

336398
BIBLIOTECA



TECNOLOGICO 28 NOV 2013
DE MONTERREY.

**ADMINISTRACIÓN DE PROYECTOS DE SOFTWARE
APLICANDO RESTRICCIONES:
UN ENFOQUE MULTIDISCIPLINARIO
CASO DE ESTUDIO**

TESIS QUE PRESENTA

SERGIO AGUILERA MALANCHE

**MAESTRÍA EN CIENCIAS COMPUTACIONALES
MCC02, SISTEMAS INTELIGENTES**

DICIEMBRE, 2010



**TECNOLÓGICO
DE MONTERREY.®**

**ADMINISTRACIÓN DE PROYECTOS DE SOFTWARE
APLICANDO RESTRICCIONES:
UN ENFOQUE MULTIDICIPLINARIO
CASO DE ESTUDIO**

TESIS PARA OPTAR EL GRADO DE
MAESTRO EN CIENCIAS COMPUTACIONALES
PRESENTA

SERGIO AGUILERA MALANCHE

Asesor: Dr. RALF EDER

Comité de tesis: Dr. MIGUEL GONZALEZ MENDOZA
Dr. RALF EDER LANGE
Dr. RAUL ANTONIO TREJO RAMIREZ

Jurado: Dr. MIGUEL GONZALEZ MENDOZA Presidente
Dr. RAUL ANTONIO TREJO RAMIREZ Secretario
Dr. RALF EDER LANGE Vocal

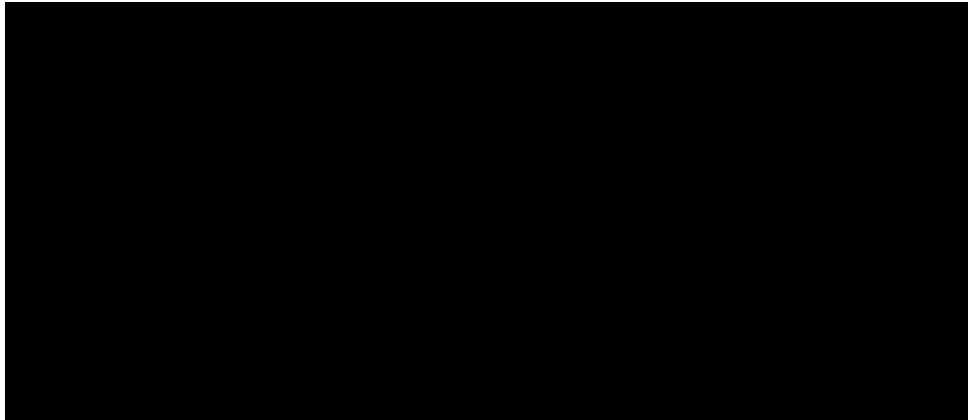
Atizapán de Zaragoza, Edo. México., DICIEMBRE de 2010.

Tabla de Contenido

AGRADECIMIENTOS	5
DEDICATORIA	5
1. INTRODUCCION	5
1.1. OBJETIVOS DE LA INVESTIGACIÓN	6
1.2. JUSTIFICACIÓN	6
1.3. TIPO DE INVESTIGACIÓN	8
1.4. MÉTODO A UTILIZAR	8
1.5. DISEÑO / PROCEDIMIENTO DETALLADO	9
1.6. UNIDAD DE ANÁLISIS / MUESTRA	11
1.7. INSTRUMENTOS A UTILIZAR	12
CAPÍTULO 2. FUNDAMENTE TEÓRICO.....	12
2.1. ANTECEDENTES	12
2.2- ADMINISTRACIÓN DE PROYECTOS CON PMI.....	16
2.3- TEORÍA DE RESTRICCIONES	17
2.4- TEORÍA DE RESTRICCIONES APLICADA A LA ADMINISTRACIÓN DE PROYECTOS DE SOFTWARE	18
2.5- ADMINISTRACIÓN DE PROYECTOS CON TSP Y PSP.....	22
2.6- COMPORTAMIENTO ORGANIZACIONAL.....	23
2.6.1- DIVERSIDAD DENTRO DE LAS ORGANIZACIONES.....	25
2.6.2 ACTITUD Y SATISFACCIÓN DE TRABAJO	27
2.6.3 EMOCIONES Y HUMOR.....	27
2.6.4 PERSONALIDAD Y VALORES	29
2.6.5 PERCEPCIÓN INDIVIDUAL Y TOMA DE DECISIONES	31
2.6.6 CONCEPTOS DE MOTIVACIÓN	32
2.6.7 APLICACIÓN DE LA MOTIVACIÓN	33
2.6.8 FUNDAMENTOS DE COMPORTAMIENTO GRUPALES	34
2.6.9 EQUIPOS DE TRABAJO.....	35
2.6.10 COMUNICACIÓN	36
2.6.11 LIDERAZGO	38
2.6.12 PODER Y POLÍTICA	39
2.6.13 CONFLICTOS NEGOCIACIÓN	41
2.6.14 CULTURA ORGANIZACIONAL.....	43

2.6.15 POLÍTICAS Y PRÁCTICAS DE RECURSOS HUMANOS	44
2.6.16 CAMBIOS ORGANIZACIONALES Y ADMINISTRACIÓN DEL ESTRÉS	45
2.7 TIPOS DE DESARROLLO DE SOFTWARE:	47
CAPÍTULO 3 CASO PRÁCTICO	50
3.1 PROPUESTA DE SOLUCION CLIENTE ANONIMO	50
3.1.1 OBJETIVO	50
3.1.2. ALCANCE.....	50
3.1.3. BENEFICIOS	56
3.1.4. DESCRIPCIÓN DEL PRODUCTO.....	56
3.1.5. PLATAFORMA TECNOLÓGICA.....	58
3.1.6. PRODUCTOS A ENTREGAR	58
3.2. METODOLOGÍA.....	61
3.2.1. ETAPAS A SEGUIR EN EL MANTENIMIENTO DESARROLLO DEL SISTEMA	61
3.2.1.1. Definición del Plan Operativo	61
3.2.1.2. Definición de Análisis y Diseño del Sistema.....	62
3.2.1.3. Diseño detallado y Construcción	62
3.2.1.4. Implantación del sistema	62
3.2.2. ESFUERZO REQUERIDO	62
3.2.3 TIEMPO REQUERIDO.....	63
3.2.4. EQUIPO DE TRABAJO	63
3.2.5. PLAN CALENDARIO	63
3.2.6. METODOLOGÍA DE PROGRAMACIÓN.....	63
3.2.7. CAPACITACIÓN Y TRANSFERENCIA DE HABILIDADES.....	63
3.3. DEPENDENCIAS Y RESPONSABILIDADES	64
3.3.1. REQUERIMIENTOS DE DESARROLLO E INSTALACIÓN.....	64
3.3.2. REQUERIMIENTOS PARA LA OPERACIÓN	64
3.4.- LECCIONES APRENDIDAS	65
3.5.- ADMINISTRACIÓN DE PROYECTOS.	70
3.6.- TEORÍA DE RESTRICCIONES.	71
CAPÍTULO 4 GENERACIÓN DE ANÁLISIS Y SÍNTESIS	71
4.1 ADMINISTRACIÓN DE PROYECTOS.....	71
4.2 ADMINISTRACIÓN DE PROYECTOS CON TSP Y PSP.....	72
4.3 ADMINISTRACIÓN DE PROYECTOS CON TEORÍA DE RESTRICCIONES (TOC).....	73
4.4 COMPORTAMIENTO ORGANIZACIONAL (OB).....	75

CAPÍTULO 6 TRABAJO FUTURO	86
CAPÍTULO 7 REFERENCIAS.....	86
7.1 REFERENCIAS	86
7.2 GLOSARIO DE TÉRMINOS:	87



1. INTRODUCCION

Una de las formas de administrar proyectos de software es mediante el modelo que propone el PMI (Project Management Institute) que con sus nueve áreas de conocimiento y con sus cinco grupos de ciclo de vida de un proyecto, ha intentado que la administración de un proyecto de cualquier industria sea controlada, sin embargo a mi juicio y por mi experiencia en la industria de software en donde los proyectos no se terminan en tiempo, se salen del presupuesto planeado, se salen de las especificaciones acordadas y fuera del alcance original, los proyectos se salen de control. La mayoría de las veces es porque existen restricciones que no fueron contempladas y que el modelo del PMI no especifica. Todas las restricciones las canaliza hacia la parte de riesgos, sin un tratamiento adecuado, por lo que algunos proyectos siguen fallando y en específico en lo que refiere a la industria de software.

Por otro lado dentro de la industria de manufactura existe un modelo que se llama Teoría de Restricciones (TOC- Theory of Constraints) que se utiliza como una herramienta de ayuda en los procesos de Manufactura.

Algunos autores han tratado de hacer una combinación entre la Teoría de Restricciones y la Administración de Proyectos y los resultados no han sido del todo satisfactorios.

A mi juicio el pobre resultado de los pocos o muchos proyectos insatisfactorios en tiempo, presupuesto, esfuerzo y con las especificaciones acordadas es porque no se ha profundizado en el tema de cómo la teoría de restricciones puede ayudar en la administración de proyectos de software. Algunos autores han contribuido a solo Administración de Proyectos mediante Teoría de Restricciones; sin embargo, ninguno ha contribuido en mezclar los modelos (TOC Y PMI) de tal forma que ayude a la Administración de proyectos de Software.

Es por ello que revisé a profundidad dichos modelos, con lo que se propone una metodología o actividades a seguir dentro del modelo del PMI, de tal forma que permita a los proyectos realizarse en tiempo.

El PMI tarde o temprano visualizará que la teoría de restricciones es necesaria incrustarla en el modelo de PMI. Es por ello que propondré alguna forma dentro de mi tesis de como incrustar la teoría de restricciones en el modelo PMI de tal forma que ayude a mejorar la Administración de Proyectos de software.

Para ello también abordé con profundidad otros modelos y teorías como el de PSP, TSP y el Comportamiento Organizacional que ayudan en el desarrollo de Administración de proyectos de Software. Sin embargo nuestro mayor enfoque es el de ver porque siguen fallando los proyectos de software cuando se sigue un modelo como el de PMI.

La solución de propuesta se basa en como las restricciones pueden ser aplicadas a las nueve áreas de conocimiento en y cómo explotar ambos modelos.

1.1. Objetivos de la investigación

El objetivo de la presente investigación es la de proponer un enfoque multidisciplinario que junto con el modelo de teoría de Restricciones (TOC- Theory of Constraints) y el Comportamiento Organizacional nos ayude a tener mejores herramientas para la Administración de Proyectos de Software mediante una adecuación al modelo que propone el PMI (Project Management Institute) para Administración de Proyectos utilizando puntos relevantes del modelo de Teoría de Restricciones, del Comportamiento Organizacional para mejorar el desarrollo y eficientar la Administración de Proyectos de Software.

1.2. Justificación

Actualmente uno de los mayores problemas a los que se enfrenta la industria de Software es a la Administración de Proyectos y una de las mejores prácticas que existen en el mercado para Administración de proyectos es la del PMI (Project Management Institute) que contiene cinco grupos del ciclo de vida de un proyecto que son: (IPECC) [1]

- 1.- Inicio
- 2.- Planeación
- 3.- Ejecución
- 4.- Control y Seguimiento
- 5.- Cierre

Adicionalmente contiene nueve áreas de conocimiento: (IATCCRCRC)

- 1.- Integración
- 2.- Alcance
- 3.- Tiempo
- 4.- Costo
- 5.- Calidad
- 6.- Recursos Humanos
- 7.- Comunicación
- 8.- Riesgos
- 9.- Compras.

Pareciera ser que se contemplase todo lo necesario para administrar un proyecto en cualquier industria, sin embargo aplicado a la industria de Software, nos encontramos ante problemas o variables que no han sido correctamente enfocados como son:

- ✓ Certeza de terminar el proyecto en el tiempo acordado.
- ✓ Certeza de terminar el proyecto con el presupuesto acordado.
- ✓ Certeza de terminar el proyecto con las especificaciones acordadas.

Dentro de esta investigación estudié cómo ayudar a resolver y minimizar los problemas antes mencionados. Lo hice dentro de un marco de referencia de la Teoría de Restricciones (TOC) y la del Comportamiento Organizacional (Organizational Behavior OB), las cuales son ampliamente utilizadas dentro de la industria de la manufactura.

Los elementos a las que me he referido anteriormente pueden fluctuar o variar de acuerdo al esfuerzo de los recursos empleados y en específico cuando se utilizan en la administración de proyectos de software. Normalmente dichos cambios o fluctuaciones son en contra de los proyectos, lo que ha derivado en desviaciones.

Dentro del ciclo de vida clásico de Ingeniería de Software para el desarrollo de un proyecto de Software se encuentra: (IADCPIP)

- ✓ Ingeniería y Análisis del sistema (Requerimientos Globales de sistema)

- ✓ Análisis de requerimientos del Software
- ✓ Diseño del Software
- ✓ Codificación del Software
- ✓ Pruebas del Sistema
- ✓ Instalación del Software
- ✓ Puesta en Marcha.

Este ciclo seguirá siendo necesario dentro de la definición de la Estructura detallada de trabajo (EDT) en la etapa de planeación dentro de la administración de un proyecto de software. La parte de la investigación la enfoque en el análisis del modelo de Teoría de Restricciones (TOC) incrustado en la administración de proyectos de software, debido a que contempla variables que pueden ayudar a minimizar los riesgos que las mejores prácticas del PMI no contemplan. Cabe mencionar, que así como la administración de proyectos es una herramienta que es de gran ayuda a la industria del software, también puede ser aplicada a cualquier otra, pues como tal, de lo que se encarga es de establecer los parámetros necesarios que permitan disminuir los riesgos dentro de un proyecto considerando las diversas variables que se puedan presentar.

1.3. Tipo de Investigación.

En esta tesis se presenta un estudio no experimental transeccional correlacional causal con obtención de datos e investigaciones que ya existen. El propósito de este diseño correlacional causal [3] es el investigar tres variables: la Teoría de Restricciones aplicada a la manufactura, la administración de proyectos del modelo del Project Management Institute (PMI) y por último el Comportamiento Organizacional. Asimismo, me encargue de revisar cómo traslapar los elementos que cada uno de estos presentan, de tal forma que la investigación contribuya a mejorar la administración de proyectos de software mediante teoría de restricciones (TOC).

1.4. Método a Utilizar.

Me basé en la utilización de estudios históricos, literatura sobre administración de proyectos, teoría de restricciones, comportamiento organizacional y algunas experiencias propias dentro de la administración de proyectos de Software en un proyecto concreto ya realizado.

1.5. Diseño / Procedimiento detallado

El diseño y procedimiento que utilicé fue un método cualitativo donde estudié el modelo de Teoría de Restricciones (TOC) mediante estudios históricos y literatura aplicado a la manufactura, también utilicé el PMBOK y el Comportamiento Organizacional adicionalmente y de manera intuitiva con mi experiencia la administración de proyectos de software propongo un modelo que minimice las desviaciones que contribuya a mejorar la administración de proyectos de software.

El procedimiento a seguir fue:

Paso 1: Revisión del ciclo de vida clásico de Ingeniería de Software.

Dentro del ciclo de vida clásico de Ingeniería de Software para el desarrollo de un proyecto de Software se encuentra: (IADCPIP)

Paso 2: Revisión de las disciplinas de la Ingeniería de Software de TSP y PSP.

TSP (Team Software Process) .- Es una disciplina dentro de la ingeniería de software que establece grupos de ingeniería para ser utilizados en aplicaciones y conceptos integrados para el desarrollo de software. El TSP se compone de:

- ✓ Establecimiento de metas
- ✓ Definición de los roles y funciones del equipo
- ✓ Asignación y evaluación de los riesgos
- ✓ Elaboración de un plan de equipo.

TSP ayuda a la organización a establecer una práctica de ingeniería madura y disciplinada, que produce seguridad y confiabilidad en el resultado.

El uso de TSP dentro de una organización puede construir equipos auto dirigidos hacia la planeación y el seguimiento de su trabajo, estableciendo objetivos, procesos y planes. Estos equipos de software pueden ser equipos integrados de 3 a 20 ingenieros.

PSP (Personal Software Process).- Es una disciplina dentro de la ingeniería de software que establece procesos definidos para recoger mediciones y estimaciones detalladas en el tiempo para producir un producto. Dichas mediciones e indicadores se analizan con métodos estadísticos que permiten producir estimaciones muy precisas, basadas en:

- ✓ Datos históricos
- ✓ Seguimiento de los procesos
- ✓ Calidad de un proyecto en curso

- ✓ Predicción de los efectos del calendario
- ✓ Predicción de la calidad de un producto de software terminado.
- ✓ Reducción de los defectos de los productos.

Paso 3: Estudié el modelo de TOC aplicado a la Manufactura.

La Teoría de restricciones se resume en cinco pasos

1. Identificar la restricción. .- identificar la restricción significa la habilidad de identificar el problema dispersar el problema en pequeños problemas y priorizar el impacto.
2. Decidir cómo explotar la restricción.- una vez identificada la restricción y priorizada se clasifica para decidir cómo descomponerla.
3. Subordinar todo acerca de la decisión. Una vez tomada la decisión de descomponerla se subordina para ver cómo se atacará la decisión.
4. Elevar la restricción.- verbalizar un poco mejor la restricción para elevar el grado de análisis y no volver a caer en ella y se toma la solución. En este punto se utiliza el método llamado efecto-causa –efecto. Dentro de este punto existen tres estados, llamados
 - a. Clasificación
 - b. Correlación
 - c. Efecto-causa-efecto.

Clasificación.- Se revisan las características específicas de la restricción para navegar sobre ésta y reconocer el tipo de restricción

Correlación.- son las turbulencias por las cuales esta restricción existe, puede estar relacionado con otras variables basada en una cuidadosa observación con otros objetos inherentes a la restricción. Dentro de la observación se sustituye la pregunta de por qué (why) por la de cómo (how).

Efecto-Causa-Efecto.- Una vez encontrados todos los objetos del estado de la correlación se utiliza la lógica de deducción o explicación reconociendo los objetos del tema.

5. Si el paso número cuatro ha sido roto y se vuelve a caer en la restricción lo que se debe hacer es volver al paso uno, para volver a identificar la restricción con otro grado de clasificación e identificar las nuevas habilidades para identificar el problema.

Por naturaleza dentro de los tipos de restricciones que existen en la manufactura hay tres tipos de restricciones:

- ✓ Restricciones en las políticas
- ✓ Restricciones en los recursos
- ✓ Restricciones en las materiales

Paso 4.- Revisé el Modelo de Administración de Proyectos del PMI (Project Management Institute).

Administración de proyectos del PMI (Project Management Institute) se resumen en cinco grupos del ciclo de vida de un proyecto que son: (IPECC) y adicionalmente con nueve áreas de conocimiento: (IATCCRRC).

Paso 5.- Revisé experiencias de un proyecto de desarrollo Software de Administración Recursos aplicando el Modelo de PMI.

Paso 6.- Revisé experiencias de un proyectos de desarrollados software aplicando algunos puntos de Teoría de Restricciones.

Paso 7.- Revisé todo lo relacionado con el Comportamiento Organizacional

Paso 8.- Elaboré Conclusiones

1.6. Unidad de Análisis / Muestra.

La unidad de muestra que analicé y comprende tres elementos:

- ✓ Eventos: Resultados de eventos pasados como desarrollos de software.
- ✓ Fenómenos: Revisé la siguiente literatura y bibliografía probada como:
 - El modelo de Administración de Proyectos por modelos de PMI
 - El modelo de TOC
 - Comportamiento Organizacional.
 - Disciplinas de Ingeniería de Software. TSP y PSP
- ✓ Personas: Entrevistas a personas que han estado en desarrollos de software con el modelo de administración de proyectos del PMI.

El tamaño de la muestra para cada unidad fue la siguiente:

- ✓ Eventos: Se restringirá a un proyecto de más de 12,000 horas/hombre donde se haya implemento Administración de Proyectos mediante el modelo de PMI con la finalidad de revisar sus fallas.
- ✓ Fenómenos: Bibliografía para la comprensión de los modelos de PMI y el modelo de TOC, así como las disciplinas de ingeniera de software TSP y PSP, y el Comportamiento Organizacional.
- ✓ Personas: Se entrevistó a las persona en los diferentes de roles de un proyecto.

1.7. Instrumentos a utilizar.

Los instrumentos utilizados fueron:

1.-Literatura:

¿Qué?

Recolección acerca de la siguiente información:

- ✓ El modelo de Administración y proyectos del PMI
- ✓ El modelo de Teoría de Restricciones.
- ✓ Disciplinas de Ingeniería de Software
- ✓ Comportamiento Organizacional

¿Cómo?

Análisis de las mismas.

El modelo de Administración y proyectos del PMI

El modelo de Teoría de Restricciones.

Disciplinas de Ingeniería de Software

El Comportamiento Organizacional

¿Dónde?

En referencias bibliográficas y últimos escritos sobre el tema.

2.-Entrevistas:

¿Quién?

A las personas relacionadas con un proyecto de administración de proyectos

¿Cómo?

Mediante un cuestionario donde revisé el porqué de las desviaciones del proyecto.

Máximo de 10 preguntas.

¿Dónde?

En las oficinas de Sinersys ubicadas en: Poniente 140 No. 839 OF. 305 Col. Industrial Vallejo C.P. 02300 México, D.F. Delegación Azcapotzalco.

Capítulo 2. FUNDAMENTO TEÓRICO.

2.1. Antecedentes

El modelo de Teoría de Restricciones fue conceptualizado en los ochentas (80) a partir de los programas de Optimización de la Producción (OPT) en las fábricas de manufactura e industria automotriz. Está basada en el simple hecho de que los procesos multitarea, de cualquier ámbito, solo se mueven a la velocidad del paso más lento. La manera de acelerar el proceso es utilizar un catalizador del paso más lento y lograr que trabaje hasta el límite de su capacidad para acelerar el proceso completo de la manera más eficiente y rápida como sea posible. La teoría enfatiza los hallazgos y apoyos del principal factor limitante. En la descripción de esta teoría estos factores limitantes se denominan restricciones o "cuellos de botella".

