

Administración de Proyectos de Tecnología



Proyecto Final

Guía para la Administración de Proyectos de Desarrollo de Software

Preparado por : Equipo #33

Integrante 1

- Matrícula1

Integrante2

- Matrícula2

Integrante3

- Matrícula3

Integrante4

- Matrícula4

Sede Nogales y Navojoa

Fecha de Publicación: _____

Tabla de Contenidos

	Página
Documentación del Proyecto	
Carta de Presentación / Resumen Ejecutivo	3
Análisis de Variaciones con respecto al Baseline	5
Gráfica de Gantt	8
Metodología de Trabajo	11
Organización del Equipo de Trabajo	12
Análisis del esfuerzo aplicado	13
Análisis de la aplicación de la Administración de Proyectos	16
Minutas de las juntas realizadas por el equipo	18
Desarrollo de la Guía de Admón de Proyectos de Desarrollo de Software	
Introducción	34
¿Todos los proyectos de desarrollo de software son iguales en todas partes?	35
¿Cómo iniciamos un proyecto de desarrollo de software?	36
¿Cómo planeamos un proyecto de desarrollo de software?	42
¿Qué aspectos debemos de considerar al ejecutar un proyecto de desarrollo de software?	53
¿Qué alternativas tenemos para controlar un proyecto de desarrollo de software?	57
¿Cuándo y cómo debemos de concluir un proyecto?	63
¿Qué debemos hacer para que nuestra empresa“ nos compre” esta nueva forma de trabajar?	66
Checklist	67
Bibliografía	69

Carta de Presentación / Resumen ejecutivo

A __ de _____ del 200__

Estimado Cliente:

El área de desarrollo de software se ha caracterizado por una deficiente administración de sus proyectos, lo que ha ocasionado cuantiosas pérdidas a las compañías que los llevan a cabo. Los principales consecuencias a las que nos enfrentamos por una mala administración de proyectos de desarrollo de software son:

- 1) Software de mala calidad
- 2) Pérdida de oportunidades por entregas tardías
- 3) Gastos reales muy por encima del presupuesto
- 4) Desmotivación por parte del equipo de trabajo.

El desarrollo de software involucra en gran parte la interacción y comunicación entre personas, y el problema es que el administrador de proyectos tradicionalmente proviene de alguna formación educativa y profesional mayormente técnica, en donde lo abstracto toma relevancia sobre el aspecto social, lo que le dificulta relacionarse con las personas y realmente administrar lo que hacen.

Por esta razón creímos necesario desarrollar esta "**Guía de Administración de Proyectos de Desarrollo de Software**" donde se expongan de forma resumida los pasos que debería de seguir y las etapas que debería contener la administración de un proyecto de este tipo.

Con esta guía se pretende apoyar al administrador del proyectos de desarrollo de software a través de:

1. Una referencia práctica de las actividades que necesita realizar para incrementar las probabilidades de éxito de sus proyectos.
2. Una sugerencia de los punto que debería documentar en su proyecto,
3. Un mejor entendimiento del por qué necesita llevar a cabo las etapas del ciclo de vida de Administración de Proyectos: Integración, Planeación, Ejecución, Control y Cierre.
4. La elaboración de un checklist con los puntos de vital importancia en la administración de proyectos.

Los puntos que quedan **fuera del alcance de la guía** son:

1. Una completa descripción de las herramientas disponibles para la administración de proyectos de desarrollo de software. Simplemente nos enfocamos en sugerir una forma práctica de administrar este tipo de proyectos. Para un mayor detalle de qué otras cosas podríamos incluir dentro de la administración favor de revisar la bibliografía utilizada/
2. Una receta de cocina para tener éxito en la administración de proyectos de desarrollo de software. Esto sería imposible ya que cada proyecto es diferente y requieren de poner mayor énfasis en tal o cuál área (algunos proyectos son más riesgosos que otros, otros tienen un presupuesto más ajustado, etc.)
3. Un libro con ejemplos prácticos donde se expongan casos de éxito aplicados y casos de fracaso y sus causas. Sin embargo, para su elaboración se realizó una investigación bibliográfica y de campo en donde se identificaron estos tipos de casos para su resumen genérico dentro de la guía.

Esperamos que esta guía sea de gran apoyo para usted y su empresa y agradeceríamos cualquier comentario o sugerencia que usted tuviera con respecto a ésta.

Atentamente

Equipo #33

Curso de Administración de Proyectos de Tecnología
Universidad Virtual del Sistema Tecnológico de Monterrey

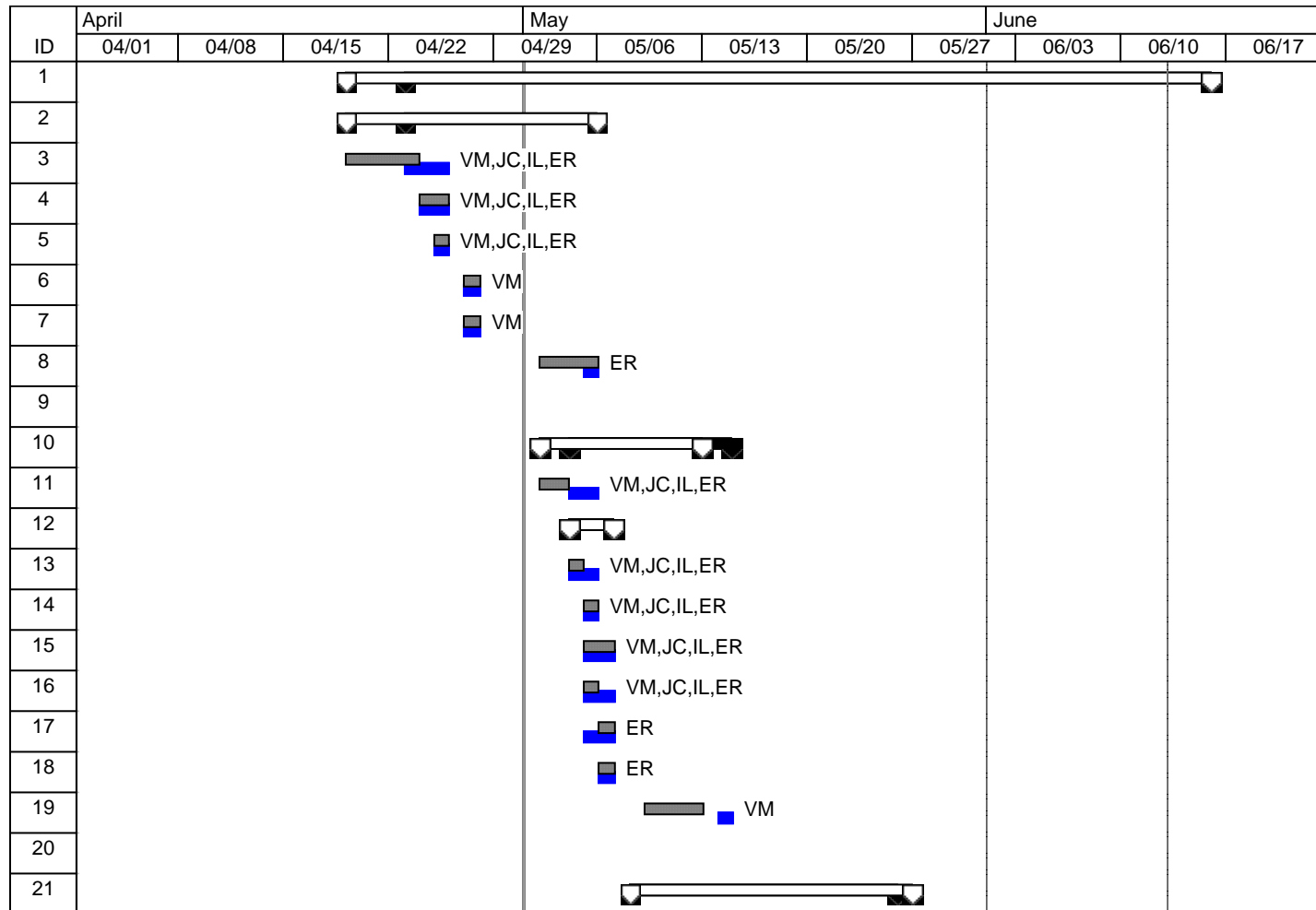
Análisis de Variaciones con respecto al Baseline

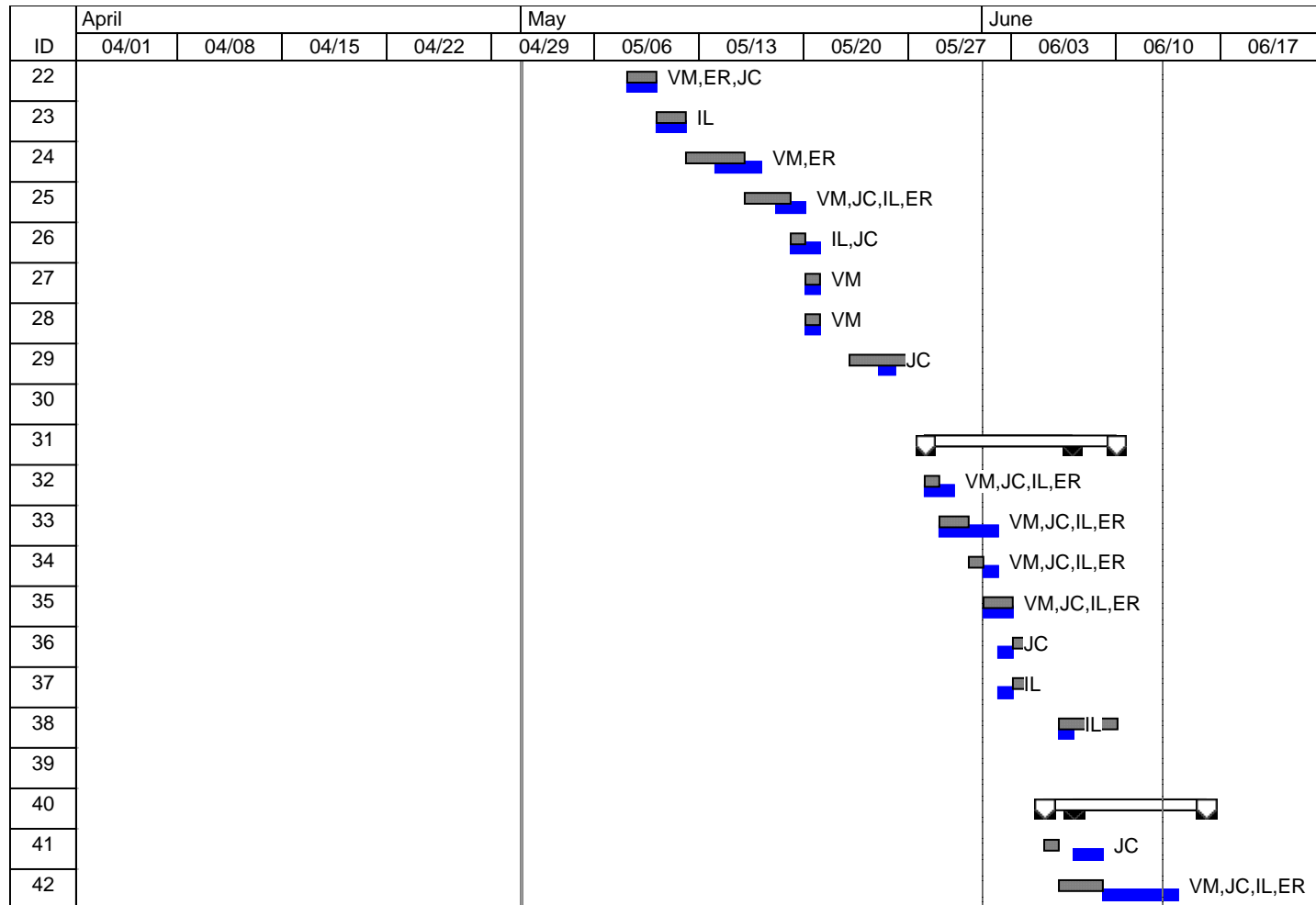
	Task Name	Start	Finish	Baseline Start	Baseline Finish	Start Var.	Finish Var.
1	Guía de administración de proyectos de tecnología	23-Apr-01	15-Jun-01	19-Apr-01	15-Jun-01	2 days	0 days
2	Etapa 1. Integración del equipo	23-Apr-01	05-May-01	19-Apr-01	05-May-01	2 days	0 days
3	1.1 Definición de integrantes y lema	23-Apr-01	25-Apr-01	19-Apr-01	23-Apr-01	2 days	2 days
4	1.2 Definición de criterios de evaluación	24-Apr-01	25-Apr-01	24-Apr-01	25-Apr-01	0 days	0 days
5	1.3 Definición de formato de minutas	25-Apr-01	25-Apr-01	25-Apr-01	25-Apr-01	0 days	0 days
6	1.4 Entrega de integrantes del equipo y lema	27-Apr-01	27-Apr-01	27-Apr-01	27-Apr-01	0 days	0 days
7	1.5 Elaboración y entrega de minutas de la etapa 1	27-Apr-01	27-Apr-01	27-Apr-01	27-Apr-01	0 days	0 days
8	1.6 Entrega de autoevaluación del equipo	05-May-01	05-May-01	02-May-01	05-May-01	3 days	0 days
9							
10	Etapa 2. Planeación del proyecto	04-May-01	14-May-01	02-May-01	12-May-01	2 days	2 days
11	2.1 Selección del tema sobre el que se desarrollará el proyecto final	04-May-01	05-May-01	02-May-01	03-May-01	2 days	2 days
12	2.2 Elaborar plan de trabajo	04-May-01	06-May-01	04-May-01	06-May-01	0 days	0 days
13	2.2.1 Definir alcance del proyecto	04-May-01	05-May-01	04-May-01	04-May-01	0 days	1 day
14	2.2.2 Establecer actividades a realizar	05-May-01	05-May-01	05-May-01	05-May-01	0 days	0 days
15	2.2.2 Estimar tiempos y fechas para cada actividad	05-May-01	06-May-01	05-May-01	06-May-01	0 days	0 days
16	2.2.3 Asignar responsables para cada actividad	05-May-01	06-May-01	05-May-01	05-May-01	0 days	1 day
17	2.3 Entrega de plan de trabajo	05-May-01	06-May-01	06-May-01	06-May-01	-1 day	0 days
18	2.4 Elaboración y entrega de minutas de la etapa 2	06-May-01	06-May-01	06-May-01	06-May-01	0 days	0 days
19	2.5 Entrega de autoevaluación del equipo	14-May-01	14-May-01	09-May-01	12-May-01	5 days	2 days
20							
21	Etapa 3. Investigación de campo	08-May-01	25-May-01	08-May-01	26-May-01	0 days	-1 day
22	3.1 Identificación de personas y empresas a entrevistar	08-May-01	09-May-01	08-May-01	09-May-01	0 days	0 days
23	3.2 Elaboración del cuestionario a aplicar en las entrevistas	10-May-01	11-May-01	10-May-01	11-May-01	0 days	0 days
24	3.3 Entrevista a profesionales del área de administración de proyectos	14-May-01	16-May-01	12-May-01	15-May-01	2 days	1 day

