, la escala no lineal de los recursos, la medición de proyectos y productos, la incertidumbre inicial en el proyecto y el alcance del producto, y los conocimientos adquiridos como el proyecto evoluciona.
- Los requisitos del software cambian a menudo durante un proyecto de software, como se obtiene el conocimiento y el alcance del proyecto y el producto emergente.
- Los requisitos para el nuevo software y la modificación de un software a menudo influyen y son influenciados por los procesos de negocio de una organización y los procesos de flujo de trabajo de los empleados.
- El capital intelectual del personal de software es el capital activo principal para los proyectos de software y las organizaciones de desarrollo de software, porque el software es un producto directo de los procesos cognitivos humanos.

- La comunicación y la coordinación dentro de los equipos de software y con las partes interesadas del proyecto a menudo carecen de la claridad. Muchas de las herramientas y técnicas utilizadas en la ingeniería de software están destinados a mejorar la comunicación y coordinación.
- La creación de software requiere una solución innovadora de problemas para crear soluciones únicas. La mayoría de los proyectos de software desarrollan productos únicos porque la replicación de software existente es un proceso sencillo, en comparación con la replicación de artefactos físicos. Los proyectos de software son más afines a los proyectos de investigación y desarrollo de construcción o fabricación de proyectos.
- Los proyectos de software suponen un riesgo e incertidumbre, ya que el producto es intangible y requieren de la innovación, y las partes interesadas no pueden articularse o acordar las necesidades a satisfacer por el producto de software de manera efectiva.
- La planificación inicial y estimación de proyectos de software es un reto debido a que estas actividades dependen de requisitos, que son a menudo imprecisos o parte de los datos históricos que son a menudo ausente o inaplicables. La preparación de estimaciones precisas también es un reto debido a la eficiencia y eficacia de los desarrolladores de software es muy variable.
- La complejidad del producto hace que el desarrollo y la modificación del software sea un reto debido a la enorme cantidad de caminos lógicos dentro de los módulos del programa, combinados con los valores de datos que ejercen estos, y las combinaciones de los detalles de la interfaz entre los módulos del programa.
- Las exhaustivas pruebas de software no serían prácticas a causa del tiempo que sería necesario para poner a prueba todas las rutas lógicas e interfaces bajo todas las combinaciones de datos de entrada y otros estímulos de entrada.
- El desarrollo de software a menudo implica la inclusión de diferentes productos de proveedores y desarrollo de interfaces con otro software; esto puede dar lugar a problemas de integración y de rendimiento.

- Porque la mayoría del software está interconectado, las técnicas de seguridad de la información son necesarias. La seguridad del software es un reto importante y creciente.
- La objetiva cuantificación y la medición de la calidad del software es difícil debido a la naturaleza intangible de software.
- Los desarrolladores de software utilizan procesos, métodos y herramientas que están en constante evolución y se actualizan con frecuencia.
- El software es a menudo el elemento de un sistema que se cambia cuando la funcionalidad, comportamiento o atributos de calidad han de ser cambiado.
- Un producto de software puede ser requerido para operar en una variedad de plataformas de hardware y de software.
- El software ejecutable no es un producto independiente. Se ejecuta en el hardware de computación y es a menudo un elemento de un sistema que consta de un diverso grupo de componentes como el hardware, otro software, y el procedimiento manual.
- Las plataformas tecnológicas, la infraestructura de software, y el software suministrado por el proveedor con frecuencia se cambian o se actualizan, y que puede requerir cambios en el software que se está desarrollando.

Los factores anteriormente descritos son de gran importancia y relevancia para su análisis pues especifican, intentando abarcar gran cantidad, factores que implican el reto de la dirección de proyectos de software. Al igual que muchos productos de trabajo del conocimiento, el software es intangible. No es una entidad física que puede ser evaluado por medidas tradicionales (masa, volumen, conductividad, gravedad específica). Es, sin embargo, limitada por factores tales como el hardware de procesamiento, la memoria disponible y el ancho de banda de comunicación.

La naturaleza intangible de software crea dificultades para medir el estado actual del producto, que a su vez hacen el seguimiento y control de un proyecto de software. Los enfoques tradicionales, tales como la estructura de desglose del trabajo, redes de horario, y la notificación del valor ganado, están diseñados para adaptarse a las necesidades de los



proyectos de software. Estas técnicas tradicionales se aumentan con técnicas tales como el desarrollo iterativo y / o incremental con manifestaciones frecuentes de software parcialmente completado.

La naturaleza maleable de software tiene dos aspectos positivos y negativos para la dirección de proyectos de software. En el lado positivo, la maleabilidad de software hace que sea posible, a veces (pero no siempre) responder rápidamente a las cambiantes necesidades de los usuarios y otros factores ambientales, en comparación con el cambio de los elementos de hardware de la computadora u otros artefactos físicos. En el lado negativo, la interrupción de los trabajos en curso para responder a los cambios solicitados puede abrumar la planificación de tareas y las limitaciones presupuestarias de cómputo.

Cada proyecto de software es una tarea única, porque la replicación del software existente es un proceso sencillo a diferencia de la replicación de los artefactos físicos de proyectos de fabricación y construcción. Cada proyecto de software es, pues, una empresa que produce un producto único. El objetivo de la preparación es para producir repetidamente artefactos que son casi idénticos como sea posible, dadas las limitaciones de la ciencia de los materiales y los aspectos prácticos de la tecnología de fabricación y la aceptación en el mercado; mientras que el objetivo de un proyecto de software es producir una copia perfecta de un producto de software, dentro de las limitaciones de programación, presupuesto, recursos y tecnología de software y hardware.

El software es un producto directo de los procesos cognitivos de las personas que participan en el trabajo en equipo innovador. Si bien es cierto que todos los ingenieros se involucran en el trabajo en equipo, el hecho de que el software es desarrollado y modificado sin las limitaciones de los procesos que intervienen los medios de comunicación o de fabricación físicas hace que el trabajo en el equipo de software sea de naturaleza diferente a otros trabajos en equipo de otras disciplinas de ingeniería. Como resultado, muchos de los procedimientos y técnicas utilizados en la dirección de proyectos de software están diseñados para facilitar la comunicación y la coordinación entre los miembros del equipo que participa, coordinados en estrecha colaboración, en el trabajo intensivo del intelecto.

Una planificación precisa y la estimación de costos y el cronograma, es difícil para todo tipo de proyectos, pero es particularmente difícil para los proyectos de software debido a que: 1) el software es desarrollado y modificado por las actividades de trabajo cognitivas de los desarrolladores, 2) la productividad de los desarrolladores de software individuales varía ampliamente (en cantidad y calidad del trabajo), 3) los requisitos en que se basan las estimaciones son a menudo mal definidos, y 4) la continua evolución de la tecnología pueden hacer que los datos históricos para nuevos proyectos sean inexactos. Por estas razones, los métodos de desarrollo de software actuales se concentran en el desarrollo de un conjunto mayor de incrementos de productos para que las compensaciones entre programación, presupuesto, recursos, funcionalidad y atributos de calidad se puedan ajustar continuamente.

La productividad en proyectos de software incluye tanto la cantidad como la calidad del trabajo. La cantidad de software escrito (número de líneas de código) no es una buena medida de la productividad del programador; un programador que escribe un programa pequeño, eficiente, es más productivo para contribuir a un resultado exitoso que un programador que escribe un programa grande, ineficiente. Del mismo modo, un programador que corre a través de una tarea, pero que comete muchos errores requiriendo de trabajo correctivo, es menos productivo que un programador que avanza más lentamente, pero produce un programa que tiene un menor número de errores. La productividad de los programadores con experiencia y la experiencia similar, medida por la cantidad y calidad del software resultante, se ha demostrado en varias ocasiones.

Con el advenimiento de la interconexión generalizada, la seguridad del software se ha convertido en una consideración importante en la construcción o la modificación de un producto de software. Al igual que otros atributos de calidad, los atributos de seguridad deben ser planificados, diseñados, construidos, verificados y validados. Al igual que otros atributos de calidad, la seguridad del software no puede ser "probado".

Los proyectos de software también son difíciles porque el software es siempre parte de un sistema. El software como una entidad independiente es inútil; para ser de utilidad, el software se ejecuta en el hardware digital. En algunos casos, un proyecto de software es un elemento de un programa de desarrollo que implica el desarrollo de acompañamiento de los componentes o desarrollo de software relacionados con el software para otros proyectos. En otros casos, el desarrollo de software puede estar limitado al desarrollo de software de aplicación para una población de usuarios conocidos. En estos casos, el sistema incluye equipo especificado, el sistema operativo y lenguaje de programación o lenguajes que proporciona las capacidades necesarias.