Existen dos fenómenos que se encuentran presentes en las plantas de fabricación que se llaman eventos dependientes y fluctuaciones estadísticas. Como medida para ayudar a que los eventos dependientes y las fluctuaciones estadísticas de la fábrica no afecten el desempeño balanceado de una planta, se establece el concepto de "amortiguador". Este amortiguador protegerá el throughput (velocidad a la que el sistema genera dinero a través de las ventas) de la planta de cualquier perturbación que se produzca en los factores denominados cuellos de botella. Este amortiguador o buffer puede ser un intervalo de tiempo expresado en horas, días o meses que sirve como protección contra las perturbaciones desconocidas que tarde o temprano ocurrirán y finalmente, para asegurarse que el inventario no crezca más allá del nivel dictado por el amortiguador, deberá limitarse la velocidad a la cual se liberan materiales de una planta de producción. El amortiguador predeterminado dentro de las líneas de manufactura es el espacio en una cinta transportadora que existe entre dos operaciones. Este amortiguador le dice al operario cuando trabajar y cuando deje de trabajar, este enfoque hace que el flujo de trabajo este sincronizado de manera que el stock sea bastante bajo en comparación con un modo tradicional de operar. Los amortiguadores de stock regulan el ritmo de la producción en las líneas de producción, los amortiguadores son indicadores que le permiten a la línea avanzar de una manera sincronizada. Hay que tener mucho cuidado porque cualquier perturbación hará que el sistema entero se detenga. Deberá de amarrarse como una cuerda desde el cuello de botella hasta la primera operación; en otras palabras la velocidad a la cual se liberarán materiales a la planta será gobernada por la velocidad a la cual está produciendo el cuello de botella. Esto finalmente también debe y puede ser aplicado dentro de una Fábrica de desarrollo de Software.

La acumulación de fluctuaciones son las que pueden afectar las variables tiempo, presupuesto y especificaciones como pueden ser la calidad del desarrollo de los programas, el ausentismo del personal, la rotura de máquinas, el corte de energía eléctrica, el faltante de materia prima que aplicado para el desarrollo de software como ejemplo puede ser la falta de especificaciones bien definidas, la capacidad de trabajo mal balanceado, la rotura de equipos, el ausentismo del personal, el corte de energía eléctrica, etc..

Administración de Proyectos.- Es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer los requisitos del proyecto que se logra mediante la aplicación e integración de los procesos de inicio, planeación, ejecución, seguimiento, control, y cierre.

Dentro de la administración de proyectos de software la dirección de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para satisfacer los requisitos del mismo. La dirección de proyectos se logra mediante la ejecución de procesos, usando conocimientos, habilidades, herramientas y técnicas de dirección de proyectos que reciben entradas y generan salidas.

Según la Administración de proyectos del PMBOK para que un proyecto tenga éxito, el equipo del proyecto debe:

- ✓ Seleccionar los procesos apropiados dentro de los Grupos de Procesos de la Dirección de Proyectos (también conocidos como Grupos de Procesos) que sean necesarios para cumplir con los objetivos del proyecto.
- ✓ Usar un enfoque definido para adaptar las especificaciones del producto y los planes de tal forma que se puedan cumplir los requisitos del proyecto y del producto.
- ✓ Cumplir con los requisitos para satisfacer las necesidades, deseos y expectativas de los interesados.
- ✓ Equilibrar las demandas concurrentes de alcance, tiempo, costos, calidad, recursos y riesgos para producir un producto de calidad.

Esto no significa que el conocimiento, las habilidades y los procesos deban aplicarse siempre de manera uniforme en todos los proyectos. El director del proyecto, en colaboración con el equipo del proyecto, siempre es responsable de determinar qué procesos son apropiados, y el grado de rigor apropiado para cada proceso, para cualquier proyecto dado.

En TOC uno de los métodos de decisión cuando se utilizan buffers o amortiguadores muy holgados o un buffer estrecho de tiempo es llamado método de aceleración o zona de aceleramiento, en donde en la administración de proyecto de software se mete a revisar el detalle del avance forzando a un aceleramiento del proceso donde la protección o amortiguador del Ingeniero desarrollador pasa a un total control del administrador del desarrollo del Software, de tal forma que los trabajos llegarán a su objetivo en el tiempo necesario. Deberá ir cambiando la planeación del proyecto minimizando los tiempos de los buffers, a lo cual TOC llama "aceleración planificada." Este método consiste primordialmente en conseguir que el trabajo llegue antes de su tiempo inicial planeado.

Las acciones que se llevan a cabo para acelerar un trabajo ("aceleración planificada") son:

Como primera acción es determinar qué trabajos se supone que están en el origen del buffer, identificando donde se ha atascado el trabajo que se ha retrasado.

Como segunda acción, una vez encontrado el atraso del trabajo actuamos y tomamos las acciones que correspondan para que éste trabajo pueda seguir adelante. En un ejemplo práctico dentro de la parte de un desarrollo de software simplemente un programador se atora o se detiene por una cuestión técnica por falta de experiencia, la intervención del administrador de proyectos es necesaria para ver como se le apoya con un arquitecto o alguien de nivel técnico mayor si fuera el caso. Este tiempo debe de haberse planeado.

Como tercera acción se registra dónde hemos encontrado el trabajo atascado así como también registrar el tiempo de atraso con sus respectivas características y la solución que se dio a ese atraso. En el caso práctico se puede llevar una bitácora de incidencias.

Como cuarta acción se repite el proceso desde la primera acción hasta la tercera acción sacando una lista de recursos a los problemas y soluciones que se dieron.

La quinta acción consiste en identificar las causas de posibles problemas en función de la frecuencia de los recursos utilizados en cada atraso mediante la lista de registro de la acción anterior (cuarta acción). Esto que acabamos de mencionar si lo enfocamos a procesos de manufactura o de producción pueden ser procesos defectuosos por:

- ✓ Ajustes poco fiables
- ✓ Falta de capacidad de protección
- ✓ Que los recursos este mal gestionado

Cuando se presentan dichos problemas lo que se analiza a fondo es el nivel de recursos y el nivel de trabajo que se está desarrollando, no tendremos que acelerar más de una estación el trabajo en cuestión del atraso, eliminando hasta cierto punto la aceleración. Esto aplicado a la industria de software significa que un desarrollador puede que no esté desarrollando programas correctos pudiendo ser la causa la falta de nivel técnico del desarrollador (capacidad) junta con la poca claridad de la especificación, o una especificación no bien definida. Si llevamos a cabo las acciones anteriores podremos ir reduciendo la longitud de los buffers o tiempos de proceso (lead – time).

Como sexta acción una vez localizados los sitios donde se encuentran los trabajos retrasados, se asignan prioridades de acuerdo con el número de veces que aparece en la lista cada recurso. Podremos utilizar un factor de ponderación y en función de ello podemos aplicar mejora en la productividad aplicando los diagramas de Pareto.

Los diagramas de Pareto nos permitirán ver un tipo específico de histograma, ordenado por frecuencia de ocurrencia, que muestra cuántos defectos se han generado por tipo o categoría de causa identificada. La técnica de Pareto se usa principalmente para identificar y evaluar incumplimientos. En los diagramas de Pareto, el ordenamiento por categoría se usa para guiar la acción correctiva. El equipo del proyecto debería llevar a cabo acciones para solucionar primero los problemas que están causando la mayor cantidad de defectos. Los diagramas de Pareto están relacionados conceptualmente con la ley de Pareto, que sostiene que una cantidad relativamente pequeña de causas provoca generalmente la mayor parte de los problemas o defectos. Esto comúnmente se denomina principio 80/20, donde el 80 por ciento de los problemas se debe al 20 por ciento de las causas. Los diagramas de Pareto también se pueden usar para resumir todos los tipos de datos para los análisis 80/20.

Dentro de las principales acciones que TOC propone es utilizar estaciones de trabajo eficientes en donde se pueden eliminar las actividades que no agregan valor y tratar de hacer la manufactura más esbelta incrementando o disminuyendo capacidades de acuerdo a la estación de trabajo y el trabajo a realizar. Para la parte de desarrollo de proyectos nosotros clasificaremos a los desarrolladores o programadores de acuerdo a su nivel de experiencia como pueden ser ingenieros junior, advisories, senior y arquitectos que a la hora de asignar el trabajo simple, mediano o complejo puede ser clasificado en función de la experiencia y la

capacidad de cada miembro del equipo, la experiencia de cada desarrollador y así como la de asignar trabajo correspondiente debiera ser de acuerdo a su capacidad. Sin embargo en ocasiones esto no es tomado en cuenta dentro de la planeación y suceden atrasos inesperados.

Actualmente existe el concepto "Administración Ágil para ingeniería de Software aplicando Teoría de Restricciones para resultados de negocios". Este concepto se focaliza en la parte de la ingeniería de software y no tanto en la parte de administración de proyectos de software. Mi enfoque está basado utilizando el modelo de Project Management Institute como principal herramienta para la Administración de Proyectos de Software apoyado y ayudando al concepto de Teoría de Restricciones y la razón principal es porque todavía hoy en día no hemos podido resolver la problemática inicial de:

- ✓ Certeza de terminar el proyecto en el tiempo acordado.
- ✓ Certeza de terminar el proyecto en el presupuesto acordado.
- ✓ Certeza de terminar el proyecto con las especificaciones acordadas

2.2- Administración de proyectos con PMI.

La administración de proyectos de acuerdo al PMI es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer los requisitos del proyecto. La administración de proyectos se logra mediante la aplicación e integración de los procesos de dirección de proyectos de inicio, planificación, ejecución, seguimiento, control, y cierre. El administrador de proyecto es la persona responsable de alcanzar los objetivos del proyecto.

La Administración de un proyecto incluye:

- Identificar los requisitos.
- Establecer objetivos claros y posibles de realizar.
- Equilibrar las demandas concurrentes de calidad, alcance, tiempo y costo.
- Adaptar las especificaciones, los planes y el enfoque a las diversas inquietudes y expectativas de los diferentes interesados.

Los Administradores del proyecto a menudo hablan de una "triple restricción":

- 1) Alcance
- 2) Tiempo
- 3) Costo

A la hora de gestionar los requisitos concurrentes de un proyecto, la calidad de proyecto se ve afectada por el equilibrio de estos tres factores. Los proyectos de alta calidad entregan productos, servicios o resultados requeridos con el alcance solicitado, en tiempo y dentro del presupuesto. La relación entre estos tres factores es tal que si cambia cualquiera de ellos, se ve afectado por lo menos otro de los factores. Los administradores de proyectos también

gestionan los proyectos en respuesta a la incertidumbre. El riesgo de un proyecto es un evento o condición inciertos que, si ocurre, tiene un efecto positivo o negativo al menos en uno de los objetivos de dicho proyecto.

El equipo de administración del proyecto tiene una responsabilidad profesional ante sus interesados, incluidos los clientes, la organización ejecutante y el público. Los miembros del PMI acatan un "Código de Ética", y quienes tienen la certificación de Profesional de la Dirección de Proyectos (PMP[®]) acatan un "Código de Conducta Profesional". Los miembros del equipo del proyecto que son miembros PMI o PMP están obligados a respetar las versiones actualizadas de estos códigos.

Es importante destacar que muchos de los procesos incluidos en la administración de proyectos son repetitivos debido a la existencia o a la necesidad de elaborar gradualmente el proyecto durante el ciclo de vida del proyecto. Esto significa que, a medida que un equipo de dirección del proyecto conoce más a profundidad un proyecto, el equipo puede luego dirigirlo con un mayor nivel de detalle.

El término "administración de proyectos" se usa a veces para describir un enfoque de la organización o de dirección respecto a la gestión de los proyectos y de algunas operaciones continuas, que pueden ser redefinidas como proyectos, que también se denomina "dirección por proyectos". Una organización que adopta este enfoque define sus actividades como proyectos de una manera que sea coherente con la definición de proyecto. Ha habido una tendencia en los últimos años a gestionar más actividades de más áreas de aplicación utilizando dirección de proyectos

2.3- Teoría de Restricciones.

Una de las teorías que propuse en esta investigación es como la Administración de proyectos de Software puede ser desarrollada mediante la ayuda de la Teoría de Restricciones la cual abarca conceptos como:

- ✓ Fluctuaciones Estadísticas
- ✓ Eventos dependientes e independientes.
- ✓ Cuellos de botella
- ✓ Amortiguadores
- ✓ Balance de Cargas
- ✓ Ruido en los procesos.

La Teoría de restricciones la podemos resumir en cinco pasos

1. Identificar la restricción. - identificar la restricción significa la habilidad de identificar el problema dispersar el problema en pequeños problemas y debe priorizar el impacto.
2. Decidir cómo explotar restricción.- una vez tomada identificada la restricción y priorizada se clasifica para decidir cómo explotarla.

3. Subordinar todo acerca de la decisión. Una vez tomada la decisión para explotarla se subordina para ver cómo se atacara la decisión.
4. Elevar la restricción.- verbalizar un poco mejor la restricción para elevar el grado de análisis y no volver caer en ella con la solución más adecuada. En este punto se utiliza el método llamado efecto-causa –efecto. Dentro de este punto existen tres estados, llamados
 - a. Clasificación
 - b. Correlación
 - c. Efecto-causa-efecto.

Clasificación.- Se revisan las características específicas de la restricción para navegar sobre estas y ver el tipo de restricción

Correlación.- son las turbulencias por las cuales esta restricción existe, puede estar relacionado con otras basadas en una cuidadosa observación con otros objetos inherentes a la restricción.

Efecto-Causa-Efecto.- Una vez encontrado todos los objetos del estado de la correlación se utiliza la lógica de deducción o explicación reconociendo los objetos de la materia la ciencia.

5. Si el paso número cuatro ha sido roto y se vuelve a caer en la restricción lo que se debe hacer es volver al paso uno, para volver a identificar la restricción con otro grado de clasificación e identificar las nuevas habilidades para identificar el problema.

Por naturaleza dentro de los tipos de restricciones que existen en la manufactura encontramos tres tipos de restricciones

- ✓ Restricciones en las políticas
- ✓ Restricción en los recursos
- ✓ Restricciones en los materiales

2.4- Teoría de Restricciones aplicada a la Administración de Proyectos de software.

Del amortiguador o Buffer donde por definición es el tiempo de protección contra cualquiera de las perturbaciones desconocidas de un proyecto, la pregunta que debemos resolver es ¿Cuánto tiempo debe ser la longitud del amortiguador o buffer o tiempo expresado en horas, días, meses? Aunque tuviéramos el tiempo y los recursos necesarios para recopilar datos estadísticos, solo conseguiríamos una curva de probabilidad en función de la probabilidad contra el tiempo, lo importante de la curva es enfocarnos al impacto que tendrá sobre el proyecto. A medida que pasamos a intervalos de tiempo más y más largos,

aumenta la probabilidad de superar las perturbaciones, o bien dicho de otra forma disminuye la probabilidad de ocurrencia de las perturbaciones desconocidas, pero nunca llegaremos a la certeza absoluta. A este enfoque la teoría de restricciones le llama que hemos caído en el ruido. El ruido es la perturbación que puede llegar al infinito. Ver figura 1

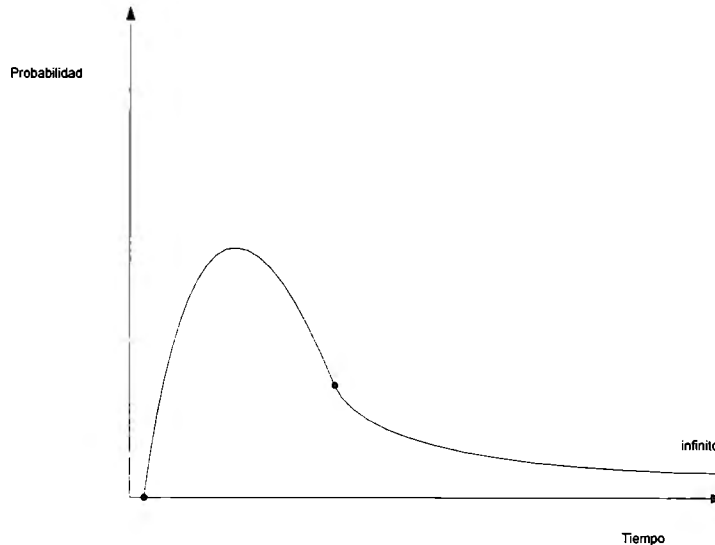


Figura 1.

Es aquí donde las variables de tiempo, presupuesto y especificaciones acordadas caen y difícilmente pueden salir a menos que se tome la decisión. La determinación de la longitud del buffer de tiempo es una cuestión de juicio. Si queremos ser precavidos y elegir un buffer o amortiguador muy largo en el tiempo, podremos absorber cualquier perturbación, sin embargo la pregunta sería ¿a qué precio? Cuando un ingeniero de sistemas desarrolla un programa y se le pregunta ¿cuánto tiempo tardará en desarrollar un programa? Como respuesta a esta pregunta se pueden utilizar los modelos de estimación como:

- ✓ COCOMO.- Constructive Cost Model este modelo presenta tres formas la cual ejemplifica dicho modelo.[2]
 - Modelo Básico.- Es estático simplemente evaluado que calcula el esfuerzo y costo del desarrollo del software en función del tamaño del programa expresado en líneas de código (LDC) estimadas.
 - Modelo intermedio.- Calcula el esfuerzo del desarrollo de software en función del tamaño del programa y un conjunto de guías de costos que incluyen una evaluación subjetiva de los atributos del producto, de los atributos del hardware, de los atributos del personal y de los atributos del proyecto.
 - Atributos del producto:

- Fiabilidad del software requerida
- Tamaño de la base de datos de la aplicación
- Complejidad del producto
- Atributos del hardware
 - Restricciones de rendimiento en tiempo de ejecución
 - Restricciones de memoria
 - Volatilidad del entorno de la máquina virtual
- Atributos del personal
 - Capacidad de análisis
 - Capacidad del ingeniero de software
 - Experiencia con las aplicaciones
 - Experiencia con la máquina virtual
 - Experiencia con el lenguaje de programación
- Atributos del proyecto
 - Utilización de las herramientas de software
 - Aplicación de los métodos de ingeniería de software
 - Planificación temporal del desarrollo requerida.
- El modelo avanzado.- incorpora todas las características de la versión intermedia con una evaluación del impacto de las guías de costo en cada fase (análisis, diseño, etc.) del proceso de ingeniería de software.
- ✓ Modelo de estimación de Putnam.- El modelo de estimación de Putnam es un modelo que asume una distribución específica del esfuerzo a lo largo de la vida de un proyecto de desarrollo de software.
 La distribución del esfuerzo puede ser caracterizada por una gran cantidad de curvas descritas analíticamente por Lord Raleigh, dicha curva es utilizada para obtener una ecuación del software que relaciona el número de líneas de código esperadas con el esfuerzo y el tiempo de desarrollo:

$$L = CkK^{1/3} t^{4/3}$$
 Donde Ck es una constante del estado de la tecnología y refleja las "restricciones directas que frenan el progreso. Valores típicos pueden ser valor bajo para un entorno pobre de desarrollo de software (por ejemplo sin metodología, sin documentación, y revisiones pobres, con un modo de ejecución no interactivo),
 Valor medio puede ser donde la metodología, documentación y revisiones sean buenas que se aproximan a la realidad.
- ✓ Modelo de puntos de función.- Este modelo utiliza las métricas del software orientadas a la función medidas indirectamente del proceso de desarrollo los cuales son :
 - Número de entradas del usuario
 - Número de salidas del usuario
 - Número de peticiones del usuario
 - Números de archivos



- Numero de interfaces externas
- Factor de ponderación (Simple , mediano, complejo)
- ✓ Modelos de estudio temporal.- Este es un modelo que propone la productividad que considera las características microscópicas del entorno de trabajo. Dentro del cual se encuentran los siguientes parámetros.
 - La fracción medio del trabajo diario gastado en trabajos administrativos o indirectos
 - Duración media de las interrupciones del trabajo (minutos)
 - Tiempo de recuperación medio después de la recuperación
 - Número de interrupciones/día
 - Costo indirecto por persona expresado en como una fracción de la paga base.
 - Paga diferencial para las horas extras expresadas como una fracción de paga base.
 - Coeficiente de tiempo de calendario en personas-días para completar el proyecto.

Sin embargo aunque se utilicen estas técnicas llegará el momento en donde se deberá usar el juicio para determinar el tiempo más preciso. La pregunta es ¿Qué hacemos?

El objetivo más preciso de la presente investigación es la de proponer el modelo de Teoría de Restricciones (TOC- Theory of Constraints) y el modelo del Comportamiento Organizacional como herramientas de ayuda para la Administración de Proyectos de Software mediante una adecuación al modelo que propone el PMI (Project Management Institute) para administración de proyectos utilizando puntos relevantes de la modelo teoría de Restricciones para mejorar el desarrollo y hacer eficiente la administración de proyectos de Software.

Dentro de los tipos de restricciones que encontramos en la industria en general dentro del modelo de PMI. (Project Management Institute) nos encontramos con las siguientes restricciones:

- ✓ Restricciones en el Alcance
- ✓ Restricciones en la Calidad.
- ✓ Restricciones en el Programa
- ✓ Restricciones en el Tiempo
- ✓ Restricciones en el Presupuesto
- ✓ Restricciones en los Recursos.
- ✓ Restricciones en el Riesgo.
- ✓ Restricciones en la Comunicaciones

2.5- Administración de Proyectos con TSP y PSP.

[14] PSP ayuda a los desarrolladores individuales mejorar su el rendimiento por llevar la disciplina a la manera en que desarrollan software. Muestra claramente a los desarrolladores cómo gestionar la calidad de sus productos, cómo hacer un plan, y cómo hacer compromisos. También les ofrece los datos para justificar sus planes. Ellos pueden evaluar su trabajo y sugerir mejoras en la dirección de analizar y revisar el desarrollo tiempo, los defectos, y los datos de talla. PSP consiste en una serie de siete procesos personales que proporciona la secuencias de comandos diferentes, medidas, formas, plantillas, listas de control y las normas. Estos procesos se clasifican de la siguiente manera:

- ✓ PSP0, PSP0.1 - Recoger datos actuales
- ✓ PSP1, PSP1.1 - Estimación a partir de datos históricos
- ✓ PSP2, PSP2.1 - Gestión de calidad
- ✓ PSP3 - PSP2.- Aplicar en función del ciclo

PSP0 proporciona una constante fundamental para medir el progreso y una base definida para mejorar.

PSP0.1 extiende PSP0 añadiendo elementos tales como el estándar de codificación, la medición del tamaño PIP (Propuesta de Mejora de Procesos).

PSP1 se centra en la gestión de proyectos personales.

PSP1 añade estimación de software de tamaño y presentación de informes de ensayo a PSP.

PSP1.1 añade la planificación de tareas y el calendario la planificación de PSP1.

PSP2 se extiende PSP1 al introducir un control técnicas tales como la revisión de código y revisión del diseño. Estas técnicas de revisión de ayudar a los ingenieros de software encontrar defectos tan pronto como sea posible.

PSP2.1 y PSP2 establecer criterios de diseño y examinar la integridad del diseño verificación y coherencia de estas técnicas.

PSP3 introduce métodos a las personas a usar cuando son para el desarrollo de software a gran escala. En la escala la PSP2 hasta proyectos de mayor envergadura, el enfoque es subdividir el proceso personal de desarrollo más grande programas en pedazos del tamaño de PSP2.