	Task Name	Start	Finish	Baseline Start	Baseline Finish	Start Var.	Finish Var.
25	3.4 Analizar la información recolectada	18-May-01	19-May-01	16-May-01	18-May-01	2 days	1 day
26	3.5 Síntesis y documentación de la información recolectada	19-May-01	20-May-01	19-May-01	19-May-01	0 days	1 day
27	3.6 Entrega del reporte de investigación de campo	20-May-01	20-May-01	20-May-01	20-May-01	0 days	0 days
28	3.7 Elaboración y entrega minutas de la etapa 3	20-May-01	20-May-01	20-May-01	20-May-01	0 days	0 days
29	3.8 Entrega de autoevaluación del equipo	25-May-01	25-May-01	23-May-01	26-May-01	2 days	-1 day
30							
31	Etapa 4. Investigación Bibliográfica	28-May-01	06-Jun-01	28-May-01	09-Jun-01	0 days	-3 days
32	4.1 Identificación y selección de fuentes	28-May-01	29-May-01	28-May-01	28-May-01	0 days	1 day
33	4.2 Levantamiento de información	29-May-01	01-Jun-01	29-May-01	30-May-01	0 days	2 days
34	4.3 Análisis y selección de información relevante	01-Jun-01	01-Jun-01	31-May-01	31-May-01	1 day	1 day
35	4.4 Elaboración de abstracts y documentación	01-Jun-01	02-Jun-01	01-Jun-01	02-Jun-01	0 days	0 days
36	4.5 Entrega del reporte de investigación bibliográfica	02-Jun-01	02-Jun-01	03-Jun-01	03-Jun-01	-1 day	-1 day
37	4.6 Elaboración y entrega minutas de la etapa 4	02-Jun-01	02-Jun-01	03-Jun-01	03-Jun-01	-1 day	-1 day
38	4.7 Entrega de autoevaluación del equipo	06-Jun-01	06-Jun-01	06-Jun-01	09-Jun-01	0 days	-3 days
39							
40	Etapa 5. Elaboración del Documento Final del Proyecto	07-Jun-01	15-Jun-01	05-Jun-01	15-Jun-01	2 days	0 days
41	5.1 Diseño del entregable	07-Jun-01	08-Jun-01	05-Jun-01	05-Jun-01	2 days	3 days
42	5.2 Análisis global de la información recolectada - Guía	09-Jun-01	13-Jun-01	06-Jun-01	08-Jun-01	3 days	5 days
43	5.3 Desarrollar gráfica de Gannt	13-Jun-01	13-Jun-01	08-Jun-01	08-Jun-01	5 days	5 days
44	5.4 Documentar la metodología del trabajo	12-Jun-01	14-Jun-01	08-Jun-01	08-Jun-01	4 days	6 days
45	5.5 Establecer modelo de organización del equipo de trabajo	13-Jun-01	13-Jun-01	08-Jun-01	08-Jun-01	5 days	5 days
46	5.6 Análisis del esfuerzo aplicado	13-Jun-01	13-Jun-01	08-Jun-01	08-Jun-01	5 days	5 days
47	5.7 Análisis de resultados	14-Jun-01	14-Jun-01	09-Jun-01	09-Jun-01	5 days	5 days
48	5.8 Establecer checklist para la administración de proyectos de tecnología	14-Jun-01	14-Jun-01	10-Jun-01	10-Jun-01	4 days	4 days
49	5.9 Elaborar carta de presentación, resumen ejecutivo y alcance del proyecto	11-Jun-01	12-Jun-01	11-Jun-01	11-Jun-01	0 days	1 day

	<i>Task Name</i>	<i>Start</i>	<i>Finish</i>	<i>Baseline Start</i>	<i>Baseline Finish</i>	<i>Start Var.</i>	<i>Finish Var.</i>
50	5.10 Integración de los documentos elaborados en todas las etapas	14-Jun-01	14-Jun-01	11-Jun-01	11-Jun-01	3 days	3 days
51	5.11 Revisión final del documento final del proyecto	14-Jun-01	15-Jun-01	12-Jun-01	13-Jun-01	2 days	2 days
52	5.12 Elaboración y entrega minutas de la etapa 5	15-Jun-01	15-Jun-01	14-Jun-01	14-Jun-01	1 day	1 day
53	5.13 Entrega del documento final del proyecto	15-Jun-01	15-Jun-01	15-Jun-01	15-Jun-01	0 days	0 days
54							
55	Cierre de proyecto	15-Jun-01	15-Jun-01	15-Jun-01	15-Jun-01	0 days	-1 day

Gráfica de Gantt para el proyecto de Elaboración de una Guía de Desarrollo de Software





ID	April				May					June			
	04/01	04/08	04/15	04/22	04/29	05/06	05/13	05/20	05/27	06/03	06/10	06/17	
43											■ VM		
44											■ ER		
45											■ JC		
46											■ IL		
47											■ VM,JC,IL,ER		
48											■ VM,JC,IL,ER		
49											■ VM,JC		
50											■ ER,IL		
51											■ VM,JC,IL,ER		
52											■ VM		
53											■ ER		
54													
55											◼ 06/15		

Metodología de trabajo

El objetivo de nuestro proyecto es desarrollar una guía para ayudar a administrar proyectos de desarrollo de software. Una vez integrado el equipo de trabajo, nuestro proyecto se basó en la siguiente metodología:

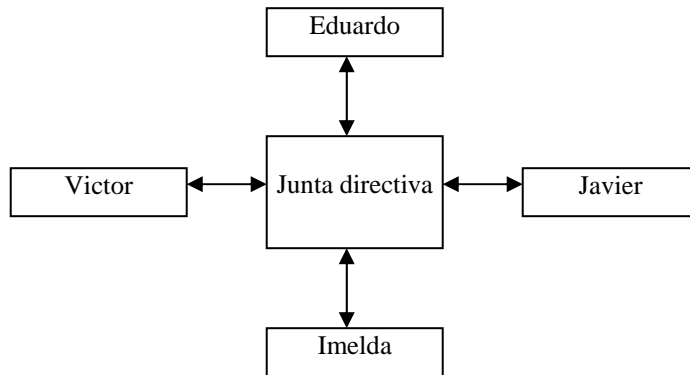
- **Definición del alcance del proyecto.**
En esta etapa se nos definió el objetivo del proyecto. En base al objetivo, es necesario determinar el alcance del proyecto, para definir qué queda dentro o fuera del proyecto. La metodología utilizada durante esta etapa se basó en reuniones del equipo de trabajo y retroalimentación del equipo docente.
- **Investigación de campo.**
El objetivo de esta etapa fue conocer las técnicas utilizadas por profesionales dedicados a administrar proyectos de desarrollo de software a través de un diálogo directo con ellos. La metodología utilizada durante esta etapa se basó la aplicación de entrevistas a administradores de proyectos de desarrollo de software en nuestra región. Se diseño un cuestionario cuidando que todas las preguntas que se realizaran dieran un valor agregado a nuestra investigación.

Después de realizar las entrevistas se analizaron los resultados obtenidos, los cuales fueron reportados durante la segunda etapa del proyecto.

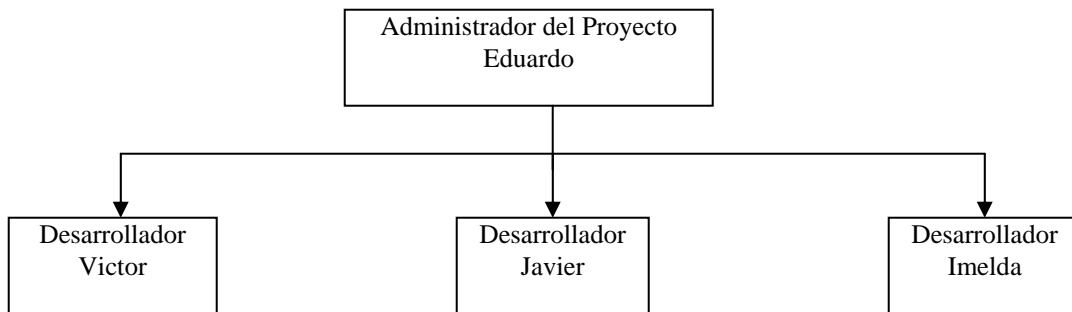
- **Investigación bibliográfica.**
El objetivo de esta etapa fue conocer parte de la literatura existente acerca de la administración de proyectos de desarrollo de software a través de la lectura de textos y artículos existentes ya sea en forma de libros, revistas o artículos de internet. La metodología se basó en la investigación y selección de fuentes relevantes, la cual se documentó con la generación de abstractos del material.
- **Desarrollo de la guía.**
El objetivo de esta etapa es generar una guía que ayude a administrar proyectos de desarrollo de software. La guía deberá basarse en las experiencias aprendidas durante las entrevistas en la investigación de campo, la documentación seleccionada sobre la administración de proyectos de desarrollo de software, los conocimientos adquiridos a través de las sesiones satelitales y la participaciones del Tea Room, y sobre todo, la perspectiva que el equipo se ha formado acerca de administración de proyectos de desarrollo de software.

Modelo de Organización del equipo de trabajo

Para el desarrollo de nuestro proyecto, nos organizamos en un equipo de trabajo con cuatro integrantes, tres de ellos de la Sede Nogales y uno más de la Sede Navojoa. La estructura organizacional que adoptamos en un principio fue la de un equipo autodirigido que representamos en el siguiente diagrama:



Los problemas que enfrentamos es que el liderazgo difícilmente puede transferirse entre los miembros del equipo y se pierde el enfoque del proyecto. Una vez aprendida esa lección nos reorganizamos de acuerdo a la siguiente estructura:



Análisis del esfuerzo aplicado

Nombre del Recurso / Actividad	Esfuerzo Real
Víctor Miranda (VM)	77 hrs
1.1 Definición de integrantes y lema	3 hrs
1.2 Definición de criterios de evaluación	2 hrs
1.3 Definición de formato de minutas	2 hrs
1.4 Entrega de integrantes del equipo y lema	1 hrs
1.5 Elaboración y entrega de minutas de la etapa 1	1 hrs
2.1 Selección del tema sobre el que se desarrollará el proyecto final	2 hrs
2.2.1 Definir alcance del proyecto	7 hrs
2.2.2 Establecer actividades a realizar	3 hrs
2.2.2 Estimar tiempos y fechas para cada actividad	2 hrs
2.2.3 Asignar responsables para cada actividad	2 hrs
2.5 Entrega de autoevaluación del equipo	1 hrs
3.1 Identificación de personas y empresas a entrevistar	2 hrs
3.3 Entrevista a profesionales del área de administración de proyectos	2 hrs
3.4 Analizar la información recolectada	5 hrs
3.6 Entrega del reporte de investigación de campo	2 hrs
3.7 Elaboración y entrega minutas de la etapa 3	1 hrs
4.1 Identificación y selección de fuentes	3 hrs
4.2 Levantamiento de información	2 hrs
4.3 Análisis y selección de información relevante	2 hrs
4.4 Elaboración de abstracts y documentación	2 hrs
5.2 Análisis global de la información recolectada - Guía	12 hrs
5.3 Desarrollar gráfica de Gannt	4 hrs
5.7 Análisis de resultados	2 hrs
5.8 Establecer checklist para la administración de proyectos de tecnología	4 hrs
5.9 Elaborar carta de presentación, resumen ejecutivo y alcance del proyecto	4 hrs
5.11 Revisión final del documento final del proyecto	3 hrs
5.12 Elaboración y entrega minutas de la etapa 5	1 hrs
Cierre de proyecto	0 hrs
Javier Castillo (JC)	49 hrs
1.1 Definición de integrantes y lema	2 hrs
1.2 Definición de criterios de evaluación	2 hrs
1.3 Definición de formato de minutas	1 hrs
2.1 Selección del tema sobre el que se desarrollará el proyecto final	2 hrs
2.2.1 Definir alcance del proyecto	7 hrs
2.2.2 Establecer actividades a realizar	2 hrs
2.2.2 Estimar tiempos y fechas para cada actividad	2 hrs
2.2.3 Asignar responsables para cada actividad	1 hrs
3.1 Identificación de personas y empresas a entrevistar	2 hrs
3.4 Analizar la información recolectada	1 hrs
3.5 Síntesis y documentación de la información recolectada	1 hrs
3.8 Entrega de autoevaluación del equipo	1 hrs
4.1 Identificación y selección de fuentes	2 hrs
4.2 Levantamiento de información	2 hrs

Nombre del Recurso / Actividad	Esfuerzo Real
4.3 Análisis y selección de información relevante	2 hrs
4.4 Elaboración de abstracts y documentación	2 hrs
4.5 Entrega del reporte de investigación bibliográfica	1 hrs
5.1 Diseño del entregable	3 hrs
5.2 Análisis global de la información recolectada - Guía	4 hrs
5.5 Establecer modelo de organización del equipo de trabajo	2 hrs
5.7 Análisis de resultados	2 hrs
5.8 Establecer checklist para la administración de proyectos de tecnología	2 hrs
5.9 Elaborar carta de presentación, resúmen ejecutivo y alcance del proyecto	1 hrs
5.11 Revisión final del documento final del proyecto	2 hrs
Cierre de proyecto	0 hrs
Imelda Luzanía (IL)	53 hrs
1.1 Definición de integrantes y lema	2 hrs
1.2 Definición de criterios de evaluación	3 hrs
1.3 Definición de formato de minutas	1 hrs
2.1 Selección del tema sobre el que se desarrollará el proyecto final	1 hrs
2.2.1 Definir alcance del proyecto	5 hrs
2.2.2 Establecer actividades a realizar	3 hrs
2.2.2 Estimar tiempos y fechas para cada actividad	2 hrs
2.2.3 Asignar responsables para cada actividad	1 hrs
3.2 Elaboración del cuestionario a aplicar en las entrevistas	3 hrs
3.4 Analizar la información recolectada	3 hrs
3.5 Síntesis y documentación de la información recolectada	2 hrs
4.1 Identificación y selección de fuentes	2 hrs
4.2 Levantamiento de información	1 hrs
4.3 Análisis y selección de información relevante	2 hrs
4.4 Elaboración de abstracts y documentación	1 hrs
4.6 Elaboración y entrega minutas de la etapa 4	1 hrs
4.7 Entrega de autoevaluación del equipo	1 hrs
5.2 Análisis global de la información recolectada - Guía	5 hrs
5.6 Análisis del esfuerzo aplicado	3 hrs
5.7 Análisis de resultados	3 hrs
5.8 Establecer checklist para la administración de proyectos de tecnología	3 hrs
5.10 Integración de los documentos elaborados en todas las etapas	3 hrs
5.11 Revisión final del documento final del proyecto	2 hrs
Cierre de proyecto	0 hrs
Eduardo Romero (ER)	91 hrs
1.1 Definición de integrantes y lema	3 hrs
1.2 Definición de criterios de evaluación	3 hrs
1.3 Definición de formato de minutas	2 hrs
1.6 Entrega de autoevaluación del equipo	1 hrs
2.1 Selección del tema sobre el que se desarrollará el proyecto final	2 hrs
2.2.1 Definir alcance del proyecto	8 hrs
2.2.2 Establecer actividades a realizar	5 hrs
2.2.2 Estimar tiempos y fechas para cada actividad	5 hrs

Nombre del Recurso / Actividad	Esfuerzo Real
2.2.3 Asignar responsables para cada actividad	2 hrs
2.3 Entrega de plan de trabajo	1 hrs
2.4 Elaboración y entrega de minutas de la etapa 2	1 hrs
3.1 Identificación de personas y empresas a entrevistar	2 hrs
3.3 Entrevista a profesionales del área de administración de proyectos	2 hrs
3.4 Analizar la información recolectada	5 hrs
4.1 Identificación y selección de fuentes	2 hrs
4.2 Levantamiento de información	6 hrs
4.3 Análisis y selección de información relevante	3 hrs
4.4 Elaboración de abstracts y documentación	4 hrs
5.2 Análisis global de la información recolectada - Guía	15 hrs
5.4 Documentar la metodología del trabajo	4 hrs
5.7 Análisis de resultados	3 hrs
5.8 Establecer checklist para la administración de proyectos de tecnología	4 hrs
5.10 Integración de los documentos elaborados en todas las etapas	3 hrs
5.11 Revisión final del documento final del proyecto	3 hrs
5.13 Entrega del documento final del proyecto	2 hrs
Cierre de proyecto	0 hrs
Horas Totales Invertidas	270 hrs

El costo del proyecto fué de 270 horas. En este caso, por tratarse de un proyecto que se realiza como complemento de un curso de aprendizaje, no tuvimos un costo en unidades monetarias, ya que para esto sería necesario que estimáramos cuánto vale el tiempo que le dedicamos a cumplir con las tareas propias de un estudiante de maestría.