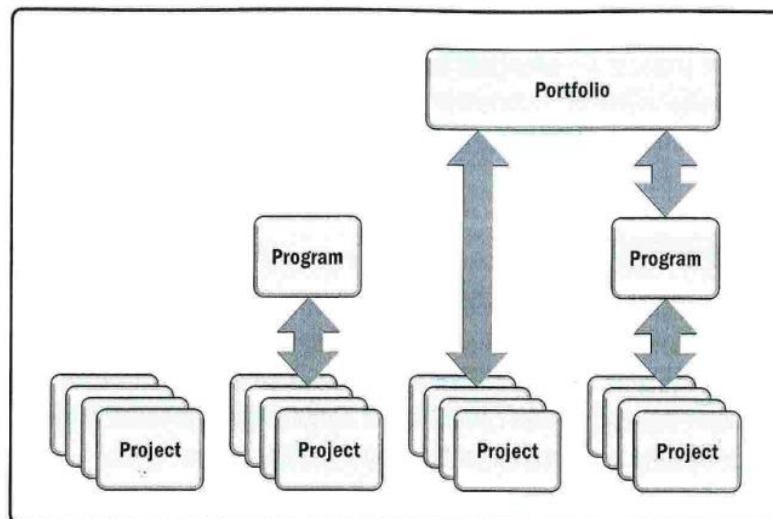
### **Relaciones entre Dirección de Portafolios, Dirección de Programas, Dirección de Proyectos y Dirección Organizacional de Proyectos**

Para entender los conceptos de dirección de portafolios, dirección de programas y dirección de proyectos es importante reconocer las similitudes y las diferencias que existen entre cada una de estas disciplinas. También resulta útil comprender cómo se relacionan con la dirección organizacional de proyectos (OPM por sus siglas en inglés). OPM es un marco para la ejecución de estrategias a través de la dirección de proyectos, la dirección de programas y la dirección de portafolios, conjuntamente con una serie de prácticas organizativas, destinadas a generar de manera consistente y predecible estrategias organizacionales capaces de producir un mejor desempeño, mejores resultados y una ventaja competitiva sostenible.

La dirección de portafolios, la dirección de programas y la dirección de proyectos se alinean o son impulsadas por las estrategias organizacionales. Sin embargo, la dirección de portafolios, la dirección de programas y la dirección de proyectos difieren en la manera en que cada una contribuye al logro de los objetivos estratégicos. La dirección de portafolios se alinea con las estrategias organizacionales mediante la selección de los programas o proyectos adecuados, el establecimiento de prioridades con respecto al trabajo a realizar y la provisión de los recursos necesarios, mientras que la dirección de programas adecua sus proyectos y componentes de programas y controla las interdependencias a fin de lograr los

beneficios estipulados. La dirección de proyectos desarrolla e implementa planes para lograr un alcance determinado, que viene dado por los objetivos del programa o del portafolio al cual está vinculado, y, en último término, por las estrategias organizacionales. OPM potencia la capacidad de la organización para la consecución de los objetivos estratégicos mediante la vinculación de los principios y prácticas de la dirección de proyectos, la dirección de programas y la dirección de portafolios con los facilitadores de la organización (p.ej., prácticas estructurales, culturales, tecnológicas y de recursos humanos). Una organización mide sus capacidades para posteriormente planificar e implementar mejoras encaminadas a la adopción sistemática de las mejores prácticas.

Las relaciones entre los proyectos, programas y portafolios se ilustran a continuación:



Muchos proyectos de software no son parte de los programas y no toda la gestión de las organizaciones en los proyectos de software está en un portafolio base. En estos casos, cada proyecto de software existe como una entidad independiente. Algunos proyectos de software pueden ser parte de los programas y algunos programas pueden ser incluidos en los portafolios.

Las líneas de productos de software son similares a los programas. Una línea de productos consiste en un componente base que apoyan adiciones y extensiones, que dan lugar a

productos específicos dentro de la línea de productos. Por ejemplo, una línea de productos podría consistir en un componente base para la contabilidad financiera a la que se añaden diferentes interfaces de usuario para dar cabida a la lengua y cultura diferente. Una línea base de producto puede evolucionar con el tiempo. Las líneas de productos se incluyen a veces en los portafolios.

Mientras que el software tiene un fuerte impacto en las organizaciones y sus operaciones (tanto de software de infraestructura y software de aplicaciones), hay diferentes maneras en las que el software puede generar un valor, por ejemplo, valor económico, valor social, el bienestar público, y el impacto en los lugares de trabajo y ambientes de esparcimiento. Por lo tanto, el establecimiento de criterios de priorización de los programas y proyectos dentro de un portafolio puede ser un difícil acto de equilibrio entre los diferentes criterios de valor.

### **Dirección de Portafolio**

Un portafolio consiste en proyectos, programas, subconjuntos de portafolio y operaciones gestionados como un grupo con objeto de alcanzar los objetivos estratégicos. Los proyectos o programas del portafolio no son necesariamente interdependientes ni están necesariamente relacionados de manera directa. Por ejemplo, una compañía de infraestructuras que tiene el objetivo estratégico de “maximizar el rendimiento de sus inversiones” puede incluir en un portafolio una combinación de proyectos en el ámbito del petróleo y el gas, la energía, el agua, los caminos, los ferrocarriles y los aeropuertos. A partir de esta combinación, la compañía podría optar por gestionar los proyectos relacionados como un único programa. Todos los proyectos energéticos podrían ser agrupados en un programa de energía. Del mismo modo, todos los proyectos hídricos podrían ser agrupados en un programa hídrico. De este modo los programas de energía e hídrico serían componentes integrantes del portafolio de la compañía de infraestructuras.

La dirección de portafolios consiste en la gestión centralizada de uno o más portafolios con objeto de alcanzar los objetivos estratégicos. La dirección de portafolios se centra en asegurar que los proyectos y programas se revisen a fin de establecer prioridades para la

asignación de recursos, y en que la dirección del portafolio sea consistente con las estrategias de la organización y esté alineada con ellas.

Las organizaciones cuya misión principal es el desarrollo y modificación del software a veces da tratamiento a los proyectos de software como elementos de un portafolio con el fin de aumentar la eficiencia y la eficacia de sus actividades de trabajo y participar en las iniciativas de mejora de procesos que serán de beneficio para todos los proyectos del portafolio de las organizaciones. Dentro de un portafolio, los proyectos de software son priorizados para su ejecución en base a parámetros tales como: la complejidad, grado de incertidumbre, el valor del negocio, retorno de la inversión, y así sucesivamente. Los marcos del ciclo de vida estandarizados que se pueden adaptar a cada proyecto son elementos importantes de la gestión del portafolio para las organizaciones de software.

### **Dirección de Programas**

Un programa se define como un grupo de proyectos relacionados, subprogramas y actividades de programas, cuya gestión se realiza de manera coordinada para obtener beneficios que no se obtendrían si se gestionarían de forma individual. Los programas pueden incluir elementos de trabajo relacionado que están fuera del alcance de los proyectos específicos del programa. Un proyecto puede o no formar parte de un programa, pero un programa siempre consta de proyectos.

La dirección de programas consiste en la aplicación de conocimientos, habilidades, herramientas y técnicas a un programa para satisfacer los requisitos del mismo y para lograr unos beneficios y un control que no es posible obtener dirigiendo los proyectos de manera individual.

Los proyectos de un programa se relacionan a través del resultado común o de la capacidad colectiva. Si la relación entre los proyectos viene dada únicamente por un cliente, vendedor, tecnología o recurso en común, el esfuerzo se debería gestionar como un portafolio de proyectos, en lugar de hacerlo como un programa.

La dirección de programas se centra en las interdependencias entre los proyectos y ayuda a determinar el enfoque óptimo para gestionarlas. Las acciones relacionadas con estas interdependencias, pueden incluir:

- Resolver restricciones y/o conflictos de recursos que afectan a múltiples proyectos del programa,
- Alinear la dirección de la organización/estratégica que afecta las metas y los objetivos de los proyectos y del programa, y
- Resolver la gestión de incidentes y cambios dentro de una estructura de gobernabilidad compartida.

Un ejemplo de programa sería un nuevo sistema de comunicaciones vía satélite, con proyectos para el diseño y construcción del satélite y las estaciones terrestres, la integración del sistema y el lanzamiento del satélite.

El software es a veces considerado como un componente secundario del sistema en programas que implican el desarrollo de diversos componentes. Como resultado, puede no haber ningún gerente de proyectos de software designado explícitamente. Dado el papel central del software en el actual sistema y el impacto del software que tiene sobre las características del sistema (así como el software es un componente principal que puede afectar a la finalización de un sistema en la planificación predeterminada), un director de proyecto de software designado debe ser un miembro del equipo de la dirección del programa.

### **Proyectos y Planificación Estratégica**

Los proyectos se utilizan a menudo como medio para alcanzar directa o indirectamente los objetivos recogidos en el plan estratégico de una organización. Por lo general, los proyectos se autorizan como resultado de una o más de las siguientes consideraciones estratégicas:

- Demanda del mercado (p.ej., una compañía automotriz que autoriza un proyecto para construir más automóviles de bajo consumo en respuesta a la escasez de combustible);
- Oportunidad estratégica/necesidad del negocio (p.ej., un centro de formación que autoriza un proyecto de creación de un curso nuevo para aumentar sus ingresos);
- Necesidad social (p.ej., una organización no gubernamental en un país en vías de desarrollo que autoriza un proyecto para dotar de sistemas de agua potable, baños y educación sanitaria a comunidades que padecen altos índices de enfermedades infecciosas);
- Consideraciones ambientales (p.ej., una empresa pública que autoriza un proyecto para crear un nuevo servicio que consista en compartir automóviles eléctricos a fin de reducir la contaminación);
- Solicitud de un cliente (p.ej., una empresa eléctrica que autoriza un proyecto para construir una nueva subestación a fin de abastecer un nuevo parque industrial);
- Avance tecnológico (p.ej., una compañía de productos electrónicos que autoriza un proyecto nuevo para desarrollar un ordenador portátil más rápido, más económico y más pequeño sobre la base de los avances en materia de memorias y de tecnología electrónica); y
- Requisito legal (p.ej., un fabricante de productos químicos que autoriza un proyecto para establecer pautas para la correcta manipulación de un nuevo material tóxico).

