



A.F. 133283



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL  
Maestría en Sistemas de Información Gerencial  
Facultad de Ingeniería en Electricidad y Computación

"Adaptación de una empresa a una metodología  
de trabajo de mantenimiento del ciclo de vida  
de aplicaciones y sistemas basado en MSF"

**TESIS DE POSTGRADO**

Previa a la obtención del Título de:

**MAGISTER EN SISTEMAS DE INFORMACION GERENCIAL**

Presentada por:

*Ing. Roger Salinas Robalino*

**Guayaquil - Ecuador**

**2011**



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**MAESTRÍA EN SISTEMAS DE INFORMACIÓN GERENCIAL**

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**

**"ADAPTACIÓN DE UNA EMPRESA A UNA METODOLOGÍA DE TRABAJO DE  
MANTENIMIENTO DEL CICLO DE VIDA DE APLICACIONES Y SISTEMAS BASADO EN MSF"**

**TESIS DE POSTGRADO**

Previa a la obtención del Título de:

**MAGISTER EN SISTEMAS DE INFORMACIÓN GERENCIAL**

Presentada por:

**ING. ROGER SALINAS ROBALINO**

**GUAYAQUIL – ECUADOR**

**2011**

# AGRADECIMIENTO

A mi Dios, en quién me he refugiado, mi roca y mi fortaleza, mi esperanza, seguridad mía desde mi juventud; cuya bondad no tiene límites y cuyo amor a nosotros es un abismo inconmensurable.

A la Virgencita Santísima de La Merced, co-redentora y mediadora de los cautivos, enfermos, afligidos y a todos aquellos que sufren algún tipo de prisión, como el error, el pecado y el vicio. Contigo voy Virgen pura, y en Tu poder voy confiado, pues yendo de Ti amparado, mi alma volverá segura. Dulce Madre, no te alejes, Tu vista de mí no apartes, ven conmigo a todas partes, y nunca sólo me dejes. Y ya que me proteges tanto, haz que me bendiga, el Padre, el Hijo y el Espíritu Santo, Amén.

A mi mami, quien hasta sus últimos días se preocupó de nosotros y mostró su cariño, y quien desde el Cielo, seguirá cuidándonos hasta que nos encontremos con ella nuevamente.

A mi mamá, por su incondicional y constante apoyo. Es cierto que los homenajes se hacen en vida, así que la presente tesis se la dedico en especial a ella, quien a pesar de los riesgos previstos al inicio de este programa de estudios, estuvo como un soporte moral, tanto en las adversidades previstas como en las que surgieron en el transcurso de este periodo. Gracias por todo.

# CERTIFICADO

---

Por medio del presente se deja por sentado que se ha revisado y aprobado la Tesis de Post-Grado del **Ing. Roger Salinas Robalino** para el título de **MAGISTER EN SISTEMAS DE INFORMACIÓN GERENCIAL**, bajo el tema:

***“ADAPTACIÓN DE UNA EMPRESA A UNA METODOLOGÍA DE TRABAJO DE MANTENIMIENTO DEL CICLO DE VIDA DE APLICACIONES Y SISTEMAS BASADA EN MSF”.***



**MSIG Lenín Freire**  
Director del Programa



**MSIG Robert Andrade**  
Director de Tesis



**MSIG Carlos Martín**  
Miembro del Tribunal



**MSIG Juan C. García**  
Miembro del Tribunal

# DECLARACIÓN EXPRESA

“La responsabilidad por los hechos, ideas y doctrinas expuestos en esta tesis, nos corresponden exclusivamente; y el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”.

(Reglamento de Exámenes y Títulos profesionales de la ESPOL)

A handwritten signature in black ink, reading "Roger Salinas Robalino", written over a horizontal dotted line.

Ing. Roger Salinas Robalino.