Análisis de los resultados de aplicar las técnicas de administración de proyectos

Al desarrollar el proyecto de la “Guía para Administración de Proyectos de Desarrollo de Software utilizamos las siguientes técnicas de administración de proyectos:

1. Se empezó por organizar el equipo de trabajo a través de un tipo de liderazgo peer to peer donde cada uno tomaba las veces de líder ante diferentes situaciones. Sin embargo con el tiempo este modelo fue evolucionando hacia otros diferentes tipos de liderazgo al darnos cuenta de la personalidad de cada uno de nosotros y de las diferentes situaciones en que nos encontrábamos (algunos casados otros solteros, alguien en una ciudad diferente a la de los demás miembros del equipo, algunos más inclinados a recibir órdenes para empezar a trabajar mientras que otros trabajaban más independientemente, etc.). Aunque no se nombró formalmente, el líder del equipo de trabajo surgió espontáneamente al ir progresando en el proyecto y tomando la iniciativa para mover al equipo en la dirección deseada para alcanzar el objetivo.
2. Se estableció una misión / visión para la guía. Esta misión sirvió para mantener enfocado al equipo y tomar decisiones acerca de lo que debíamos o no incluir.
3. Se utilizó MS Project para documentar el plan maestro del proyecto, y posteriormente darle seguimiento de una forma más fácil y rápida. Como experiencia podemos decir que para nuestro proyecto fué una herramienta útil ya que nos permitió detectar fácilmente variaciones, nivelar cargas de trabajo, y ver de forma separada y global como se iban comportando los recursos. Anteriormente habíamos utilizado MS Project solamente para calendarizar actividades, sin embargo, al desarrollar este proyecto nos dimos cuenta de las diferentes perspectivas en las que puedes ver el proyecto a través de las diferentes vistas y reportes que este software proporciona para ejercer un mejor control.
4. Algunas herramientas para el control de proyectos que nos parecieron interesantes para ser utilizadas, sin embargo decidimos mejor no llevarlas a cabo dada la relativa simplicidad del proyecto. Es decir, dado que el proyecto fue parte del curso de administración de proyectos tecnología, las fechas de entrega ya estaban dadas y tuvimos que hacer una planeación hacia atrás para determinar cuando deberíamos empezar a trabajar en cada parte. La concurrencia de actividades que se desarrollaban al mismo tiempo no fue mucha porque en una gran parte de ellas todo el equipo estaba involucrado. Por otro lado, fue difícil adaptarse a nuevas herramientas de control (aunque fueran mejores) una vez que ya habíamos empezado a trabajar y estábamos acostumbrados a formas de control menos organizadas y proactivas.

5. Aunque a veces llegaron a ser solamente un compromiso, las minutas en ocasiones resultaron ser buenas referencias para formalizar las decisiones y acuerdos que se tomaron por parte del equipo. Esta herramienta llegó a ser útil para recordar que había pasado con tal o cual actividad que se había comprometido y en donde aún no teníamos respuesta o solución.
6. El Work Breakdown Structure es una herramienta tan sencilla de comprender y que a veces parece tan obvia, que difícilmente pudimos creer que nunca hubiéramos utilizado algo así. Quizá todos hayamos hecho algo parecido al WBS alguna vez, sin embargo el problema podía haber radicado en que no hacíamos las sub-actividades lo suficientemente pequeñas para poder hacer una estimación acertada de tiempo y de costos.
7. El modelo de colaboración virtual a distancia con el equipo (ya que una integrante del equipo se encuentra en otra ciudad) sirvió para comunicar de forma asíncrona los acuerdos y actividades que deberíamos desempeñar. Sin embargo, la comunicación en tiempo real (a través de chat o teléfono) que a veces fue necesaria para actuar rápidamente ante imprevistos, no funcionó del todo bien por la dificultad de coincidir en tiempo disponible y el costo que representa (específicamente de teléfono).

Minutas de las juntas

Minuta # 1.

Minuta de la reunión llevada a cabo el día 24 de abril del 2001.

Asistentes

Imelda Luzanía
Eduardo Romero
Javier Castillo
Victor Miranda

Realizó

Javier Castillo

Objetivo

Definición de criterios de evaluación, lema de equipo y manejo de minutas.

Orden de la reunión

- Comunicación entre los miembros del equipo.
- Definición de criterios.
- Elección del lema del equipo.
- Manejo de minutas

Comunicación entre los miembros del equipo

Debido a que un integrante del equipo esta en la sede de Navojoa Son., y los tres integrantes restantes son parte de la sede de Nogales Son., se tomo el acuerdo de comunicarnos por medio de e-mail ó por ICQ. Si surgiera algún evento de urgencia se tendrá que realizar una conferencia telefónica entre todos los integrantes del equipo para poder solucionar la oportunidad de mejora que se este presentando en ese momento.

Definición de Criterios

Se presentaron varias ideas para los criterios de evaluación, presentándose 4 borradores, gracias a la lluvia de ideas de todos los integrantes, de las cuatro ideas surgió nuestro plan de criterio de evaluación.

Elección del lema del equipo

El lema del equipo fue presentado por el compañero Eduardo, este lema fue recibido con mucho agrado y es desde ahora en adelante nuestro estandarte.

Manejo de minutas

Javier se ofreció a tomar notas en las reuniones ya sean virtuales o presenciales con el fin de elaborar las minutas. Se hará una minuta por cada reunión que se lleve a cabo de este modo tendremos presente los acuerdos a los que llegamos como equipo. Se tomarán en cuenta los e-mails que nos enviemos para llegar a los acuerdos, de esta forma nos mantendremos comunicados.

Acuerdos y compromisos.

- Comunicación por e-mail o ICQ.
- Los criterios de evaluación son puntualidad, participación, responsabilidad y liderazgo. Se ponderarán en una escala del 10 al 7.
- Nuestro lema será "Un esfuerzo sincero en equipo construye el éxito de todos".
- Javier realiza la minuta, a más tardar el 26/Abr/2001.
- Victor sube la lista de integrantes, criterios de evaluación y la minuta de la primera parte, a más tardar el 27/Abr/2001.
- Todos verificamos correos y/o ICQ para confirmar nuestro acuerdo a las actividades o enviamos correcciones según sea el caso.
- A más tardar el 26 de abril todos habremos revisado los elementos de la 1ª etapa del proyecto.

Se programa reunión para 30/Abr/2001

Minuta # 2

Minuta de la reunión llevada a cabo el día 3 de Mayo del 2001.

Asistentes

Imelda Luzanía
Eduardo Romero
Javier Castillo
Victor Miranda

Realizó

Eduardo Romero

Objetivo

Establecer las bases para el desarrollo del plan de trabajo para el proyecto final.

Orden de la reunión

Determinar alcance del proyecto.
Confirmar el tema que se tomará como base para el desarrollo del proyecto final.
Nuevos acuerdos y compromisos.

Determinar el alcance del proyecto final

Debido a que surgieron muchas dudas acerca de lo que era el proyecto final, cada integrante del equipo expresó su punto de vista con respecto al proyecto. De aquí llegamos a la conclusión de que con "Guía" se refería a un documento que se pudiera aplicar a cualquier proyecto de tecnología, pero siempre tomando como base un tema en específico. El alcance viene siendo dado por todas aquellas actividades que estén relacionadas con la administración de proyectos de tecnología en las áreas de informática dentro de las empresas maquiladoras del área de Nogales. Aunque como la administración de proyectos es un área que se puede aplicar en diferentes proyectos o situaciones, los resultados del proyecto bien podrían aplicar a otras empresas diferentes fuera de la región.

Confirmar el tema que se tomará como base para el desarrollo del proyecto final

El tema propuesto por Víctor Miranda fué el de "Proyectos para el Desarrollo de Software", estando todos los integrantes del equipo en acuerdo debido a que es el tema en el que la mayoría de nosotros tiene mayor experiencia.

Nuevos acuerdos y compromisos

- Eduardo Romero confirmará con el asesor de la clase si la idea que tenemos del proyecto final es la correcta.
- Cada integrante del equipo deberá sugerir las actividades que debe incluir el proyecto, así como las fechas de compromiso y responsables a más tardar el Viernes 4 de Mayo del 2001.

- Se programa una nueva junta para el Viernes 4 de Mayo del 2001 con el objetivo de revisar las propuestas y sintetizar la información generada por los diferentes miembros del equipo.

Minuta # 3

Minuta de la reunión llevada a cabo el día 4 de Mayo del 2001.

Asistentes

Imelda Luzanía
Eduardo Romero
Javier Castillo
Victor Miranda

Realizó

Eduardo Romero

Objetivo

Desarrollar un borrador del plan de trabajo para el proyecto final.

Orden de la reunión

Analizar y sintetizar la información generada por los miembros del equipo.
Preparar borrador con el plan sugerido y revisarlo nuevamente.
Nuevos acuerdos y compromisos.

Analizar y sintetizar la información generada por los miembros del equipo

Se revisaron las actividades comunes sugeridas por todos los miembros del equipo y se consideraron para ser incluídas en el borrador. Se discutieron las actividades en las que algunos de nosotros no estuvimos de acuerdo y se sometió a votación si se incluían o no. De igual forma, cada uno de nosotros expresó su punto de vista con respecto a otras actividades sugeridas por los demás y que nos llamaron la atención para considerar su inclusión en el borrador.

Preparar borrador con el plan sugerido y revisarlo nuevamente

Con el resultado de la actividad anterior, se fué preparando un borrador con las principales ideas encontradas. Una vez terminado el borrador se volvió a revisar como un solo documento con el objetivo de encontrar posibles huecos o redundancias en el plan de trabajo, y corregirlos.

Nuevos acuerdos y compromisos

- Eduardo Romero entregará la autoevaluación del equipo de la primera semana a más tardar el Sábado 5 de Mayo del 2001.
- Cada integrante revisaría individualmente el borrador y enviaría cualquier comentario que pudiera surgir a Eduardo Romero.
- Eduardo Romero se comprometió a entregar el documento de la segunda etapa del proyecto (basado en el borrador previamente elaborado) y sus minutas correspondientes a más tardar el Domingo 6 de Mayo del 2001. Si hubiera cualquier comentario por parte de cualquier miembro del equipo éste también sería incluído en el documento.

Minuta # 4

Minuta de la reunión llevada a cabo el día 10 de Mayo del 2001.

Asistentes

Imelda Luzanía
Eduardo Romero
Javier Castillo
Victor Miranda

Realizó

Victor Miranda

Objetivo

Establecer la estrategia para el desarrollo de la investigación de campo para el proyecto final.

Orden de la reunión

Determinar las personas a ser entrevistadas.
Determinar las preguntas que se realizarán durante la entrevista.
Nuevos acuerdos y compromisos.

Determinar las personas a ser entrevistadas

Cada quien entrevistará a una persona que se dedique a la administración de proyectos de desarrollo de software.

Determinar las preguntas que se realizarán durante la entrevista

Cada integrante aportará las preguntas que considere que sean necesarias realizar durante la entrevista. Se seleccionarán las preguntas comunes y además las preguntas que surjan durante el análisis del cuestionario. Se estima que el tiempo requerido para realizar el cuestionario debe ser entre 45 minutos y una hora y media.

Nuevos acuerdos y compromisos

- Las preguntas propuestas deben recibirse a más tardar el Viernes 11 de Mayo.
- La confirmación para las entrevistas debe recibirse a mas tardar el Sábado 12 de Mayo.
- La próxima reunión del equipo es el Sábado 12 de Mayo a las 9:00 A.M. en la sede de la UV.

Minuta # 5

Minuta de la reunión llevada a cabo el día 12 de Mayo del 2001.

Asistentes

Imelda Luzanía
Eduardo Romero
Javier Castillo
Victor Miranda

Realizó

Victor Miranda

Objetivo

Definir el cuestionario que será aplicado a los entrevistados durante la investigación de campo.

Orden de la reunión

Análisis de las propuestas de cuestionario y síntesis del cuestionario final.
Nuevos acuerdos y compromisos.

Análisis de las propuestas de cuestionario y síntesis del cuestionario final

Se analizaron las preguntas propuestas por cada integrante del equipo y se seleccionaron las preguntas comunes y además las preguntas que surgieron durante el análisis del cuestionario. Se estimó que el tiempo requerido para realizar el cuestionario estaría entre 45 minutos y una hora y media.

Nuevos acuerdos y compromisos

- Los resultados de las entrevistas deberán entregarse a más tardar el Jueves 17 de Mayo.
- La próxima reunión del equipo es el Jueves 17 de Mayo al finalizar la clase.

Minuta # 6

Minuta de la reunión llevada a cabo el día 17 de Mayo del 2001.

Asistentes

Imelda Luzanía
Eduardo Romero
Javier Castillo
Victor Miranda

Realizó

Victor Miranda

Objetivo

Definir el marco de referencia para el análisis de las entrevistas realizadas.

Orden de la reunión

Análisis de las respuestas obtenidas durante la entrevista.
Definición del marco de referencia que se utilizará para la presentación de la investigación de campo.
Nuevos acuerdos y compromisos.