Los proyectos incluidos en programas o portafolios constituyen un medio para alcanzar las metas y los objetivos de la organización, a menudo en el contexto de un plan estratégico. Si bien dentro de un programa un grupo de proyectos puede tener beneficios específicos, estos proyectos también pueden contribuir a los beneficios del programa, a los objetivos del portafolio y al plan estratégico de la organización.

Las organizaciones gestionan los portafolios sobre la base de su plan estratégico. Uno de los objetivos de la dirección de portafolios consiste en maximizar el valor del portafolio mediante un examen cuidadoso de sus componentes: los programas, los proyectos y otros trabajos relacionados que lo constituyen. Los componentes cuya contribución a los



objetivos estratégicos del portafolio es mínima, pueden ser excluidos. De esta manera, el plan estratégico de una organización se convierte en el principal factor que guía las inversiones en los proyectos. Paralelamente, los proyectos realimentan a los programas y portafolios mediante informes de estado, lecciones aprendidas y solicitudes de cambio que pueden ayudar a identificar posibles impactos sobre otros proyectos, programas o portafolios. Las necesidades de los proyectos, incluidas las necesidades de recursos, se recopilan y se comunican nuevamente a nivel del portafolio, lo que a su vez orienta la planificación de la organización.

Los proyectos de software a veces se llevan a cabo para explorar la viabilidad de utilizar un nuevo proceso de desarrollo dentro de un contexto específico (como un modelo de adaptación del ciclo de vida del proyecto), para explorar y aprender una nueva tecnología (como la computación en la nube), para desarrollar un prototipo de un nuevo estilo de interfaz de usuario (tal como una pantalla holográfica o en tres dimensiones) o para explotar una innovación basada en software (por ejemplo, una interfaz de multimedia incluyendo una aplicación de software). En estos casos, el valor de negocio del proyecto de software no es el producto de salida, pero sí el conocimiento adquirido institucional en el proyecto.

### **Oficina de Dirección de Proyectos**

Una Oficina de Dirección de Proyectos (PMO, por sus siglas en inglés) es una estructura de gestión que estandariza los procesos de gobierno relacionados con el proyecto y hace más fácil compartir recursos, metodologías, herramientas y técnicas. Las responsabilidades de una PMO pueden abarcar desde el suministro de funciones de soporte para la dirección de proyectos hasta la responsabilidad de la propia dirección de uno o más proyectos.

Existen diferentes tipos de estructuras de PMOs en las organizaciones, en función del grado de control e influencia que ejercen sobre los proyectos en el ámbito de la organización. Por ejemplo:

- **De apoyo.** Las PMOs de apoyo desempeñan un rol consultivo para los proyectos, suministrando plantillas, mejores prácticas, capacitación, acceso a la información y lecciones aprendidas de otros proyectos. Este tipo de PMO sirve como un repositorio de proyectos. Esta PMO ejerce un grado de control reducido.
- **De control.** Las PMOs de control proporcionan soporte y exigen cumplimiento por diferentes medios. Este cumplimiento puede implicar la adopción de marcos o metodologías de dirección de proyectos a través de plantillas, formularios y herramientas específicos, o conformidad en términos de gobierno. Esta PMO ejerce un grado de control moderado.
- **Directiva.** Las PMOs directivas ejercen el control de los proyectos asumiendo la propia dirección de los mismos. Estas PMOs ejercen un grado de control elevado.

La PMO integra los datos y la información de los proyectos estratégicos corporativos y evalúa hasta qué punto se cumplen los objetivos estratégicos de alto nivel. La PMO constituye el vínculo natural entre los portafolios, programas y proyectos de la organización y los sistemas de medida corporativos (p.ej., cuadro de mando integral).

Puede que los proyectos que la PMO apoya o dirige no guarden más relación entre sí que la de ser gestionados conjuntamente. La forma, la función y la estructura específicas de una PMO dependen de las necesidades de la organización a la que ésta da soporte.

Una PMO puede tener la autoridad para actuar como un interesado integral y tomar decisiones clave a lo largo de la vida de cada proyecto, hacer recomendaciones, poner fin a proyectos o tomar otras medidas, según sea necesario, a fin de mantenerlos alineados con los objetivos de negocio. Asimismo, la PMO puede participar en la selección, gestión e utilización de recursos de proyectos compartidos o dedicados.

Una función fundamental de una PMO es brindar apoyo a los directores del proyecto de diferentes formas, que pueden incluir, entre otras:

- Gestionar recursos compartidos a través de todos los proyectos dirigidos por la PMO;
- Identificar y desarrollar una metodología, mejores prácticas y estándares para la dirección de proyectos;
- Entrenar, orientar, capacitar y supervisar;
- Monitorear el cumplimiento de los estándares, políticas, procedimientos y plantillas de la dirección de proyectos mediante auditorías de proyectos;
- Desarrollar y gestionar políticas, procedimientos, plantillas y otra documentación compartida de los proyectos (activos de los procesos de la organización); y
- Coordinar la comunicación entre proyectos.

Los directores de proyecto y las PMOs persiguen objetivos diferentes y, por lo tanto, responden a necesidades diferentes. Todos estos esfuerzos están alineados con las necesidades estratégicas de la organización. A continuación se relacionan algunas de las diferencias entre los roles de directores de proyecto y PMO:

- El director del proyecto se concentra en los objetivos específicos del proyecto, mientras que la PMO gestiona los cambios significativos relativos al alcance del programa, que pueden considerarse como oportunidades potenciales para alcanzar mejor los objetivos de negocio.
- El director del proyecto controla los recursos asignados al proyecto a fin de cumplir mejor con los objetivos del mismo, mientras que la PMO optimiza el uso de los recursos de la organización compartidos entre todos los proyectos.
- El director del proyecto gestiona las restricciones (alcance, cronograma, costo, calidad, etc.) de los proyectos individuales, mientras que la PMO gestiona las metodologías, estándares, riesgos/oportunidades globales, métricas e interdependencias entre proyectos a nivel de empresa.

Además de las funciones y objetivos citados anteriormente, una oficina de gestión de proyectos (PMO) que gestiona una colección de proyectos de software (tal vez en adiciones a la gestión de otros tipos de proyectos) que pueden incluir, entre otras:

- Proporcionar un repositorio común para los datos relativos al esfuerzo, costos, planificación, los defectos, los agentes y factores de riesgo recogidos de un proyecto de software dentro de una organización,
- Utilizar el repositorio de datos para desarrollar uno o más modelos de costos,
- Utilizar el repositorio de datos para analizar las fortalezas y debilidades de los proyectos de software como una base para las iniciativas de mejora de procesos y para analizar los resultados de las actividades de mejora de procesos,
- Ayudar a los administradores de proyectos en la realización de estimaciones de costos y programas y la preparación de los planes del proyecto,
- Proporcionar plantillas, formularios y la recopilación de datos automatizados,
- Adquirir y armonizar el uso de nuevas herramientas y plataformas para el desarrollo de software, dirección de programas y dirección de portafolios en toda la organización,
- Una biblioteca con los módulos de código reutilizables,
- Administrar los recursos compartidos,
- Asegúrese del valor de negocio de cada proyecto de software,
- Difundir las tendencias en factores tales como los métodos, herramientas y técnicas, dirección del ciclo de vida y patrones de usabilidad y técnicas en toda la organización, y
- Capacitar a los administradores de proyectos y equipos de proyecto.

En algunas organizaciones, una PMO también puede estar involucrada en las auditorías de proceso de cumplimiento de la dirección del proyecto, la evaluación de la dirección de la madurez del proyecto, y las iniciativas de mejora de procesos. La Oficina de Dirección de Proyectos, al igual que los proyectos de software, puede estar sujeta a limitaciones de la organización. Una PMO para proyectos de software puede ser una entidad independiente o un elemento de una PMO más grande. Algunas organizaciones de TI tienen una oficina de dirección de proyectos de tecnología de información que manejan múltiples proyectos (por ejemplo, infraestructura, telecomunicaciones, redes, etc.)

## **Valor del Negocio**

El valor del negocio es un concepto único para cada organización. El valor del negocio se define como el valor del negocio en su totalidad, como la suma total de sus elementos tangibles e intangibles. Como ejemplos de elementos tangibles se pueden citar los activos monetarios, los equipos, la participación de los accionistas y los servicios. Como ejemplos de elementos intangibles se pueden citar la buena voluntad, el reconocimiento de marca, el beneficio público y las marcas registradas. Dependiendo de la organización, el alcance del valor del negocio puede ser a corto, mediano o largo plazo. Se puede crear valor a través de la gestión eficaz de las operaciones permanentes. No obstante, a través del uso eficaz de la dirección de portafolios, la dirección de programas y la dirección de proyectos, las organizaciones tendrán la capacidad de emplear procesos establecidos y confiables para cumplir con los objetivos estratégicos y obtener mayor valor de negocio a partir de sus inversiones en proyectos. Si bien no todas las organizaciones están orientadas al negocio, todas ellas desarrollan actividades relacionadas con el negocio. Ya sea que se trate de una agencia gubernamental o de una organización sin fines de lucro, todas las organizaciones se centran en lograr valor de negocio para sus actividades.

Un producto de software puede ser un producto de propiedad de la empresa y proporcionar una importante fuente de ingresos para esa unidad de negocio. En algunos casos, la infraestructura y el software de soporte al cliente se podrán capitalizar y se deprecian con el tiempo. Los productos de software a veces son desarrollados para su uso en múltiples sistemas, aumentando así el valor comercial de los productos.