## Contenido

Índice de figuras.....	7
Introducción.....	9
Capítulo 1: Antecedentes.....	10
Situación actual de la Empresa.....	10
Causas que motivan un cambio de metodología.....	11
Consecuencias del marco de trabajo actual.....	12
Capítulo 2: Planificación de la Implementación de la Metodología basada en MSF.....	13
Estructura de Trabajo para el Plan de Acción.....	13
Contraste de Metodologías. Investigación.....	14
Cronograma de Trabajo.....	32
Capítulo 3: Ejecución del Proceso de Implementación de la Metodología	34
Capacitación del Personal Actual de Sistemas en MSF.....	34
Coaching del Proyecto Piloto.....	38
Modelo de Equipos de MSF.....	39
Escalamiento hacia abajo – Combinación de roles para equipos más pequeños.....	46
Desarrollo de Procedimientos y Políticas acordes a MSF.....	47
Principios fundamentales aplicados al Modelo de Equipos y Conceptos clave o prácticas probadas.....	47
Visionamiento (Microsoft Corporation - MSF Visionamiento, 2009) .....	50
Planificación (Microsoft Corporation - MSF Planificación, 2009)....	60
Desarrollo (Microsoft Corporation - MSF Desarrollo, 2009).....	65
Estabilización (Microsoft Corporation - MSF Estabilización, 2009) .	72
Instalación/Liberación (Microsoft Corporation - MSF Instalación, 2009).....	78
Capítulo 4: Estabilización del Proceso de Implementación.....	84
Evaluación de la Capacitación en MSF.....	84

Resultados e Influencia de MSF en el Proyecto Piloto .....	86
Barreras Administrativas y Organizacionales de la Implementación de la Metodología .....	87
Capítulo 5: Procedimientos y Políticas. Metodología MSF adaptada .	90
Versión Final de los Procedimientos a Implementarse .....	90
Visionamiento .....	91
Planificación .....	99
Desarrollo.....	104
Estabilización.....	105
Instalación .....	107
Políticas de Soporte a la Metodología MSF.....	109
Directivas Generales .....	110
De la Cultura Organizacional en el Ciclo de Vida de Desarrollo.....	110
De los Proyectos MSF.....	111
De las Reuniones.....	111
Aprobación de la Solución desarrollada.....	112
Herramientas de Software de Apoyo a la Metodología .....	113
Microsoft™ Project Server* .....	113
Mantis Bug Tracker.....	114
WBS Chart Pro.....	114
PERT Chart Expert .....	114
Apéndices.....	117
Apéndice 1. Metodologías de desarrollo y programación (Wikipedia, 2010)	
.....	117
Metodologías de desarrollo de software.....	117
Enfoques Generales de desarrollo de software .....	118
Modelo en cascada .....	118
Prototipado .....	119
Incremental.....	119
Espiral.....	119
Rapid Application Development (RAD).....	120
Otros enfoques de desarrollo de software .....	120

Apéndice 2. Descripción de las plantillas de MSF y su contenido.....	121
Visionamiento .....	121
Planificación .....	123
Desarrollo.....	130
Estabilización.....	131
Instalación .....	133
Bibliografía .....	134
Glosario de términos.....	136

## Índice de figuras

Figura 2.1 Flujos de trabajo en cada fase de RUP (WillyDev).....	18
Figura 2.2 Relación entre las áreas de Proceso de CMMI – Ing. Software (Monografías, 2008).....	21
Figura 2.3 Tabla de valores Sigma de defectos permisibles (Mercadeo.com).....	26
Figura 2.4 Curva de media y varianza de validación de defectos en 6 $\sigma$ (Mercadeo.com) .....	27
Figura 2.5 Vista general de MSF (Microsoft Corporation - MSF Estructura del Equipo, 2009).....	28
Figura 2.6 Modelo de Equipos de MSF (Microsoft Corporation - MSF Estructura del Equipo, 2009).....	29
Figura 2.7 Modelo de Procesos de MSF (Microsoft Corporation - MSF Ciclo de Vida, 2009).....	29
Figura 2.8 Triángulo de Negociación de MSF (Microsoft Corporation - MSF Visionamiento, 2009).....	30
Figura 3.1 Roles del Modelo de Equipos de MSF (Microsoft Corporation - MSF Estructura del Equipo, 2009) .....	40