Análisis de las respuestas obtenidas durante la entrevista

Se analizaron las respuestas obtenidas por cada una de las entrevistas realizadas, tratando de determinar que aspectos eran comunes y cuales presentaban alguna peculiaridad.

Se hizo un análisis del perfil de los entrevistados y el ambiente en el que se desempeñaban.

Marco de referencia

En base al análisis de los perfiles de los entrevistados y de las similitudes y diferencias encontradas en sus respuestas a la entrevista, se determinó un marco de referencia para la realización del análisis final de las entrevistas, el cual se basa en 5 aspectos de la administración de proyectos: inicio, planeación, ejecución, control y conclusión.

Nuevos acuerdos y compromisos

- Cada quien realizará un análisis de las entrevistas en base al marco de referencia establecido.
- Los resultados del análisis deberán entregarse a más tardar el Viernes 18 de Mayo.
- La próxima reunión del equipo es el Viernes 18 de Mayo a las 5:30 P.M. en la sede de la UV.

Minuta # 7

Minuta de la reunión llevada a cabo el día 18 de Mayo del 2001.

Asistentes

Imelda Luzanía
Eduardo Romero
Javier Castillo
Victor Miranda

Realizó

Victor Miranda

Objetivo

Concluir el documento final de la tercera etapa del proyecto.

Orden de la reunión

Revisión de los análisis de las entrevistas usando el marco de referencia establecido.

Realización del análisis definitivo de las entrevistas en base a los análisis individuales.

Nuevos acuerdos y compromisos.

Revisión de los análisis de las entrevistas usando el marco de referencia establecido.

Se revisaron los análisis realizados por cada integrante del equipo y se destacaron los aspectos comunes encontrados en los análisis.

Realización del análisis definitivo de las entrevistas en base a los análisis individuales.

Se realizó el análisis definitivo basándonos en los aspectos comunes de los análisis individuales.

Nuevos acuerdos y compromisos

- Eduardo genera el archivo final de la tercera etapa a más tardar el Sábado 19 de Mayo.
- Victor le da presentación al documento final y se encarga de subirlo a la página del curso a más tardar el Domingo 20 de Mayo.
- Victor realiza las minutas de las reuniones y se encarga de subirlas a la página del curso a más tardar el 20 de Mayo.
- Imelda y Javier se pondrán de acuerdo para la realización de la autoevaluación de la tercera etapa del proyecto.

Minuta # 8

Minuta de la reunión llevada a cabo el día 28 de Mayo del 2001.

Asistentes

Imelda Luzanía
Eduardo Romero
Javier Castillo
Víctor Miranda

Realizó

Imelda Luzanía

Objetivo

Establecer la estrategia de la investigación bibliográfica para la cuarta etapa del proyecto.

Orden de la reunión

Determinar tipo de bibliografía a investigar.
Determinar cantidad de bibliografías por cada integrante del equipo.
Nuevos acuerdos y compromisos.

Determinar tipo de bibliografía a investigar

Se determinó que la bibliografía a investigar tiene que ver con la administración de proyectos en el desarrollo de software.

Determinar cantidad de bibliografías por cada integrante del equipo.

Se estableció que por lo mínimo tres bibliografías por integrante. Y que estas pueden ser de páginas de Internet, libros, revistas, etc.

Nuevos acuerdos y compromisos

- Se pone como límite de entrega de documentos bibliográficos para el miércoles 30 de mayo, incluyendo la fuente del documento.
- Se programa nueva reunión para el 30 de mayo. Reunión virtual.

Minuta # 9

Minuta de la reunión llevada a cabo el día 30 de Mayo del 2001.

Asistentes

Imelda Luzanía
Eduardo Romero
Javier Castillo
Víctor Miranda

Realizó

Imelda Luzanía

Objetivo

Definición de lo que se va a hacer con los documentos bibliográficos que se obtuvieron.

Orden de la reunión

Actividades a realizar de los documentos recopilados.
Nuevos acuerdos y compromisos.

Actividades a realizar de los documentos recopilados.

Se estableció que como marca la cuarta etapa del proyecto, se hará un abstract de cada documento con un máximo de 5 líneas, así como la definición de la relación de cada fuente bibliográfica con el proyecto. Así como la fuente de donde fue obtenido.

Nuevos acuerdos y compromisos

- Se establece el límite de entrega de abstracts, relación y fuente para el viernes 1 de junio.
- La próxima reunión se dispone para el 1 de junio para arreglar últimos detalles.

Minuta # 10

Minuta de la reunión llevada a cabo el día 1 de Junio del 2001.

Asistentes

Imelda Luzanía
Eduardo Romero
Javier Castillo
Víctor Miranda

Realizó

Imelda Luzanía

Objetivo

Concluir el documento final de la cuarta etapa del proyecto.

Orden de la reunión

Revisión de los documentos elaborados por cada integrante del equipo.
Integración de los abstracts para dar forma al documento final de la cuarta etapa del proyecto.
Nuevos acuerdos y compromisos.

Revisión de los documentos elaborados por cada integrante del equipo.

Se analizó si eran afines a la investigación que se pide como cuarta etapa del proyecto y si cumplían con los requisitos.

Integración de los abstracts para dar forma al documento final de la cuarta etapa del proyecto.

Se integraron todas las fuentes bibliográficas con sus respectivos abstracts y la relación que tienen con el proyecto.

Nuevos acuerdos y compromisos

- Imelda genera el documento final de la cuarta etapa del proyecto a más tardar para el sábado 2 de junio.
- Imelda le da presentación al documento final y se encarga de subirlo a la página del curso a más tardar el Domingo 20 de Mayo.
- Imelda realiza las minutas de las reuniones y se encarga de subirlas a la página del curso a más tardar el 20 de Mayo.
- Imelda y Javier se pondrán de acuerdo para la realización de la autoevaluación de la cuarta etapa del proyecto.

Minuta # 11

Minuta de la reunión llevada a cabo el día 9 de Junio del 2001.

Asistentes

Imelda Luzanía
Eduardo Romero
Javier Castillo
Victor Miranda

Realizó

Victor Miranda

Objetivo

Revisión de las etapas de investigación de campo y bibliográfica para definir la estrategia de desarrollo de la guía.

Orden de la reunión

A partir de los resultados de la revisión de las etapas de investigación de campo y bibliográfica, se definió el marco de referencia que se utilizará para el desarrollo y presentación de la guía. Se asignaron responsabilidades y fechas compromiso.

Nuevos acuerdos y compromisos

- Los puntos de la guía tendrán un enfoque interrogativo. Esto es, se pregunta por ejemplo cuales son los controles necesarios para la administración de un proyecto de desarrollo de software.
- El desarrollo de la guía se basa en cinco procesos, inicio, planeación, ejecución, control y cierre.
- Eduardo envía la tabla de contenidos y la asignación de actividades a cada uno, también desarrollará los puntos de planeación y ejecución de proyectos de la guía.
- Imelda desarrollará la introducción a la guía, desarrollará la descripción de los diferentes tipos de proyectos, desarrollará la guía para iniciar un proyecto y desarrollará la guía para la conclusión de proyectos.
- Javier entregará las gráficas de Gannt, el resumen ejecutivo, las minutas de las juntas, el análisis del esfuerzo, el análisis de la aplicación de administración de proyectos, la metodología de trabajo, el modelo organizacional y las minutas de la quinta etapa.
- Victor desarrollará la parte de control de proyectos, la forma de aplicarlo en nuestras empresas,

Minuta # 12

Minuta de la reunión llevada a cabo el día 12 de Junio del 2001.

Asistentes

Imelda Luzanía
Eduardo Romero
Javier Castillo
Victor Miranda

Realizó

Victor Miranda

Objetivo

Revisión del avance del desarrollo de la guía.

Orden de la reunión

Se revisan los avances en el desarrollo de la guía y se redistribuyen actividades.

Nuevos acuerdos y compromisos

- Eduardo apoya en el desarrollo del resumen ejecutivo y el análisis del esfuerzo aplicado.
- Imelda continúa con los puntos asignados en la reunión anterior.
- Javier entregará las gráficas de Gannt, el resumen ejecutivo, las minutas de las juntas, el análisis del esfuerzo, el análisis de la aplicación de administración de proyectos.
- Victor apoya en el desarrollo de la metodología de trabajo, el modelo organizacional y las minutas de la quinta etapa,

Minuta #13

Minuta de la reunión llevada a cabo el día 14 de Junio del 2001.

Asistentes

Imelda Luzanía
Eduardo Romero
Javier Castillo
Victor Miranda

Realizó

Victor Miranda

Objetivo

Se revisan los avances en el desarrollo de la guía y se integran en el documento final.

Orden de la reunión

Se recibieron las aportaciones de los integrantes del equipo y Eduardo las integrará en el documento final.

Nuevos acuerdos y compromisos

- Eduardo integra y sube el documento final.

Guía para la Administración de Proyectos de Desarrollo de Software

I. Introducción

Al desarrollar esta Guía para la Administración de Proyectos de Desarrollo de Software nos fijamos la siguiente misión: *"Incrementar al máximo la probabilidad de éxito de todos aquellos proyectos de desarrollo de software de cualquier empresa, a través de la implementación de ciertas prácticas orientadas a la detección oportuna de variaciones en el desempeño del proyecto y la ejecución de las acciones apropiadas para corregirlas."*

Esta misión nos llevó a identificar y describir el conjunto de actividades que son generalmente aceptadas para administrar un proyecto de desarrollo de software. El conocimiento y práctica de estas actividades son aplicables para la mayoría de los proyectos considerando su valor y utilidad ya demostrados para el fin que se tiene. Sin embargo, cada proyecto debe ser evaluado de forma independiente, ya que tanto el Administrador del Proyecto como su equipo de Trabajo deben determinar el grado de control que es apropiado (o factible) para un proyecto dado.

La administración de proyectos es la aplicación de los conocimientos, herramientas, habilidades y técnicas adquiridas para la integración, planeación, ejecución, control y cierre de las actividades de un proyecto con el fin de alcanzar o exceder las expectativas de los patrocinadores de éste. Es por ello que se analizan aspectos tales como alcance, tiempo, costo, calidad, necesidades, riesgos, etc., lo que a fin de cuentas es lo que permite llevar a cabo una planeación real de las actividades a realizar para cumplir con el proyecto en el presupuesto, tiempo y calidad programados.

Aunque los principios de administración de proyectos son los mismos dentro de cualquier área operativa, la orientación hacia "Desarrollo de Software" de esta guía pretende facilitar a los profesionales de ésta área el acceso a los conocimientos necesarios (tales como cultura, formación de equipos, liderazgo, planeación, presupuestos, riesgos y control) para llevar al éxito un proyecto dado.

II. ¿Todos los proyectos de desarrollo de software son iguales en todas partes?

1. ¿Cuáles son los tipos de proyectos de desarrollo de software?

Los proyectos se llevan a cabo en cualquier nivel de la organización y pueden requerir desde una sola persona hasta miles. Pueden necesitar menos de 100 horas o hasta 10 millones de horas para ser completado, y así como puede impactar en un solo departamento, también puede tener resultados en toda la empresa ya que podría utilizarse globalmente.

Enfocándonos específicamente al área de desarrollo de software encontramos dos situaciones muy particulares que merecen analizarse por separado antes de planear el proyecto:

- El proyecto de desarrollo de software se realiza por una área de sistemas o de tecnologías de información que da soporte interno a una empresa con un giro diferente al de desarrollo de software.
- El proyecto de desarrollo de software se realiza por una empresa cuyo giro es precisamente el desarrollo y venta de software con fines meramente comerciales

En el primer caso, el proyecto se puede volver un poco más restrictivo dado que este tipo de ambiente de desarrollo generalmente cuenta con recursos muy limitados que impactan en el posible alcance del proyecto. Esto se debe a que en muchas ocasiones los presupuestos con los que se cuenta en este tipo de empresas, se encuentran ya pre-establecidos antes de realizar el análisis del alcance del proyecto. En caso de que no se haga un buen análisis en este tipo de proyectos se corre el riesgo de quedar corto en el presupuesto y optar por alternativas que afecten la calidad del software que se desarrolla y consecuentemente su éxito.

En el segundo caso, la empresa es un poco más libre de establecer los presupuestos de acuerdo al alcance de sus proyectos. La principal razón de esto es que en este caso, la empresa depende de la innovación, creatividad, originalidad y funcionalidad del software que desarrolla y posteriormente comercializa, por lo que tienen mayor facilidad de lograr la aprobación del presupuesto para el desarrollo de sus proyectos, ya que se encuentra una liga bien definida entre una área de oportunidad o ventaja competitiva para la empresa y el desarrollo del proyecto propuesto.

III. ¿Cómo iniciamos un proyecto de desarrollo de software?

1. ¿Cuáles desarrollos debemos considerar como proyecto y cuáles como rutina de soporte?

Existe una diferencia muy marcada entre lo que es un proyecto y lo que es un programa o como comúnmente se le llama "una rutina de soporte":

Proyecto es una serie de actividades que se ejecutan temporalmente para crear un producto o un servicio único. El término "*Temporalmente*" significa que cada proyecto tiene un inicio y un fin definidos. Con "*Único*" se refiere a que el producto o servicio es distinguible entre productos y servicios similares. Ejemplos de proyectos podrían ser "El desarrollo del sistema de ventas para una compañía", "La migración de una versión a otra de un software específico", etc.. Todas estas actividades se asemejan en que tienen un inicio y un fin bien definidos, eliminando la posibilidad de que estas mismas actividades se desarrollen teniendo el mismo objetivo en mente.

Programa es una serie de actividades que se ejecutan recurrentemente para lograr ciertas condiciones que deseamos que existan. Algunos ejemplos pueden ser: "El programa de soporte a usuarios", "El programa de limpieza y respaldos del sistema", "El programa de mantenimiento preventivo a PCs", etc.. El denominador común para todos estos conjuntos de actividades es que se repiten una y otra vez y vuelven a empezar sin llegar a un final en el desarrollo de las actividades.

2. ¿Cómo debemos seleccionar a los miembros del equipo que trabajará en el proyecto?

Para tener un eficiente equipo de desarrollo, el administrador del proyecto debe tomar en cuenta que no se debe basar en la apariencia de las personas para contratarlas, sino en sus habilidades. No se deben guiar por su físico, si lucen bien o no, la vestimenta que usan, si tienen cabello corto o largo, o si va con la imagen de la empresa en cuanto a uniformidad ya que eso es un signo de inseguridad en lo que se refiere a administración.

Si ocurre lo anterior, las personas más valiosas empiezan a imaginar que no son apreciadas por su esfuerzo y sus contribuciones al trabajo, entonces es cuando surgen los problemas y el desempeño disminuye.

Para lograr una integración exitosa de los miembros del equipo del proyecto se pueden suponer los siguientes elementos que explicamos a continuación:

Recopilación de ejemplos de proyectos terminados

Este conjunto de ejemplos incluyen rutinas elaboradas en algún proyecto de software como el uso de ciclos y la ayuda que significó para dar por terminado el sistema. Así como la estructura y especificaciones que siguieron de forma que

ejemplifiquen la realidad de sus habilidades para hacer el trabajo que la empresa demanda.