El objetivo principal del TSP es crear un equipo para establecer y mantener un proceso de equipo auto dirigido, y el apoyo a cada disciplina de trabajo como base del marco PSP. Equipos auto dirigidos significa que el equipo maneja planes y el seguimiento de su trabajo, gestiona la calidad de su trabajo y trabaja proactivamente para alcanzar los objetivos del equipo. TSP tiene dos componentes principales: la creación de equipos y el trabajo en equipo. La creación de equipos es un proceso que define las funciones de cada miembro del equipo y se establece el trabajo en equipo hasta el lanzamiento de TSP y periódica relanzamiento. El trabajo en equipo es un proceso que se ocupa de los procesos de ingeniería y las prácticas utilizadas por el equipo. TSP, en definitiva, proporciona a los ingenieros y gerentes una forma que establece y gestiona su equipo para producir el software de alta calidad a tiempo y presupuesto.

[13] El TSP es descrito en dos dimensiones: organizacional y operacional. El organizacional con un propósito administrativo y el operacional con un propósito técnico

El TSP organizacional es descrito en tres partes generales:

- ✓ Equipos de formación
- ✓ Equipos de lanzamiento
- ✓ Equipos de trabajo

Equipos de formación.- Formación de equipos comienza desde la contratación y selección del equipo orientado principalmente a la capacitación y a las necesidades del equipo. Las principales actividades las podemos dividir en cuatro: recursos, localización, necesidad de las habilidades, reclutamiento y proporcionar capacitación. [14]

Equipos de lanzamiento.- A través de los equipos de lanzamiento el equipo produce los siguientes resultados: Metas del equipo, roles de cada miembro del equipo, desarrollar el plan de trabajo, un plan de calidad un programa de trabajo, valoración de cada riesgo y reporte de estatus de los mismos.

Equipos de trabajo.- Dentro de esta parte se encuentran cinco partes generales como son: La colección de datos, el plan de seguimiento, un equipo de retroalimentación, un balance de cargas de trabajo y una re planeación del trabajo.

El TSP operacional es prácticamente dividido en tres:

- ✓ El TSP estructurado y de flujo
- ✓ El TSP procesos (Requerimientos, Diseño, Implementación , Integración y Pruebas del sistema)
- ✓ El TSP elementos del proceso (Script , Formatos, Roles especificados, Valoración de la estrategia, el checklist y especificaciones no relacionadas con los roles, cursos de entrenamiento y autorización de actividades).

2.6- Comportamiento Organizacional.

El Comportamiento Organización es importante para un proyecto porque investiga el impacto de los individuos grupos y estructuras organizacionales que tengan sobre un proyecto. Este comportamiento organizacional se focaliza en cómo mejorar la productividad, reducir el ausentismo, reducir la rotación de personal, incrementar la satisfacción del personal en los proyectos y sobre todo encaminado a proyectos exitosos en alcance, tiempo y presupuesto. [13 página. 5]. El modelo del Comportamiento Organizacional que estudiamos se encuentra contenido en la figura 2.

Basic OB Model, Stage II

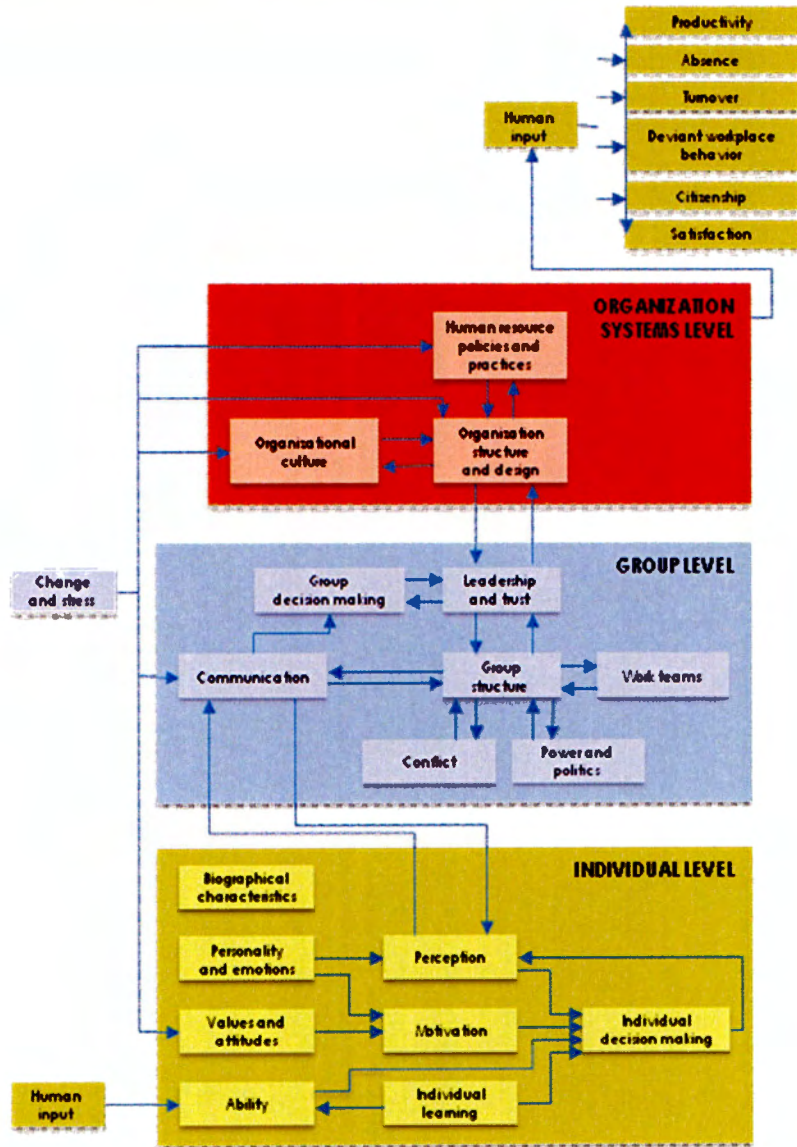


Figura 2

Y de una forma general las ares que estudiamos se encuentran en la figura 3:

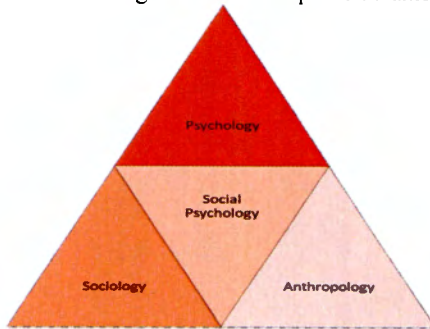


Figura 3

La figura 4 nos muestra del impacto que causan los empleados en las organizaciones y los motivos.

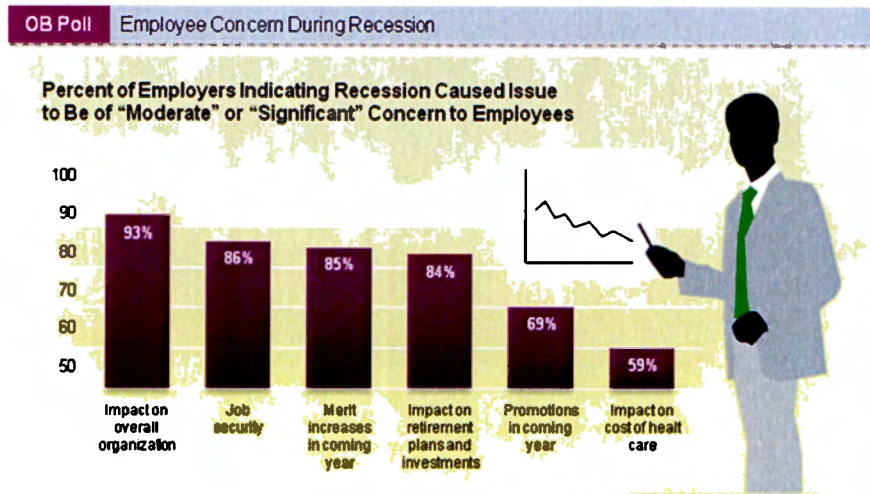


Figura 4

2.6.1- Diversidad dentro de las organizaciones.

Uno de los puntos importantes dentro de la diversidad organizacional considera la edad, raza, género, religión, estatus de capacidad, la discriminación de los grupos hacia las personas son importantes a considerar. [13 pagina 38]

La diversidad dentro de los proyectos consiste en tres aspectos:

- 1.- Habilidades de la persona
- 2.- Características de la persona
- 3.- El aprendizaje de la persona.

1.- Habilidades de la persona:

Unos de los puntos que el administrador de proyectos debe tener en cuenta cuando arma a su equipo de trabajo es revisar las habilidades de cada persona y la influencia que tendrá en el nivel de ejecución del trabajo encomendado. Se puede basar en las siete dimensiones de las habilidades intelectuales como son:

- a) Numero de aptitudes
- b) Comprensión Verbal
- c) Percepción y rapidez
- d) Razonamiento inductivo
- e) Razonamiento deductivo
- f) Visualización espacial
- g) Memoria

De acuerdo a la figura 5

En una plática con ellos se deben revisar las habilidades para comunicarse con las demás personas



Figura 5.

2.- Características de la persona.- Las características de la persona debe ser revisadas cuando un administrador de proyectos arma su equipo. En específico para el administrador de un proyecto debe tener en cuenta el carácter de las personas. De cómo esta influirá a lo largo del proyecto, será como en un equipo de football americano, quien debe ser fuerte para enfrentar los desarrollados pesados, quien será el estratégico que empujará al proyecto en la parte técnica, deberá tener que utilizar mucho la teoría de actitud de salir siempre adelante.

2.6.2 Actitud y satisfacción de trabajo

Uno de los principales aspectos que debe de cuidar el administrador de un proyecto es la actitud de todo su equipo de trabajo porque pueden representar avisos o problemas potenciales de comportamiento que influyen en el equipo y en el proyecto. Dentro de un proyecto uno debe planear las vacaciones de las personas o bien que tan desgastada esta la persona en comparación con su último proyecto, tendrá que entrevistar a cada uno para saber el estado en el que se encuentra cada persona, la creatividad deber ser usada siempre en cada momento por el administrador de proyecto para no bajar la actitud durante el proyecto, la actitud es muy buena al iniciar un proyecto, sin embargo gran parte se deteriora cuando se llega a la última fase del proyecto debido tal vez al desgaste del mismo. El administrador de proyecto debe preocuparse por la salida, inclusive planear y avisar a quien corresponda de la integración de la salida de la gente al término del proyecto a su nuevo proyecto. El administrador de proyectos debe asegurar siempre que su equipo de trabajo este siempre motivado con una buena actitud hacia el trabajo.[13 pagina 70]

2.6.3 Emociones y Humor

Los aspectos importantes que debe de considerar el administrador de un proyecto de su gente es acerca del humor de las personas y sobre todo la estructura de la definición del humor y como debe enfrentar esta situación, los diferentes niveles del humor son:

Alto positivo

Alerta

Excitado

Eufórico

Feliz

Bajo positivo

Triste

Deprimido

Aburrido

Cansado

Alto negativo

Tenso

Nervioso

Estresado
Enojado
Bajo Negativo
Contento
Sereno
Relajado
Calma

De acuerdo a la figurar 6 que nos muestra la estructura del humor.

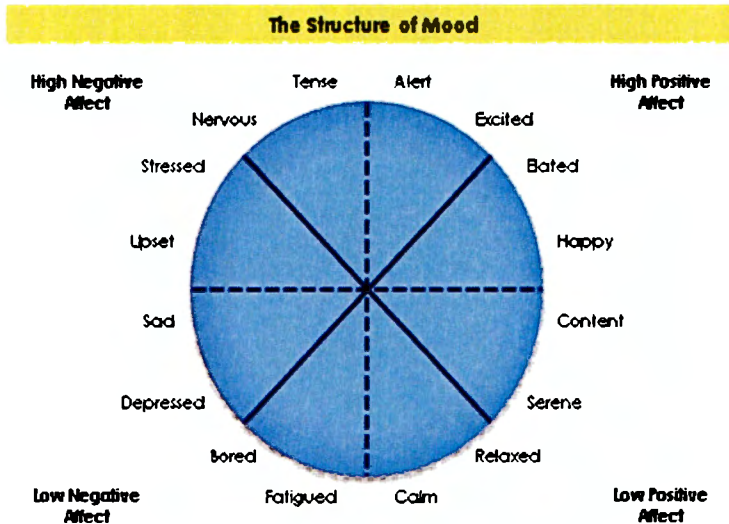


Figura 6

Todos estos niveles son los que debe observar el administrador de proyectos y la forma de mitigar cuando una persona no esté en un nivel medio que permita un medio ambiente sano, algunos aspectos importantes que afectan el humor son :

- 1.-Ejercicio
- 2.-Dormir su horas adecuadas
- 3.- Actividades sociales
- 4.- La personalidad
- 5.- Clima
- 6.- Stress (derivado de un exceso carga de trabajo o todo lo que afecte en la tensión de la persona)

La forma de cómo mejorar este estado de ánimo es considerando los siguientes puntos:

1.- La selección.- La selección de personal para la tarea adecuada de acuerdo a su capacidad, y en específico revisar la personalidad cuando se trate de un rol con alto grado de relación con los demás o un rol de alto grado de investigación en donde se requiera trabajar solo.

2.- Creatividad.- Dejar a la gente que utilice su creatividad es una de las formas de encontrar su equilibrio dejando cierta tolerancia al error.

3.- Motivación.- Ejercer retroalimentación en el momento adecuado con las palabras correctas antes de que ocurra una desviación fuera de los niveles de riesgo del proyecto son necesario para la motivación del equipo de trabajo , duro en el fondo y suave en la forma de decir las cosas.

El trabajo en equipo es una de las formas de cómo lograr la motivación cuando estos han alcanzado metas cortas y todos apoyaron al lograr el resultado.

Una de las formas de la motivación que ayuda a los colaboradores del equipo es sentirse bien, el administrador de proyectos debe procurar revisar con cada subordinado si hay algún sentimiento en el cual no esté cómodo y afecte directamente al proyecto.

4.- Liderazgo.- Un liderazgo adecuado cambia las emociones de las personas, cuando un liderazgo es entusiasta, excitante y activo, la mayoría de las veces ejercerán una energía sobre sus subordinados de competente, optimista, un trabajo placentero y satisfactorio. El administrador de proyectos debe trabajar en los aspectos más relevantes de las características de liderazgo, dando una clara visión, la probabilidad de que todos acepten cualquier cambio dentro del proyecto será sin problema y alta.[13 pagina 98]

2.6.4 Personalidad y Valores

La personalidad es la manera de cómo una persona reacciona al interactuar con otros.

Dentro de los aspectos de la personalidad del administrador de proyectos como la de los colaboradores que trabajan en el proyecto están los siguientes grandes rasgos:

1.-Emocionalmente estable.- Lo más relevante que afecta el proyecto es que son menos negativos en pensamientos, menos negativos en sus emociones y esto afecta en un alto grado de satisfacción en su vida y en el proyecto el nivel de stress es bajo.

2.-Extrovertido.- Los más relevante es que tiene mejor interrelación con los demás, es emocionalmente más expresivo, es asertivo y socialmente dominante. Esto afecta positivamente en un alto grado de ejecución en su trabajo, mejora el liderazgo, el trabajo es alto y se tiene una vida satisfactoria.

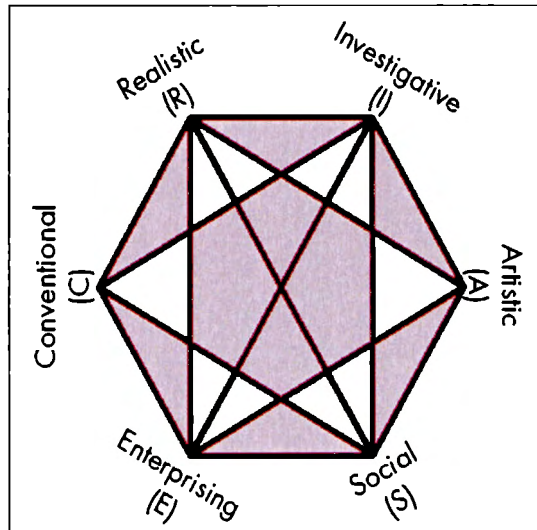
3.- Abierto y sincero: Lo más relevante de esta personalidad es que aumenta el aprendizaje aumenta la creatividad, es más flexible y autónomo. Esto afecta positivamente en la ejecución de entrenamiento, más adaptable en los cambios.

4.-Agradable.- lo más relevante es la gente cooperativa, tranquila que inspire confianza.

5.-Consciente.- una persona que es responsable, confiable organizado y persistente.

6.-Narcisista.- tiende a ser arrogante, un grandioso sentido de importancia, requiere excesiva admiración. [13 pagina 132]

En la figura 7 nos muestra la relación entre los tipos de personalidad profesional



***Need to match personality
type with occupation.***

Figura 7

2.6.5 Percepción Individual y toma de decisiones

En la administración de proyectos la percepción es importante porque está basada en el comportamiento de las personas la cual se puede definir como la realidad que cada quien ve.

Los factores que afectan la percepción son:

Contexto o situacionales:- los cuales refieren al tiempo (no es lo mismo trabajar un día entre semana que un sábado o domingo por la noche), al trabajo (un trabajo sencillo o complejo) y al aspecto social (si hay o no compañerismo en el equipo) que le fueron otorgados.

Interpretación de las Características personales: Entre estas se encuentran las aptitudes de las personas, su motivación, su interés, su experiencia y sus expectativas.

La tendencia hacia al objetivo.-Que tanto la persona es novedosa, como ha sido su antecedente de la persona, como ha sido su trayectoria de la persona en los últimos proyectos, las características de la persona en cuanto a presentación (limpio, atractivo, educado, aseado), su voz (tenue, gritón , seguro de sí mismo).

Uno de los puntos importantes dentro de la percepción es incrementar la probabilidad de acierto de la percepción principalmente porque es imposible asimilar todo lo que nosotros vemos. La forma de incrementar esta percepción es con las entrevistas y con la evaluación de las expectativas que nosotros esperamos de las personas y con esto aumentará la percepción de las personas, sobre todo en la medida en que más conozcamos de sus rasgos culturales y ambientales.

La toma decisiones como administrador de proyectos es una forma de vida que debe de llevar el APS (Administrador de Proyectos de Software) y una de las formas de tomar decisiones son:

1.- Analizar la situación.- la forma en lo que debe el APS de analizar la situación es en ambiente, tiempo, espacio, comportamiento organizacional en el que se encuentra, riesgos que correrá, la decisión deberá ser enfocada en el entorno cultural de la organización.

2.- Buscar Alternativas.- Lo que no se debe perder nunca de vista es el objetivo focalizándolo adecuadamente, no debe ser racional, no debe conocer toda la información en caso de que no exista, puede o no conocer la información relevante, debe uno de buscar todas las alternativas posibles, inclusive la decisión desde no hacer nada como alternativa, debe soportarse en base a su experiencia y revisar el modelo causa y efecto, preguntándonos ¿Qué pasa si se toma esta alternativa? ¿Qué consecuencias tendrá? Debe uno de buscar toda la información que exista sobre el tema en cuestión, así como aumentar el número de opciones para tomar la decisión.

3.- Combinar el análisis racional con la intuición.- Es importante que las decisiones sean en base a un análisis racional pero si todas las decisiones fueran llevadas a cabo con estricto apego análisis racional, el tiempo podría ser un factor en contra de estos análisis,

debemos confiar en las personas y debemos confiar en el APS. El APS por su parte debe usar su intuición que en base en su experiencia.

4.- Mejorar la creatividad.- El APS debe encontrar la forma y los espacios de tiempo para que su gente sea creativa, específicamente cuando enfrenten cualquier problema y no tanto el seguir con un método o metodología de trabajo porque un proyecto siempre es diferente a cualquiera de los demás, podrán ser similares pero no iguales. [13 pagina 166]
En base a la figura 8 nos muestras todos los factores que influyen en la percepción.

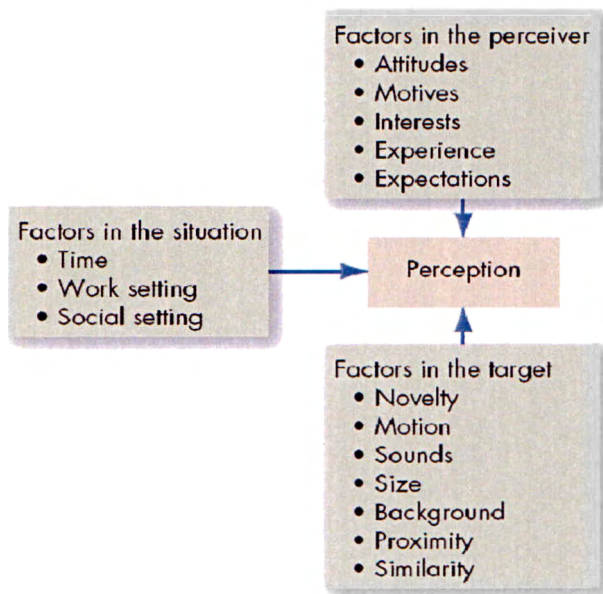


Figura 8

2.6.6 Conceptos de Motivación

La motivación es el proceso mediante el cual los individuos miden el grado de intensidad dirección, persistencia y esfuerzo hacia el objetivo. El APS constantemente debe estar midiendo esta intensidad, dirección y persistencia. Existen diferentes teorías de motivación que ayudan con la productividad de cada persona.

- ✓ Teoría de necesidades
- ✓ Teoría de reconocimiento y evaluación
- ✓ Teoría de conseguir el objetivo
- ✓ Teoría del reforzamiento.
- ✓ Teoría de la igualdad y justicia [13 pagina 202]

En figura 9 nos muestra una integración de las teorías de motivación.

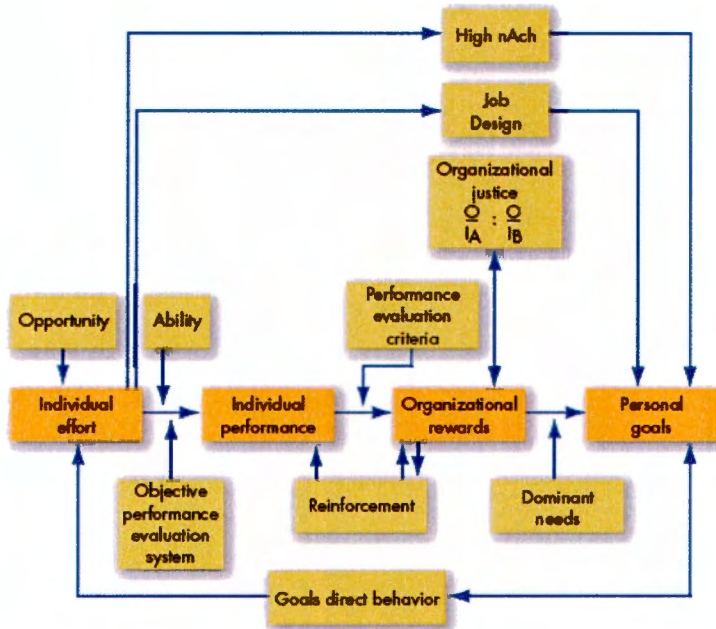


Figura 9

2.6.7 Aplicación de la motivación

Algunas características para el diseño del trabajo y la forma de evaluar deben ser importantes para la motivación y alguna de los rasgos importantes son:

Habilidades.- Es el grado en el cual el trabajo incorpora una serie de habilidades y de talento para llevarse a cabo, en este caso el APS, debe revisar las habilidades como ejemplo las habilidades de un arquitecto de desarrollo de Software requiere un grado de especialización y talento.