Figura 3.2 Roles combinables para equipos pequeños (Microsoft Corporation - MSF Estructura del Equipo, 2009) .....	46
Figura 3.3 Matriz de Negociación (Microsoft Corporation - MSF Visionamiento, 2009)	52
Figura 3.4 Disciplina de Administración del Riesgo (Microsoft Corporation - MSF Manejo de Riesgos, 2009) .....	55
Figura 3.5 Declaración del Riesgo (Microsoft Corporation - MSF Manejo de Riesgos, 2009).....	56
Figura 3.6 Herramienta para análisis cualitativo de Riesgos (Microsoft Corporation - MSF Manejo de Riesgos, 2009) .....	57
Figura 3.7 Herramienta para la gestión de Riesgos (Microsoft Corporation - MSF Visionamiento, 2009).....	59
Figura 3.8 El Ciclo de una Liberación Interna (Microsoft Corporation - MSF Desarrollo, 2009).....	68
Figura 3.9 Las pruebas no se limitan a una única fase (Microsoft Corporation - MSF Desarrollo, 2009) .....	69
Figura 3.10 Subconjuntos representativos de errores, defectos, etc. (Microsoft Corporation - MSF Desarrollo, 2009).....	71
Figura 3.11 Convergencia de Errores (Microsoft Corporation - MSF Estabilización, 2009) .....	74
Figura 3.12 Rebote de Cero Errores.....	75
Figura 4.1 Evaluación de Capacitación Formal en MSF .....	85
Figura 5.1 Fase de Visionamiento.....	93
Figura 5.2 Construcción del Equipo MSF .....	96

Figura 5.3 Redacción del Documento de Visión y Alcance.....	99
Figura 5.4 Fase de Planificación.....	101
Figura 5.5 Diseño Conceptual.....	102
Figura 5.6 Conversión de Aproximaciones en Planes.....	104
Figura 5.7 Fase de Desarrollo.....	105
Figura 5.8 Fase de Estabilización.....	107
Figura 5.9 Fase de Instalación.....	109
Figura 6.1 Metodologías de desarrollo.....	117

## Introducción

Existen factores comunes en la forma como se llevan a cabo los proyectos de desarrollo de software y el mantenimiento de los sistemas y aplicaciones existentes: aquello que se denomina el Ciclo de Vida de los Sistemas y Aplicaciones, y son entre otros: No existe una descripción clara y concisa de lo que se requiere que se haga, no se declara expresamente hasta donde entregar una solución; es decir, el alcance de un proyecto; y no se efectúa un control eficiente y proactivo de los errores de aplicaciones, tanto en proceso de construcción como cuando se encuentra la solución en producción.

El efecto de estos problemas es que no se cumplan las expectativas, los proyectos se entreguen tarde y fuera de presupuesto, lo que se haya construido no era lo que se necesitaba, no se entienda la información que se necesita para saber qué hacer, no se consigue una cohesión entre los miembros del equipo. Finalmente, la solución entregada sea impredecible en el sentido que siempre se descubre nuevos problemas; y en el mejor de los casos, es muy difícil de usar, y operar bien en el ambiente de producción.

Adicionalmente, las entidades de control, tales como la Superintendencia de Compañías y de Bancos, ejercen presión en las empresas privadas al incluir entre sus criterios de calificación organizacional, el hecho de contar con una metodología de mantenimiento de aplicaciones.

A continuación se barajan algunas alternativas de metodologías de Ciclo de Vida de Sistemas y Aplicaciones, y se elige *"Microsoft Solutions Framework (MSF)"* como metodología base, por razones que posteriormente se exponen en la presente tesis.

*"MSF"* puede ser usado no solamente para proyectos de desarrollo de software, sino que puede ser usado para proyectos de cualquier índole; más aún, incorpora y extiende criterios que promulga PMI, además de buenas prácticas de gestión de proyectos, y se integra con implementaciones de *"ITIL"*; como por ejemplo *"MOF"*.

Se espera que la presente tesis sirva de guía de referencia práctica que englobe perspectivas de alto y bajo nivel de cómo implementar *"MSF"* para creación y mantenimiento de Software dentro de una empresa cuya misión principal o razón de ser de la empresano es la tecnología.

## Capítulo 1: Antecedentes

### Situación actual de la Empresa

La empresa que va a ser objeto del presente estudio –a la cual en adelante llamaremos "la empresa"- es una compañía de prestigio nacional cuyo modelo de negocio; tal como la gran mayoría de empresas de hoy en día, se apalanca extensivamente en un manejo eficiente y efectivo de la información de sus clientes, y los servicios que esta les ofrece. Esto no solo permite modelos predictivos de las tendencias y preferencias de sus clientes, sino que también es imprescindible para la operativa diaria de la empresa: Existen sistemas y aplicativos de misión crítica que necesitan estar habilitados y en completamente perfecto estado 24horas x7días x365 días; característica sin la cual, no sólo que se pone en perjuicio el bienestar de sus clientes, sino que merma

significativamente la imagen y prestigio de la empresa, frente a sus competidores.