Exámenes de aptitud

Estos exámenes son importantes porque buscan identificar si el prospecto tiene las habilidades para integrarse al equipo del proyecto y además, permiten analizar las aptitudes y características peculiares de cada individuo como su personalidad, habilidades interpersonales y valores. Aunque están más orientados a evaluar las tareas que la persona desempeñará después de ser contratado, también evalúan si la persona es eficiente para análisis, programación, o lo que es requerido en el puesto.

Entrevista

Es importante sugerir al prospecto que prepare una presentación oral de 10 a 15 minutos donde narre algunas experiencias de sus trabajos anteriores. Como ejemplo alguna tecnología que desarrolló y las experiencias que vivió con sus compañeros de trabajo. El candidato escoge el tema de su presentación y se establece la fecha en la que el administrador organiza una pequeña reunión, formada por las personas que ya forman parte del equipo y que podrían ser sus compañeros. Hay que hacerle notar que su nerviosismo es normal y que los demás están pasando por lo mismo, y que uno de los aspectos que quiere analizar es su habilidad para comunicarse.

Al final de la entrevista y después que el candidato haya salido, se debe analizar lo que presentó cada prospecto y ver en cual puesto encaja dentro del equipo del proyecto. Al final, el último que tiene la palabra es el administrador de proyectos.

Algunos criterios de selección son:

- ✓ Compromiso al proyecto, al equipo, a la organización, al líder.
- ✓ Habilidad para comunicarse con los demás.
- ✓ Flexibilidad
- ✓ Competencias técnicas
- ✓ Orientación: a la tarea y/o a resultados
- ✓ Adaptabilidad para tener más de un jefe
- ✓ Conocimiento de las herramientas de Administración de Proyectos
- ✓ Confianza (ambos sentidos)
- ✓ Actitud de admitir los errores
- ✓ Actitud para compartir
- ✓ Orientación al logro

3. ¿Por qué debemos de seleccionar a un Administrador de Proyectos de Desarrollo de Software para nuestros proyectos?

El Administrador de Proyectos es uno de los elementos más importantes del equipo y es responsabilidad de los directivos el contratarlo y de encontrar el

momento adecuado para hacerlo. Su selección también se debe a que se necesita de una persona con las habilidades suficientes para dirigir un grupo de personas. Una persona capaz de definirle a los integrantes el alcance y propósito del proyecto, la razón por la que fue elegida cada persona para formar parte del equipo, lo que se espera de la persona, el apoyo que se le proporcionará, y lo que se ganará por participar en el proyecto.

Teniendo un Administrador de Proyectos el o los patrocinadores se sienten más seguros de que se hará el esfuerzo por finalizar el proyecto ya que se encuentra una persona que cumple con las características necesarias para dirigir un proyecto de tal magnitud. Sin él no sería posible llegar a finalizar el proyecto de una forma satisfactoria.

4. ¿Qué características debe poseer este Administrador de Proyectos de Desarrollo de Software?

- ✓ Experiencia: Debe tener experiencia, es decir, haber trabajado antes en proyectos de desarrollo de software, ya que es muy cierta la frase que dice “una gota de experiencia es mejor que una tonelada de teoría”.
- ✓ Educación: La persona debe tener una preparación profesional y un título con el cual respalde que es profesionalista en el área que se requiere.
- ✓ Capacidad de liderazgo y “expertise” estratégico: Debe tener las habilidades para guiar a otras personas para que cumpla con las expectativas que el patrocinador demanda del proyecto. De modo que visualice el todo sin olvidar el detalle, que esté orientado a resultados, hábil para capacitar formar a sus subordinados.
- ✓ “Expertise” técnico. Que la persona sea capaz de tomar decisiones basándose en los aspectos técnicos del proyecto. Asimismo, que tenga la suficiente experiencia y conocimientos en algún área de la tecnología. Conocimiento de las técnicas, productos, métodos, tecnología, proveedores, tendencias.
- ✓ Competencias interpersonales/humanas. Que sea capaz de crear un ambiente de confianza entre él y su equipo.
- ✓ Habilidades de liderazgo: Que sepa motivar, inspirar, tutorear, escuchar activamente, dar y recibir retroalimentación, ser asertivo, es decir, entender las necesidades, preocupaciones e intereses del individuo y de las relaciones entre las personas. Así como prevenir y resolver conflictos, ser sensible y flexible.
- ✓ Habilidades administrativas – gerenciales: Que disponga de los recursos que se destinan al proyecto terminando a tiempo, dentro del presupuesto,

y bajo las restricciones que se impusieron. Capaz de dirigir emprendedores, dar y recibir reconocimiento por lo que se realiza. Comunicarse tanto hacia arriba como hacia abajo. Responsabilizarse por los errores y sus consecuencias.

5. ¿Qué tipo de liderazgo será necesario ejercer en un proyecto de desarrollo de software?

Un equipo necesita una estructura para que pueda tener resultados y para ello ocupa de un líder que tenga las habilidades suficientes para tratar con gente, delegar responsabilidades, entre otras actividades. Son diferentes los tipos de liderazgos que puede ejercer un administrador de proyectos que se explican a continuación:

Liderazgo directivo (llevar de la mano)

Se le llama Directivo cuando un equipo se encuentra en la etapa de formación. El líder debe ayudar a los integrantes del equipo a conocerse entre ellos y explicarles el rol que cada uno juega en el equipo, es decir las actividades que desarrollará como parte del proyecto. Clarificarle las metas del equipo, roles, y responsabilidades. Se debe enfocar también en formar relaciones interpersonales y no sólo ponerlos a trabajar porque es en ese momento cuando surgen los conflictos, ya que no todo es trabajo porque para poder trabajar en equipo debe existir antes un ambiente de confianza. Esto se puede lograr con una comida o fiesta.

Liderazgo de influencia

Este tipo de liderazgo debe manejarse usando la influencia y la persuasión para asegurar que los integrantes estén haciendo su trabajo. Los miembros del equipo necesitan mucha ayuda psicológica por el hecho de estarse adaptando a una nueva relación interpersonal. El líder debe actuar como para finiquitar los conflictos ya que si deja que se sigan generando puede ser muy destructivo.

Liderazgo participativo

Este tipo de liderazgo es democrático ya que motiva a que todos los integrantes del equipo aporten ideas para que el proyecto siga adelante. Esto se llega a realizar porque el equipo ya está formado, ya pasaron por las primeras etapas y se sienten unidos, tienen un sentido de identidad personal de pertenencia al grupo. Ya están muy involucrados en el trabajo por lo que el líder actúa también como apoyo, como sucede entre los mismos miembros del equipo. Ya puede compartir la toma de decisiones.

Liderazgo delegativo

A este tipo de liderazgo se le atribuye el nombre debido a que los miembros del equipo ya saben lo que deben hacer, por lo que el líder se puede enfocar a realizar otras actividades como concentrarse para analizar el progreso del equipo, planear para futuros trabajos, y algunas otras actividades. El equipo ya

obtiene resultados y los integrantes se enorgullecen de lo que hacen continuamente, lo cual los mantiene muy motivados, hasta llegar a hacer bromas amigables entre ellos mismos.

El Administrador del Proyecto debe tener en cuenta las características individuales de cada miembro de su equipo de trabajo para aplicar en cada caso el tipo de liderazgo que sea el más adecuado a éste. Por ejemplo, con un programador que es nuevo y que tiene poca experiencia se podría aplicar el liderazgo directivo para moverlo a que aprenda y conozca otras cosas; sin embargo para una persona con mucha experiencia que es autosuficiente un tipo de liderazgo delegativo sería el más adecuado.

6. ¿Qué debemos documentar en esta etapa de Inicio del Proyecto?

Descripción del problema

Antes de empezar cualquier proyecto debemos tener por escrito de una forma muy clara cuál es el problema u oportunidad que se pretende atacar con el desarrollo del proyecto, esto facilitará y permitirá enfocar mejor las actividades posteriores del proyecto.

Descripción de la funcionalidad del software

Una vez que se tiene enfocado el problema hay que hacer una propuesta de solución de una forma general de forma que sirva de orientación para en una etapa posterior definir el alcance del proyecto.

Limitaciones al proyecto

Cualquier restricción que limite al equipo de trabajo en el desarrollo del proyecto debe ser documentada. Algunas limitantes pueden ser un presupuesto pre-establecido, un número reducido de gentes en el equipo de trabajo, disposiciones contractuales, desarrollo en un lenguaje de programación en específico, etc.

Supuestos que serán considerados en el proyecto

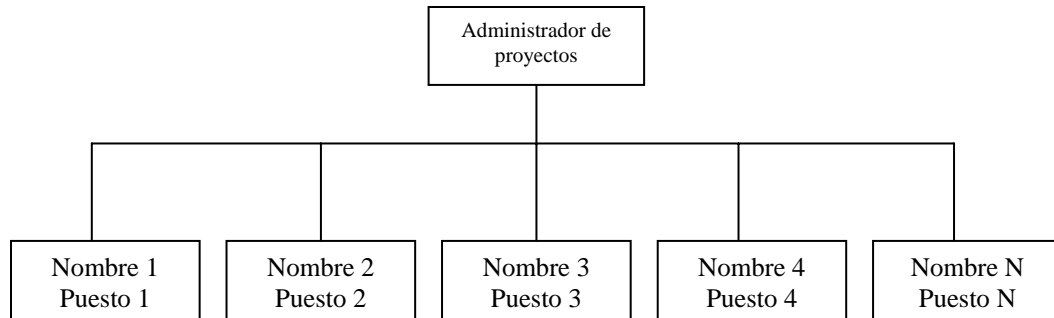
Los supuestos son todas aquellas afirmaciones que se tomarán como válidas para el proyecto en cuestión. Por ejemplo, un proyecto puede suponer que la disponibilidad de los programadores será de 30 horas a la semana, o puede suponer que el costo de un determinado recurso será tal o cual y permanecerá sin variación.

Organización del equipo de trabajo

Se debe poner por escrito la gráfica organizacional en donde se establezca quien es el líder del proyecto, así como los puestos de cada integrante del equipo. Asimismo, una lista de actividades detalladas de lo que llevará a cabo cada miembro dentro del equipo, es decir, actividades específicas, en el desarrollo del proyecto.

La gráfica tiende a ser horizontal, donde el líder o el Administrador de Proyectos se encuentra a la cabeza y sus subordinados abajo en un nivel igual para todos.

Ejemplo:



IV. ¿Cómo planeamos un proyecto de desarrollo de software?

1. ¿En qué consiste la planeación de un proyecto?

La planeación de un proyecto consiste en identificar con anticipación suficiente:

- Las tareas que debemos realizar
- Las personas que deben realizarlas
- La forma en que estas tareas serán logradas
- Las fechas en que serán logradas
- Los recursos que necesitaremos para lograrlas
- El costo en que incurriremos al desarrollarlas

El desarrollo de los puntos anteriores se documentará dentro de un plan maestro al cual se anexarán otros documentos importantes relacionados con la etapa de planeación como veremos más adelante. Por el momento solamente mencionaremos que el plan maestro formará parte de un "libro de proyecto".

El libro de proyecto deberá contener todos los documentos de relevancia para el proyecto como lo son la descripción del problema, los requerimientos, las especificaciones, los planes, las hojas de monitoreo y control, las aprobaciones, las minutas de las juntas, la documentación técnica y de usuario del software, el registro de los cambios y problemas que se presentaron durante el desarrollo del proyecto, la documentación del cierre del proyecto.

2. ¿Por qué es necesario planear las actividades a realizar en un proyecto de desarrollo de software?

Si no hacemos una planeación no podemos ejercer un control sobre un proyecto. En otras palabras, no podemos saber si estamos bien o mal si ni siquiera sabemos donde estamos ni donde deberíamos estar en este momento con respecto al proyecto.

En un proyecto de desarrollo de software muchas de las actividades pueden ser muy abstractas lo que dificulta la percepción del avance del proyecto. Si no hacemos un buen plan, solamente nos daríamos cuenta que el proyecto tuvo problemas cuando éste llegue a su fecha programada de terminación, fecha para la cual puede ser muy tarde.

Otro punto importante a tener en cuenta es que algunos software pueden llegar a ser demasiado complejos (demasiadas líneas de códigos, módulos, interfaces, procesos, bases de datos, etc.), que la mayor parte de las personas involucradas en el proyecto solamente conocen una parte del sistema que entre todos están desarrollando. Si no existe una planeación en donde exponamos la visión del todo, algunas de las actividades podrían llegar a olvidarse, lo cual podría reflejarse en retrabajos, mala calidad del software, retrasos en los tiempos de entrega, etc.

3. ¿Cómo hacemos un plan para un proyecto de desarrollo de software?

Al realizar un plan de un proyecto, corremos el riesgo de irnos a tanto detalle que llegue a ser tedioso y tardado darle seguimiento a tal plan. Pero entonces, ¿hasta que nivel de detalle debemos planear?.

En administración de proyectos existe un concepto llamado WBS (por sus siglas en inglés, Work Breakdown Structure). Un WBS consiste en quebrar un problema complejo en sub-problemas más sencillos de resolver, éstos a su vez en otros sub-problemas y así en adelante. La subdivisión de problemas continúa hasta que se considere que el subproblema ya se puede estimar con cierto grado de exactitud, o hasta que la duración del subproblema sea igual a la unidad de tiempo más pequeña en la que se desea hacer la calendarización del proyecto. Por ejemplo, si queremos calendarizar en semanas, debemos detenernos en el WBS hasta que las actividades del proyecto duren aproximadamente una semana.

El WBS nos ayuda a hacer una mejor estimación de tiempos y de costos porque entre más sencilla sea la actividad existe una mayor certidumbre en cuanto al estimado realizado. Finalmente, la suma de los tiempos de todas las actividades del proyecto nos dará el estimado total del esfuerzo necesario para llevarlo a cabo.

Ahora que sabemos hasta que nivel de detalle queremos llegar en la planeación vamos a devolvemos un poco para explicar como debemos realizar el proceso de planeación de un proyecto de desarrollo de software. *La planeación de un proyecto de este tipo debe incluir.*

- *Reuniones con los clientes del proyecto de desarrollo de software en cuestión.* Estas reuniones servirán para entender cuales son las necesidades reales y los requerimientos que ellos tienen hacia el área de desarrollo. Los analistas que participen en estas reuniones deben ser lo suficientemente capaces de entender el problema, así como de sugerir acciones alternativas a las sugeridas por el cliente de forma de minimizar el impacto en el proceso de desarrollo de software.

Las reuniones se deben de repetir hasta que quede bien claro que es lo

que el cliente está solicitando, y además las fechas límites dentro de las cuales se espera el proyecto sea implementado. En cuanto a las fechas límite, sería preferible si éstas se programaran por el equipo del proyecto, sin embargo esto muchas veces no sucede así, forzando de esta manera al equipo de desarrollo a trabajar horas extra o a traer nuevos recursos de programación.