Dado que el software es el producto del trabajo en equipo coordinado estrechamente que a menudo es innovadora, el capital intelectual de una organización de software (el desarrollador de software, encargado, y otro personal de software) es un elemento especialmente importante de valor para el negocio. Otro elemento importante del valor de negocio de las organizaciones de software es el conjunto de procesos, procedimientos y técnicas que permiten que el director del proyecto de software y desarrollador de productos sean eficientes y eficaces mediante la mejora de la comunicación y coordinación entre los

individuos, equipos, proyectos, procesos, procedimientos. Las técnicas que proporcionan el negocio, también agregan valor al negocio.

### **Rol del Director del Proyecto**

El director del proyecto es la persona asignada por la organización ejecutora para liderar al equipo responsable de alcanzar los objetivos del proyecto. El rol del director del proyecto es diferente del de un gerente funcional o del de un gerente de operaciones. Por lo general, el gerente funcional se dedica a la supervisión gerencial de una unidad funcional o de negocio y la responsabilidad de los gerentes de operaciones consiste en asegurar que las operaciones de negocio se llevan a cabo de manera eficiente.

Dependiendo de la estructura de la organización, un director de proyecto puede estar bajo la supervisión un gerente funcional. En otros casos, el director del proyecto puede formar parte de un grupo de varios directores de proyecto que dependen de un director de programa o del portafolio, que es el responsable en última instancia de los proyectos de toda la empresa. En este tipo de estructura, el director del proyecto trabaja estrechamente con el director del programa o del portafolio para cumplir con los objetivos del proyecto y para asegurar que el plan para la dirección del proyecto esté alineado con el plan global del programa. El director del proyecto también trabaja estrechamente y en colaboración con otros roles, como los de analista de negocio, director de aseguramiento de la calidad y expertos en materias específicas.

Sin embargo, el trabajo del proyecto de software puede ser organizado por el componente del producto (por ejemplo, la interfaz de usuario, base de datos, computación y software de comunicación), funcionalmente por componente de proceso (por ejemplo, el análisis, el diseño, la construcción, los procesos de pruebas, y la instalación / formación), o por subsistema (por ejemplo, tiempo, radar, pantalla del tráfico aéreo). El equipo de proyecto podrá realizarse de una manera funcional, por ejemplo, el director del proyecto de software puede ser el gerente de las unidades funcionales del proyecto durante la planificación y ejecución de ese proyecto. Además, un proyecto de software grande puede ser tratado como

un programa de software que se descompone en múltiples proyectos; cada una con un jefe de proyecto cuyos productos trabajo se fusionan un solo producto.

Los directores de proyectos de software proporcionan liderazgo en:

- Iniciar, planificar y desarrollar las estimaciones y la planificación tanto de la iniciación de forma continua como las condiciones de cambio,
- Seguimiento y control de los hitos del cronograma, los gastos de los presupuestos, requisitos de estabilidad, rendimiento del personal, utilización de recursos, y los factores de riesgo identificados utilizando un proceso sistemático de control de versiones,
- Liderar y dirigir mediante la definición de la visión del proyecto y su mantenimiento como los requisitos y otros cambio de limitaciones, el día a día el liderazgo de los jefes de equipo, desarrolladores de software, y el personal de apoyo que están ocupados en el trabajo en equipo innovador,
- Mantener el cumplimiento de las políticas de la organización y los requisitos contractuales,
- La gestión del riesgo mediante la identificación, análisis, priorización y responder a factores de riesgo de manera permanente y continua,
- Facilitar, moderar, monitorear, inspirar, el trabajo con los trabajadores del conocimiento de ingeniería de software para obtener el resultado deseado, y
- La comunicación con las partes interesadas para cerrar la "brecha tecnológica" mediante el uso de términos y conceptos que son familiares para los interesados.

En un proyecto pequeño (por ejemplo, Menos de diez personas) el director del proyecto pueden tener funciones adicionales, como líder del equipo y / o diseñador de software, arquitectura de software, analista de negocios, u otras funciones que contribuyen. Por otra parte, el director de un proyecto de software pequeño puede gestionar simultáneamente uno o varios pequeños proyectos; Sin embargo, se debe tener cuidado de no sobrecargar el director de un pequeño gestor con otras funciones.

## **Habilidades Interpersonales de un Director de Proyectos**

Un director de proyecto de software necesita asegurarse de que la comunicación efectiva y la coordinación se produzcan dentro del equipo del proyecto y con los interesados externos al proyecto.

- Liderazgo,
- Humildad,
- Escucha efectiva,
- Trabajo en equipo,
- Motivación,
- Comunicación,
- Colaboración y el intercambio de conocimientos,
- Influencia,
- Gestión de conflictos,
- Toma de decisiones,
- Conocimientos de política y cultural,
- Negociación.

Algunos puntos claves del conocimiento de la Dirección de Proyectos de Software:

- Dado que el software es un producto intangible, hay muchas posibilidades para los ciclos de vida de proyectos de software y muchos enfoques diferentes para la aplicación de las técnicas de dirección de proyectos de software a los ciclos de vida.
- El principio de "la mayoría de los proyectos, la mayor parte del tiempo" es modificarse para adaptarse a las variaciones dentro de los continuos ciclos de vida de desarrollo de software; los directores de proyectos de software se adaptan los métodos, herramientas y técnicas para la gestión de proyectos y a los aspectos particulares de diversos ciclo de vida del proyecto de software.
- No se espera que los directores de proyectos de software tengan el conocimiento y las habilidades de sus miembros del equipo en profundidad, pero deben entender los



problemas y preocupaciones de sus miembros del equipo, tratar y deben de estar familiarizados con la terminología utilizada por los miembros del equipo. Los directores de proyectos de software y también deben entender diferentes enfoques para la dirección de proyectos de software dentro de la continuidad de los ciclos de vida de los proyectos de software.

- los directores de proyectos de software pueden tener antecedentes técnicos, pero no siempre en el software. Esos directores de proyectos que no tienen fuertes habilidades de software pueden tener que trabajar en estrecha colaboración con un líder técnico en su proyecto de software. Las personas con conocimientos de software fuertes podrían tener que concentrarse en el desarrollo de sus negocios, gestión de proyectos y habilidades interpersonales.

## **Gestión de la Calidad**

La gestión de la calidad del software es cada vez más importante debido a que la protección, la seguridad y el bienestar del público en general son cada vez más dependientes de software.

Cada uno de los entregables del proyecto debería tener una calidad aceptable, según lo determinado por las necesidades de los usuarios y otras partes interesadas. La gestión de la calidad, control de calidad, verificación y validación de elementos son importantes en la dirección de proyectos de software.

Los atributos de calidad de software que son importantes para los usuarios y otras partes interesadas incluyen:

- Seguridad,
- Fiabilidad,
- Resistencia,
- Fiabilidad,
- Escalabilidad,
- Rendimiento,

- Facilidad de aprendizaje,
- Facilidad de uso (usabilidad),
- Interpretación de los mensajes de error,
- Disponibilidad,
- Accesibilidad,
- Eficiencia,
- Flexibilidad,
- Interoperabilidad, y
- Robustez.

Los atributos de calidad de software que son importantes para los desarrolladores de software incluyen:

- Capacidad de prueba,
- Mantenibilidad,
- Portabilidad,
- Extensibilidad, y
- Reutilización

## **CAPÍTULO II**

### **INFLUENCIA DE LA ORGANIZACIÓN**

La descripción de los ciclos de vida de los proyectos de software, y cómo los proyectos de software interactúan con el trabajo operativo en curso y otros elementos de una organización y la influencia de los actores más allá del equipo de desarrollo de software y, cómo la estructura organizativa, afecta la forma en que se inicia un proyecto, planificado, ejecutado, supervisado y controlado, y su cierre. Se hace hincapié en las herramientas y técnicas que se han desarrollado para dar cabida a los aspectos especiales y únicos de los proyectos y la dirección de proyectos de software.

## **Influencia de la Organización en la Dirección de Proyectos**

La cultura de la organización, la estructura y el estilo de liderazgo tienen una fuerte influencia en cómo se administran y llevan a cabo los proyectos de software puesto que los ingenieros de software son trabajadores del conocimiento que desarrollan y modifican el software mediante la participación en el trabajo en equipo, coordinado estrechamente a los proyectos de software.

### **Culturas y Estilos de la Organización**

Existe una amplia gama de culturas organizacionales y estilos de liderazgo dentro de las organizaciones que desarrollan y modifican el software. La cultura, la estructura y el estilo de liderazgo para el proyecto de software están modelados e influidos por factores tales como la misión, visión y valores de la organización. Las normas de conducta de la organización, el dominio de los productos, las interacciones con otras unidades funcionales de la organización y el tamaño de la empresa y, las relaciones con los clientes y otras partes interesadas del proyecto.

Además, debido a la naturaleza única del proyecto de software (producto intangible y trabajo en equipo coordinado estrechamente), los factores de la organización que influyen en el estado de ánimo y la motivación de los trabajadores de software son un poco diferentes que para otras personas que trabajan en las organizaciones.

Los factores de la organización que tienden a aumentar la motivación, el compromiso y la productividad del personal de software incluyen factores tales como:

- Lugar de trabajo sin interrupción externa,
- Problemas técnicos desafiantes,
- La autonomía para resolver problemas,
- Capacidad de controlar un horario de trabajo,
- Aprender cosas nuevas,

- Líderes técnicos competentes,
- Oportunidades para experimentar con nuevas ideas,
- Visión convincente o estado final,
- Formación y la orientación adecuada, y
- Herramientas de software adecuadas y tecnología de la computación.