Identificación de tareas.- Identificar las taras de acuerdo a su complejidad, de las tareas, la cual debe de ser una forma en la cual APS revise y asigne tareas.

Importancia de la tarea.- La importancia de las tareas es la de revisar el impacto que tendrá el trabajo en la vida y trabajo de esa persona así como la de las personas que le rodean.

Autonomía.- Es el grado en el cual el trabajo provee al trabajador libertad o independencia. En esta parte lo que tiene que hacer el APS es definir la libertad que tendrá el proyecto desde trabajar en la casa hasta la libertad en el tiempo, siempre y cuando se cumplan con los objetivos.

Retroalimentación.- La retroalimentación es el grado en la cual se debe llevar a cabo de acuerdo a la ejecución de las tareas. Esta actividad el APS debe ser apoyado por alguien con especialidad de arquitecto (o técnico) hacia los desarrolladores. La retroalimentación a tiempo despeja dudas acerca del buen desempeño del trabajo, una retroalimentación fuera de tiempo, causa tensión y desmotivación aunque sea positiva.

Definición de la forma de comunicación inicial.- Un administrador de proyectos debe tener un 90% de su tiempo en comunicación por la cual debe definir las diferentes técnicas de comunicación (de manera personal, por mail, una vez al día, una vez a la semana, etc.) y darlas a conocer, de tal forma que todo su equipo conozca dicha forma para minimizar los problemas de comunicación. [13 página 249]. De acuerdo a la figura 10

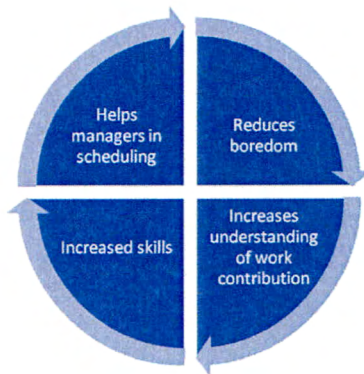


Figura 10

2.6.8 Fundamentos de comportamiento grupales

El comportamiento del colaborador, los sentimientos están estrechamente relacionados y las influencias de un grupo (normas) son significativos para afectar el comportamiento individual. Las normas del grupo (normas) son altamente eficaces en el establecimiento del trabajo de cada colaborador. El dinero es un factor menor que determina la salida de los trabajadores y son más bien las normas del grupo, los sentimientos y la seguridad de los equipos de trabajo. El APS debe formar equipos de trabajo para fortalecer tareas que de manera individual se podrían llevar más del tiempo establecido.

Los siguientes estados de grupo ayudan al desarrollo:

- ✓ La formación.- Los miembros sienten mucho más la incertidumbre dentro de esta etapa empiezan a pensar que formaran parte de un equipo.
- ✓ Tormenta de ideas:- En esta etapa los conflictos son muchos entre los miembros del grupo.
- ✓ Normalización: Dentro de esta etapa los miembros han desarrollado una estrecha relación y la cohesión.
- ✓ Ejecución.- Dentro de esta etapa el grupo es totalmente funcional
- ✓ Levantar.- En los grupos temporales, que se caracteriza por la preocupación con el ajuste de las actividades más que el rendimiento que escuchar. [13 pagina 274]

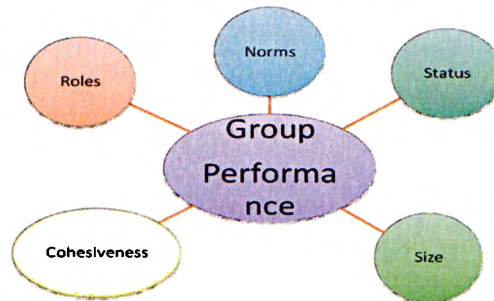


Figura 11

2.6.9 Equipos de trabajo

El trabajo en equipo ayuda a las personas a crear una identidad en el trabajo que le permita desarrollar su talento y habilidades así como el de compartirlos con los demás, de esta forma le permite también aprender de las demás personas, creando un desarrollo profesional en cada persona. Esto es basado en la teoría de identificación social que se compone de cuatro elementos:

Categorización: a menudo ponemos a los demás (y a nosotros mismos) dentro de categorías. Etiquetar a alguien como bueno malo son formas de decir otras cosas acerca de los demás.

Identificación: también nos asociamos con determinados grupos (nuestros grupos), para reafirmar nuestra autoestima.

Comparación: comparamos nuestros grupos con los demás grupos, percibiendo un sesgo favorable hacia el grupo al que pertenecemos.

Distinción psicosocial: deseamos que nuestra identidad sea a la vez distinta positivamente comparable con otros grupos. [13 pagina 312] de acuerdo a la figura 12 que nos muestra los diferentes roles del equipo.

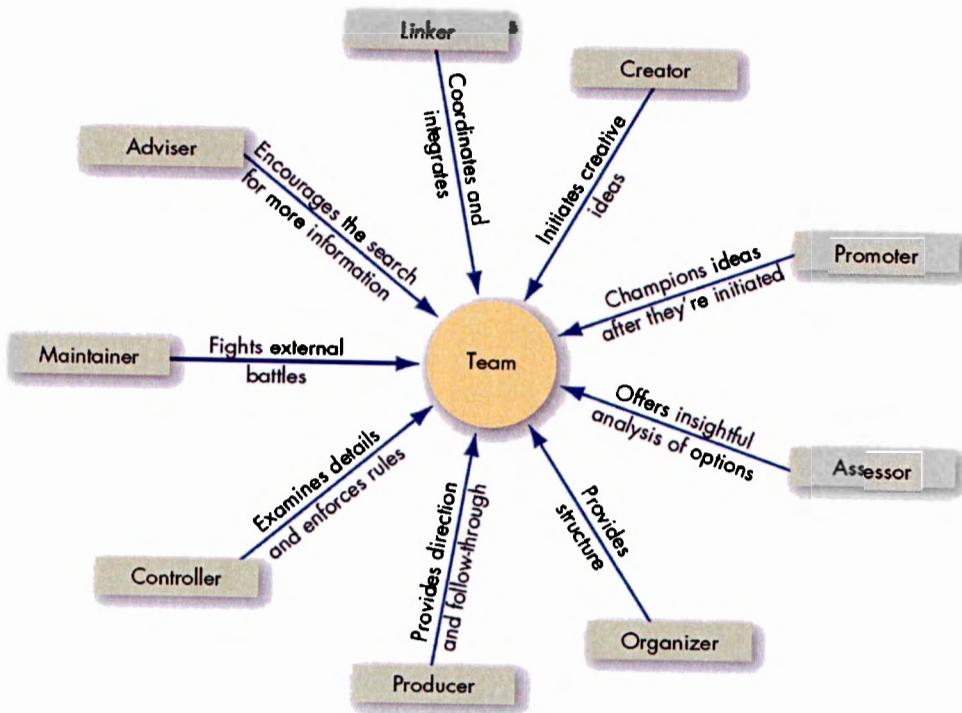


Figura 12

2.6.10 Comunicación.

El 90% de la actividad de un APS es la comunicación, una pobre comunicación lleva a los proyectos a conflictos. El personal de desarrollo de proyectos de software gasta más o menos un 60 % de su tiempo en comunicación con los clientes, con compañeros, escribiendo correos, leyendo correos, hablando y escuchando.

Las principales barreras para una eficiente comunicación se encuentran en:

- **Filtro de la información:** La distorsión de la información hacia arriba y abajo muchas veces ocurre con el propósito de manipulación para fines de algún interesado, hablar con la claridad necesaria es de vital importancia aunque cueste lo que cueste para tomar decisiones oportunas en todos los sentidos.

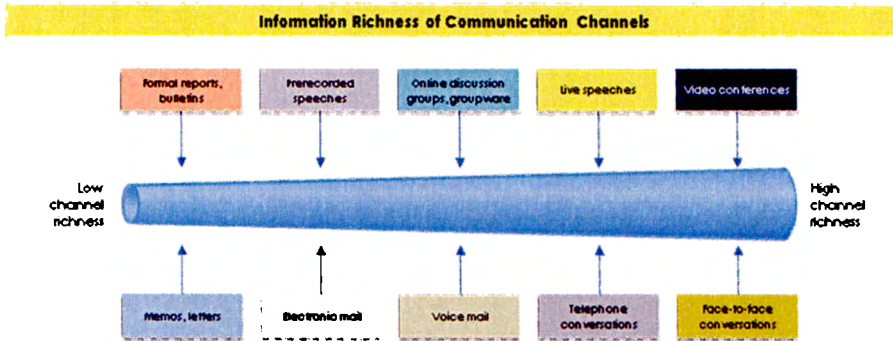
será la comunicación, es decir puede que alguna persona no esté interesada en el proyecto debido a los problemas personales que tiene esa persona en base a lo mencionado, lo que ocasionará que la comunicación no se transmita en un 100% del mensaje y llegue solo un 40% o 30%. El APS debe tener una percepción selectiva de cada uno de los miembros del equipo de trabajo y planear una estrategia de comunicación ya sea a cada miembro del equipo o por equipos o equipo en general.

- La sobrecarga de información: Cada integrante del equipo de un proyecto tiene una capacidad finita de procesar información y cuando esta capacidad es excedida tiende a ocurrir una sobre carga de información lo que ocasionará que :
 - ✓ La información se olvide
 - ✓ La información se ignore
 - ✓ La información se seleccione
 - ✓ La información se pierda.

Como resultado final de esto es una comunicación menos eficiente.

- Emociones: La interpretación del mensaje cuando uno está enojado, feliz, distraído es propenso a ignorar la información o se tendrá una tendencia de acuerdo al criterio de cada quien. El APS lo que deberá hacer será leer las emociones de cada miembro del equipo y mandar el mensaje adecuado en el tiempo, espacio y contenido.
- Lenguaje: Las palabras tienen significado diferente para diferentes personas y esto puede ser clasificado de acuerdo a la edad y al contexto de cada persona. El uso del lenguaje para el APS debe ser una forma de motivar, guiar, ejercer la autoridad sobre todo el equipo. El APS nunca debe asumir que el mensaje se ha transmitido correctamente debe asegurarse con alguna estrategia de recepción con preguntas simples. El tono de voz del APS debe ser con el temple adecuado de brindar confianza.
- Silencio: El romper con el silencio por falta de información debe ser una constante del APS y ser prudente con todo su equipo y con los patrocinadores del proyecto debe ser un estilo de trabajo que continuamente se comunique.
- Expresión corporal: El uso de la expresión corporal muchas veces es muy eficaz cuando se utiliza un adecuado tono de voz, expresión facial, lenguaje del cuerpo. La comunicación electrónica es productiva pero tiene muchas trampas tendencias de la gente cuando se quiere evitar, o evitar muchas cosas, el uso de la expresión corporal en la medida de lo posible es mejor.
- La comunicación con los patrocinadores.- La comunicación con los patrocinadores debe ser de dar seguimiento al proyecto siendo esta precisa y concisa, sin desvíos que alteren el rumbo del proyecto. [13 pagina 340]

La figura 13 nos ayuda a ilustrar lo mencionado



La Figura 13

2.6.11 Liderazgo.

El APS debe tener la habilidad de influenciar sobre el grupo de trabajo alcanzando la visión o las metas del proyecto. Debe de formular en detalle los planes, crear la eficiencia del proyecto y sobre las actividades de operación diaria.

Dentro de las principales características que debe tener un APS como líder se encuentran:

Extrovertido.- Sociable, dominante carismático, entusiasta y animoso

Consciente.- A la gente le gusta trabajar con alguien que entienda los problemas y que tenga una relación alta con el proyecto y con la gente.

Abierto.- mostrando experiencia y escuchando las ideas de los demás en base a argumentos sólidos.

Inteligencia Emocional (EI).- Es con el corazón y la empatía la que se debe de generar con todos, escuchar, entender y sentir las necesidades de cada persona del equipo de trabajo. Al entender a las personas pueden tener una perspectiva del comportamiento. [13 pagina 374]. La siguiente figura 14 nos muestra un completo modelo de liderazgo por rangos.

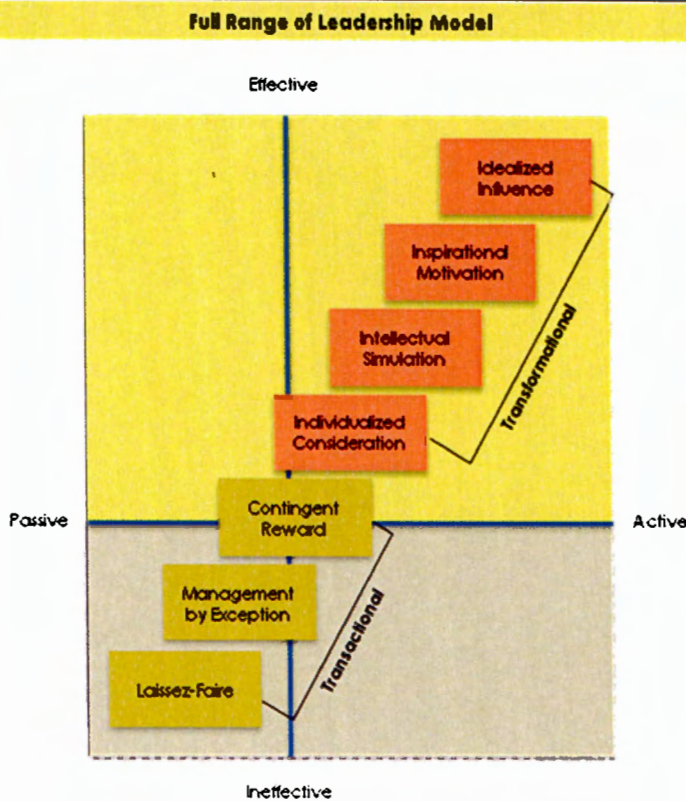


Figura 14

2.6.12 Poder y Política

Una de las razones de porque a la gente le gusta trabajar con alguien que tiene poder se debe a que generalmente son placenteros en el trabajo, ya que este poder les permite de manera fácil y hábil delegar tareas en el equipo de trabajo.

Uno de los problemas a lo que se enfrenta el APS, es el de tener poder sobre su equipo para que su equipo reaccione a la velocidad que el equipo deba reaccionar de acuerdo a las instrucciones del APS. Este poder puede ser legítimo, heredado o puesto por la organización, o bien un poder de experto.

En ocasiones cuando el poder no se gana el equipo de trabajo tiende a desviar la autoridad moral que el APS debiera tener sobre su equipo. El poder del experto es influenciado por una habilidad o conocimiento. Muchos de los problemas a lo que se

enfrenta la administración de proyectos de software es porque generalmente el APS no está a la altura del nivel técnico de cualquier desarrollador y lo que tiende a pasar es que el desarrollador de software hace caso más a un experto en la materia. En este caso lo que tiene que hacer el APS es usar la política y armar una estructura en su equipo de trabajo de tal forma que se gane a los arquitectos de software de esa forma el APS ejercerá un poder mediante los arquitectos hacia su equipo de trabajo. Para el caso de los analistas y APS debiera tener un buen nivel de analista para que en ese sentido use el poder del experto de análisis para que su equipo de analistas lo apoye.

Para administrar un proyecto de software se debe de contar con diferentes habilidades y dentro de los principales se encuentran:

- ✓ Administradores de base de datos
- ✓ Arquitectos
- ✓ Analistas de Sistemas
- ✓ Desarrolladores
- ✓ Líder de proyectos.

Un APS para que tenga la autoridad de poder sobre los analistas y sobre los arquitectos debe ser un buen analista primero para que tenga alguna autoridad de competencia sobre su equipo de trabajo. La política le ayudará a ir evaluando a su equipo de trabajo de acuerdo a las tareas y el avance del proyecto. Cuando este tenga algún atraso por cualquier cuestión, el APS debe tomar acción y ser político en la toma de acción a seguir para no afectar el poder sobre los demás, tratando de ser lo más objetivo posible para el avance del proyecto. La política ayuda a tener un alto grado de satisfacción del equipo de trabajo, decrementa la rotación de personal y los conflictos son mitigados exitosamente. [13 pagina 418]

La siguiente figura 15 nos muestran los factores que influyen en el comportamiento de la política.

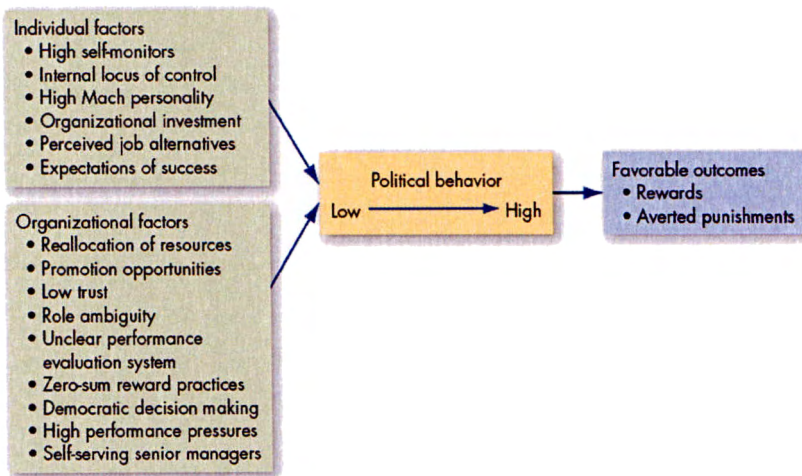


Figura 15

2.6.13 Conflictos Negociación

Un conflicto es un proceso que empieza cuando una de las partes percibe que otro le está afectando negativamente. Y los conflictos contribuyen a que un equipo de desarrollo ejecute su trabajo adecuadamente y el resultado de un conflicto viene de:

- Una pobre comunicación
- De una falta de apertura por alguna de las personas
- Falla en la respuesta de alguna necesidad en particular

Existen dos clases de conflicto

- a) Conflicto Funcional.- que soporta las metas de un grupo y mejora su ejecución
- b) Conflicto Disfuncional.- que dificulta el desempeño de grupo

El proceso de conflicto tiene cinco estados:

1.- Incompatibilidad.-

- Comunicación.-Dificultades semánticas, malos entendidos, sobre la comunicación y el "ruido"
- Estructura.-El tamaño y la especialización de los puestos de trabajo, claridad jurisdiccionales / ambigüedad, Miembros / meta incompatibilidad, Estilos de liderazgo (cercano o participativo), Sistemas de recompensa (ganar-perder), la dependencia / interdependencia de los grupos
- Las variables personales.-Diferentes sistemas de valores individuales, Tipos de personalidad

2.- Cognición y personalización.-

- El conflicto puede ser percibido y no significa que se tome personal.
- El conflicto puede tener el nivel de sentirse y cuando esto llega emocionalmente a involucrarse las partes externan ansiedad, tensión, frustración y hostilidad.

3.- Intención

En la intención son las decisiones que se toman para actuar de una determinada manera donde se tiene diferentes estados conociendo el comportamiento entenderemos como responderá la contraparte. Aquí encontramos:

- El competitivo el cual busca satisfacer su propio interés en donde una persona puede ganar.
- Los que colaboran los cuales buscan dentro del conflicto el deseo de una completa satisfacción de todos y dentro del cual existe una cooperación para el beneficio mutuo dentro de este es el ganar-ganar.
- El que evita la negociación .- tratar de ignorar el conflicto y lo evitar, esta es una negociación perder-perder

- El que complace en todo lo que su contraparte dice sacrificando todo para mantener la relación, a esta negociación se le llama perder-ganar
- El comprometido el cual busca de su oponente una oportunidad para ser servicial , aceptando una solución que quizá provee una incompleta satisfacción

4.-Comportamiento.- Dentro de este estado incluye las declaraciones, acciones y reacciones que hacen que de un conflicto dentro de la administración de proyectos intenciones que desvían el propósito como los ataques verbales, las amenazas y las agresiones físicas un esfuerzo por destruir a la otra parte.

5.- Resultados. El conflicto tiene como resultado o un incremento en el desempeño del grupo o un decremento en el performance del grupo.

El conflicto es constructivo cuando este mejora la calidad de las decisiones, estimula la innovación, anima los intereses de todos y la curiosidad del grupo.

La siguiente figura 16 nos muestra los procesos de Conflicto.

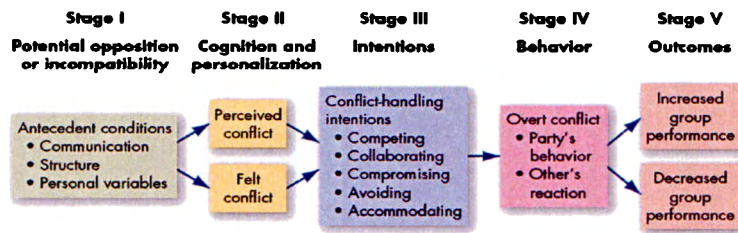


Figura 16

Negociaciones

La negociación es un proceso en el cual dos o más individuos intercambian mercancías, productos o servicios e intentan llegar a un acuerdo sobre el tipo de intercambio entre ellos.

El Administrador de proyecto debe de aprender las técnicas de negociación mediante las cual le permitirá desempeñar mejor su rol, tomando en cuenta, la meta, la motivación, el foco, los intereses y la información a compartir y la duración de la relación. [13 página 452]

La siguiente figura 17 nos muestra el Conflicto y la unidad de ejecución.