- *Crear una misión* que nos dé las bases para fijar nuestros objetivos, así como para que también sirva de referencia en la toma de decisiones relativas al proyecto. La misión debe de contener los siguientes 3 aspectos:
 - a) Lo que estamos buscando lograr
 - b) Para quién lo estamos haciendo
 - c) Cómo pensamos lograrlo

La misión debe de ser lo suficientemente amplia para no quedarnos cortos en los resultados y lo suficientemente específica para no salirnos del contexto de lo que pensamos lograr.

La misión nos servirá para mantener orientado al equipo de trabajo en cuanto a la dirección que los proyectos deben tomar y el tipo de resultados que esperamos de ellos.

- *Establecer los objetivos del proyecto.* Uno de los principales problemas cuando establecemos objetivos en el área de desarrollo de software es que no somos muy claros en su definición. Ustedes podrán recordar objetivos como "Desarrollar un sistema para el área de materiales en la compañía" o "Integración de los sistemas de ventas con los sistemas de manufactura". Estos objetivos (si así los podemos llamar) son muy abiertos; por ejemplo: ¿Qué funciones del área de materiales se implementarán con el sistema?, ¿Para cuándo se implementará el sistema?, ¿En qué consiste la integración de ventas y manufactura?, ¿Cuáles son los sistemas que abarcan tal integración?, etc..

Un objetivo demasiado abierto no nos sirve para nuestro plan porque ocasionaría problemas para su seguimiento. Los objetivos deben ser:

- a) Específicos
- b) Medibles
- c) Realizables
- d) Realistas
- e) Delimitados en el tiempo

- *Establecer el alcance del proyecto identificando las actividades necesarias para alcanzar los objetivos.* La forma más conveniente de lograr esto es a través del desarrollo de un WBS (Work Breakdown Structure) donde contemplemos todas y cada una de las actividades que abarcará el proyecto. En este punto debemos considerar que para llegar a un WBS completo necesitaremos empezar con la etapa de diseño del sistema, de forma que podamos saber aproximadamente cuantas pantallas, reportes o interfaces necesitaremos diseñar, cuantos programas necesitaremos escribir, cuantas tablas manejaremos, cuantas relaciones, etc.
- *Identificar a las personas que conformarán nuestro equipo de trabajo.* Una vez que tenemos nuestro WBS completo conocemos que tipo de habilidades serán necesarias para realizar cada una de las actividades ahí expuestas. En base a esto, necesitamos identificar los recursos humanos que posean las habilidades requeridas. Sin embargo, esta actividad va muy de la mano con la estimación de tiempos y la calendarización de actividades, ya que debemos asegurar la disponibilidad del recurso para cuando éste sea requerido por el proyecto, no antes ni después. Esto es muy difícil de lograr porque un departamento de desarrollo generalmente procura mantener a todos sus recursos siempre ocupados en diferentes proyectos. Es por eso que en ocasiones tendremos que decidir entre disponibilidad, nivel de experiencia, nivel de conocimientos y aptitudes de la persona para realizar cierto desarrollo.

Algo muy importante a tener en cuenta es pensar con anticipación si existe algún peligro de que algún miembro del equipo de trabajo no pueda llegar a la conclusión del proyecto. Esto por diferentes razones: inestabilidad en el trabajo, vacaciones, otros proyectos de alta prioridad con grandes posibilidades de llevarse a cabo, etc.. Si encontramos algún problema de este tipo lo mejor es evitar integrar a tal persona en el equipo de trabajo.

- *Identificar los recursos de hardware, software, y cualquier otro que serán necesarios para el desarrollo del proyecto.* Este punto al igual que el punto anterior de recursos humanos va muy de la mano con la estimación de tiempos y con la calendarización de actividades porque en algunos casos será necesaria la adquisición de recursos que no están disponibles en el inicio del proyecto. Pongamos como ejemplo un desarrollo de una aplicación de Extranet que ya ha concluido. En un mundo ideal, el plan de trabajo indicaría que ya debemos empezar

el proceso de implementación en un ambiente productivo; sin embargo, en el mundo real podría ocurrir que la implementación no empezara sino hasta dentro de una semana, ya que es este tiempo el que falta para que nos entreguen el servidor que compramos.

- *Estimar los tiempos y el costo de llevar a cabo las actividades definidas.* Una vez que conocemos cada una de las partes de software que tenemos que desarrollar, es necesario calcular los costos en que incurriremos. Los costos vienen dados por los salarios de las personas que trabajan en el proyecto multiplicado por el tiempo que estarán trabajando en éste, adicionando el costo de la infraestructura u otros recursos que sean necesarios adquirir y con los cuales se espera trabaje de forma satisfactoria el software que se desarrolle. Por otro lado, al costo también debemos adicionar otros gastos como viajes, comidas, servicios, etc.

Es importante tener en cuenta que en el proceso de estimación deben participar los desarrolladores del software, ya que ellos son quizá las personas con más bases para dar un estimado más exacto de cada una de las tareas que se tienen que llevar a cabo. Otra fuente a través de la cual podemos generar estimados más exactos es usando los libros de proyectos similares. No hay nada mejor que evitar reinventar la rueda, es decir, si ya habíamos gastado un tiempo en resolver un problema similar, lo mejor es usar como base el conocimiento ya adquirido.

Otro punto importante es el NO SUPONER que los programadores dedicarán el 100% de su tiempo al desarrollo del software. Esto es un error muy común que cometen los administradores de proyectos porque no toman en cuenta que sus recursos humanos necesitan dedicar tiempo a otras actividades que también son importantes. Una sugerencia de como repartir el tiempo de la gente es:

- a) Dedicar 65% del tiempo al trabajo efectivo de proyectos
 - b) Dedicar 15% del tiempo a cuestiones administrativas
 - c) Dedicar 10% del tiempo al desarrollo de habilidades
 - d) Dedicar 10% a cuestiones sociales (recordemos que somos humanos y necesitamos relacionarnos con los demás)
- *Hacer una calendarización de actividades donde se definan actividades, responsables, tiempos, fechas límites y costos.* Una vez

que tenemos la estimación de tiempos de las actividades, y que hemos identificado los recursos que necesitamos para llevar a cabo el proyecto y su disponibilidad, es necesario armar el rompecabezas de forma que aprovechemos al máximo el tiempo y los recursos. Para lograr este propósito, algunas veces es posible programar "paralelamente" dos o más actividades independientes con diferentes recursos.

Una recomendación en este punto es realizar la calendarización utilizando 3 diferentes estimados:

1. Estimación optimista en donde todo sale como se planeó
2. Estimación pesimista en donde todo lo que pasa todo lo que pueda salir mal.
3. Estimación Satisfactoria en donde consideremos solamente algunos de los riesgos más probables a suceder.

La estimación pesimista es la que desde el punto de vista de programación deberíamos proporcionar a nuestros clientes. Si existe presión por parte del cliente para adelantar la fecha de terminación utilizaremos la estimación satisfactoria u optimista dependiendo de la urgencia del cliente por tener listo el software.

- *Identificar y resaltar la ruta crítica del plan de trabajo.* La ruta crítica viene siendo dada por la serie de actividades a realizarse de principio a fin del proyecto, dependientes cada una de éstas de su antecesora, que en suma conforman el camino más tardado para concluir el proyecto. La identificación y conocimiento de esta ruta crítica es indispensable porque cualquier retardo o anticipación en alguna de las actividades que la componen, invariablemente tendrá un efecto en la fecha de terminación del proyecto y consecuentemente en su costo.
- *Identificar los riesgos que podría enfrentar el proyecto de desarrollo de software.* Podemos atacar los riesgos utilizando diferentes enfoques:
 1. Apagar fuegos. Corregir solamente cuando aparezcan problemas
 2. Detectar y corregir problemas rápidamente cuando se presenten

3. Planear tiempo y acciones para reducir el impacto del riesgo en caso de que llegara a ocurrir
4. Prevenir los riesgos a través de la incorporación de planes de mitigación dentro del proyecto mismo.
5. Eliminación de las causas de los riesgos

De los puntos anteriores debemos de buscar siempre la eliminación de las causas de riesgo, aunque habrá situaciones que no se puedan evitar y en las cuales se podrá actuar usando alguno de los otros cuatro enfoques. Algunas de las prácticas que se utilizan para planear, minimizar y/o evitar los riesgos son:

- a) Todos aquellos riesgos que pudieran ocurrir por causas diferentes a la negligencia de alguna persona, deben ser identificados y documentados, incluyendo la probabilidad de que tal evento ocurra, sus consecuencias, su costo, la calendarización y los detalles técnicos del soporte. La ruta crítica del proyecto debe ser revisada con un cuidado especial para eliminar, dentro de lo posible, cualquier factor de riesgo que exista en las actividades que la componen. Todos aquellos riesgos que no puedan ser eliminados pasarán a formar parte de un plan de contingencias en el que se especifiquen las acciones que tendrán que llevarse a cabo, su duración, los recursos que involucrará y los costos que involucrará.
- b) Un punto crítico para eliminar riesgos de retrabajos es el acuerdo de interfases con el usuario. Estas interfases se deben planear antes del análisis y diseño a través de la creación de un prototipo que muestre la interfase gráfica del sistema y la navegación a través de él.
- c) Una gran parte de los costos de desarrollo son por defectos introducidos en la programación y no detectados. Este costo se incrementa de forma significativa cada vez que el defecto se propaga a la siguiente fase del proyecto. Por esto es necesario planear pruebas formales en cada etapa del proyecto para minimizar la posibilidad de que se filtren defectos.
- d) La planeación de tareas de corta duración que produzcan productos que se puedan medir facilita la detección de defectos en el software. Esta tarea solamente puede estar bien o mal. No se aceptan puntos intermedios.
- e) No sobrecargar de trabajo a la gente. El tiempo que pretendemos ganar con tiempos extras se paga con los defectos introducidos en el software por cansancio y frustración. Es mejor planear el tiempo adecuado para cada recurso en el proyecto respetando el tiempo personal de cada uno de ellos.

- *Identificar y dar a conocer la forma en que se comunicarán los diferentes aspectos relevantes de un proyecto.* Una de las principales características de la administración de proyectos de desarrollo de software es la falta de información con respecto al estatus en que se encuentran en un momento dado. Si llegase a ocurrir algún problema a mitad del proyecto, este puede pasar desapercibido hasta que llega la fecha de conclusión del proyecto y éste refleje el problema. Una comunicación efectiva podría ayudar a una toma de decisiones mejor y oportuna en cuanto al proyecto. Es por esto que una planeación de la comunicación se hace necesaria.

La planeación de la comunicación se puede realizar de la siguiente manera:

- a) Analizando las necesidades de información de las diferentes personas relacionadas con el proyecto y las opciones que tenemos para su distribución. Aquí se detalla como se recolectará la información que se necesita comunicar así como los responsables de su extracción y archivo, su publicación a las personas adecuadas y su constante actualización. Por otro lado también se detalla el tipo de información a ser distribuída, incluyendo formato, contenido, nivel de detalle, y convenciones a ser utilizadas.
- b) Definiendo los medios a través de los cuales se dará a conocer la información. Existen dos formas de que la gente apropiada obtenga la información que requiere: una es a través de la publicación de algún documento o exposición verbal de la información; la otra es proveyendo el acceso a las personas adecuadas para que ellos mismos consulten la información que necesitan en cualquier momento. Dependiendo de las necesidades de cada individuo se deberá decidir entre implementar una o la otra o las dos.

La comunicación tiene muchas dimensiones:

Escrita y oral.

Interna (dentro del proyecto) y externa (al cliente, los medios, el público, etc.)

Formal (a través de reportes) e informal (a través de memos y conversaciones)

Vertical (arriba y abajo dentro de la organización) y horizontal (con los compañeros de trabajo)

En un proyecto de desarrollo de software muy probablemente utilizamos todas estas dimensiones de la comunicación, sin embargo es necesario utilizarlas adecuadamente. El plan de comunicación debe de contener que

tipo de comunicación utilizar en cada caso. Por ejemplo, al encontrar un problema que cambie el alcance del proyecto quizá se opte por realizar una junta de comunicación en lugar de enviar un correo electrónico explicando el problema.

- c) Proporcionando los accesos adecuados para consultar el libro del proyecto, los documentos de aceptación de cada fase del proyecto y las lecciones aprendidas durante su desarrollo. La correcta administración de tal información servirá de base para sacar provecho de la administración de conocimientos ya adquiridos dentro de la organización.

Nota importante: Hay que tener en cuenta que los proyectos pueden ser de diferentes magnitudes, y dependiendo de ésta deberán determinarse qué aspectos son los más importantes planear. Las técnicas de administración de proyectos se pueden emplear a cualquier proyecto no importando su magnitud, pero en la vida real no es tan fácil justificar un tiempo o un costo por cosas tan sencillas. Pongamos como ejemplo la elaboración de un software de consulta de transacciones de inventario; hemos platicado con los usuarios y en base a eso se han elaborado especificaciones para el desarrollo; nos hemos reunido con el programador y hemos determinado que la duración del proyecto es una semana. ¿Valdría la pena realizar una visión/misión? ¿Valdría la pena realizar un plan de contingencias? ¿Valdría la pena realizar una planeación de riesgos? ¿Valdría la pena realizar una planeación de comunicaciones? Quizá la respuesta a todas estas preguntas sea sí, sin embargo algunos de estos planes no serían necesarios formalizarlos sino simplemente acordarlos verbalmente dada la magnitud mínima del proyecto. Cada proyecto tiene condiciones diferentes y por eso es que el equipo de trabajo deberá decidir cuales son los documentos que se deberían de formalizar en cada caso.

4. ¿Cuánto tiempo es necesario dedicar a la planeación de un proyecto?

Normalmente las empresas que desarrollan software, principalmente aquellas en donde esta actividad no es su fuente de ingresos sino que es una actividad de soporte o de generación de oportunidades, no planean lo suficiente ya que lo consideran como un costo en el que no es necesario incurrir.

Quizá el principal problema por el que se tiene esta percepción de la planeación es que no hay un resultado tangible aparente durante esta etapa. La solución podría ser trabajar más estrechamente con las diferentes personas relacionadas con el proyecto (cliente, equipo de trabajo, representantes de las áreas de negocio involucradas, etc.) manteniendo una comunicación constante con ellos, como se explica en la sección anterior

acerca de "Planeación de la Comunicación". Cuando la gente está involucrada en la planeación entiende lo importante de realizar correctamente esta etapa para lograr un éxito del proyecto.

El tiempo que una organización asigne para la planeación del proyecto no es lo importante. Lo que importa es que queden bien establecidas las bases para el control del proyecto. La estimación del tiempo de esta etapa solamente viene dado por la experiencia en los diferentes tipos de proyectos que una organización pueda tener.

5. ¿Qué herramientas podemos utilizar para soportar la etapa de planeación?