La empresa invierte intensivamente en actualizaciones de tecnología, y tanto su parque tecnológico como su infraestructura de software, se mantienen al día en cuanto a las últimas liberaciones de hardware y software.

Sin embargo, y muy a pesar de las dos premisas citadas anteriormente, el esquema organizacional y estructura de talento humano ha permanecido rezagada en el tiempo: el personal de tecnología de información (llamada Área de Sistemas) existente se ha mantenido casi inalterada desde las pasadas dos décadas, y su esquema de trabajo es el mismo que ha dado resultados exitosos en aquel entonces, puesto que el entorno era diferente, las necesidades eran menos apremiantes y la competencia era mucho más manejable y la actual forma de trabajo se acoplaba perfectamente y era suficiente para la demanda de información de aquella época.

## Causas que motivan un cambio de metodología

La empresa cuenta en su organigrama con las siguientes subáreas o departamentos dentro del área de Sistemas: El Departamento de Desarrollo de Sistemas, de Soporte Técnico, de Soporte a Productos (que luego tomó el nombre de Soporte a Servicios), y el de Producción e Infraestructura.

De manera teórica, si existe alguna necesidad que *debe* ser cubierta mediante una solución tecnológica (una aplicación, un sistema o una infraestructura que requiera tanto hardware como software), debería partir de una solicitud de manera "formal" (mediante un e-mail), o mediante un conjunto de reuniones y constancias en forma de Actas de Reuniones, acerca de los términos generales bajo los cuales se espera funcione la solución. Se hace hincapié en el término *debe*, pues para este punto, alguien del área comercial respectiva debió haber evaluado si una solución tecnológica es la más apropiada para traer valor al negocio y en qué términos esto se va a dar.

Si se está hablando de un requerimiento de cambio a una aplicación existente, la(s) única(s) constancias formales de este requerimiento, serán e-mails de algún usuario

(llamado en ciertas ocasiones Usuario Experto) que hable por el área comercial respectiva y bajo cuya responsabilidad recae el costo-beneficio de dicho cambio; típicamente un sub-gerente o usuario de mayor jerarquía dentro del ámbito de operación del producto.

Existe una muy alta probabilidad que hasta este momento la claridad de los requerimientos es muy vaga, pues no se tiene completamente claro el alcance o implicaciones de los mismos. Esto produce que constantemente se realicen cambios o "afinaciones" de los mismos en el transcurso del desarrollo, quedan asunciones cuya interpretación de los mismos queda bajo la responsabilidad del Departamento de Desarrollo; o la aclaración de cualquier duda, se dilucidará mediante llamadas telefónicas, mails o reuniones formales o informales. No se establecen criterios de aceptación ni se cierra el alcance que tendrá la solución, que no sea hasta un nivel macro de casos de uso de negocio, y este es realizado por los mismos clientes internos o usuario de la futura/actual solución.

Finalmente, cuando la solución ha completado su alcance y pasa a revisión y certificación de sus usuarios o se gestiona su paso a producción, se solicitan generalmente cambios de forma que a menudo, por sus características, se convierten en cambios de fondo que retrasan el tiempo de proyecto hasta que sean completadas. Estas solicitudes provienen ya sea del área comercial que solicitó el cambio, como del área de Infraestructura, el cual hasta el momento del paso a producción, pasa virtualmente al margen de la construcción y arquitectura de la solución; y la única oportunidad de objetar algún detalle de esta será en este momento.

## Consecuencias del marco de trabajo actual

Las primeras consecuencias refieren a la imagen del grupo de trabajo ante los clientes internos y externos, puesto que el peso de una solución con defectos o no entregada a tiempo caerá en el equipo de desarrollo, y dado que nadie tiene bien definidas sus responsabilidades y competencias, es el programador al que se le asigna la responsabilidad y culpabilidad de lo desarrollado y puesto en producción.