## **Comunicaciones en la Organización**

El éxito en la dirección de proyectos de una organización depende en gran medida de un estilo de comunicación efectiva dentro de la organización, sobre todo si se considera la globalización de la profesión de dirección de proyectos. Las capacidades de comunicación dentro de la organización tienen gran influencia en la forma en que se llevan a cabo los proyectos. En consecuencia, los directores de proyecto en ubicaciones distantes pueden comunicarse de manera más efectiva con todos los interesados relevantes dentro de la estructura de la organización para facilitar la toma de decisiones. Los interesados y miembros del equipo del proyecto también pueden utilizar comunicaciones electrónicas (incluido correo electrónico, mensajería de texto, mensajería instantánea, redes sociales, videoconferencia y conferencia por Internet y otros medios electrónicos) para comunicarse formal o informalmente con el director del proyecto.

En particular, la representación de los productos de software en medios electrónicos y el crecimiento de Internet y la infraestructura web han hecho posible la globalización del desarrollo de software. Los directores de proyectos de software manejan cada vez más proyectos dispersos geográficamente.

## **Activos de los Procesos de la Organización**

Los activos de los procesos de la organización son los planes, los procesos, las políticas, los procedimientos y las bases de conocimiento específicos de la organización ejecutora y utilizados por la misma. Estos incluyen cualquier objeto, práctica o conocimiento de alguna

o de todas las organizaciones que participan en el proyecto y que pueden usarse para ejecutar o gobernar el proyecto. Los activos de procesos también incluyen bases de conocimiento de la organización como lecciones aprendidas e información histórica. Los activos de los procesos de la organización pueden incluir cronogramas completados, datos sobre riesgos y datos sobre el valor ganado. Los activos de los procesos de la organización constituyen entradas para la mayoría de los procesos de planificación. A lo largo del proyecto, los miembros del equipo del proyecto pueden efectuar actualizaciones y adiciones a los activos de los procesos de la organización, según sea necesario.

### **Procesos y Procedimientos**

Los procesos y procedimientos de muchas organizaciones de desarrollo de software y de servicios se basan en la norma ISO / IEC e IEEE para la ingeniería de software, y en los Modelos de Madurez de la Capacidad (por el CMMI, Capability Maturity Model Integration) y la biblioteca de activos de proceso del Instituto de Ingeniería de Software (SEI por sus siglas en inglés).

Algunos estándares ISO / IEC e IEEE, armonizados y emitidas como normas conjuntas (ISO / IEC / IEEE) incluyen:

- ISO / IEC / IEEE 12207: Ingeniería de software y de sistemas - Procesos de los ciclos de vida del software y del proyecto, y
- ISO / IEC / IEEE 16326: Ingeniería de software y de sistemas - procesos del ciclo de vida - Gestión de proyectos.

### **Factores Ambientales de la Empresa**

Los factores ambientales de la empresa hacen referencia a condiciones que no están bajo el control del equipo del proyecto y que influyen, restringen o dirigen el proyecto. Los factores ambientales de la empresa se consideran entradas para la mayor parte de los

procesos de planificación, pueden mejorar o restringir las opciones de la dirección de proyectos, y pueden influir de manera positiva o negativa sobre el resultado.

Los factores ambientales de la empresa varían ampliamente en cuanto a tipo o naturaleza.

Los factores ambientales de la empresa, incluyen entre otros:

- La cultura, estructura y gobierno de la organización,
- La distribución geográfica de las instalaciones y recursos,
- Los estándares de la industria o gubernamentales (p.ej., reglamentos del organismo de control, códigos de conducta, estándares de producto, estándares de calidad y estándares de fabricación),
- Las infraestructuras (p.ej., instalaciones existentes y bienes de capital),
- Los recursos humanos existentes (p.ej., habilidades, disciplinas y conocimientos como los relacionados con el diseño, el desarrollo, las leyes, las contrataciones y las compras),
- La gestión de personal (p.ej., pautas de selección y retención de personal, revisión del desempeño de los empleados y registros de capacitación, política de incentivos y horas extras y registro de horas trabajadas),
- Los sistemas de autorización de trabajos de la compañía,
- Las condiciones del mercado,
- La tolerancia al riesgo por parte de los interesados,
- El clima político,
- Los canales de comunicación establecidos en la organización,
- Las bases de datos comerciales (p.ej., datos para estimación estandarizada de costos, información de estudios de los riesgos de la industria y bases de datos de riesgos), y
- El sistema de información para la dirección de proyectos (p.ej., herramientas automáticas, tales como una herramienta de software para programación, un sistema de gestión de configuraciones, un sistema de recopilación y distribución de la información o las interfaces web a otros sistemas automáticos en línea).

Los factores anteriores descritos son aplicables por igual a proyecto de software, y en adición los siguientes:

- Organización, estructura, cultura y procesos;
- Estándares gubernamentales o industriales;
- Leyes, reglamentos y políticas;
- Infraestructura (por ejemplo, instalaciones y bienes de equipo);
- Recursos humanos;
- Administración de personal;
- Clima político; y
- Sistemas de información de dirección de proyectos.

## **Interesados y Gobierno del Proyecto**

### **Interesados del Proyecto**

Los participantes del proyecto de software son cualquier persona o entidad organizativa que afecte o se vea afectado por un proyecto de software o el producto de software resultante. Las partes interesadas incluyen tanto los grupos de interés internos y externos. Las partes interesadas internas incluyen el equipo del proyecto y otras entidades orgánicas tales como un departamento de administración de marketing o contrato.

Debido a la naturaleza abstracta de software, las prestaciones del proyecto de software están sujeta a la interpretaciones más amplias y más de las variables por parte de los participantes del proyecto que son las entidades físicas. Es importante participar, coordinar, integrar y gestionar proactivamente las partes interesadas pertinentes en cuestiones de interés para ellos tan a menudo como sea apropiado con el fin de manejar las expectativas de los resultados del proyecto y de la gobernabilidad del proyecto. Algunos interesados pueden designarse en las principales partes interesadas para diferentes aspectos de un proyecto de software y en diferentes momentos durante un proyecto.

Así como los interesados pueden tener un impacto positivo o adverso en los objetivos de un proyecto, un proyecto puede ser percibido como positivo o negativo por los interesados. Por ejemplo, los líderes empresariales de una comunidad que se beneficiará con un proyecto de expansión industrial, verán los beneficios económicos para la comunidad en forma de nuevos puestos de trabajo, infraestructuras de apoyo e impuestos. En el caso de los interesados con expectativas positivas en el proyecto, sus intereses serán mejor atendidos si logran el éxito del proyecto. Por el contrario, los interesados afectados negativamente, como los propietarios de viviendas cercanas o pequeñas empresas que pudieran perder su propiedad, verse obligados a trasladarse a otro lugar o aceptar cambios no deseados en el entorno local, verán sus intereses mejor atendidos si impiden el avance del proyecto. Ignorar los intereses de los interesados afectados negativamente puede traer como consecuencia un aumento en la probabilidad de fallas, retrasos u otras consecuencias negativas para el proyecto.

Una parte importante de las responsabilidades del director de proyecto consiste en gestionar las expectativas de los interesados, lo cual puede ser difícil ya que a menudo los objetivos de los interesados son muy diferentes o contradictorios. Una parte de las responsabilidades del director del proyecto consiste en equilibrar estos intereses y asegurarse de que el equipo del proyecto interactúe con los interesados de una manera profesional y cooperativa. Los directores de proyecto pueden involucrar al patrocinador del proyecto o a otros miembros del equipo de diferentes ubicaciones para identificar y gestionar a los interesados que pueden estar dispersos en el mundo.

A continuación se presentan algunos ejemplos de interesados del proyecto:

- **Patrocinador.** Un patrocinador es la persona o grupo que provee recursos y apoyo para el proyecto y que es responsable de facilitar su éxito. El patrocinador puede ser externo o interno a la organización del director del proyecto. Desde la concepción inicial hasta el cierre del proyecto, el patrocinador promueve el proyecto.
- **Clientes y usuarios.** Los clientes son aquellas personas u organizaciones que aprobarán y gestionarán el producto, servicio o resultado del proyecto. Los usuarios son aquellas personas u organizaciones que utilizarán el producto, servicio o



resultado del proyecto. Clientes y usuarios pueden ser internos o externos a la organización ejecutora y pueden existir en diferentes niveles.

- **Vendedores.** Los vendedores, también llamados proveedores, suplidores o contratistas, son compañías externas que celebran un contrato para proporcionar componentes o servicios necesarios para el proyecto.
- **Socios de negocios.** Los socios de negocios son organizaciones externas que tienen una relación especial con la empresa, obtenida en ocasiones mediante un proceso de certificación. Los socios de negocios proporcionan experiencia especializada o desempeñan un rol específico, tales como una instalación, personalización, capacitación o apoyo.
- **Grupos de la organización.** Los grupos de la organización son interesados internos que se ven afectados por las actividades del equipo del proyecto. Entre los ejemplos de diversas partes del negocio de una organización que pueden verse afectadas por el proyecto, se incluyen marketing y ventas, recursos humanos, área legal, finanzas, operaciones, producción y servicio al cliente.
- **Gerentes funcionales.** Los gerentes funcionales son personas clave que desempeñan el rol de gestores dentro de un área administrativa o funcional de una empresa, tal como recursos humanos, finanzas, contabilidad o compras/adquisiciones. Cuentan con personal permanente propio asignado para la realización del trabajo en curso y tienen la clara misión de gestionar todas las tareas dentro de su área funcional de responsabilidad. El gerente funcional puede aportar al proyecto su experiencia en la materia, o bien su función puede proporcionar servicios al proyecto.
- **Otros interesados.** Otros interesados, tales como entidades contratantes, instituciones financieras, organismos reguladores, expertos en la materia, consultores y otros, pueden tener interés financiero en el proyecto, realizar contribuciones al proyecto o tener interés en el resultado del proyecto.