La planeación de un proyecto puede llevarse a cabo de una forma manual muy simple. Sin embargo, a medida que aumenta la complejidad de los proyectos, se vuelve muy difícil hacer los cálculos de fechas y costos teniendo en cuenta múltiples recursos con diferentes calendarios de trabajo, diferente disponibilidad y diferentes salarios, entre otros factores. Adicional a esto, si hay cambios en el alcance, se necesitan reprogramar muchas actividades. Esto resta mucho tiempo valioso al Administrador del Proyecto, el cuál podría ser utilizado en otras áreas del proyecto. Es por esto que se recomienda utilizar alguna herramienta de software para la administración de proyectos. Una de las más conocidas es MS Project, sin embargo también la mayor parte de las veces es solamente usada en forma empírica dejando mucha de la funcionalidad del software sin utilizar. Por esta razón, se recomienda tomar un curso del software antes de empezar a trabajar con él.

6. ¿Qué tipo de planes son necesarios al desarrollar un proyecto de desarrollo de software?

A manera de resumen estos son los, llamémosle, sub-planes, dentro del plan maestro del proyecto:

Plan de actividades a realizar. Involucra la secuencia de actividades a realizar, las fechas de inicio y terminación de cada tarea y los recursos asignados a cada actividad.

Plan Organizacional. Involucra definir los roles, las responsabilidades, y las relaciones jerárquicas para el reporte de resultados, así como la fuente de los recursos (interna en la organización o externa a ésta). El plan organizacional está muy relacionado con el plan de comunicación.

Plan de recursos. Involucra la identificación de los recursos físicos necesarios para el desarrollo del proyecto así como las cantidades en que se requerirán. Esto servirá de base para la estimación de costos del proyecto así como para la planeación de compras del hardware o

software necesario.

Plan de comunicaciones. Involucra la identificación de las necesidades de información y comunicación hacia las personas involucradas en el proyecto. En otras palabras, determinar quién necesita qué información, cuando la van a necesitar y de qué forma se les va a proporcionar.

Planes de calidad y de riesgos. Involucra la identificación de los tipos de actividades que se realizarán para asegurar la calidad del software que se está desarrollando, entre ellas las revisiones técnicas del software y las pruebas funcionales del mismo. El aseguramiento de la calidad está muy relacionado con la planeación de riesgos ya que en ésta se identifican los riesgos que se consideren probables de presentarse dentro del desarrollo del proyecto, y muchos de éstos son por una mala calidad en la codificación del software.

Plan de soporte después de la implementación. Idealmente este plan no debería de existir si se hace una buena planeación y eliminación de riesgos. Sin embargo siempre existirán un menor porcentaje de problemas que se pudieron filtrar hasta la implementación del proyecto. Se espera que el impacto de este tipo de problemas sea mínimo y no generen un gran costo dentro del proyecto.

V. ¿Qué aspectos debemos de considerar al ejecutar un proyecto de desarrollo de software?

1. ¿Cuáles son las principales funciones a cubrir al ejecutar un proyecto de desarrollo de software?

El resultado de la ejecución de un proyecto de desarrollo de software sirve de base para la etapa de control del proyecto. Es por eso que cualquier actividad realizada dentro de esta etapa debe ser reportada en períodos de tiempo preestablecidos. Dentro de la etapa de ejecución de un proyecto de desarrollo de software encontramos las siguientes funciones:

La administración del proyecto de software. Se refiere al seguimiento del proyecto para observar como avanza con respecto al plan e identificar desviaciones a tiempo para tomar las acciones correctivas que sean necesarias.

La comunicación del avance del proyecto y los problemas encontrados, a las personas involucradas en el proyecto. A medida que el proyecto avanza cambian las condiciones en las que se lleva a cabo como resultado del trabajo ya hecho. Estos cambios podrían llegar a ser de interés para algunas personas relacionadas al proyecto, y por lo tanto deben ser comunicados a medida que vayan sucediendo. Por ejemplo, si el trabajo de un programador depende de el de otro, es muy posible que el programador dependiente desee estar informado de la fecha de terminación del desarrollo del programador independiente para poder empezar con su trabajo.

El desarrollo del proyecto de software. A medida que avanza la programación del proyecto, los programadores deben de llevar un registro del tiempo invertido en cada etapa que sirva para alimentar a la etapa de control del proyecto. Por otro lado, deben ir documentando y comunicando los cambios hechos a módulos que sean comunes a varios programadores. Esto evitará malos entendidos, fallas en el sistema y mejorará la coordinación de los esfuerzos aplicados en el proyecto por parte de cada uno de los recursos desarrolladores.

La coordinación y ejecución de pruebas con los clientes del proyecto. Los analistas de negocios o personas que estén más cercanas al usuario final deberán de ir coordinando pruebas de los módulos cuya programación ya haya concluído. Las pruebas de los módulos de software deberán ser efectuadas inmediatamente después de que se haya terminado su programación, de esta forma el programador tendrá la oportunidad de atender el problema cuando aún lo tiene en mente, de otra forma, la resolución del problema podría tomar más tiempo.

La negociación con los clientes del proyecto de cualquier cambio en el

alcance del proyecto. A medida que se ejecuta el proyecto podrían surgir cambios en su alcance, los cuales podrían a su vez tener o no bases sólidas. El administrador del proyecto debe negociar con el cliente las alternativas que se tienen para considerar tales cambios. Si no hay ninguna alternativa, el administrador del proyecto se reúne con su equipo de trabajo para re-definir el proyecto y calcular las nuevas fechas, presupuesto y esfuerzo requerido para implementar los cambios propuestos.

2. ¿Cómo afecta la cultura al desempeño del equipo de trabajo?

Existe un consenso entre la gente que trabaja en el área de tecnologías de información acerca de que la cultura definitivamente afecta el desempeño del equipo de trabajo. Por ejemplo, cuando existen equipos de trabajo cuyos miembros provienen de diferentes países, a veces es necesario realizar cursos de sensibilización para evitar malos entendidos y roces entre ellos mismos. A medida que se extiende el fenómeno de Internet, este tipo de equipos virtuales se vuelve más frecuente por lo que el concepto de cultura es vuelve un aspecto relevante a ser considerado.

Pero además, dentro de las mismas organizaciones se van formando culturas a medida en que las acciones se convierten en hábitos. Para alguien que recién llega a una empresa con una cultura diferente a la que él o ella estaba acostumbrado(a), es difícil adaptarse al cambio. Por eso este tipo de aspectos deben de tenerse en cuenta en la planeación del recurso humano.

Por otro lado, cuando la empresa para la cual desarrollamos el software posee una cultura muy conservadora en sus procesos y metodología, implementar un nuevo sistema puede llegar a convertirse en un reto mayor que el que tuvimos para desarrollarlo. Se debe tener especial cuidado en este tipo de situaciones para identificar que personas podrían llegar a ser los catalizadores del cambio que nos ayuden en la implementación del nuevo sistema en la organización cliente.

3. ¿Qué tipo de ambiente de trabajo debo de propiciar para que mi equipo trabaje productivamente?

Existen diferentes prácticas que pueden ayudar a formar un buen ambiente de trabajo en el cual los integrantes de un equipo trabajen sinérgicamente, sin embargo, su aplicación no garantiza que esto se dé, ya que dependerá de algunos otros factores como las habilidades de los integrantes del equipo y sus actitudes. Algunas de estas prácticas son:

1. Es importante que los miembros del equipo de trabajo convivan en un ambiente diferente al trabajo. Por ejemplo, la formación de equipos deportivos o la convivencia en otros lugares ayuda a que los miembros se conozcan y se tengan confianza.

2. Debemos procurar que un mismo recurso no trabaje en demasiados proyectos a la vez. Entre menos proyectos atiende de forma paralela más oportunidad tendrá de enfocarse en el trabajo que realiza.
3. El trabajo de desarrollo de software requiere de cierto nivel de concentración, por lo que un ambiente ruidoso o con muchas distracciones en la mayoría de los casos no ayudará a que el equipo trabaje efectivamente. Es verdad que hay que dejar espacios para que los integrantes del equipo platiquen y tengan un poquito de vida social entre ellos mismos dentro del trabajo. Sin embargo por lo general cuando estas personas trabajan juntos toman los mismos tiempos para trabajar y descansar por lo que esto no representa un verdadero problema.
4. Un ambiente de trabajo burocrático en donde todo tenga que ser documentado llega a ser tedioso para un equipo de trabajo. Por otro lado, un ambiente donde no se deje ningún tipo de libertad de toma de decisiones a los integrantes del equipo tiende a bajar la moral de los individuos. Es necesario hacer procedimientos más flexibles que permitan una ágil respuesta ante situaciones que requieran soluciones de este tipo.
5. Se deben establecer sistemas de recompensas que no fomenten la competencia interna entre los mismos integrantes del equipo ya que esto propicia rivalidades que no permiten el trabajo sinérgico del equipo.
6. Se debe de evitar cargar de trabajo excesivo a los recursos de programación ya que esto solamente traerá problemas de calidad y retrasos como ya se había comentado anteriormente. El tiempo extra solamente se debe utilizar solamente como último recurso y con la menor frecuencia posible.
7. El administrador del proyecto muchas veces comete el error de comprometer fechas sin bases que lo respalden, lo que se convierte en proyectos imposibles de realizar. Es importante que el equipo de desarrollo esté involucrado en cualquier decisión relacionado a los compromisos del proyecto. Es frustrante trabajar en algo que sabemos que no podremos lograr.
8. Un ambiente de trabajo donde los miembros del equipo de trabajo se encuentren físicamente separados entorpece la comunicación entre ellos mismos e impide que se identifiquen el uno con el otro. Siempre que sea posible, deberá procurarse que los integrantes de un equipo se encuentren lo suficientemente cerca para el intercambio de ideas espontáneas, aún cuando estas no sean relacionadas al trabajo.

VI. ¿Qué alternativas tenemos para controlar un proyecto de desarrollo de software?

1. ¿Qué tipo de controles son necesarios al desarrollar un proyecto de desarrollo de software?

Los controles necesarios durante el desarrollo de un proyecto de desarrollo de software abarcan diferentes áreas del proyecto, tales como el control del alcance del proyecto, control de riesgos, control del avance del proyecto, control de la calidad, control de costos y control de abastecimientos. Los controles se deben poner en operación desde el inicio del proyecto, y mantenerse su funcionamiento hasta la conclusión del proyecto.

Los controles deben de influir en los factores que crean los cambios para asegurarse que dichos cambios traigan algún beneficio, los controles deben detectar la ocurrencia de los cambios y manejarlos en forma adecuada.

Los controles son procesos que se ejecutan durante el desarrollo de un proyecto, y como tales requieren entradas y generan salidas de acuerdo a reglas o técnicas establecidas.

- Control del alcance del proyecto.
Tiene el objetivo de controlar los cambios en el alcance del proyecto, para lo cual es necesario implementar un sistema que defina los procedimientos bajo los cuales es válido realizar cambios al proyecto, incluyendo los documentos necesarios y los sistemas de rastreo de cambios, así como también el ciclo de aprobación que debe seguir toda propuesta de cambio. Se debe estar conciente que todos los cambios en el alcance del proyecto pueden tener un impacto ya sea en el costo o en el avance del proyecto, y considerar que el impacto proporcional al avance del proyecto.

Las entradas a este control son el WBS (Work Breakdown Structure) del proyecto, los reportes de desempeño del proyecto, el plan de manejo del alcance del proyecto y las solicitudes de cambio. Las solicitudes de cambio responden a eventos externos, errores u omisiones en la definición del alcance ya sea del proyecto o del software a desarrollar, o a cambios que se consideran que añadirán funcionalidad al software a desarrollar.

Como salida de este proceso obtendremos los cambios en el alcance del proyecto, los cuales tienen que incorporarse al WBS. Otras de las salidas de este proceso son acciones correctivas. Una salida más de este proceso es muy importante, nos referimos a las lecciones aprendidas, las cuales tienen que respaldarse con el razonamiento que explique las variaciones presentadas y las causas que originaron las requisiciones de cambio. Las lecciones aprendidas se incorporan a la base de datos histórica del proyecto.

- **Control de riesgos.**
El objetivo del proceso de control de riesgos es la ejecución del plan de manejo de riesgos, que es el conjunto de procedimientos que serán utilizados ante la ocurrencia de eventos identificados como riesgos. El plan de manejo de riesgos debe especificar quién es el responsable del manejo de las áreas de riesgo, como se identifican los riesgos, como se implementan los planes de contingencia. El plan de manejo de riesgos puede enfocarse de 3 distintas maneras, la primera es evitar los riesgos a través de la eliminación de las causas de riesgo, la segunda es la mitigación de los riesgos a través de la disminución de la probabilidad de ocurrencia de los mismos, y la tercera es la aceptación de las consecuencias de los riesgos.

Una vez presentado un evento de riesgo, se debe ejecutar el plan de manejo de riesgos para generar acciones correctivas y actualizar el plan de manejo de riesgos a través del aprendizaje obtenido.

Un aspecto muy importante para hacer funcionar un sistema de manejo de riesgos es el proceso de identificación de riesgos, en el cual se deben identificar los factores externos de riesgo y los factores internos de riesgo, en general no se podrá tener control sobre los factores externos, sin embargo es posible tener un grado de control sobre los factores internos. Los factores de riesgo deben ser cuantificados, lo cual significa que se tiene que estimar la probabilidad de ocurrencia y la severidad del riesgo, y con esos datos es posible hacer simulaciones para tratar de predecir que tanto riesgo puede tolerar nuestro proyecto.

Para el control de riesgos se podría utilizar un formato como el siguiente:

Elemento de riesgo	Modo de ocurrencia	Probabilidad de ocurrencia	Severidad (Impacto en el Proyecto)	Peso	Manejo del riesgo
Programadores	Bajo desempeño	2	8	16	Buena selección, capacitación y motivación
	Ausentismo	2	8	16	Supervisión
	Renuncia	1	10	10	Candidatos alternos
Red	Baja velocidad				
	Pérdida de conexión				
Servidores					
Software					

Bajo la columna de “Elemento de riesgo” se ponen todos los elementos que uno considere que puedan fallar para el desarrollo del proyecto. Bajo la columna de “Modo de ocurrencia” se ponen las formas en las cuales pueden fallar los elementos de riesgo. Bajo la columna “Probabilidad de ocurrencia” se pone un número entre 1 y 5 para ponderar qué tan probable es que ocurra la falla. Bajo la columna “Severidad” se pone un número entre 1 y 10 para indicar qué tanto impactaría al proyecto la ocurrencia de la falla. Bajo la columna “Peso” se pone el producto de la Probabilidad y la Severidad. Bajo la columna “Manejo de riesgo” se ponen los planes para evitar o mitigar el riesgo.

- **Control del avance del proyecto.**
Para controlar el avance de un proyecto es necesario monitorear el avance real y compararlo con el avance programado, en caso de encontrar variaciones buscar una explicación para las mismas y generar acciones correctivas que eviten que las variaciones se hagan más grandes.