El segundo grupo de consecuencias repercuten en la satisfacción del empleado, las tensiones en el equipo de trabajo, y el desmedro en la productividad y espíritu de compromiso generalizado, fruto del descontento que produce la falta de procedimientos y políticas claras y justas en las que cada persona dentro del entorno laboral aporte con lo que le corresponde de manera sinérgica, en lugar de delegar erróneamente la responsabilidad de las actividades al compañero más próximo, bajo argumento de una lógica muy subjetiva y empírica, haciendo que cada desarrollo de aplicativos se convierta en ciertas ocasiones en una pugna de poderes desmesurada.

En la gran mayoría de las veces, es el programador el que investiga los requerimientos de negocio, interactúa directamente con los clientes, desarrolla, establece los escenarios y bancos de pruebas, corrige y valida los incidentes de calidad, y en muchas ocasiones es el que pasa a producción e incluso el que usa su propia aplicación.

Generalmente, y dado que la necesidad de mantener la continuidad de negocio disponible, hace que el programador deba realizar sus pasos a producción a altas horas de la noche o en la madrugada sin tener necesidad alguna, pudiendo haber sido evitada involucrado a operaciones desde el comienzo del proyecto y estableciendo un plan asertivo de liberación de versiones.

## **Capítulo 2: Planificación de la Implementación de la Metodología basada en MSF**

### **Estructura de Trabajo para el Plan de Acción**

Lo que principalmente se elaborará en este punto es la conformación de un comité de trabajo voluntario dentro del mismo equipo de trabajo del área, especialmente para elaborar las directrices de esta nueva metodología y encargarse de difundirla entre su entorno más próximo: sus propios compañeros.

El trabajo a cargo de este comité será el siguiente: Investigar acerca de las ventajas y desventajas, así como criterios de aplicabilidad de la mayor cantidad de metodologías y

marcos de trabajo disponibles. Elegir la que mejor se acople a la realidad de la empresa y a las necesidades de la demanda del mercado; en otras palabras, que esté exitosamente vigente hoy en día. Finalmente realizar las gestiones apropiadas de capacitación, replicación interna de información y ejecución o implantación de la metodología dentro de la institución, previo sometimiento a revisión y aprobación de los funcionarios a los que se vean afectados por este cambio. Se trata de un cambio radical de forma de trabajo no sólo del área de Sistemas sino de la empresa en su conjunto; por lo que su implementación posterior deberá tener su plan y cronograma respectivo de trabajo, sin que esto no afecte el funcionamiento normal del modelo de negocio de la empresa.

Se requiere como factor clave de éxito que este grupo de trabajo se comprometa voluntariamente con esta causa, bajo la asunción de que esto no los exime en primera instancia de su desenvolvimiento y tareas cotidianas de su trabajo diario, pero se tenga la voluntad expresa de establecer un cambio definitivo para bien de la empresa.

## Contraste de Metodologías. Investigación

En el siguiente apartado se analizarán los pros y contras de cuatro de las metodologías exitosamente usadas y comprobadas, exponiendo sus características, ventajas y desventajas. Estas son: *Rational Unified Process (RUP)*, *Capability Maturity Model Integration (CMMI)*, *Six Sigma (6σ)* y *Microsoft Solutions Framework (MSF)*.

En el procedimiento actual de puesta a producción de sistemas, así como control de cambios e incidentes o problemas se basa en el apego a ciertos tipos de metodologías, dependiendo de varios factores incidentes y decidores; tales como: impacto en la línea de negocio al cual será destinado, nivel de urgencia o fecha límite de realización, disponibilidad de recursos humanos asignados a cada proyecto o requerimiento, tanto del área de Sistemas como de otras áreas que desempeñan en cada caso el rol de fuente de información útil para el desarrollo de cada proyecto, adicional al de fuente misma del requerimiento. Estas metodologías actualmente empleadas incluyen: Método de la cascada (Winston W. Royce, 1970 (Cascada)), Espiral (Barry Boehm,

1988(Espiral)), y métodos de Desarrollo de Aplicaciones Rápidas (RAD), Programación Extrema (XP, Kent Beck, 1990), Joint o de Sesiones Periódicas (JAD, Chuck Morris y Tony Crawford, 1979), y Lean Development (James Martin, 1991 (RAD)). La presente comparativa, y en general, el presente estudio, no centra su atención en la metodología de creación de software o programación en sí misma, la cual es sólo una de las fases requeridas para ejecutar un proyecto para la construcción de una solución de tinte tecnológico; sino, se enfoca en el modelo de procesos utilizado por los diferentes marcos de trabajo promulgados para ejecución de proyectos de software. En la fase correspondiente, se puede recurrir a cualquiera de estas metodologías de desarrollo. Para más información, consúltese el Apéndice1. Metodologías de desarrollo y programación.