## **Gobernabilidad del Proyecto**

La gobernabilidad del proyecto es una función de supervisión que está alineada con el modelo de gobierno de la organización y que abarca el ciclo de vida del proyecto. La gobernabilidad se refiere a cuestiones tales como la toma de decisiones, el establecimiento de prioridades, y la alineación de la visión y estrategia del trabajo de la organización. Gobernanza de la organización para el proyecto de software puede incluir elementos tales como una oficina de dirección de proyectos, dirección de portafolio de proyectos, o un grupo de estrategia de TI. La naturaleza intangible de software puede resultar en un alto nivel de formalidad en el modelo de gobierno, en un intento de dar visibilidad a un producto inherentemente invisible. Los proyectos de software suelen incluir el descubrimiento de requisitos y limitaciones dentro de un ambiente de aprendizaje en cuanto los proyectos evolucionan. Los modelos formales de gobierno que tratan el desarrollo de software como un proceso lineal y predictivo pueden ejercer un impacto negativo sobre el proyecto de software realizado por la organización. Si bien los distintos tipos de proyectos requieren diferentes niveles de formalidad de gobierno, es importante que los modelos de gobierno sean adecuados para la no linealidad, y un entorno de aprendizaje adaptativo de desarrollo de software.

## **Equipo del Proyecto**

### **Composición del Equipo del Proyecto**

La composición de un equipo de proyecto de software es a menudo un equilibrio entre las consideraciones ideales y las limitaciones prácticas. Las consideraciones ideales para la composición de los equipos de desarrollo de software incluyen:

- Miembros del equipo dedicados y no dedicados.
- Equipo de colaboración versus división funcional.
- Especialistas versus Generalistas.

- Interno versus provisional.

### Equipo Colaborativo

El beneficio para los proyectos de software desde la estructuras de equipos de proyectos que mejoran la colaboración dentro y entre los equipos es tal y como se presenta en la siguiente tabla, las colaboraciones están destinadas a aumentar la productividad y facilitar la innovadora solución de problemas. La formación de un equipo de colaboración incurre en algún momento costos iniciales, que son recuperados durante la duración del proyecto.

Atributo	Objetivo
Recursos dedicados	<ul style="list-style-type: none"> <li>● Aumentar el enfoque y la productividad</li> </ul>
Equipos interdisciplinarios	<ul style="list-style-type: none"> <li>● Acelerada integración de las distintas actividades laborales.</li> <li>● La incorporación de una mayor frecuencia en la realimentación.</li> </ul>
Locación	<ul style="list-style-type: none"> <li>● Mejor comunicación.</li> <li>● Equipo dinámico mejorado.</li> <li>● Compartir el conocimiento.</li> <li>● Reducción de costos de aprendizaje</li> </ul>
Especialistas y Generalistas	<ul style="list-style-type: none"> <li>● Experiencia dedicada y flexibilidad de las asignaciones de trabajo.</li> </ul>
Ambiente de trabajo estable	<ul style="list-style-type: none"> <li>● Planificación de recursos humanos simplificada.</li> <li>● Preservación y expansión del intelecto</li> </ul>

A pesar de los beneficios de la colaboración también se aplican a la predicción del ciclo de vida de los equipos. Los equipos de colaboración a menudo son críticos para el éxito de la adaptación del ciclo de vida de los proyectos, porque los equipos requieren adaptarse a un entorno de trabajo colaborativo para dar cabida a una evolución dinámica de trabajo del proyecto.

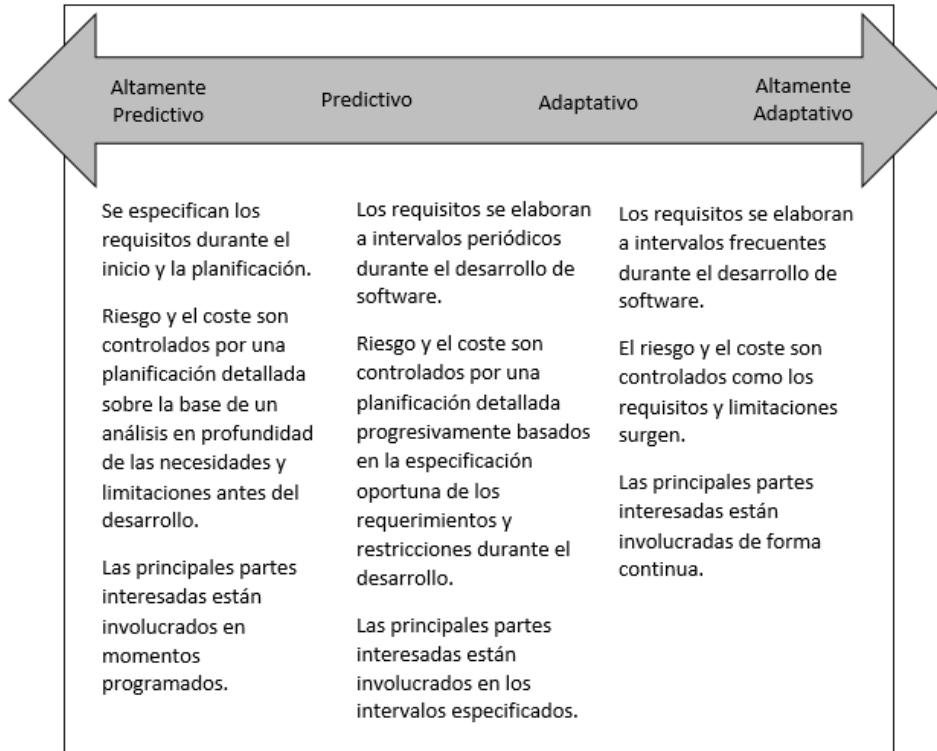
## **CAPÍTULO III**

### **CICLO DE VIDA DEL PROYECTO**

El ciclo de vida de un proyecto es la serie de fases por las que atraviesa un proyecto desde su inicio hasta su cierre. Las fases son generalmente secuenciales y sus nombres y números se determinan en función de las necesidades de gestión y control de la organización u organizaciones que participan en el proyecto, la naturaleza propia del proyecto y su área de aplicación. Las fases se pueden dividir por objetivos funcionales o parciales, resultados o entregables intermedios, hitos específicos dentro del alcance global del trabajo o disponibilidad financiera. Las fases son generalmente acotadas en el tiempo, con un inicio y un final o punto de control. Un ciclo de vida se puede documentar dentro de una metodología. Se puede determinar o conformar el ciclo de vida del proyecto sobre la base de los aspectos únicos de la organización, de la industria o de la tecnología empleada. Mientras que cada proyecto tiene un inicio y un final definido, los entregables específicos y las actividades que se llevan a cabo variarán ampliamente dependiendo del proyecto. El ciclo de vida proporciona el marco de referencia básico para dirigir el proyecto, independientemente del trabajo específico involucrado.

El ciclo de vida de proyecto de software y el ciclos de vida de productos de software son conceptos distintos. Un ciclo de vida del producto de software incluye un ciclo de vida del proyecto inicial de software, pero también incluye los procesos para la implementación, soporte, mantenimiento, evolución, sustitución y retirada de un producto de software. Mejora y adaptación del software entregado inicialmente pueden implicar varios ciclos de vida de proyectos más allá de la inicial.

Los enfoques de los ciclos de vida de los proyectos pueden variar continuamente desde enfoques predictivos u orientados a un plan hasta enfoques adaptativos u orientados al cambio. La continuidad de los ciclos de vida de proyectos de software se ilustra en el siguiente gráfico.



### **Continuidad de los Ciclos de Vida de Proyectos de Software.**

Los ciclos de vida de proyectos de software altamente predictivo se caracterizan por el énfasis en la especificación de requisitos y la planificación detallada durante la iniciación y planificación de las fases de un proyecto de software. Los planes detallados basados en los requisitos conocidos y sus limitaciones reducen el riesgo y costo. También se han previsto los hitos para la participación de las principales partes interesadas.

Los ciclos de vida altamente adaptables para proyectos de software se caracterizan por la especificación progresiva de los requisitos, basados en los ciclos de desarrollo iterativos cortos. Los riesgos y costes se reducen en la evolución progresiva de los planes iniciales; las principales partes interesadas están involucradas de forma continua.

Las siguientes consideraciones se aplican a la zona media de la serie de continuidad predictivo-adaptativo: a) los riesgos y costes se reducen por la evolución iterativa de los planes iniciales; y b) las principales partes interesadas tienen más oportunidades de participar en ciclos de iteración predictivo-adaptación que las partes interesadas en el hito

normalmente poco frecuente de los ciclos de vida altamente predictivo. Sin embargo, las partes interesadas en los proyectos predictivo-adaptativo tienen menos oportunidades de participar de las principales partes interesadas que participan continuamente en ciclos de vida de gran capacidad de adaptación.

Los ciclos de vida de proyectos de software en la región media del continuo ciclo de vida tienden a alinearse más estrechamente con el lado predictivo o el lado de adaptación del continuo dependiendo de la forma en que los requisitos se especifican, cómo se gestionan los riesgos y costes, y la naturaleza de los grupos claves de los interesados.

Los principales interesados son aquellos individuos y grupos cuya participación es esencial para un resultado exitoso. Las principales partes interesadas pueden incluir, pero no se limitan a los usuarios potenciales, clientes, ingenieros de sistemas, integradores de sistemas, adquisición, operación y mantenimiento. Diferentes actores clave pueden necesitar estar involucrados en diferentes momentos durante el desarrollo software.