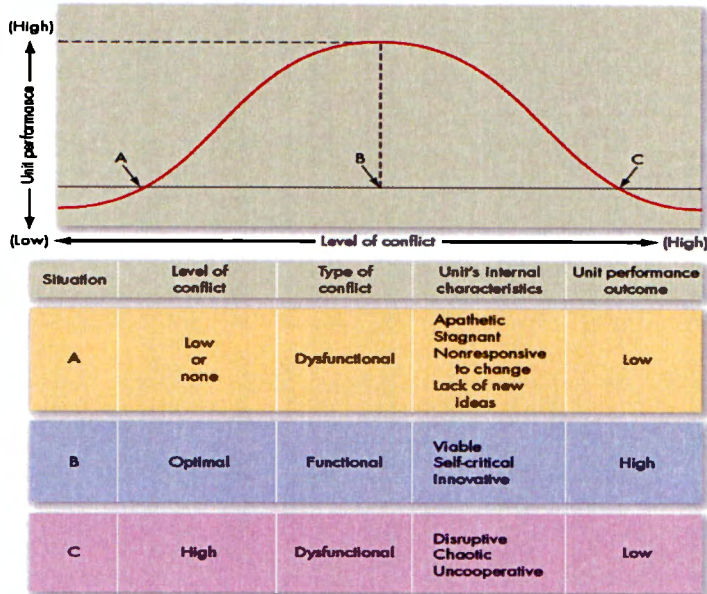


Figura 17

2.6.14 Cultura organizacional

La cultura organizacional es importante dentro de la administración de proyectos porque ayuda a los miembros que laboran en el equipo a distinguirse de otras organizaciones

Existen siete primeras características que son la esencia de la cultura organizacional

- 1.- El asumir riesgos y la innovación.- El cual es el grado en cual los empleados animados a innovar y a tomar riesgos.
- 2.- Atención a los detalles.- es el grado en cual se espera que todos los empleados exhiban la precisión, análisis y atención de los detalles.
- 3.- Orientado a resultados.- El grado en el cual los empleados tiende a dar resultados de acuerdo a la tarea o trabajo asignando.
- 4.- Orientada a Gente.- Es el grado el cual los empleados toman en consideración las decisiones de la alta dirección en este enfoque seria del administrador de proyectos.
- 5- Equipo orientado.- Es el grado en cual las actividades de trabajo son organizadas alrededor de los equipos
- 6.- Agresivos.- Es el grado en el cual la gente es competitiva y agresiva para desarrollar su trabajo en lugar de ser pasivos y tolerantes.

7.- Estabilidad.- El grado en el cual la organización mantiene la estabilidad de la organización o del proyecto. [13 pagina 518]

2.6.15 Políticas y Prácticas de recursos Humanos

Reclutamiento y Selección de personal.

El administrador de proyectos debe apoyarse del área de recursos humanos desde la selección de personal que trabajará en el proyecto hasta cumplir con la habilidad adecuada para desempeñar la actividad encomendada.

Recursos Humanos debe aplicar un adecuado proceso de selección de personal, ayudándose de los antecedentes de cada empleado, así como los exámenes psicométricos y técnicos que deba cubrir la vacante en cuestión. Desde ahí comienza el Administrador de proyectos a trabajar desde la revisión del personal a colaborar en el proyecto.

Algunos ejecutivos de compañías japonesas realizan la entrevista inicial en uno de las montañas más altas de Japón para medir la perseverancia y resistencia de los empleados.

Capacitación y desarrollo

Una vez seleccionado al personal, puede revisar alguna carencia de habilidades que haya encontrado durante el proyecto y tome la decisión inmediata de capacitar al personal mediante diferentes técnicas:

- 1.- Un curso para reforzar la habilidad técnica.
- 2.- Un curso de relaciones humanas.
- 3.- Que colaboré en un equipo en cual tienda a aprender de los demás mediante tareas sencillas e ir aumentando el grado de complejidad de la tarea hasta ponerlo a punto de los resultados que se debe entregar en el proyecto.
- 4.- Trabajo en equipo mediante tareas fuera del proyecto, como escalar una montaña en equipo, pasear en grupo a lugar, organizar un equipo de football interno. Entrenamientos flexibles que puedan hacerse en cualquier lugar y en cualquier tiempo.

Evaluación y desempeño

Una de las tareas que el Administrador de proyectos debe de hacer es la de evaluar a su personal mediante las diferentes técnicas de evaluación de recursos humanos. Se debe apoyar de Recursos Humanos para realizar esta tarea. (Las evaluaciones pueden ser en función de resultados, 360 grados etc.). Una vez realizada la evaluación se debe de retroalimentar a los colaboradores que trabajan en el proyecto con el fin de que mejoren su desempeño. [13 pagina 553]

2.6.16 Cambios organizacionales y Administración del estrés

Cambios organizacionales

Los proyectos y las organizaciones pueden tener cambios organizacionales que afecten a la organización o los proyectos para mejorar. Algunos puntos que deben ser considerados para que esto impacten menos son:

- 1.- Establecer una sensación de urgencia y crear una razón convincente de porque el cambio es necesario, se puede hablar de cambiar o morir en la organización
- 2.- Formar una coalición con el poder suficientemente para dirigir el cambio
- 3.- Crear una nueva visión del cambio con las estrategias adecuadas.
- 4.- Comunicar la visión en toda la organización.
- 5.- Capacitar a otros para actuar en la visión correcta removiendo las barreras necesarias y animando al cambio, tomando los riesgos necesarios e implementando soluciones creativas del cambio.
- 6.- Planear, crear y animar en términos de ganar hacia la nueva visión
- 7.- Consolidar las mejoras, reevaluar los cambios y hacer los ajustes necesarios dentro de un nuevo programa.
- 8.- Reforzar los cambios poniendo la demostración de la relación entre el nuevo comportamiento y el éxito de la organización o del proyecto.

Administración de Estrés

Uno de los principales puntos que un administrador de proyectos debe controlar es el estrés de su equipo de trabajo. El estrés es una condición dinámica dentro de la cual los individuos se ven confrontados ante una oportunidad, una demanda de trabajo o una relacionada con los recursos que los individuos desean acerca de los resultados y que son percibidos como inciertos e importantes.

Las principales fuentes de desgaste son:

Factores ambientales:- Económicos inciertos, Políticos inciertos, cambios tecnológicos.

Factores Organizacionales.- Demandantes tareas, demandantes roles, demandante relación interpersonal.

Factores Familiares.- problemas familiares, problemas económicos, personalidad.

Diferencias individuales.- percepciones, experiencia de trabajo, soporte social, hostil auto eficaz.

La figura 18 nos muestra las resistencias para el cambio tanto en los individuos como en las organizaciones.

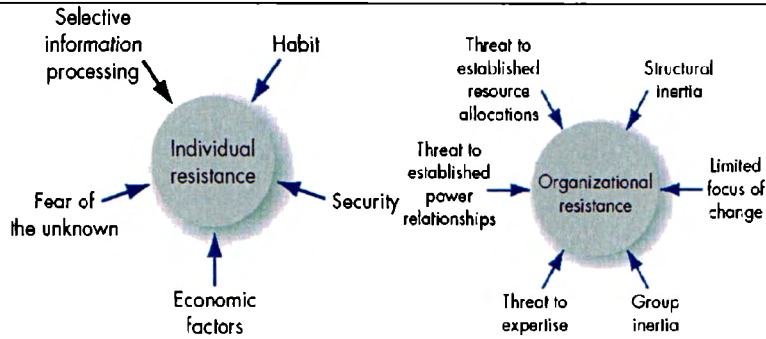


Figura 18.

Las consecuencias del stress son:

Síntomas fisiológicos.- Dolores de cabeza, Presión arterial, enfermedades del corazón.

Síntomas psicológicos.- Ansiedad, depresión, decremento en la satisfacción del trabajo.

Síntomas de comportamiento.- productividad, ausentismo, falta de personal.

Una de las primeras causas del estrés son las condiciones de trabajo y las causas son las diferencias culturales de las personas. Un ejemplo puede que una persona no se desenvuelve bien por estar lejos de su familia o el lugar de trabajo le quede excesivamente lejos.

Otra de las principales causas del estrés es el exceso de horas trabajadas que se puede reducir con diferentes recursos como amigos (soporte social), hablar con la familia.

Es de vital importancia que el administrador de proyectos controle el estrés mediante la observación de cada colaborador del proyecto. En ocasiones hay miembros del equipo que no hablan y no comentan la jornada de horas tan larga que se ha tenido, y cuando se le pide apoyo por lo general estos tiende a explotar. [13 pagina 558].

La figura 19 nos muestra el modelo del Stress.

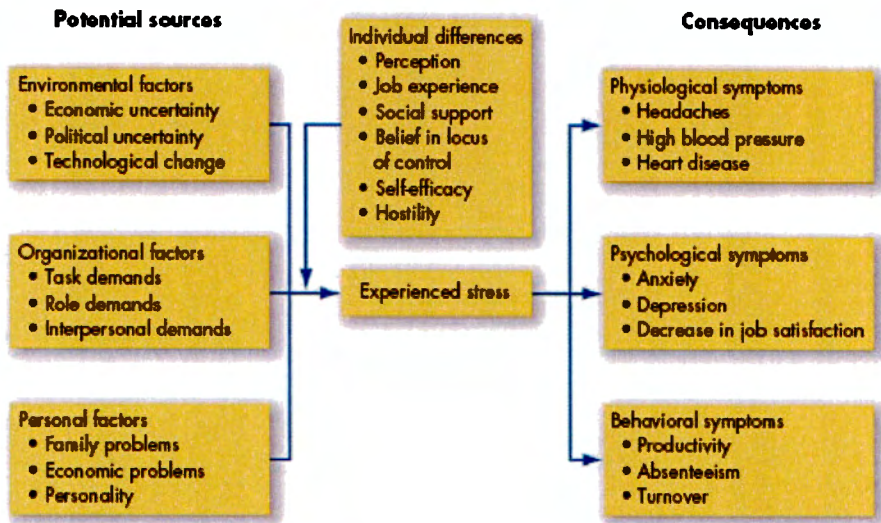


Figura 19

2.7 Tipos de desarrollo de Software:

2.7.1 Software de sistema:

El objetivo del software del sistema es desvincular adecuadamente al usuario y al programador de los detalles de la computadora en particular que se use, aislándolo especialmente del procesamiento referido a las características internas de: memoria, discos, puertos y dispositivos de comunicaciones, impresoras, pantallas, teclados, etc.

El software de sistema le procura al usuario y programador adecuadas interfaces de alto nivel, herramientas y utilidades de apoyo que permiten su mantenimiento. Incluye entre otros:

- ✓ Sistemas operativos
- ✓ Controladores de dispositivos
- ✓ Herramientas de diagnóstico
- ✓ Herramientas de Corrección y Optimización
- ✓ Servidores
- ✓ Utilerías de apoyo.

2.7.2.-Software de programación:

El software de programación es el conjunto de herramientas que permiten al programador desarrollar programas informáticos, usando diferentes alternativas y lenguajes de programación, de una manera práctica. Incluye entre otros:

Editores de texto

Compiladores

Intérpretes

Enlazadores

Depuradores

Entornos de Desarrollo Integrados (IDE): Agrupan las anteriores herramientas, usualmente en un entorno visual, de forma tal que el programador no necesite introducir múltiples comandos para compilar, interpretar, depurar, etc. Habitualmente cuentan con una avanzada interfaz gráfica de usuario (GUI).

2.7.3.-Software de aplicación:

Es aquel que permite a los usuarios llevar a cabo una o varias tareas específicas, en cualquier campo de actividad susceptible de ser automatizado o asistido, con especial énfasis en los negocios. Incluye entre otros:

Aplicaciones para Control de sistemas y automatización industrial

Aplicaciones ofimáticas

Software educativo

Software empresarial

Bases de datos

Telecomunicaciones (por ejemplo Internet y toda su estructura lógica)

Videjuegos

Software médico

Software de Cálculo Numérico y simbólico.

Software de Diseño Asistido (CAD)

Software de Control Numérico (CAM)

2.7.4 Proceso de creación del software

Se define como proceso al conjunto ordenado de pasos a seguir para llegar a la solución de un problema u obtención de un producto, en este caso particular, para lograr la obtención de un producto software que resuelva un problema.

El proceso de creación de software puede llegar a ser muy complejo, dependiendo de su porte, características y criticidad del mismo. Por ejemplo la creación de un sistema operativo es una tarea que requiere proyecto, gestión, numerosos recursos y todo un equipo disciplinado de trabajo. En el otro extremo, si se trata de un sencillo programa (por ejemplo, la resolución de una ecuación de segundo orden), éste puede ser realizado por un solo programador (incluso aficionado) fácilmente. Es así que normalmente se dividen en tres categorías según su tamaño (líneas de código) o costo: de Pequeño, Mediano y Gran porte. Existen varias metodologías para estimarlo, una de las más populares es el sistema COCOMO que provee métodos y

un software (programa) que calcula y provee una estimación de todos los costos de producción en un "proyecto software" (relación horas/hombre, costo monetario, cantidad de líneas fuente de acuerdo a lenguaje usado, etc.).

Considerando los de gran porte, es necesario realizar complejas tareas, tanto técnicas como de gerencia, una fuerte gestión y análisis diversos (entre otras cosas), por lo cual se ha desarrollado una ingeniería para su estudio y realización: es conocida como Ingeniería de Software.

En tanto que en los de mediano porte, pequeños equipos de trabajo (incluso un avezado analista-programador solitario) pueden realizar la tarea. Aunque, siempre en casos de mediano y gran porte (y a veces también en algunos de pequeño porte, según su complejidad), se deben seguir ciertas etapas que son necesarias para la construcción del software. Tales etapas, si bien deben existir, son flexibles en su forma de aplicación, de acuerdo a la metodología o Proceso de Desarrollo escogido y utilizado por el equipo de desarrollo o por el analista-programador solitario (si fuere el caso).

Se estima que, del total de proyectos software grandes emprendidos, un 28% fracasan, un 46% caen en severas modificaciones que lo retrasan y un 26% son totalmente exitosos. [2]

Cuando un proyecto fracasa, rara vez es debido a fallas técnicas, la principal causa de fallos y fracasos es la falta de aplicación de una buena metodología o proceso de desarrollo o en la buena APS.

Crisis del software

- ✓ El 30% de los proyectos software son cancelados
- ✓ El 50% de los proyectos son abandonados o exceden los costos previstos
- ✓ El software falla a menudo dada su baja calidad en un 60%
- ✓ La entrega del software es tardía en 9 de cada 10 proyectos

Causas del Fracaso

- ✓ Falta de gestión
- ✓ Estimación inexacta
- ✓ Comunicación del estado del proyecto
- ✓ Falta de información histórica
- ✓ Incapacidad de gestionar el proceso software.

Capítulo 3 Caso Práctico

3.1 PROPUESTA DE SOLUCION CLIENTE ANONIMO

3.1.1 Objetivo

Apoyar a CLIENTE en el desarrollo e implantación del Sistema de Administración de Recursos, el cual satisfaga en forma permanente los requerimientos actuales, permitiendo simplificar las operaciones y funciones que de esta se deriven.

CLIENTE desea establecer un sistema que le permita una mejor administración y un mejor control de cada una de las etapas del proceso de fabricación de producto es y de sus puntos de venta.

3.1.2. Alcance

El alcance del desarrollo incluirá las principales funciones que se manejan en el Sistema de Administración de Recursos de productos.

3.1.2.1.- El desarrollo de los siguientes módulos:

➤ PRODUCCION

- Planeación de la producción
 - Generación de plan de producción (pedidos especiales, stock de tiendas)
 - Modificación del plan de producción
 - Consulta del plan de producción
 - Baja del plan de producción
 - Aprobación del plan de producción
 - Generar ordenes de producción
 - Generar requisiciones al almacén general
- Ordenes de producción
 - Alta de ordenes de producción
 - Modificación de ordenes de producción
 - Consulta de ordenes de producción
 - Cancelación de ordenes de producción
 - Aprobación de ordenes de producción

- Seguimiento a las ordenes de producción
- Aprobación de estación de trabajo
- Costeo de productos para recetas
- PUNTO DE VENTA
- Caja
 - Registro de Ventas
 - Cancelación de Ventas
 - Registro de Ventas Perdidas
 - Impresión de Ticket's
 - Impresión de Facturas
 - Impresión de Remisiones
 - Reimpresiones
 - Devoluciones y Cancelaciones de Clientes
 - Corte de Caja
 - Retiros de Efectivo
 - Factura Global del día con ventas de ticket's o documentos NO fiscales
 - Facturación de Remisiones
 - Pedidos especiales
 - Generación de Pedidos Especiales
 - Sustitución de Pedidos Especiales
 - Pedidos a entregar
 - Generación de orden de producción
- Control del inventario
 - Registro de entradas
 - Registro de salidas
 - Consulta de inventario
 - Control de Vida de Anaquel
 - Ajustes
 - Traspasos
- Reportes

- Emisión de Lista de Precios
- Reporte Conteo Inicial Existencias Iniciales
- Formato de Pedido de Producción
- Formato de Entradas de Producción a Tiendas
- Formato de Salidas por Baja
- Vales
- VENTAS
 - Descuentos a Clientes de Mayoreo
 - Promociones de Temporada
 - Vendedores
 - Altas a Vendedores
 - Administración de Asistencias
 - Comisiones sobre ventas
 - Reportes
 - Hoja de Producción
 - Tablero de Indicadores
- ADMINISTRACIÓN DE ALMACENES
 - Alta de almacenes
 - Modificación de almacenes
 - Consulta de almacenes
 - Baja de almacenes
 - Inventarios
 - Entradas a los almacenes
 - Salidas de los almacenes
 - Ajustes
 - Traspasos
 - Generación de stock's
- ALMACÉN GENERAL
 - Requisiciones
 - Alta de requisiciones
 - Modificación de requisiciones

- Consulta de requisiciones
- Cancelación de requisiciones
- Aprobación de requisiciones
- Seguimiento a las requisiciones
- Entradas
- Salidas
- Ajustes
- Traspasos
- Reporte de Costos por área
- Clasificación de productos ABC
- COMPRAS
 - Ordenes de compra
 - Alta de órdenes de compra
 - Modificación de órdenes de compra
 - Consulta de órdenes de compra
 - Cancelación de órdenes de compra
 - Aprobación de órdenes de compra
 - Seguimiento a las órdenes de compra
 - Generación de Cuentas por Pagar
- DISTRIBUCIÓN
 - Ordenes de entrega
 - Alta de ordenes de entrega
 - Modificación de ordenes de entrega
 - Consulta de ordenes de entrega
 - Cancelación de ordenes de entrega
 - Aprobación de ordenes de entrega
 - Seguimiento a las ordenes de entrega
 - Definición de Políticas de Entrega
 - Control de Implementos de Distribución
 - Control de Incidencias
 - Clasificación Conductor

- Rutas
 - Generación de rutas
 - Modificación de rutas
 - Asignación de prioridades
- Entrega / Recepción en Sucursal
- Administración de la Flotilla
 - Captura de Kilometraje
 - Consumo de Combustible
 - Alertas de Mantenimiento
- CONTABILIDAD
 - Concentrado de Ingresos por Tienda y por Mes
 - Formato de Conteo Fácil de Efectivo
 - Pólizas (Movimientos en Caja, Producción, Traspasos)
 - Generación de Cuentas por Pagar
 - Archivo de Entradas a Almacén
- ADMINISTRACIÓN DEL SISTEMA
 - Productos (Recetas)
 - Alta de productos
 - Modificación de productos
 - Consulta de productos
 - Baja de productos
 - Materiales (Materia prima, consumibles, otros)
 - Alta de materiales
 - Modificación de materiales
 - Consulta de materiales
 - Baja de materiales
 - Estaciones de trabajo
 - Alta de estaciones de trabajo
 - Modificación de estaciones de trabajo
 - Consulta de estaciones de trabajo
 - Baja de estaciones de trabajo

- Relación Productos-Estaciones de trabajo
 - Alta de relación
 - Modificación de relación
 - Consulta de relación
 - Baja de relación
- Proveedores
 - Alta de proveedores
 - Modificación de proveedores
 - Consulta de proveedores
 - Baja de proveedores
- Políticas de Proveedores
 - Alta de políticas de proveedores
 - Modificación de políticas de proveedores
 - Consulta de políticas de proveedores
 - Baja de políticas de proveedores
- Vehículos
 - Alta de vehículos
 - Modificación de vehículos
 - Consulta de vehículos
 - Baja de vehículos
- Conductores
 - Alta de conductores
 - Modificación de conductores
 - Consulta de conductores
 - Baja de conductores
- Clientes
 - Alta de Clientes
 - Modificación de Clientes
 - Consulta de Clientes
 - Baja de Clientes

3.1.2.2.- Mantenimiento de información.

- Procesos de respaldo y recuperación de información
- Proceso de migración de información a base de datos histórica

3.1.3. Beneficios

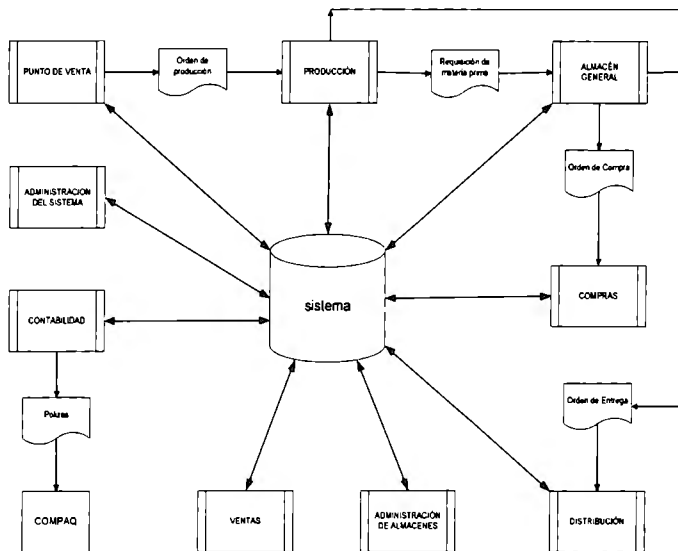
Dentro de los principales beneficios se encuentran:

- Centralización de la información
- Información real y oportuna para la toma de decisiones
- Disminución en tiempos de proceso en las tareas a automatizar
- Eliminación de recaptura de información.
- Conexión lógica de las áreas para un flujo efectivo del proceso.

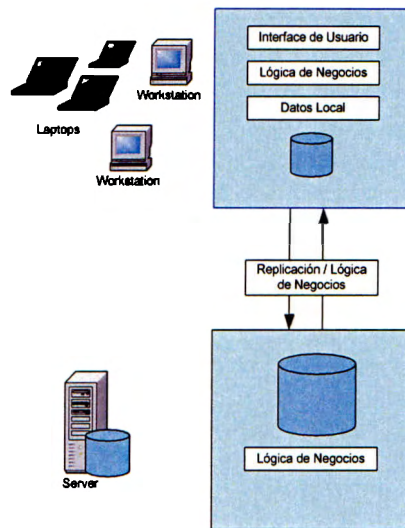
3.1.4. Descripción del producto

A continuación se muestra el diagrama conceptual que define el Sistema de Administración de Recursos de producto:

SISTEMA DE ADMINISTRACIÓN DE RECURSOS



La solución tecnológica para los puntos de venta se basa en una solución Smart Client la cual se define con los siguientes diagramas.