Una de las posibles consecuencias de las variaciones, implica el ajuste de las fechas programadas para las actividades subsecuentes. Otra posible consecuencia es la necesidad de obtener más recursos para tratar de alcanzar las fechas comprometidas para las siguientes etapas, lo cual implica un incremento en el costo del proyecto. Una vez que se ha ejecutado una acción de control para ajustar el programa del proyecto, es necesario establecer las lecciones aprendidas de tal situación, para tratar de explicar las causas de las variaciones y obtener experiencia en las etapas siguientes del proyecto.

Una condición para tener un buen control del avance del proyecto es la existencia de un buen plan de las actividades involucradas en el proyecto, su definición, su duración, secuencia y los recursos disponibles.

La parte más importante es el reconocimiento de que sin planeación no puede haber control.

Se sugiere que se utilice un formato como el que se muestra a continuación para controlar el avance del proyecto:

	Inicio			Fin			Avance %	Responsable
	Plan	Estimado	Real	Plan	Estimado	Real		
Actividad 1								
Actividad 2								

Una vez definido el plan de trabajo, las fechas de inicio y fin de cada actividad se ponen bajo la columna de “Plan”, y a la vez se pone el nombre del responsable de la ejecución de cada actividad bajo la columna de “Responsable”. El administrador del proyecto debe monitorear el avance de las actividades y actualizar las fechas de acuerdo a las siguientes reglas, si por ejemplo, una actividad se inicia de acuerdo al plan pero hay alguna razón que nos indique que la actividad no se concluirá en la fecha planeada, se deberá poner una fecha estimada de terminación bajo la columna “Estimado”, si la actividad se concluye antes de la fecha estimada, la fecha de terminación se pondrá bajo la columna “Real”. Todas las demás fechas se ajustan de acuerdo a la dependencia de actividades.

Como se puede ver el proceso involucra mucho trabajo si se lleva en forma manual. Afortunadamente existe software que puede hacer esta tarea, sin embargo, para que en realidad sea bien utilizado, el administrador del proyecto debe ingresar su plan base “Baseline” e ir ingresando las fechas estimadas y reales para obtener el efecto neto de la variación de alguna actividad.

- Control de la calidad.
El control de la calidad de un proyecto involucra el monitoreo de resultados específicos del proyecto para determinar si cumplen con los estándares de calidad establecidos.

Se controla la calidad del software a desarrollar y también la calidad de la administración del proyecto de desarrollo del software.

El control de la calidad es la ejecución del plan de manejo de la calidad, el cual debe establecer la política de calidad que se llevará en el proyecto, los estándares que deben de cumplirse y las definiciones operacionales, las cuales establecen qué es cada cosa y como debe medirse. A estas definiciones operacionales también se les conoce como métricas.

Las políticas modernas de calidad establecen que la calidad no se inspecciona, sino que se produce.

Las técnicas usadas para el control de la calidad abarcan los análisis costo/beneficio, diagramas de Ishikawa, el diseño de experimentos, checklists, inspecciones, gráficas de control, diagramas de pareto y muestreo estadístico entre otras.

A través del control de la calidad se generan mejoras de la calidad, decisiones de aceptación, retrabajos, checklists completados y ajustes en los procesos de desarrollo de software.

- **Control de costos.**
El objetivo del control de costos es mantener el costo del desarrollo del proyecto de software dentro del presupuesto asignado. Las actividades que deben realizarse para controlar el costo involucran el monitoreo de los costos generados para detectar variaciones del presupuesto, asegurarse que todos los cambios apropiados sean registrados fielmente en el costo de partida, prevenir que se incluyan cambios no autorizados en el costo de partida, informar de cambios sin autorización. El control de costos debe encontrar el porque de las variaciones, tanto positivas como negativas.

Para tener control sobre los costos es indispensable contar con un plan de manejo de costos, el cual debe describir como manejar las variaciones de los costos, este plan se genera a partir de la estimación de costos, la cual se basa en la planeación de los recursos necesarios. La asignación de los presupuestos también se basa en la estimación de los costos.

- **Control de abastecimientos.**
El objetivo del control de los abastecimientos es garantizar que los recursos estén disponibles en cantidad, calidad y en el momento necesarios.

Para garantizar la disponibilidad de los recursos humanos es necesario desarrollar tablas de roles y asignaciones que especifiquen en qué momento se requieren tales recursos.

Para garantizar la disponibilidad de recursos materiales es necesario desarrollar un plan de control de abastecimientos, el cual debe especificar cuales son las políticas de selección de proveedores, las políticas de aceptación de los materiales y las políticas de pagos.

El control de los abastecimientos materiales es la ejecución del plan de abastecimientos.

2. **¿Qué herramientas utilizamos para llevar el control de un proyecto de desarrollo de software?**

Las herramientas para el control de un proyecto de desarrollo de software pueden ser procesos manuales para proyectos pequeños, pero cuando los proyectos son de una magnitud y complejidad considerable es necesario utilizar herramientas integrales de software que nos permitan controlar las variables descritas anteriormente. Las herramientas más conocidas son MS Project y Primavera. Cabe decir que a pesar de lo poderoso que son estas

herramientas, muchos administradores de proyectos no las utilizan en un 100%.

VII. ¿Cuándo y cómo debemos de concluir un proyecto?

1. ¿Qué criterios debemos usar para dar por terminado un proyecto?

Normalmente un proyecto termina cuando se han concluido todas las actividades planeadas para llevarlo a cabo. Sin embargo un proyecto también puede terminar por otras causas antes de llegar a completarse. De acuerdo a una encuesta realizada a ejecutivos de TI por Standish Group en 1995, estas son las 10 principales causas por las que un proyecto puede terminar, digámoslo así, de una forma anormal:

- Requerimientos incompletos
- Falta de involucramiento del usuario
- Falta de recursos (gente, equipo, dinero, etc.)
- Expectativas no realistas
- Falta de soporte del ejecutivo
- Cambios en los requerimientos
- Falta de planeación
- Ausencia de la necesidad (Ya no existe el problema)
- Falta de administración de TI
- Desconocimiento de la tecnología

Un proyecto que no llega a completarse puede llegar a considerarse como un fracaso, sin embargo, como se puede llegar a observar en la lista de causas anteriormente listadas algunas de éstas podrían aplicar igualmente a proyectos bien administrados y a proyectos pobremente administrados. En nuestra opinión, se deben analizar detenidamente las causas por las que se termina un proyecto para saber si debemos considerarlo como un fracaso.

Hoy en día vivimos en un mundo de cambios muy rápidos en tecnología, mercados, y organizaciones. Aquellos proyectos de tecnología que requieren de períodos muy extendidos para su desarrollo deben tomar en cuenta estos cambios ya que, de no hacerlo, sería más factible darlos por terminado que continuar con ellos.

Por otro lado, si pensamos en un proyecto de desarrollo de software "en casa", que venga a cambiar los estilos de trabajo de la gente, podríamos compararlo con la introducción de un nuevo producto que necesita mercadearse y venderse. No podemos esperar que este tipo de proyectos tengan éxito el 100% de las veces ya que existe una gran incertidumbre debido a la independencia de otros factores que no podemos controlar y que pueden tener un efecto negativo en el proyecto.

2. ¿Qué documentos acompañan la conclusión de un proyecto?

Toda la documentación relacionada al desempeño del proyecto

Incluye todos los documentos generados durante las etapas de integración,

planeación, ejecución y control del proyecto.

La documentación técnica y de usuario del software

Incluye toda la documentación generada durante la fase de análisis y diseño del software, la programación del software, y su manual de operación.

Documentación relevante relacionada con el proyecto

Incluye memos, reportes, correos, comunicados, y cualquier otro tipo de documento que haya tenido relevancia en el proyecto, pero que en un inicio no se haya planeado su desarrollo. Es decir, son todos los documentos que se generan de forma informal a medida que avanza el proyecto pero que pudieron haber tenido relevancia en la toma de decisiones. Se recomienda organizar lógicamente todos estos documentos de forma que se facilite su comprensión cronológica durante el proyecto.

Aprobación formal de la implementación del proyecto

Puede ser una carta firmada por el (los) cliente(s) del proyecto donde se especifique claramente el alcance del software y su aceptación de que tal software ha cumplido satisfactoriamente con las pruebas necesarias para que pueda ser implementado en un ambiente productivo.

Lecciones Aprendidas durante el desarrollo del proyecto

El administrador del proyecto, durante el desarrollo de éste tendrá que enfrentarse ante diferentes problemas en los cuales tendrá que tomar decisiones de gran importancia. Algunas veces tomará la decisión correcta, pero habrá veces en las que la decisión tomada nos lleve a resultados no satisfactorios. Sin embargo, toda esta historia de acciones se puede aprovechar a través de una base de conocimientos en donde se almacenen problemas, acciones y consecuencias, con el fin de aprovechar todo este conocimiento para evitar cometer el mismo error en el futuro. Es importante tener en cuenta para este punto los indicadores de éxito de un proyecto para que cualquier fracaso pueda ser analizado y sus causas puedan ser determinadas.

3. ¿Cómo podemos medir el éxito de un proyecto?

El éxito o fracaso de un proyecto se puede medir en las siguientes dimensiones:

Costo: El proyecto es exitoso cuando al concluir no rebasa el presupuesto que se había planeado para éste.

Tiempo: El proyecto es exitoso cuando termina en el tiempo que se había establecido en el plan del proyecto o antes.

Alcance: El proyecto es exitoso cuando tiene el nivel de profundidad y rango

requerido por el alcance establecido para el proyecto.

Calidad: El proyecto es exitoso cuando el software que se desarrolló durante la ejecución del proyecto hace lo que el cliente solicitó y de la forma en que se había solicitado.

Algunas veces algunos proyectos se consideran como exitosos con el cumplimiento de solamente algunos de estos indicadores. Sin embargo, es necesario tenerlos en cuenta todos ya que la carencia de alguno de ellos indica que algo salió mal en el proyecto y debería ser evitado en el futuro.

VIII. Y ya que sabemos como administrar un proyecto de desarrollo de software, ¿Qué debemos hacer para que nuestra empresa “nos compre” esta nueva forma de trabajar?

Lo primero que debemos hacer es seguir el consejo del Dr. Edwards Demming, involucrar a la alta gerencia. También es importante entrenar a todo el equipo en los aspectos de la administración de proyectos, buscando objetivos específicos para cada uno, por ejemplo, es importante el entrenamiento para la alta gerencia para lograr que sus expectativas sean realistas.

Lo más importante es demostrar que el enfoque de administración de proyectos es benéfico para la empresa, y la forma de lograrlo es seleccionar primero proyectos pequeños con altas probabilidades de éxito, aplicar las técnicas de administración de proyectos, mostrar los resultados y compararlos con los resultados de los proyectos que no aplican una administración.

IX. Checklist para la administración de proyectos de desarrollo de software

1. Entablar reuniones con los usuarios para definir el problema o área de oportunidad.
2. Definir claramente quienes serán nuestros clientes.
3. Definir la misión / visión del proyecto.
4. Definir los objetivos del proyecto.
5. Seleccionar a un administrador del proyecto con la experiencia adecuada.
6. Realizar un WBS (Work Breakdown structure) para definir el alcance del proyecto.
7. Establecer los criterios por los cuales debería darse por terminado un proyecto (aún cuando no se haya concluído aún).
8. Establecer los procedimientos para llevar a cabo la comunicación entre los integrantes del equipo de trabajo. Al mismo tiempo facilitar los medios para que ésta se dé.
9. Definir y comunicar a los clientes los supuestos y restricciones válidos para este proyecto.
10. Estimar el esfuerzo requerido para llevar a cabo el proyecto en cuanto a tiempos, fechas, costos, y otros recursos, utilizando las herramientas adecuadas.
11. Determinar los recursos físicos y humanos que serán necesarios comprar o contratar para que apoyen el desarrollo del proyecto. Al mismo tiempo establecer el plan de estas adquisiciones.
12. Analizar la dependencia con otros proyectos en curso y la disponibilidad de recursos que se necesiten.
13. Seleccionar a los miembros del equipo de trabajo con los conocimientos y habilidades adecuadas al proyecto.
14. Seleccionar a los miembros del equipo que harán pruebas a los desarrollos del proyecto.
15. Establecer la estructura del equipo y los procedimientos de reporte.

16. Realizar el plan maestro del proyecto que incluya todas las actividades a llevar a cabo.
17. Determinar cuales son los riesgos más probables a ocurrir y planear su solución preventiva o correctiva según sea el caso (Plan de contingencias).
18. Obtener la aprobación del ejecutivo para el desarrollo del proyecto en base a los planes realizados.
19. Asegurar la disponibilidad y operabilidad del equipo que se usará para desarrollar los sistemas de software necesarios.
20. Preparar actividades recreativas en donde los integrantes del equipo de trabajo se conozcan y convivan para lograr una sinergia en el trabajo.
21. Ejecutar la programación de los sistemas software necesarios para cumplir con los objetivos.
22. Dar seguimiento a las acciones realizadas por el equipo de desarrollo del sistema para identificar variaciones y problemas a través de formatos pre-establecidos.
23. Administrar los posibles cambios que puedan surgir durante el desarrollo del proyecto a través de la evaluación del impacto y la toma de decisiones para la ejecución de las acciones apropiadas.
24. Asegurar la calidad del software desarrollado a través de la ejecución de pruebas individuales e integrales de los programas que ya hayan sido terminados en desarrollo.
25. Comunicar los avances y/o problemas en el proyecto a las gentes involucradas en éste (stakeholders) periódicamente.
26. Realizar la documentación técnica y de usuario del software desarrollado.
27. Analizar y documentar los problemas y/o oportunidades que se presentaron durante el proyecto para tenerlos en cuenta para futuros proyectos de cierta similitud al actual.
28. Integrar la documentación de la administración del proyecto.
29. Obtener la aprobación del cliente del desarrollo de software presentando la documentación tanto del proyecto como del software.
30. Cerrar el proyecto

X. Bibliografía

Boehm, Barry, *Project Termination Doesn't Equal Project Failure*, Vol. 33 No. 9, Computer, Septiembre de 2000, pp. 94-96

Brown, Norm, Industrial-Strength, *Management Strategies*, Vol. 13 No. 4, IEEE Software, Julio de 1996, pp. 94-103

Contacto de Unión empresarial. *La administración del conocimiento*. México, D.F. Biblioteca Digital: Infolatina

Duncan, William R. A guide to the project management, body of knowledge. PMI, 1996.

DeMarco Tom, Lister, Timothy. *Peopleware*. Dorset House, 1999.

Haynes, Marion Marion E. *Administración de Proyectos*. Editorial Iberoamérica, 1992.

Islas, Maricela. *Las 10 mejores prácticas en las pruebas de software*. Red. México, D.F. Biblioteca Digital: Infolatina

Lewis, James P. *Fundamentals of project management*. AMACOM, 1997.

Maddux, Robert B. *Cómo formar equipos de trabajo*. Editorial Iberoamérica, 1994.

Mcconnell, Steve, *The Software Manager's Toolkit*, Vol. 17 No. 4, IEEE Software, Julio/Agosto de 2000, pp. 5-7