Además, hay que destacar que los ciclos de vida de proyectos de software son complejos y multidimensionales. Ellos incluyen procesos para las actividades tales como la gestión de la configuración, el proceso y la garantía de calidad del producto, pruebas independientes, y otras actividades como apropiados y necesarios. Además, los proyectos de software pueden ser elementos de los programas, y pueden incluir interfaces con otras unidades funcionales de la organización de desarrollo de software o empresa de negocios, a los proyectos afiliados, y / o al grupo o proyectos subcontratados.

Hay una distinción entre la planificación y la realización de un proyecto de software para la entrega y el producto resultante. El cliente y otras partes interesadas clave para un proyecto dentro del ciclo de vida del software, podrían elegir: a) aceptar una sola entrega del producto al final del proyecto; b) aceptar la entrega de incrementos planificados previamente, de la funcionalidad del producto, en momentos específicos durante el desarrollo; c) aceptar la entrega al final del proyecto, pero con presentaciones periódicas de los incrementos de los productos durante el desarrollo; o d) no tener ninguna implicación

entre la especificación de requisitos inicial y la entrega del producto. Del mismo modo, un cliente y otras partes interesadas clave para un proyecto de software dentro del ciclo de vida de adaptación podrían optar por presenciar demostraciones de la evolución de las capacidades en los extremos de los ciclos de iteración con una sola entrega del producto al final del proyecto, o para aceptar la entrega de incrementos de productos a intervalos planificados o emergentes.

Otras posibles variaciones son: a) un director de proyectos de software podría realizar planos detallados de los requisitos (de manera predictiva) con ciclos de iteración frecuentes durante la construcción de software (de una manera adaptativa); b) un proyecto de software podría llevarse a cabo de forma iterativa durante el análisis y el diseño con el producto construido como un único incremento del producto; c) un producto de software podría desarrollarse gradualmente con o sin iteraciones durante el desarrollo de los incrementos; ó d) un producto de software podría desarrollarse sin iteraciones y solo un incremento del producto.

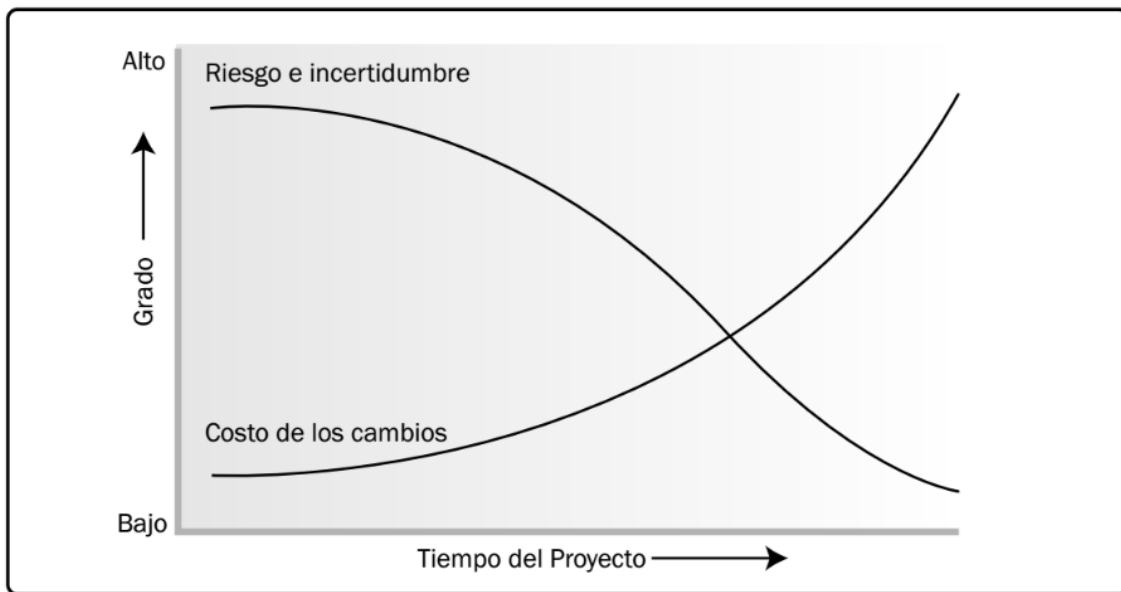
### **Características del Ciclo de Vida de un Proyecto**

La creación de entregas de software por lo general requiere de una gran variedad de procesos de los ciclos de vida del proyecto. De acuerdo con la norma ISO / IEC / IEEE 12207, el desarrollo de software incluye los siguientes procesos:

- Análisis: Procesos de análisis de requisitos del software.
- Arquitectura: Procesos de diseño de la arquitectura de software.
- Diseño: Procesos de diseño de los detalles de software.
- Construcción: Procesos de construcción del Software.
- Integración: Procesos de Integración de Software, y
- Prueba: Pruebas de Calificación de Software.

La siguiente figura ilustra los niveles de costos y del personal a través de los ciclos de vida del proyecto. La figura representa un perfil que se eleva durante el cierre del proyecto. Este perfil es típico del software predictivo en los ciclos de vida del proyecto. Los ciclos de vida

del software tienden a disminuir el pico de costo y el nivel de personal durante la ejecución y el seguimiento y control de las fases y el cambio de todo el perfil de las etapas anteriores. Esto es posible debido a que los ciclos de vida validan incrementos de software que trabaja en forma continua para minimizar el impacto y el costo de los cambios posteriores, además, los ciclos de vida que mantienen un nivel de personal constante durante la ejecución y el seguimiento y control tienden a aplanar el perfil durante aquellos elementos de los ciclos de vida de proyectos de software.



## Fases del Proyecto

### Relaciones entre Fases

Cuando los proyectos constan de más de una fase, las fases son parte de un proceso generalmente secuencial, diseñado para asegurar el control adecuado del proyecto y para obtener el producto, servicio o resultado deseado. Sin embargo, en determinadas situaciones, un proyecto puede beneficiarse de la implementación de fases superpuestas o simultáneas.



Existen dos tipos básicos de relaciones entre fases:

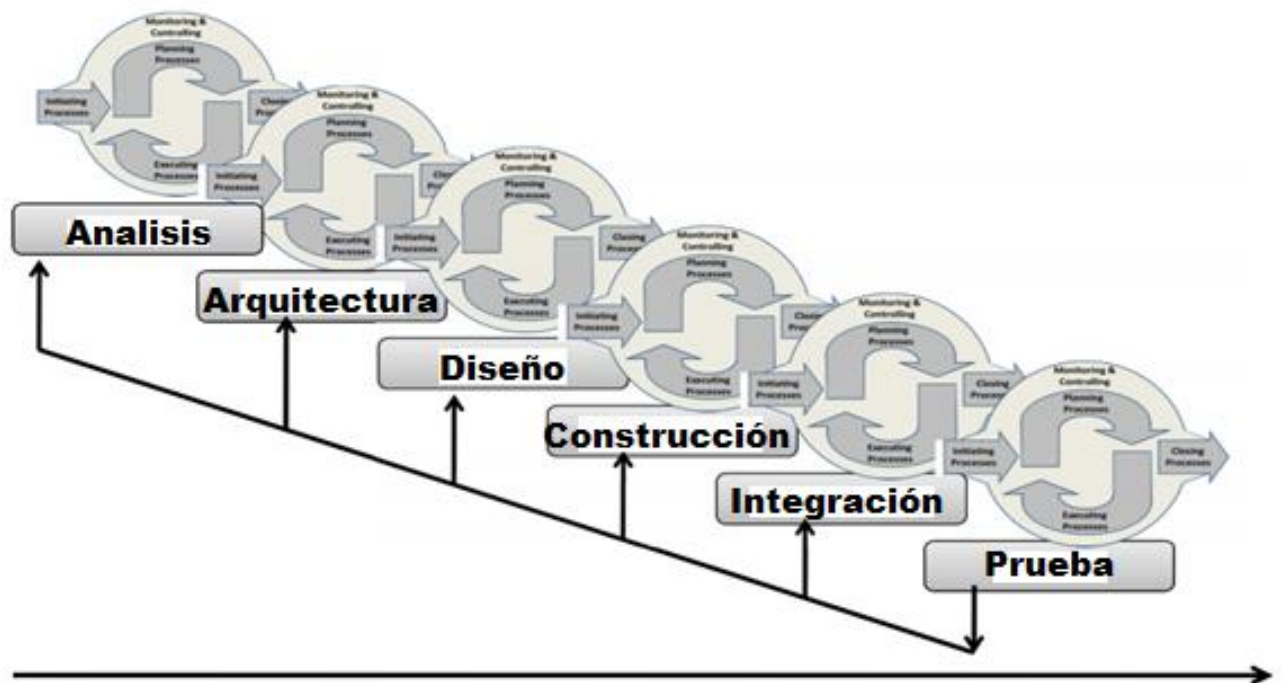
- **Relación secuencial.** En una relación secuencial, una fase sólo se inicia cuando se completa la fase anterior. La naturaleza paso a paso de este enfoque reduce la incertidumbre, pero puede eliminar opciones para acortar el cronograma general.
- **Relación de superposición.** En una relación de superposición, una fase se inicia antes de que finalice la anterior. Esto puede aplicarse algunas veces como un ejemplo de la técnica de compresión del cronograma, conocida como ejecución rápida. La superposición de fases puede requerir recursos adicionales para permitir que el trabajo se realice en paralelo, puede aumentar el riesgo y hacer preciso repetir partes de un proceso, si la fase siguiente avanza antes de que se disponga de información precisa de la fase previa.