I.- MODULOS DE DESARROLLO.

Se propone desarrollar el sistema en la siguiente secuencia:

1. PRODUCCION
2. PUNTO DE VENTA
3. VENTAS
4. ADMINISTRACIÓN DE ALMACENES
5. ALMACÉN GENERAL
6. COMPRAS
7. DISTRIBUCIÓN
8. CONTABILIDAD
9. ADMINISTRACIÓN DEL SISTEMA

II Mantenimiento de Información

Contar con procesos de respaldo y restauración de información así como un proceso de migración de información a base de datos histórica.

3.1.5. Plataforma Tecnológica

Hardware

- LAN para la operación en la planta, comunicación con las sucursales a través de Internet.
- Punto de venta
- Dispositivos de reconocimiento de huella digital
- Dispositivos de lectura e impresión de código de barras

Software

- Front End JSP
- Reglas de negocio Java
- Base de datos PostgreSQL
- Sistema Operativo Linux Suse (por definir)

- **Software de Desarrollo** (no requerido por CLIENTE)

- Eclipse
- Cliente de PostgreSQL
- Magic

3.1.6. Productos a entregar

Se entregará como parte del desarrollo del "Sistema de Administración de Recursos":

Entregable / Hito	PUNTO DE VENTA	PRODUCCION	OTROS
Planeación			
Estimados	29/02/2008	29/02/2008	30/05/2008
Plan de trabajo	29/02/2008	29/02/2008	30/05/2008
Plan Operativo	29/02/2008	29/02/2008	30/05/2008
Análisis			
Especificación de requerimientos	29/02/2008	29/02/2008	30/05/2008
Casos de uso	31/03/2008	31/03/2008	30/06/2008
Diseño			
Modelo arquitectónico	30/04/2008	30/04/2008	31/07/2008
Interfaz de usuario	30/04/2008	30/04/2008	31/07/2008
Diagramas de clases	30/04/2008	30/04/2008	31/07/2008
Diagramas de secuencia	30/04/2008	30/04/2008	31/07/2008
Desarrollo			
Codificación	20/06/2008	11/07/2008	19/09/2008
Pruebas unitarias	20/06/2008	11/07/2008	19/09/2008
Pruebas			
Casos de prueba	20/06/2008	11/07/2008	19/09/2008

Pruebas de aceptación de usuario	11/07/2008	31/07/2008	30/09/2008
Liberación			
Paquete del producto	31/07/2008	29/08/2008	31/10/2008
API del sistema	31/07/2008	29/08/2008	31/10/2008
Manual de usuario	31/07/2008	29/08/2008	31/10/2008
Manual técnico	31/07/2008	29/08/2008	31/10/2008
Capacitación	31/07/2008	29/08/2008	31/10/2008
Archivos de instalación	31/07/2008	29/08/2008	31/10/2008
Instalación	31/07/2008	29/08/2008	31/10/2008
Carta de aceptación	31/07/2008	29/08/2008	31/10/2008

RELACION DE ENTREGABLES VS FECHAS.

Primer Entregable:

Planeación de Punto de Venta y Producción

- ✓ Estimados
- ✓ Plan de trabajo
- ✓ Plan Operativo

Análisis de Punto de Venta y Producción.

- ✓ Especificación de requerimientos

Fecha: 29 de Febrero del 2008

Segundo Entregable::

Análisis de Punto de Venta y Producción.

- ✓ Casos de Uso

Fecha: 31 de Marzo del 2008

Tercer Entregable:

Diseño de Punto de Venta y Producción.

- ✓ Modelo arquitectónico
- ✓ Interfaz de usuario
- ✓ Diagramas de clases
- ✓ Diagramas de secuencia

Fecha: 30 de Abril del 2008

Cuarto Entregable:

Planeación de otros (Almacén General, Compras, Distribución, Ventas, Contabilidad Administración de almacenes y Administración del sistema):

- ✓ Estimados
- ✓ Plan de trabajo
- ✓ Plan Operativo

Análisis de otros (Almacén General, Compras, Distribución, Ventas, Contabilidad Administración de almacenes y Administración del sistema):

- ✓ Especificación de requerimientos

Fecha: 31 de Mayo del 2008

Quinto Entregable:

Análisis de otros (Almacén General, Compras, Distribución, Ventas, Contabilidad Administración de almacenes y Administración del sistema):

- ✓ Casos de Uso

Desarrollo de Punto de Venta.

- ✓ Codificación
- ✓ Pruebas unitarias

Pruebas de Punto de Venta

- ✓ Casos de prueba

Fecha: 30 de Junio del 2008

Sexto Entregable:

Diseño de otros (Almacén General, Compras, Distribución, Ventas, Contabilidad Administración de almacenes y Administración del sistema):

- ✓ Modelo arquitectónico
- ✓ Interfaz de usuario
- ✓ Diagramas de clases
- ✓ Diagramas de secuencia

Desarrollo de Producción.

- ✓ Codificación
- ✓ Pruebas unitarias

Pruebas de Producción.

- ✓ Casos de prueba

Pruebas de Punta de venta y Producción

- ✓ Pruebas de aceptación de usuario.

Liberación de Punto de Venta.

- ✓ Paquete del producto
- ✓ API del sistema
- ✓ Manual de usuario
- ✓ Manual técnico
- ✓ Capacitación
- ✓ Archivos de instalación
- ✓ Instalación
- ✓ Carta de aceptación

Fecha: 31 de Julio del 2008

Séptimo Entregable:

Liberación de Producción.

- ✓ Paquete del producto
- ✓ API del sistema
- ✓ Manual de usuario
- ✓ Manual técnico
- ✓ Capacitación
- ✓ Archivos de instalación
- ✓ Instalación
- ✓ Carta de aceptación

Fecha: 30 de Agosto del 2008

Octavo Entregable:

Desarrollo de otros (Almacén General, Compras, Distribución, Ventas, Contabilidad Administración de almacenes y Administración del sistema):

- ✓ Codificación
- ✓ Pruebas unitarias

Pruebas de otros (Almacén General, Compras, Distribución, Ventas, Contabilidad Administración de almacenes y Administración del sistema):

- ✓ Casos de prueba
- ✓ Pruebas de aceptación de usuario

Fecha: 31 de Septiembre del 2008

Noveno Entregable:

Liberación de otros (Almacén General, Compras, Distribución, Ventas, Contabilidad Administración de almacenes y Administración del sistema: Paquete del producto

- ✓ API del sistema
- ✓ Manual de usuario
- ✓ Manual técnico
- ✓ Capacitación
- ✓ Archivos de instalación
- ✓ Instalación
- ✓ Carta de aceptación

Fecha: 30 de Octubre del 2008

3.2. METODOLOGÍA

3.2.1. Etapas a seguir en el Mantenimiento desarrollo del sistema

Al iniciar el proyecto se llevará a cabo en conjunto con CLIENTE una sesión de planeación detallada de las actividades del mismo.

Para implementar sistemas de información se han identificado estos pasos a seguir, los cuales deberán de ejecutarse en la secuencia que se marca, aunque es importante mencionar que algunos de estos deberán de llevarse en paralelo para asegurar el éxito de la puesta en marcha del sistema.

3.2.1.1. Definición del Plan Operativo

El objetivo de esta fase es definir el modelo de ciclo de vida apropiado a la magnitud y complejidad del proyecto para establecer los procesos, actividades, recursos, infraestructura y tareas necesarias para desarrollar el producto de software, e incluye los siguientes puntos:

- Definición conceptual y operacional del sistema
- Descripción de requerimientos de software y de sistema
- Descripción de la arquitectura del sistema
- Definición del ciclo de vida del proyecto

3.2.1.2. Definición de Análisis y Diseño del Sistema

El objetivo de esta fase es el definir por escrito las características detalladas del sistema, generar la estructura general del sistema a desarrollar, identificando los módulos, interfaces, funciones y el esquema general de la base de datos..

- Definición operacional del sistema
- Generación de prototipo de pantallas.
- Descripción de diseño general de la Base de Datos

3.2.1.3. Diseño detallado y Construcción

En esta fase se detallan sucesivamente las funciones hacia niveles de mayor detalle, obteniendo las especificaciones de programa de cada componente; produciendo unidades que puedan ser codificadas, compiladas, y probadas. También se generan planes de desarrollo e integración de software, se construyen los programas que deben de cumplir con las funciones definidas para el sistema a partir de las especificaciones técnicas detalladas de cada uno. Deben probarse los programas individualmente, generando un Check List (lista de casos de prueba) los cuales se implementan para asegurar el buen funcionamiento de los mismos.

- Generación de especificaciones de programas
- Definición de interfaces
- Construcción de programas
- Construcción de la Base de Datos
- Prueba unitaria del sistema

3.2.1.4. Implantación del sistema

Integrar cada componente de software tomando en cuenta los requerimientos de prueba, procedimientos, datos, responsabilidades y calendarización de las pruebas, de acuerdo a los requerimientos de certificación del sistema. Asegurar que la implantación de cada requerimiento del sistema sea probada, y que el sistema se encuentre listo para la entrega.

- Integración del sistema
- Pruebas de Integración del sistema
- Pruebas de aceptación del sistema
- Generación de documentación
- Capacitación
- Instalación del sistema

3.2.2. Esfuerzo requerido

De acuerdo con el Análisis de la información de las funciones detectadas para este proyecto y de acuerdo a la experiencia de Sinersys en el Desarrollo de Sistemas de

Información, a los promedios de tiempo de desarrollo aplicados por tipo y complejidad de procesos; hemos estimado el esfuerzo requerido por fase del proyecto y sería de :

12,600 horas/hombre equivalentes a 75 meses/hombre.

3.2.3 Tiempo requerido

El tiempo calendario requerido para realizar el esfuerzo antes mencionado es de 9 meses

3.2.4. Equipo de trabajo

Para llevar a cabo el proyecto se ha considerado el integrar en las distintas fases del ciclo de desarrollo del proyecto a los siguientes recursos:

1 Líder de proyecto por un tiempo de 9 meses
1 arquitecto por un tiempo parcial
8 Ingenieros de Sistemas por un tiempo de 8 meses

3.2.5. Plan calendario

De 1 Febrero del 2008 a 30 Octubre del 2008

3.2.6. Metodología de programación

En lo que se refiere a desarrollo de aplicaciones, se cuenta con una metodología de programación orientada a objetos propia, así como estándares de programación y para este proyecto podrían utilizarse los estándares de Sinersys de común acuerdo con CLIENTE.

3.2.7. Capacitación y transferencia de habilidades

Para proporcionar la capacitación y transferencia de habilidades y conocimientos, se propone programar sesiones en el lugar de trabajo una vez liberado el sistema a producción. La capacitación se proporcionará al grupo de personas determinado por CLIENTE, además, Sinersys permanecerá una semana en las instalaciones de CLIENTE, después de instalar el software para proporcionar una capacitación en el trabajo a los operadores y usuarios.

Se recomienda que el personal coordine su horario para asistir a las sesiones apropiadas de acuerdo con los programas de capacitación, los cuales serán programados y descritos de común acuerdo con CLIENTE.

3.3. DEPENDENCIAS Y RESPONSABILIDADES

3.3.1. Requerimientos de desarrollo e instalación

La infraestructura de hardware, software y networking necesarios para las pruebas integrales del desarrollo del servicio serán proporcionados por y en las instalaciones de CLIENTE.

Las áreas de trabajo, salas de juntas y servicio telefónico, así como el acceso a las mismas las 24 horas serán proporcionados por CLIENTE en las localidades que la misma designe en caso de ser necesario.

Infraestructura de hardware y software necesaria.

- Servidor
 - Procesador Dual Core Intel Xenon 3050, 2MB Cache, 2.13 Ghz
 - 8 GB RAM
 - 500 GB DD
- Sistema Operativo
 - Linux Suse (por definir)
- Servidor de Aplicaciones
 - Tomcat
- Base de datos
 - PostgreSQL
- Infraestructura de red

- Proporcionar casos de prueba suficientes que aseguren el correcto funcionamiento del sistema y las interfaces del "SISTEMA DE ADMINISTRACIÓN DE RECURSOS" tanto en su operación normal como excepcional.

3.3.2. Requerimientos para la operación

- Establecer las responsabilidades de cada área involucrada en el manejo de la información.
- Captura total y completa de los datos requeridos por el producto por parte de CLIENTE con excepción de los datos de los usuarios y los catálogos.
- Apego total a los procedimientos de operación establecidos de CLIENTE por parte del usuario final.
- CLIENTE deberá llevar a cabo las acciones necesarias para el acceso al servidor del "SISTEMA DE ADMINISTRACION DE RECURSOS", así como la coordinación para la capacitación a su personal, en el proceso de captura y procedimientos a seguir en el uso del sistema.

3.4.- Lecciones aprendidas

Comentarios de lecciones aprendidas de un desarrollador del proyecto (tiene características hábiles en la parte de desarrollo):

- Como programador, es bueno siempre probar todo el módulo cuando éste sufre un cambio, por mínimo que sea.
- Nunca debe obviarse nada, mucho menos en el análisis (Casos de Uso) o en el diseño (diagramas).
- El papel de un buen líder jamás queda por encima de un buen programador, es decir, si hubiera un mal líder, no importaría lo bueno que sean los programadores, las cosas no saldrían ni en tiempo ni en forma.
- El cliente siempre tiene la razón, pero nunca sabe lo que quiere, o no conoce su propio negocio, o no sabe de sistemas.
- Precisamente por eso, nunca debe decirse que 'Sí' a todo lo que el cliente pide.
- Precisamente por eso también, el papel de un consultor es mostrarse experto, asesorarle y así brindarle confianza en nuestro trabajo.
- Además, no se debe decir 'Sí' a todo lo que el cliente pide si no se sabe con certeza (mínima por lo menos) de que es posible, funcional y eficiente (porque puede ser funcional pero tardar 3 horas en ejecutarse y así, por muy bueno que sea, pierde la eficacia, o puede ser rápido, pero la programación tarda mucho más tiempo del esperado).
- Se predica con el ejemplo, no basta con pedir, también hay que hacer.
- La buena comunicación empieza por que los de arriba comuniquen bien.
- Un buen desarrollo es el arma que se tiene para que la instalación sea fácil, rápida y eficiente.
- Un buen diseño es el arma que se tiene para que el desarrollo sea bueno.
- Un buen análisis es el arma que se tiene para que el diseño sea bueno.
- Un buen levantamiento de requerimientos es el arma que se tiene para que el análisis sea bueno.
- Una buena negociación es el arma que se tiene para que el levantamiento sea bueno.
- Para algo existen los procesos, ¡hay que usarlos!
- Los casos de prueba son fundamentales para que en el desarrollo se pueda probar todo bien y como se debe.
- Los casos de uso son fundamentales para que la generación de los casos de prueba se de bien y como se debe.
- Lo que es obvio para unos, no lo es para los demás.
- No importa qué tan mínimo sea el detalle o cuántas veces se haya mencionado, siempre es necesario repetirlo una vez más, porque no siempre se recuerda si 'X' persona lo sabe, lo escuchó o ya se le dijo pero simplemente lo olvidó, así que, ahorras tiempo a la larga perdiendo 5 min. más explicando, por muy lógico que esto parezca.

- La palabra directa no es el medio de confirmación ideal, el medio escrito (como el email) es el medio de confirmación ideal.
- Para algo la ingeniería de sistemas es eso, una ingeniería, no se analiza por nada, ni se diseña por nada, ni se implementa por nada, ni se documenta por nada: **TODO ES IMPORTANTE.**
- Un análisis, diseño y desarrollo que piensan que son perfectos a la primera, provocan el sistema más imperfecto que se pueda imaginar. **SIEMPRE HAY QUE PENSAR EN EL MANTENIMIENTO.**
- Además, pensar en el mantenimiento es un acto de humildad y caridad, o cuando menos de auto-ayuda, para quien tenga que darle mantenimiento a nuestro proyecto en el futuro.
- Además, uno nunca sabe lo que al cliente se le puede ocurrir que haga mañana SU sistema.
- Con una buena administración del proyecto, si el cliente pide algo, se sabe a la primera: 1) negociar, 2) determinar su alcance y 3) ubicarlo como requerimiento o como hallazgo de lo que ya se tenía.
- Si el proyecto es orientado a objetos, vamos a analizar, diseñar y no solamente desarrollar orientado a objetos, pero con diseñar no solamente me refiero a hacer diagramas de clases ¡hay que pensar orientado a objetos para hacer un proyecto orientado a objetos!
- No es un defecto que no conozcamos la tecnología para el desarrollo, es un defecto no querer aprenderla o no querer sacrificar tiempo en hacerlo. Al fin y al cabo todos crecemos si aprendemos algo nuevo.
- Un Framework (como JSF, Spring o Hibernate) nos facilita la vida, sólo hay que saber utilizarlo y no por desconocido o complicado desecharlo y descalificarlo.

Lecciones aprendidas de un Analista del proyecto:

No.	Área	Descripción	Sugerencia
1	Elaboración de la Propuesta técnica	Al hacer la propuesta técnica considero que se debe aondar un poco más (ser más específicos) ya que normalmente se hace la negociación a nivel "Conceptos" del usuario y negociador habla a nivel "Casos de Uso", se cierra la negociación y al realizar el análisis esos Conceptos pueden incluir <u>n</u> casos de uso y que al plasmarlo en el plan de trabajo ya no cuadran los tiempos y estos quedan muy reducidos para su elaboración.	Realizar las propuestas técnicas mas específicas, si es necesario cubrir algunos puntos de la Especificación de requerimiento para que el alcance sea mas real.
2	Documentación de Calidad	Los documentos solicitados por el área de Calidad deben de ser revisados, ya que es información redundante que implica tiempo para el líder de proyecto. En el caso de cliente x como se dividió en 3 Módulos (FV, Producción y Otros) implicaba llenar 15 documentos a la semana, sin contar los que se debían actualizar diario. Nota: Estos formatos no cumplieron su Objetivo, ya que no se percataron del Impacto del Atraso.	Revisar documentación del CMMI y modificar y/u optimizar formatos
3	Poco interes de las personas interesadas	La división por áreas ocasionó que los Interesados (directivos) no prestaran atención a los proyectos y la información sobre los proyectos no tuvo el ninguna reacción hasta que el cliente los involucro.	Dar impotancia a los retrasos para tomar deiciones oportunamente.
4	Inexperiencia del Líder de proyecto	la inexperiencia del líder del proyecto por suponer que cosas que iban saliendo eran correctas y suponer que los dirigentes estaban enterados de la situación. Deficiencias y Errores: - No informar que los casos de uso resultantes después del análisis fueron mucho mas de los considerados en la propuesta técnica. - No haberme asegurado de que la gente interesada estuviera enterada del estatus del proyecto, insistir personalmente. - No haber consultado con expertos acerca del diseño propuesto. - No involucrarse mas en el análisis del sistema para apoyar a los analistas. - Desconocimiento de la plataforma de desarrollo. - Inseguridad para tomar las decisiones por desconocimiento de herramientas. -	Cuando el lider de proyecto no cuenta con un líder técnico, si es importante conocer un poco la plataforma y herramientas que se usaran para tener confianza a tomar deiciones.
5	Desconocimiento de la plataforma seleccionada para el desarrollo	El uso de una herramienta nueva que muy poca gente conoce y tiene experiencia causo demasiado retraso, por los siguientes puntos: - No se encontro gente que conociera la herramienta por lo tanto se contrato gente sin experiencia. - Los recursos se encontraron con situacioens que no podían resolver y se apoyaron en conferencias de internet para buscar a alguien que pudiera resolver los problemas. las respuestas tardaban de 1 a 3 días cuando eran resueltas; algunas veces semanas. - por la desconocimiento de la herramienta se programó de cierta forma que parecia normal pero al realizar pruebas en producción causaron problemas. Ejemplo: no se cerraban las sesiones que se abrien por lo tanto se saturaba la memoria despues de un rato de trabajar en ella. - No se conocían las restricciones de las herramientas, ni la forma mas óptima de programar con ellas por lo que la programación quedo muy ineficiente y difícil de dar mantenimiento.	Buscar plataformas de programación conocidas, de la cual existan expertos, manuales e información facil de buscar para resolver problemas en corto tiempo.
6	Falta de un líder técnico.	Se tuvo líder técnico por un periodo muy corto y solo de medio tiempo, esto perjudico mucho al sistema: La falta de este ocasionó: - Apoyo casi nulo a los programadores sin experiencia - Tiempos muy extendidos para resolver dudas. - Programación deficiente - Programación sin estandares. - Poco control en los cambios de los ambientes - Poco reuso de los componentes -	Se debería de hacer no solo una revisión a nivel documentos sino tambien programación para asegurarse que la gente programa sencilla para facilitar mantenimientos posteriores.
7	Carencia de Expertos en Sinersys	Sinersys no contaba (o cuenta) con un grupo de expertos identificado que puedan apoyar en ciertas situaciones. (Entiendase por "Expertos" a una persona que ha trabajado con la herramienta mas de 2 años). - Los "Expertos" que apoyaron para revisiones contaban con apenas unos meses de experiencia, por lo tanto no sugirieron mejoras al proyecto ni visualizaron problemas posteriores. Normalmente sus comentarios eran "Esta bien".	Sinersys debería identificar gente que ha trabajado con ciertas herramientas y cuenta con la experiencia para poder apoyar o dar su punto de vista cuando se requiera. NO FAVORITISMOS.