La naturaleza del software permite una flexibilidad significativa en la superposición, el entrelazado, y la iteración de las fases de desarrollo de software. ISO / IEC / IEEE 15288 y 12207 utilizan el término "etapa" en lugar de "fases" para indicar que estas etapas se van a utilizar a lo largo de un proyecto cada vez que se necesitaban para lograr los objetivos del proyecto. Los modelos altamente adaptables de los ciclos de vida de proyectos de software, por ejemplo, se ejecutan en muchas de las etapas durante cada ciclo iterativo.

### **Ciclos de Vida Predictivos**

Los ciclos de vida predictivos (también conocidos como totalmente orientados al plan) son aquellos en los cuales el alcance del proyecto, el tiempo y costo requeridos para lograr dicho alcance, se determinan lo antes posible en el ciclo de vida del proyecto. Estos proyectos atraviesan una serie de fases secuenciales o superpuestas, donde cada fase suele enfocarse en un subconjunto de actividades del proyecto y en procesos de la dirección del proyecto. El trabajo realizado en cada fase normalmente es de naturaleza diferente al realizado en las fases anteriores y subsiguientes, y por lo tanto la composición y habilidades requeridas del equipo del proyecto puede variar de una fase a otra.

Un modelo de ciclo de vida para el proyecto de software de predicción, como se ilustra en la siguiente figura, se caracteriza por una secuencia de fases de desarrollo con retroalimentación y a la repetición de las etapas anteriores, según sea necesario.



**Fases secuenciales superpuestas de un Ciclo de Vida Predictivo de un Proyecto de Software**

Las fases pueden solaparse cuando la fase previa se ha completado parcialmente y ha proporcionado suficientes productos de trabajo para permitir la siguiente fase, y siempre que haya recursos suficientes para que puedan dos fases realizarse de forma concurrente.

La necesidad de repetir algunos procesos de las fases del proyecto previamente completadas puede ocurrir porque: a) en su naturaleza los requisitos son emergentes; b) si surgen nuevos entendimientos en torno a las expectativas de las partes interesadas en relación con el alcance el producto; c) se obtienen nuevos conocimientos sobre la tecnología; o d) los errores en el trabajo previo necesitan ser arreglados.

La planificación inicial detallada para un ciclo de vida predictivo del proyecto de software no se puede equiparar a una sola entrega "big bang" del producto resultante. Un ciclo de vida predictivo del proyecto de software puede incluir interacciones que implican una o más de las seis fases descritas en la figura anterior: **“Fases secuenciales superpuestas de un Ciclo de Vida Predictivo de un Proyecto de Software”**. Algunas de las iteraciones pueden resultar en prueba, el software entregable que puede, cuando se desee, ser incluido en el entorno del usuario.

Los ciclos de vida predictivos son de mayor éxito para los proyectos de software que tienen requisitos bien definidos, un dominio conocido del problema, tecnología estable, y un cliente conocido. Estos atributos permitir que el alcance del proyecto y el tiempo y el coste necesario para preparar dicho alcance, puedan ser determinados al inicio del ciclo de vida del proyecto.

## **CAPÍTULO IV**

### **PROYECTOS DE SOFTWARE DESDE LAS DISTINTAS ÁREAS DEL CONOCIMIENTO**

#### **Gestión del Alcance**

Para el software, el alcance del producto incluye las características y atributos de calidad que son necesarios y deseados por los usuarios, clientes y otras partes interesadas. El alcance del producto puede ser utilizado para estimar el alcance del proyecto (es decir, la planificación, el presupuesto, los recursos y la tecnología). Por otra parte, las limitaciones en el alcance del proyecto pueden determinar el alcance del producto (las características y los atributos de calidad). Las limitaciones tanto en el alcance del proyecto y el alcance del producto pueden requerir compensaciones entre las características, los atributos de calidad, la programación, el presupuesto, los recursos y la tecnología.

## **Gestión del Tiempo**

La gestión del tiempo para proyectos de software está manejada por el riesgo, la disponibilidad de recursos, el valor de negocio, y los métodos de planificación utilizados. Cuando sea posible, la planificación de un proyecto de software debe de mantenerse flexible a través de todo el proyecto para ajustar el conocimiento ganado, el mayor entendimiento del riesgo y el valor añadido. Entender los diferentes métodos de planificación y seleccionar uno o más métodos apropiados para lidiar con los riesgos de planificación es crítico para el éxito del proyecto. La mayor parte del costo de desarrollo de un proyecto es el esfuerzo humano, y el esfuerzo es el producto de las personas y el tiempo.

## **Gestión de Costos**

La gestión de costos de un proyecto, incluye los procesos que intervienen en la estimación, presupuesto, financiación, gestión y control de los costos, de modo que un proyecto pueda ser completado dentro del presupuesto aprobado inicialmente.

Grandes corporaciones y agencias gubernamentales desarrollan muchos productos nuevos de software y modifican cientos de productos existentes cada año. Las pequeñas empresas pueden desarrollar o actualizar un menor número de productos de software, pero esos productos pueden ser la esencia del negocio de la compañía. Como resultado, la gestión de costos de un proyecto es una actividad de gran importancia para todas las organizaciones y en la industria del software, se ha convertido en un proceso crítico para el éxito y la supervivencia de muchas organizaciones

El esfuerzo y la planificación, están estrechamente relacionados para proyectos de software porque el esfuerzo es el producto de personas en un tiempo determinado. Porque la hora-persona es el factor de costo primario para el desarrollo de software, la estimación del esfuerzo se utiliza como base para calcular el costo de un proyecto de software. Costos adicionales se pueden incluir como un porcentaje por encima en el costo del esfuerzo. Un gerente de proyecto de software puede gestionar los costos del proyecto en unidades de horas-personas en lugar de unidades monetarias, cuando no se proporciona el tipo de recurso para las horas-personas.

## **Gestión de la Calidad**

El aseguramiento de la calidad de software (SQA, por sus siglas en inglés) es un proceso continuo que audita otros procesos para asegurar que esos procesos están siendo seguidos (incluidos, pero no limitados por la planeación de gestión de calidad de software). SQA además determina el grado hasta el cual los resultados del control de calidad deseados del software están siendo obtenidos. El documento de SQA típicamente incluye examinar el grado hasta el cual todos los procesos usados para desarrollar y modificar software están siendo seguidos; aproximaciones para implementar estos procesos pueden ser recomendados. La guía para conducir auditorías a la calidad de software están incluidos en "IEEE Std 1028 - Software Reviews and Audits".

El Control de la Calidad de Software (SQC) se encarga de aplicar métodos, herramientas y técnicas para asegurar que los productos del trabajo de software satisfacen los requerimientos de calidad para un producto de software en desarrollo.

## **Gestión de los Interesados**

La gestión de los interesados es crítico para alcanzar resultados exitosos dado que el software es un producto intangible y es a menudo novedoso. Además, existe una brecha de expectativas entre lo que el cliente expresa y lo que los desarrolladores interpretan. Esa desalineación entre las partes interesadas representa un mayor riesgo para el éxito completo del proyecto de software.

En cuanto al ciclo de vida predictivo del proyecto, en la etapa inicial, los interesados se encuentran altamente involucrados, cuando se desarrollan los planes y requerimientos del sistema, así como en las revisiones de los principales hitos, tales como, requerimientos, diseño, revisiones de prueba y aceptación del producto.

Los proyectos de software predictivo pueden incrementar el nivel de involucramiento de los interesados mediante la construcción del software en incrementos que son periódicamente demostrados.

El ciclo de vida adaptativo del proyecto de software incluye frecuentemente demostraciones de incrementos en la evolución del entregable de software para el cliente y

otros interesados, así se mantiene la visibilidad del producto y el interesado se compromete frecuentemente en el transcurso de la duración del proyecto de software.

## **Gestión del Riesgo**

Cada proyecto desarrollo de software tiene diferentes incertidumbres, riesgos y oportunidades debido a que cada proyecto de software es una combinación única de requisitos, diseño y construcción, lo que resulta en un producto de software distinto. Los riesgos del proyecto de software y los riesgos técnicos de software afectan a todas las partes interesadas. La gestión de riesgos del software tiene como objetivo mejorar la probabilidad de lograr los objetivos del proyecto; la gestión de oportunidades de software tiene como objetivo superar los objetivos del proyecto. La gestión de oportunidades es una práctica común en la gestión de proyectos de software, especialmente en proyectos de adaptación que tienen la oportunidad de responder rápidamente a los cambios solicitados por el cliente, aplicar nuevas tecnologías, o aceptar recursos adicionales. El proceso de gestión de riesgos es "un proceso continuo para la identificación sistemática, análisis, tratamiento y seguimiento del riesgo a lo largo del ciclo de vida de un producto o servicio"

La gestión de riesgos de proyectos de software y gestión de oportunidades para la planificación de proyectos de software incluye, identificar y analizar los riesgos y oportunidades de proyectos de software; el análisis cualitativos y cuantitativos del riesgo y la oportunidad en la realización de proyectos de software; la planificación de las respuestas al riesgo y de oportunidad; y el seguimiento y control de los riesgos y oportunidades de proyectos. Comúnmente se producen riesgos para proyectos de software técnico, planificación, coste, calidad (p ej., la seguridad, la disponibilidad), la dinámica del equipo, y los factores de riesgo de clientes / interesados. Los tratamientos del riesgo incluye aceptar, evitar, transferir, o reducción del riesgo que pueda producirse por cualquiera acción inmediata o de seguimiento y acción diferida, cuando sea necesario mitigar.