8	Evaluaciones de Calidad	<p>Si se cuenta con grupo de Calidad y se revisiones estas deberían de estar mas enfocadas a la Calidad de los documentos, es decir, que estos sean claros para los lectores sin que tengan que se expertosen la materia.</p> <ul style="list-style-type: none"> - Se levantaron muchos hallazgos en donde los casos de uso no tenían la Fecha correcta, o el encabezado correcto pero jamás levantaron un hallazgo en donde se dijera que el caso de uso no era claro, o estuviera incompleto para un programador. <p>Los casos de uso son generados por analistas pero muchas veces no son tan descriptivos y esto implica mucho retrabajo y tiempo de preguntas del programador ya que no es claro.</p> <p>En el caso de cliente x: se rehicieron el 90% de los casos de uso.</p>	<p>Si se va a contar con un area de calidad entonces esta debe de ampliar su forma de evaluación.</p>
9	Programadores	<p>No solo la inexperiencia de los programadores afecto al proyecto: esto provoca:</p> <ul style="list-style-type: none"> - Frustración en ellos - Invertir mucho tiempo en el desarrollo, estar mas tiempo en la oficina - Conflictos internos (cuestiones economicas) - Los días de trabajo se extendieron por varios meses por lo cual hubo desanimo y cansancio - No recibieron motivación. <p>Deficiencias que tuvieron:</p> <ul style="list-style-type: none"> - Falta de conocimiento en la herramienta - En algunos casos inexperiencia laboral - No saben probar sus desarrollos por lo tanto los entregan con mucha deficiencias, lo cual implicaba mucho tiempo en revisiones y pruebas. en muchao casos las etapas de Prueba y Corrección consumian más tiempo el el desarrollo inicial. - No sabían medir sus tiempos para el desarrollo. Después de que supieron medir sus tiempos, estos se extendieron mucho, hubo un caso que para generar un reporte se estimó un mes y fue aprobado por el LT en turno. 	<p>Si no es posible capacitar a los programadores al menos darle opciones para que puedan resolver dudas como una gente con experiencia.</p> <p>Se debe de implementar una revision entre los programadores no solo de codigo sino de probar la funcionalidad de los programas.</p>
10	Deficiencias del cliente	<p>el cliente tenia muchas deficiencias internas, algunas de ellas:</p> <ul style="list-style-type: none"> - Carencia de procesos internos y desconocimientos de sus funciones - Una persona tenia el control de la mayonia de las areas - Mucha rotación de personal - Poco conocimiento de sistemas y no contar con una persona o area de sistemas - No se tenia una clara vision de lo que querían. - Temor de expresar sus ideas. 	<p>La documentacion como casos de uso y Diseño tiene que estar enfocado hacia ellos, es decir presentar la descripción de la funcionalidad y propuestas de pantallas asi como los reportes que recibiran para que visualicen si es lo que quieren.</p>
11	Conflictos externos al sistema	<p>Mucha rivalidad de gente externa al proyecto que se involucro y provocaron divisiones en el equipo.</p> <p>Nota: mas parecia "Lucha de poderes que Trabajo en equipo".</p>	<p>Aun no se como controlar o mitigar esto</p>

Comentarios de lecciones aprendidas de un Administrador de proyectos del proyecto:

Fortalezas	Debilidades	Recomendaciones
Utilización de nuevas herramientas	No se realizo el desarrollo en tiempo	Un curso de 1 semana en la tecnología nueva
Desempeñar papel de desarrollador en multiples modulos del sistema	Mala comunicación	Que más personas desempeñen el papel de los distintos módulos del sistema
Se comienza a usar más el WBS	Mala organización	Se utilicen mas procesos
	Le decían a todo que "si" a las peticiones del cliente	Mejor análisis y diseño del proyecto
	Falta un poco mas de análisis en muchos de los CU	Que todas las Unidades involucradas se coordinen para mejorar las propuestas, planes, desarrollo, etc.
	Falta de casos de prueba	Motivar a los colaboradores con objetivos por entregables
	Falta de planes de trabajo (o de comunicación de los mismos) a los involucrados	Dar a conocer al equipo de trabajo los objetivos y alcance general del proyecto
	Desorden en código (no se cumplieron estándares)	
	Mejores pruebas unitarias	
	Sinensys asumió la responsabilidad del manejo de la información cuando debería ser responsabilidad del cliente	
	Se desarrollaron requerimientos sin documentación	
	No se llevo un control de cambios adecuado	
	No se actualizaron los casos de uso en tiempo y forma	
	Falta de motivación al equipo de trabajo	
	Falta de control y seguimiento en hallazgos	
	Falta incluir prototipos de pantalla en CU o diagramas de procesos (d. actividades)	
	No se realizan pruebas unitarias a detalle	
	No se hace diferencia entre el concepto de negocio y técnico de lo que es un catálogo	
	Falta un curso para capacitar al equipo para tratar/negociar con el cliente (Usuario)	
	Falta involucrar al cliente para generar los casos de prueba.	
	Mala planeación del proyecto	
	Falta de integración entre las distintas unidades para presentar la propuesta	
	Mal control del proyecto	
	Falta de seguimiento y motivación del equipo de trabajo	
	Inexperiencia de los colaboradores con la tecnología propuesta	
	Mal manejo del cliente	
	No se ve la calidad de CMMI 3	
	Mayor involucramiento de la unidad Java	
	Mala comunicación con el cliente - equipo de desarrollo (Hallazgos)	
	Mala o poca comunicación GP - LP - LT - Equipo de trabajo	
	Falta de reacción al visualizar que los tiempos no se cumplirían	
	La asignación de trabajo fue asignada en base a un "Plan de trabajo" (generamente en excel) mas no fue asignado en base a una "Planeación".	

Comentarios de otro Administrador de proyectos del proyecto:

El enemigo está en casa.

Comentarios del Patrocinador del proyecto por parte del proveedor

Es necesario a la hora de cerrar con un cliente, saber que el cliente tiene la capacidad de desarrollar el proyecto esté preparado o tenga el nivel suficiente de conocimientos del sistemas.

Comentarios de lecciones aprendidas del cliente:

Muy Estimado Director.

INICIARE COMENTANDO LOS SIGUIENTES PUNTOS:

- 1- La metodología nos la mostraron después de una semana o dos, de no entender nada.
- 2- Solicité al Administrador de proyectos en la Junta del día x según mi agenda, recordándole, que hacíamos tales productos y que era necesario conocer la METODOLOGIA A USAR, por parte de nuestro Equipo.
- 3.- El levantamiento de la Información de las áreas excepto de Producción, todas se hicieron sentados en las aulas, pregunto!!! no se requería que el personal confirmara con nuestro equipo dicha información en campo????? Ya que si pides claridad, por principio de cuentas y teniendo el nivel que tenemos de profesionistas (sin ofender a nadie), no crees que su personal, pudo haberse levantado de su cómoda silla e ir a la tienda más cercana, a confirmar, lo que en palabras fue inexplicable para ustedes y/o poco claro???????, en referencia SOLO AL PUNTO DE VENTA?????
- 4.- La comunicación entre el equipo de nosotros es deficiente, eso lo sé desde hace un tiempo atrás, sin embargo NUNCA NADIE POR PARTE DE USTEDES, SE TOMO LA MOLESTIA DE HACERNOSLO SABER DURANTE LA REALIZACION DEL TRABAJO, NI SE REALIZO JUNTA ALGUNA, si no hasta ahora en que tienen que respaldar los errores en alguien más, también no????.
- 5- Por sugerencia mías se llevaron a cabo las juntas con todos, y mejoraron mucho más los resultados, pregunto?????POR QUE NO SE LES OCURRIO A USTEDES CON EL PUNTO DE VENTA HACER LO MISMO???????
- 6.- Por todo lo anterior no existe problema alguno, sin embargo el punto de venta NO ESTA FUNCIONANDO Y COBRANDO A NOSOTROS, ni la determinación de la capacidad de los equipos que nos recomendaron , ni la capacidad del servidor, ni la solución a la Problemáticas de los Pedidos Especiales, ni lo del retiro de efectivo, es RELEVANTE PARA USTEDES;

ATTE.
EL CLIENTE-

3.5.- Administración de Proyectos.

El resultado de este proyecto fue caótico, puso en riesgo a la organización. El administrador de proyecto y patrocinador de este proyecto, ambos certificados por el PMI, únicamente como referencia de que es necesario contar con un modelo que realmente apoye a las organizaciones en el desarrollo de Software, el proveedor contaba con una metodología CMMI nivel 3, la empresa cuenta con 19 años de experiencia en el mercado de desarrollo de sistemas. El proyecto estuvo fuera de presupuesto, fuera de tiempo y fuera de

especificaciones acordadas. Es diciembre del 2010 día que el proveedor no ha podido cerrar con el proyecto desde 2008. Es necesario revisar otros puntos como el comportamiento organizacional y la teoría de restricciones que contribuya a ayudar que los proyectos salgan en presupuesto, alcance y tiempo. Tomamos como referencia este proyecto, porque es un proyecto común en tiempo, presupuesto y alcance de desarrollo de software, un proyecto de término grande.

El tiempo es relevante para nuestro estudio, el proyecto debió haberse terminado con total de 12,600 horas/hombre.

3.6.- Teoría de Restricciones.

Algunos factores que no fueron tomados en cuenta en este proyecto y que influyeron en el retraso del proyecto fueron:

- ✓ Teoría de Restricciones
 - Fluctuaciones Estadísticas
 - Eventos dependientes e independientes.
 - Cuellos de botella
 - Amortiguadores
 - Balance de Cargas
 - Ruido en los procesos.
- ✓ Todo lo que contempla el Comportamiento Organizacional

Capítulo 4 Generación de Análisis y Síntesis

Las cuatro áreas de estos temas que se revisaron fueron:

- Administración de proyectos de software utilizando técnicas PMI.
- La Administración de proyectos de Software utilizando TSP y PSP.
- La Administración de proyectos con teoría de restricciones (TOC)
- Comportamiento Organizacional

4.1 Administración de Proyectos

El por qué están relacionados estos cuatro temas se debe a que en teoría con sólo aplicar la técnica del PMI pudiera decirse que el proyecto estará bajo control. De ahí que el PMI guste, pues parecería ser que tiene todo bajo control, actualmente las empresas donde se hacen desarrollos de software solicitan PMP certificados para asegurarse que sus proyectos saldrán en tiempo, El PMI tiene nuevas áreas de conocimiento y cinco grupos muy importantes, sin embargo en la práctica suceden casos como el caso práctico, incluso, gente certificada aplica la técnica del PMI y a pesar de ello, los resultados no son los que se esperan.

¿Por qué no es suficiente?

Desde mi punto de vista el PMI a todo lo que no está dentro de las nueve áreas de conocimiento lo mandan a riesgo, tratando de mitigar todo cuando en ocasiones los problemas pudieran ser mitigados con anticipación, pero más allá de la anticipación con conocimientos de otras técnicas de administración de proyectos. A mi forma de ver estos riesgos son mitigados después cuando mucho de estos pudieron haberse mitigado antes.

4.2 Administración de Proyectos con TSP y PSP

El TSP (Team Software Process) ayuda como una disciplina dentro de la ingeniería de software que aporta el estableciendo de grupos de ingeniería para ser utilizados en aplicaciones y conceptos integrados para el desarrollo de software. Dentro de las principales actividades que se establecen en TSP son:

- ✓ Establecimiento de metas
- ✓ Definición de los roles y funciones del equipo
- ✓ Asignación y evaluación de los riesgos
- ✓ Elaboración de un plan de equipo.

TSP ayudará a la organización a establecer una práctica de ingeniería madura y disciplinada, que produce seguridad y confiabilidad. El uso de TSP dentro de una organización puede construir equipos auto dirigidos al plan y al seguimiento de su trabajo, establecer objetivos, y sus propios procesos y planes. Estos equipos de software pueden ser equipos integrados con 3 a 20 ingenieros.

PSP (Personal Software Process) .- Es una disciplina dentro de la ingeniería de software que establece procesos definidos para recoger mediciones detalladas en el tiempo para producir un producto. Dichas mediciones o indicadores se analizan con métodos estadísticos que permiten producir estimaciones muy precisas, basadas en:

- ✓ Datos históricos
- ✓ Seguimiento de los procesos
- ✓ Calidad de un proyecto en curso
- ✓ Predicción los efectos del calendario
- ✓ Predicción la calidad de un producto de software terminado.
- ✓ Reducción de los defectos de los productos.

El CMMI y el norma mexicana NMX-NYCE contribuyen a tener las mejores prácticas del modelo de desarrollo y mantenimiento de software, método de trabajo contribuye a la mejora del desarrollo de software.

4.3 Administración de Proyectos con Teoría de Restricciones (TOC)

TOC es uno de los métodos de decisión y los cinco pasos para la administración de proyectos son:

1. Identificar

Identificar una restricción significa que ya tenemos alguna apreciación de la magnitud de su impacto sobre el desempeño general. De otra manera, también tendríamos algunas trivialidades en la lista de restricciones.

En el interior de la empresa se tiene varios candidatos a ser restricción y afortunadamente, más posibilidades de intervención: desde una máquina que se estropea o se utiliza con mucha frecuencia o la demanda de las partes que fabrica la máquina es mayor que su capacidad, una persona que soporta excesiva carga de trabajo, un departamento de ventas que no consigue suficientes pedidos para la capacidad potencial de la empresa, o un departamento de producción que no consigue acortar los plazos o aumentar el nivel de calidad o un departamento de proceso de datos que ofrece demasiado tarde los resultados para tomar decisiones, etc.. Para identificar los recursos internos como restricción, lo único que tenemos que hacer es calcular un perfil de recursos, en un horizonte dado, y elegir al recurso que tiene la mayor carga.

2. Explotar

Explotar simplemente significa sacarles el mayor jugo posible.

Una vez identificada una restricción o un cuello de botella y sin necesidad de invertir dinero en modificar su capacidad, salvo que la sustitución del recurso restricción sea muy económica, podemos explotarlo haciendo mejoras como por ejemplo asegurar su uso al 100% del tiempo disponible, ya que por ser una restricción determina el ritmo de la producción de la fábrica de software, o cambiar la combinación de productos o trabajos que atraviesan la restricción, de manera que se reduzca el tiempo de los mismos en la restricción, o haciendo inspección de calidad preventiva inmediatamente antes de la restricción, para que éste no pierda tiempo con productos defectuosos que después serán rechazados, o reduciendo el tamaño del lote a procesar. Esto puede ser que se detengan en la definición de las base de datos por un dato que todavía el usuario no definió. Si las no restricciones no suministran lo que las restricciones necesitan consumir, la decisión anterior se quedará en el papel, letra muerta que jamás será llevada a la práctica.

3. Subordinar

Todo lo demás a la decisión anterior.

Ahora estamos en un estado en el que estamos manejando la situación actual. Las no restricciones no son casos fortuitos, podemos hacer algo con respecto a ellas. Las no restricciones deben suministrar lo que las restricciones necesitan.

Desde esta perspectiva, de poco sirve al rendimiento global del sistema que el resto de recursos obvian la restricción y se pongan a alcanzar un alto rendimiento local. Probablemente aumente el inventario y los gastos de operación, pero no es probable que aumenten los beneficios.

4. Elevar la restricción.

La Restricciones del Sistema.

Elevar significa, "Levantar la limitación". Este es el cuarto paso, no el segundo. Tantas veces hemos sido testigos de situaciones en las que todo mundo se quejaba de una enorme restricción pero, al realizar el segundo paso, de la explotación, de no desperdiciar lo que sí se tenía, resulto que había de sobra. Así que no corramos aceleradamente a autorizaciones inadecuadas. Cuando hayamos terminado los pasos dos y tres, y todavía tengamos una restricción, será el momento de pasar al cuarto paso.

Una vez sincronizado el funcionamiento del sistema conviene empezar a superar las condiciones impuestas por la restricción, por ejemplo adquiriendo sistemas de información que nos permitan tener datos objetivos sobre lo que realmente está sucediendo en el proyecto, trasvasar recursos a otras partes del sistema, modificar la cartera de productos, o incluso efectuando cambios en el modelo organizativo y cultural de la empresa.

5. Si, en un paso previo, se ha roto la restricción, volver al paso 1 y no dejar que la INERCIA se convierta en la Restricción del Sistema.

Pero esto no es el quinto paso entero. Debemos agregarle una gran advertencia. La restricción tiene un impacto sobre el comportamiento de todos los demás recursos del proyecto. Todo debe subordinarse al nivel máximo de desempeño de la restricción. Así, a partir de la existencia de la restricción, en la compañía derivamos muchas reglas, a veces formalmente, a veces intuitivamente. Ahora se ha roto una restricción. Pero normalmente, no nos molestamos en regresar a examinar esas reglas. Se quedan ahí, y por lo tanto ahora tenemos restricciones de políticas.

El anterior es el proceso para enfocar, cinco pasos, un procedimiento intuitivamente obvio, que al mismo tiempo es el proceso de la mejora continua.

El DBR (Tambor-Amortiguador-Cuerda) es una metodología de la ejecución industrial, nombrada por sus tres componentes. El tambor es la restricción física de la planta: el centro de trabajo o máquina u operación que limitan la capacidad del sistema entero para producir más. El resto de la planta sigue el ritmo del tambor. Ellos se aseguran que el tambor tenga trabajo y lo que este ha procesado no se desperdicie.

El amortiguador protege al tambor, para que siempre tenga trabajo que fluye hacia él. Los amortiguadores en DBR tienen tiempo como su unidad de medida, en lugar de la cantidad de material. Esto hace que la prioridad del sistema sea operar estrictamente basado en el tiempo que se espera que un pedido u orden de producción esté al funcionamiento del buffer. El

DBR normalmente requiere los amortiguadores (buffers) en varios puntos del sistema: en la restricción, en el sitio de ensamble y en el lugar de entrega.

La cuerda es el mecanismo de salida de trabajo para la planta. Solo un buffer de tiempo en frente de una orden hace la debida liberación a la planta. Jalando trabajo en el sistema más temprano que un buffer de tiempo solamente se garantiza un trabajo en el proceso equivalente a un buffer de producto en proceso.

4.4 Comportamiento Organizacional (OB)

El OB es un campo de estudio que investiga el impacto que los individuos, grupos, y la estructura sobre el comportamiento en las organizaciones, con el fin de aplicar esos conocimientos para mejorar la eficacia de una organización dentro de esta tesis hemos revisado cada paso del comportamiento organizacional que contribuye en un porcentaje alto a la Administración de Proyectos de software mediante el estudio de porque los APS no son tan efectivos. Hemos estudiado con profundidad los temas de:

- Diversidad dentro de las organizaciones.
- Actitud y satisfacción de trabajo
- Emociones y Humor
- Personalidad y Valores
- Percepción Individual y toma de decisiones
- Conceptos de Motivación
- Aplicación de la motivación
- Fundamentos de grupos comportamiento
- Equipos de trabajo
- Comunicación
- Liderazgo
- Poder y Política
- Conflictos y Negociación
- Cultura organizacional
- Políticas y Prácticas de recursos Humanos
- Cambios organizacionales y Administración del estrés

Después de revisar cada punto del comportamiento organizacional veo el impacto que los individuos tienen en un proyecto. El 50% del resultado de que el proyecto salga en tiempo, con presupuesto y con el alcance acordado se debe a los individuos. Este porcentaje puede ser mayor pero el desarrollo del proyecto de software lo hace la gente. Al revisar el caso práctico la mayoría de los puntos a mi juicio se falló por no conocer a detalle el impacto que pudiera tener cada individuo en el rol que se le encomendó a cada quien en el proyecto. Desde el que vendió el proyecto hasta el que documenta, pasando por todos los roles.

4.5 Síntesis

La Administración de proyectos de desarrollo de software para las empresas la podemos tipificar los proyectos en 3 grupos.

- 1.- Administración de proyectos de desarrollos de software para las grandes empresas o corporativos.
- 2.- Administración de proyectos de desarrollos de software para las medianas empresas.
- 3.- Administración de proyecto de desarrollo de software para la micro y pequeña empresa.

1.- Para la Administración de proyecto de desarrollo de software para las grandes empresas o corporativos tales como bancos, telefónicas donde es software es vital el factor común es que se tengan equipos de hardware grandes para soportar grandes volúmenes de información. Dentro de esto punto la cultura de informática es alta y la Ingeniería de Software para el desarrollo de proyectos de software es adecuada podemos comentar que los cuatro temas que se abordaron en los capítulos anteriores:

- Administración de proyectos de software utilizando técnicas PMI.
- La Administración de proyectos de Software utilizando TSP y PSP.
- La Administración de proyectos con teoría de restricciones (TOC)
- Comportamiento Organizacional

Es necesario contar con un conocimiento adecuado de TOC desde mi punto de vista les ayudaría a conocer las restricciones adecuadas y el comportamiento organizacional. Los temas en lo que les puede ayudar son:

- ✓ Fluctuaciones Estadísticas
- ✓ Eventos dependientes e independientes.
- ✓ Cuellos de botella
- ✓ Amortiguadores
- ✓ Balance de Cargas
- ✓ Ruido en los procesos.

Los principales problemas a los enfrenta la ADPS (Administración de proyectos de Software) son:

- ✓ Prioridades diferentes entre departamentos del personal que trabaja con ellos
- ✓ Responsabilidades en conflicto
- ✓ Fecha de inicio y termino pre negociadas.
- ✓ Desempalme de recursos no protegidos
- ✓ Actividades simultáneas de un mismo recurso con fracciones de actividades fuera de los proyectos. Los recursos como los arquitectos en ocasiones están fraccionados en

otro proyectos es decir el arquitecto puede que trabaje en el más que más le es de su interés.

- ✓ No disponibilidad de los recursos en el tiempo adecuado. Es decir que puede que los recursos estén ocupando en el momento de uso de acuerdo al plan de trabajo.
- ✓ Castigos por terminar temprano.- Cuando un desarrollador termina temprano su actividad se le asigna a otra actividad de manera inmediata.
- ✓ Pocas actividades tardías descontrolan el proyecto y la ruta crítica.
- ✓ En ocasiones la ruta crítica no se cambia a pesar de que cambia el alcance.
- ✓ Ni poniendo protección (holgura de tiempo o gente muy buena) en las actividades se atina.

Algunas soluciones de TOC en la ADPS son:

- ✓ Tomar en cuenta las incertidumbres en las actividades.
- ✓ Agregar protección a las actividades y que solo ADPS conozca
- ✓ Adecuar las capacidades de cada recurso de acuerdo a su tarea.
- ✓ Cumplir con las necesidades de tiempo críticas
- ✓ La incertidumbre se maneja mejor a la hora de ejecutar la actividad
- ✓ Usar tiempos cortos para controlar la protección y agruparla al final de la actividad (amortiguador)
- ✓ Eliminar los tiempos cortos con protección de manera independiente.
- ✓ Identificar la secuencia crítica.
- ✓ Los tiempos sin protección son lo que se muestran al equipo de trabajo. (deben ser la mitad de su tiempo normal)
- ✓ Amortiguadores al final del proyecto y no a las actividades (Se protege al proyecto no a las actividades)
- ✓ Amortiguadores agrupados en las alimentaciones a la secuencia crítica.
- ✓ Hay que evitar agregar protección dependiendo de los históricos, si sabemos que se va a cortar el tiempo se aumenta y cuando un APS agrega protección también su gente.
- ✓ **EL cambio con TOC para el ADPS consiste en controlar los amortiguadores y no las actividades.**
- ✓ **El cambio con TOC para el ADPS consiste en administrar la incertidumbre.**
- ✓ **EL cambio con TOC para el ADPS consiste en administrar las capacidades.**
- ✓ **El cambio con TOC para el ADPS también incluirá revisar y resolver el comportamiento organizacional para controlar los amortiguadores:**

- **Diversidad del personal dentro del proyecto**
- **Actitud y satisfacción de trabajo**
- **Emociones y Humor**
- **Personalidad y Valores**
- **Percepción Individual y toma de decisiones**
- **Conceptos de Motivación**
- **Aplicación de la motivación**

- **Fundamentos de grupos comportamiento**
- **Equipos de trabajo**
- **Comunicación**
- **Liderazgo**
- **Poder y Política**
- **Conflictos y Negociación**
- **Cultura organizacional**
- **Políticas y Prácticas de recursos Humanos**

En lo personal y en base a mi experiencia, el no manejar adecuadamente el amortiguador del comportamiento organizacional es lo que afecta a las organizaciones. Pareciera que cae dentro del área de comunicaciones del PMI, sin embargo es un muy profundo como el que describió en el capítulo 2.

Dentro de estas empresas cuenta con adecuado manejo de TSP y PSP

2.- Administración de proyectos de desarrollos de software para las medianas empresas.

Es muy común que dentro de la empresa mediana este enfocado al negocio y se necesite un adecuado manejo de las herramientas de:

- La Administración de proyectos de Software utilizando TSP y PSP.
- La Administración de proyectos de software utilizando técnicas PMI.
- La Administración de proyectos con teoría de restricciones (TOC)
- Comportamiento Organizacional

Dentro de estas empresas existe una cultura media de informática y es probable que los desarrollos a lo que se quieren enfrentar existan muchas restricciones para la cual recomiendo la compra de productos o proyectos pequeños.

Al no contar con herramientas como TSP y PSP tal vez se haga muchas cosas de manera empírica y se tendrían muchas restricciones, como se comenta en el caso práctico.

Los problemas a los que se enfrentan las empresas medianas son:

- Por la parte de Ingeniería de Software y herramientas como TSP y PSP al querer un desarrollo de software tal vez su área de sistemas sea un equipo promedio es 5 personas, una persona que controla la red, monitorea que los procesos, desarrollo de sistemas y para la parte de desarrollos nuevos sean por una o dos personas. El problema aquí es que mantener la operación es un problema al que se enfrentan diariamente y antes de pasar a nivel de utilizar técnicas PMI o Administración de proyectos con teoría de restricciones o comportamiento organizacional primero tiene que aprender a desarrollar con metodología y procesos para después llegar a un siguiente nivel en donde se encuentran la parte de administración de proyectos PMI, a la administración de proyectos de software con TOC y por último el utilizar el comportamiento

organizacional. Inclusive aunque la empresa contrate una consultoría para que haga el desarrollo de software a la medida.

- Lo más recomendable es hacer software de menos de: 1,000 horas/hombre en donde tal vez la atención de los usuarios no sea muy demandante para el negocio y se pueda hacer, en caso de que rebase más 1,000 horas/hombre se enfrentan problemas.
- Comento que se enfrenta a un problema porque al no tener la cultura informática tal vez se necesite más tiempo de parte de los usuarios y administrador de proyectos, se tendrán que administrar más puntos de la incertidumbre y de los amortiguadores que probablemente el proyecto ya no sea costeable para la organización.
- Sin embargo esto se puede lograr moviendo cualquiera de las variables de tiempo, presupuesto y alcance. En donde lo primero que tendrá que hacer es elevar a la mediana empresa a un nivel de cultura informática mayor, y después continuar con el desarrollo del proyecto.

3.- Administración de proyectos de desarrollos de software para las pequeñas y micro empresas.

Aquí simplemente la pequeña y micro empresa estará enfocada a su negocio y difícilmente tendrá las herramientas de:

- La Administración de proyectos de Software utilizando TSP y PSP.
- de Administración de proyectos de software utilizando técnicas PMI.
- La Administración de proyectos con teoría de restricciones (TOC)
- Comportamiento Organizacional

Pero igual que la empresa mediana lo puede lograr mediante la contratación de alguna consultoría que tenga ese perfil y el proyecto no sea muy grande. En este caso el número máximo de personal para el área de sistemas máximo promedio de 1 persona. En donde el personal hace de todo. Es muy poco probable que ese tipo de empresas desarrolle software a menos que su negocio Core sea de desarrollo de software empresarial o aplicativo.

Dentro de estas empresas existe una cultura nula de informática y es probable que los desarrollos a lo que se quieren enfrentar existan muchas restricciones para la cual recomendando la compra de productos o la realización de proyectos pequeños.

Al no contar con ninguna herramienta como TSP y PSP tal vez se haga muchas cosas de manera empírica.

4.6 Análisis del caso práctico.

Los hechos de caso son:

Una empresa mediana que quería implementar un software de administración de recursos.

La empresa contrata a una consultoría especialista en el desarrollo de software de más de 19 años de experiencia en el desarrollo de proyectos de software.

El proyecto aún no se termina debido a la gran cantidad de problemas solo se terminó el módulo de punto de ventas, ventas, Compras y almacén y contabilidad.

La empresa no cuenta con ninguna técnica de administración de proyectos de software como:

- La Administración de proyectos de Software utilizando TSP y PSP.
- de Administración de proyectos de software utilizando técnicas PMI.
- La Administración de proyectos con teoría de restricciones (TOC)
- Comportamiento Organizacional

La empresa solo cuenta con una persona en el área de sistemas,

Haciendo un análisis interno y externo de la empresa de porque se fallo

La empresa de acuerdo a nuestro análisis del punto 4.5 de síntesis cae dentro de una empresa mediana, los comentarios del proyecto de lecciones aprendidas, la empresa mediana no contaba con una cultura informática, al hacer un traje a la medida y por la cantidad de horas para el proyecto no se contaba con el tiempo suficiente de parte del usuario para el análisis.

Dentro de ADPS el proveedor no gestiono adecuadamente los recursos, se vio inmerso en una problemática técnica nunca antes probada, se utilizó tecnología nueva.

Pero la mayor problemática la tuvo en el comportamiento organizacional del proyecto por todos los comentarios del personal involucrado en los diferentes roles descritos en el caso práctico de lecciones aprendidas y a pesar de la que la funcionalidad no era compleja se volvió una problemática no se administró adecuadamente la incertidumbre las capacidades ni las holguras. Otro factor importante que contribuyo fue la distancia entre la consultoría y el cliente, a 5 horas de distancia en auto y 45 minutos en avión, al no tener una cultura informática se querían resultados de acuerdo a lo establecido en la propuesta económica de entregables. El proyecto se empezó a desviar y no se controlaron las holguras adecuadamente, se tuvo un problema técnico el cual se tardó tres meses en resolver. La gente seguía avanzando de acuerdo al plan, se tenía problemas de comunicación los comentarios de lecciones aprendidas podemos concluir una falta de comunicación clara en el equipo y desvió total por no controlar la parte del comportamiento organizacional.

A ambas empresas tanto el cliente como la consultoría para el desarrollo de un nuevo proyecto le pueden ayudar:

- La Administración de proyectos de Software utilizando TSP y PSP.
- de Administración de proyectos de software utilizando técnicas PMI.
- La Administración de proyectos con teoría de restricciones (TOC)
- Comportamiento Organizacional

Estas teorías no solo son necesarias si no que deben ser el pan de cada día de ambas empresas si quieren sacar el proyecto.

Capítulo 5 Discusión de Resultados y Conclusiones

La conclusión principal es que los APS deben ahora administrar incertidumbres, holguras y capacidades y junto con ello los amortiguadores de cada actividad con sus respectivas restricciones existentes.

Como el PMI se ve complementado con nuevos enfoques.

Los APS primero que nada deben delegar todo aquello que es control de actividades en una persona que cumpla con el perfil de seguimiento de actividades pudiendo ser un perfil administrativo.

Como dentro de cualquier proyecto de ADPS tiene que ver con personas, la administración de la incertidumbre se debe ahora enfocar al comportamiento organizacional porque los proyectos se hacen con gente, de esta forma podemos asegurar el complemento de PMI, y dentro de la administración de la incertidumbre lo que debemos controlar y conocer es:

- ✓ Diversidad del personal dentro del proyecto
- ✓ Actitud y satisfacción de trabajo
- ✓ Emociones y Humor
- ✓ Personalidad y Valores
- ✓ Percepción Individual y toma de decisiones
- ✓ Conceptos de Motivación
- ✓ Aplicación de la motivación
- ✓ Fundamentos de grupos comportamiento

- ✓ Equipos de trabajo
- ✓ Comunicación
- ✓ Liderazgo
- ✓ Poder y Política
- ✓ Conflictos y Negociación
- ✓ Cultura organizacional
- ✓ Políticas y Prácticas de recursos Humanos

Tal como la administración de riesgos la administración de la incertidumbre debe ser clasificada en:

- Planear la incertidumbre
- Valorar la incertidumbre
 - ✓ Identificar la incertidumbre
 - ✓ Analizar la incertidumbre.
- Manejar la incertidumbre
- Monitorear la incertidumbre

Resolver no importa el tamaño ni el proyecto ni el tamaño de la empresa.

Uno de los puntos que debemos cubrir ADPS es la aceleración planificada este puede ser previo, durante o final de que la producción haya comenzado que consiste en ir cambiando la planeación del proyecto minimizando los tiempos de los buffers. A lo cual TOC llama "aceleración planificada." Este método consiste primordialmente en conseguir que trabajo llegue antes de su tiempo inicial planeado.

Las acciones que se llevan a cabo para acelerar un trabajo ("aceleración planificada") son:

Como primera acción es determinar que trabajos se supone que están en el origen del buffer. Identificando donde se ha atascado el trabajo que se ha retrasado que pudiera ser por falta de capacidad, ausencia en el trabajo o cualquier punto del comportamiento organizacional.

Como segunda acción, Una vez encontrado el atraso del trabajo actuamos y tomamos las acciones que correspondan para que este trabajo pueda seguir adelante. En un ejemplo práctico dentro de la parte de software simplemente un programador se atora o se detiene por una cuestión técnica por falta de experiencia, la intervención del administrador de proyectos es necesaria para ver como se le apoya con un arquitecto o alguien de nivel técnico mayor si fuera el caso.

Como tercera acción es registrar donde hemos encontrado el trabajo atascado así como también registrar el tiempo de atraso con sus respectivas características y la solución que se dio a este atraso. En el caso práctico es llevar una bitácora de incidencias.

Como cuarta acción es la de repetir el proceso desde la primera acción hasta la tercera acción sacando una lista de recursos a los problemas y soluciones que se dieron

Como quinta acción es identificar las causas de posibles problemas en función de la frecuencia de los recursos utilizados en cada atraso mediante la lista de registro de la acción anterior (cuarta acción). Esto que acabamos de mencionar si lo enfocamos a procesos de manufactura o de producción puede ser procesos defectuosos por:

- ✓ Ajustes poco fiables
- ✓ Falta de capacidad de protección
- ✓ Que el recursos este mal administrado

Cuando se presentan dichos problemas lo que se analizan a fondo es el nivel de recursos y el nivel de trabajo que se está desarrollando, en este caso hay que revisar que del comportamiento organizacional está afectando, no tendremos que acelerar más de una y otra vez el trabajo en cuestión del atraso, eliminado hasta cierto punto la aceleración. Esto aplicado a la industria de software significa que un desarrollador puede que no esté desarrollando programas correctos pudiendo ser la causa la falta de nivel técnico del desarrollador (capacidad) junta con la poca claridad de la especificación, o una especificación no bien definida., que no esté entiendo adecuadamente. Si llevamos a cabo las acciones anteriores podremos ir reduciendo la longitud de los buffer o tiempos de proceso (lead – time).

Como sexta acción una vez localizado los sitios donde se encuentran los trabajos retrasados, se asignan prioridades de acuerdo con el número de veces que aparece en lista cada recurso. Podremos utilizar un factor de ponderación y en función de ello podemos aplicar mejora en la productividad aplicando los diagramas Pareto.

Al igual que los procesos de Manufactura para la ADPS debemos de tener un sistema en dos fases preliminares,

La primera fase que tendrá que incluir nuestro sistema de información, es el procedimiento necesario para identificar y explotar las limitaciones, subordinando a ellas y la segundo fase

es la implementación y toma de decisiones de todo aquello que se encontró identificado, registrando cada acción para futuros proyectos.

Otra conclusión importante es que mientras existan fluctuaciones estadísticas, si queremos explotar las limitaciones, debemos tener en los demás recursos más capacidad de lo que sería estrictamente necesaria según la demanda.

Lo primero que debemos cubrir es la capacidad de producción, como segundo es la capacidad de protección,

La capacidad que quede después de producción y de protección es exceso de capacidad.

El buffer de tiempo es nuestra protección contra perturbaciones desconocidas. Lo que se desconoce de ellas es no es que vayan a ocurrir, su ocurrencia es algo casi garantizado. Lo que no sabemos es cuando ocurrirán, donde afectaran, ni cuánto durará la perturbación. Dada la naturaleza aleatoria de las perturbaciones, es obvio que no estamos en situaciones de estimar el buffer de tiempo con precisión. Aunque tuviéramos el tiempo y los recursos necesarios para recopilar todos los datos estadísticos, solo conseguiríamos una curva de probabilidad.

Dicha curva muestra la probabilidad de superar una perturbación en un recurso determinado en función del tiempo que se necesita.

La administración de los buffers o amortiguadores nos proporciona varios beneficios.

1.- Nos permite determinar mejor la longitud requerida de acuerdo con el nivel existente de perturbaciones: cuantificando el ruido que hay en ese proceso o estado. Esto puede traducirse en un ejemplo al dejarle una investigación alguna programador revisar los avances concretos de acuerdo a los objetivos en espacios breves de tiempo y medir si es necesario más o menos tiempo o mayor capacidad.

2.- Nos permite acelerar tareas, sistemáticamente y metódicamente, para reducir el tiempo total del proceso (lead-time).

3.- Al localizar los sitios donde se encuentran atascados y/o retrasados los trabajos, se asignan prioridades de acuerdo con el número de veces que aparece en esa lista cada recurso tomando en cuenta y definiendo un factor de ponderación que sea adecuado. Teniendo una lista de Pareto que debe guiar nuestro esfuerzo de mejora de la productividad

4.- Nos permite administrar las capacidades de protección de los recursos en función del tiempo. Esto que si un recurso no tiene la capacidad de resolver un problema en un cierto tiempo. Deben cuidar que ese tiempo no se haga infinito y resolver con un recurso con la capacidad necesaria. Como ejemplo, aquí es que si un programador no ha sido capaz de sacar un programa en un determinado tiempo, tendremos que poner el apoyo del arquitecto para resolver, guiar y apoyar al programador, sin meterse a detalle informando al APS como

va, si el APS ve que el arquitecto está dedicando mucho tiempo e inclusive está resolviendo el problema el APS debe de cuidar que esa tarea no se haga con exceso de capacidad es decir con el tiempo del programador y el tiempo del arquitecto porque son dos tiempos de dos recursos en una misma actividad lo implica un exceso de capacidad en resolver la tarea. Una de las formas de solucionar esto es cambiando al programador y cambiar de actividades al programador en curso.

Uno de los métodos para encontrar los cuellos de botella de donde las actividades o procesos tengan retraso en tiempo se pueden usar las técnicas de TOC para dar soluciones son mediante los siguientes pasos:

- 1.- Clasificación de la información
- 2.- Correlación de la información
- 3.- Efecto Causa Efecto

La clasificación de la información consistirá en clasificar toda la información referente al cuello de botella. Después del punto de clasificación está el de ver la CORRELACIÓN que existe entre diferentes variables que básicamente refiere a revisar lo común que existen en otras variables y al final revisar el EFECTO CAUSA EFECTO que está causando el problema estableciendo lógicas de deducciones y explicaciones.

Un ejemplo que normalmente ocurre en los proyectos es:

Un APS detecta un cuello de botella de tiempo en cierto personal de alto nivel para sacar los programas, por lo que comienza con esta técnica de TOC en las CLASIFICACION de la información los clasificar en de acuerdo las diferentes variables y de una primera percepción no logra identificar cual es exactamente el problema.

En el segundo paso donde empieza a correlacionar la información revisa algunas variables en común que hay dos personas desarrolladores muy buenos que tiene cuellos de botella para sacar su trabajo. Al revisar las variables observa un factor común que ambas vienen de un proyecto anterior donde en los últimos dos meses de trabajo estuvieron jornadas de trabajo muy largas. Al área de consultoría de este no aviso de esta situación al APS ni tampoco aviso al Recursos Humanos.

El tercer paso de EFECTO – CAUSA – EFECTO de este ejemplo, una vez que la correlación ha sido identificada (Cabe mencionar que la correlación siempre existe, el trabajo del APS es encontrar la variables correlacionadas en base a la observación y deducción), el APS puede que haya encontrado el problema en base a la deducción de la correlación que existen entre las variables como fue el que este personal estuvo trabajando jornadas largas de trabajo incluyendo sábados y domingos y no se le dio tiempo de relajación o de recuperación entre un proyecto y otro. Revisar el tema de la desmotivación, una vez encontrada la causa debe buscar si existe más personal dentro del proyecto que se encuentre

en misma situación de esas variables relacionada. Para tomar la acción con los futuros cuellos de botella posibles. Avisando a los patrocinadores del proyecto para hacer un cambio en el proyecto de tiempo dejando descansar al personal de las variables correlacionadas en cuestión. El tiempo de descanso pudiera ser una posible solución sin embargo habría que revisar que del comportamiento organizacional está afectando esa desmotivación para dar solución de raíz a este problema.

Como resultado de esta investigación podemos decir que se necesitan las cuatro teorías para APS en tiempo, alcance y en el presupuesto acordado.

- La Administración de proyectos de Software utilizando TSP y PSP.
- de Administración de proyectos de software utilizando técnicas PMI.
- La Administración de proyectos con teoría de restricciones (TOC)
- Comportamiento Organizacional

Capítulo 6 Trabajo Futuro

Esas guías se pueden formalizar y ser implementadas en el PMI.

Crear una rama para certificación exclusiva para ayudar a administrar proyectos de Software. Como trabajo futuro se puede proponer un modelo que se llamado Administración de proyecto utilizando técnicas del PMI con TOC y comportamiento organizacional para administración de proyectos de Software.

Se revisarán los datos para procesarlos para que se puedan usar para generar resultados mediante:

- ✓ El análisis de incrustación de la información del modelo de TOC en el PMI.
- ✓ Revisar si hay errores o inconsistencias tanto en las entrevistas como la propuesta del modelo
- ✓ Clasificar la información.

Capítulo 7 Referencias

7.1 Referencias.

[1] PMBOK®Guide—Fourth Edition of PMI 01-01-2008

[2] Ingeniería de Software un Enfoque Practico Roger Roger S. Pressman Segunda Edición 1988 Editorial Mc Graw Hill

[3] El libro La meta .- Eliayahu. M. Goldratt y Jeff Cox, Séptima Edición de 1998, Ediciones Castillo S. A de C.V.

- [4]El libro "Necesario pero no suficiente" de : Eliyahu M. Goldratt. 2000, publicado por North River Press.
- [5]El libro "El Síndrome del Pajar", Eliyahu M. Goldratt 1990 Ediciones Diaz Santos
- [6] Metodología de la investigación Cuarta Edición por Roberto Hernández Sampieri , Carlos Fernández-Collado, Pilar Baptista Editorial Mc Graw Hill Enero del 2009
- [7] Project Management Eight edition Horlad jKerzner Ph.D 2003
- [8] CMMI 2003 by Pearson Education In Mark Beth, Mike Knorad, Sandy Shrum.
- [9]Increasing Value of project by leveraging limited capacity through the application of the theory of constraints (TOC) IEEE 2003
- [10]Using TOC Thinking Process Tools to Improve Safety Performance ZHANG LIN Department of Computer Science, Shangluo Unive Authorized licensed use limited to: Tec de Monterrey. Downloaded on February 22,2010 at 02:37:20 EST from IEEE Xplore. Restrictions apply. rsity , Shaanxi, China.
- [11]Bousfield A.K. and Bousfield W.a A , Measuring of clustering and of repeated constancias in repeatead , Psychol, Reports 19906, 19,935.
- [12] www.xpromagrmmin.com 2010
- [13] Orgazational Behavior.- Stephn P. Robbing Tomothy A Judge Fourteenth Edition
- [14] Research and Improved of team software process Computer society 2009 World Congress on Computer Scieincie and information Engineering.

7.2 Glosario de términos:

Proyecto.-De acuerdo a la definición del PMBOK es el esfuerzo temporal que se lleva a cabo para crear un producto servicio o resultado único en un tiempo determinado.

ADPS.- Administración De Proyectos de Software

APS.- Administrador de Proyectos de Software

EDT.- Estructura detalla de trabajo. WBS (Work Break Down Structure)

Throughput.- Es la velocidad a la que el sistema genera dinero a través de las ventas.

Inventario.- Es todo Aquello que el sistema ha invertido en comprar cosas que pretende vender

Gasto de Operación.- es todo el dinero que el sistema gasta en transformar el inventario en Throughput.

Eventos Dependientes: un evento o una serie de eventos deben llevarse a cabo antes de que otro pueda comenzar.

Fluctuaciones Estadísticas: Existen fluctuaciones que afectan los niveles de actividad de los distintos recursos productivos, como ser: calidad de la materia prima, ausentismo del personal, rotura de máquinas, corte de energía eléctrica, faltante de materia prima e incluso disminución de la demanda.

Alcance: La suma de productos, servicios y resultados que se proporcionarán como un proyecto.

Amortiguador.- Es el un tiempo de protección contra las perturbaciones desconocidas de un proyecto.

Ruido en un proceso.- Es cuando el proceso cae en un estado cíclico o en una perturbación tendiendo al infinito el cual difícilmente sale del ciclo sin la intervención de un ajuste o cambio en la decisión sin cambiar el proceso.

Tsp (Team Software Process) .- Es una disciplina dentro del ingeniera de software que establece grupos de ingeniería para ser utilizados en aplicaciones y conceptos integrados para el desarrollo de software.. Dentro de las principales actividades que se establecen en TSP son:

- ✓ Establecimiento de metas
- ✓ Definición de los roles y funciones del equipo
- ✓ Asignación y evaluación de los riesgos
- ✓ Elaboración de un plan de equipo.

TSP ayudará a la organización a establecer una práctica de ingeniería madura y disciplinada, que produce seguridad y confiabilidad.

El uso de TSP dentro de una organización puede construir equipos auto dirigidos dar seguimiento a planes y el seguimiento de su trabajo, establecer objetivos, y sus propios procesos y planes. Estos equipos de software pueden ser equipos integrado de 3 a 20 ingenieros.

PSP (Personal Software Process).- Es una disciplina dentro del ingeniera de software que establece procesos definidos para recoger mediciones detalladas en el tiempo para producir un producto. Dichas mediciones o indicadores se analizan con métodos estadísticos que permite producir estimaciones muy precisas, basadas en:

- ✓ Datos históricos
- ✓ Seguimiento de los procesos
- ✓ Calidad de un proyecto en curso
- ✓ Predecir los efectos del calendario
- ✓ Predecir la calidad de un producto de software terminado.
- ✓ Reduce los defectos de los productos.

Restricción.- Todo aquella que limita al sistema para alcanzar una alta ejecución para lograr la meta.