### *RationalUnifiedProcess (RUP)*

Fue creado por la inicialmente separada de IBM división "Rational Software Corporation" desde 1993. Se caracteriza por utilizar extensivamente UML. Se definen como mejores prácticas:

1. Desarrollar software iterativamente.
2. Administrar requerimientos
3. Arquitectura basada en componentes
4. Software visualmente modelado.
5. Verificar calidad de software
6. Controlar cambios en el software

Adicionalmente se definen los siguientes elementos de contenido en el desarrollo de software:

- Roles: Se definen perfiles, competencias y responsabilidades, quién se va a dedicar a qué dentro del flujo de trabajo.
- Productos de trabajo: Representa los resultados de una tarea, incluyen documentos y modelos producidos a través de todo el proceso.

- Tareas: Unidad de trabajo asignado a un rol y cuyo resultado es un producto de trabajo.

El desarrollo de software pasa por las siguientes fases:

- Incepción, se validan costos y presupuestos, modelos de casos de uso, plan de proyecto, restricciones y características clave. Al terminar esta fase se concuerda el alcance, riesgos y costos expuestos por los interesados en el proyecto.
- Elaboración, el proyecto en esta etapa comienza a tener forma y la arquitectura obtiene una forma básica. El modelo de casos de uso en el que se identifican claramente los actores debe estar completado por lo menos en un 80%.
- Construcción, aquí se construye el software, se desarrolla el código y en proyectos grandes se dividen iterativamente los casos de uso produciendo prototipos demostrables.
- Transición, Es el paso a producción en sí, las actividades incluye el entrenamiento de los usuarios finales y pruebas del sistema para validar las expectativas de los usuarios.

Se definen asimismo seis disciplinas de ingeniería:

- Disciplina de modelamiento del negocio: Se establece el valor que la tecnología puede aportar por medio de TI. Explica como describir una visión de la organización en que el sistema será implantado y como entonces se usará esta visión como una base para soportar el proceso, roles y responsabilidades.
- Disciplina de Requerimientos: Explica cómo establecer las peticiones de los accionistas e interesados y transformarlas en productos de trabajo que enmarcan el sistema a ser construido y proveen requerimientos detallados de lo que el sistema debe hacer.
- Disciplina de Análisis y Diseño: La meta es demostrar cómo el sistema que será

realizado cumplirá todos sus requerimientos, en un ambiente específico, las tareas y funciones especificadas en los casos de uso. Es fácil de cambiar cuando los requerimientos funcionales cambian. Sirve como una abstracción del código fuente y describe como las clases, agrupadas en paquetes, con interfaces bien definidas, desarrollarán los casos de uso.

- Disciplina de Implementación, sus propósitos son definir la organización del código en términos de clases, código fuente, archivos binarios, ejecutables, componentes como unidades, e integrarlos como un sistema ejecutable.
- Disciplina de Pruebas, sus propósitos son: verificar la interacción con objetos de negocio y con otros componentes de software, que todos los requerimientos sean implementados correctamente, identificar defectos antes de que sean puestos a producción, y que estos sean arreglados, probados nuevamente y cerrados. Estas pruebas, así como el resto, tienen un enfoque iterativo, esto implica que se pueden detectar defectos tan pronto como esto sea posible, reduciendo costos de corregirlos. Estas pruebas se pueden efectuar en cualquiera de las cuatro fases, en la que se apunta a medir cuatro principales dimensiones: confiabilidad, funcionalidad, rendimiento de la aplicación y rendimiento del sistema.
- Disciplina de despliegue o puesta en producción, Su propósito es producir lanzamientos de productos, y entregar software a usuarios finales. Cubre las actividades de: empaquetamiento, distribución e instalación de software, proveyendo ayuda y asistencia a los usuarios.

Finalmente existen tres Actividades de Apoyo:

- Administración de Cambios, en concordancia con lo que recomienda ITIL (en su versión actual 3.0), en la que se lleva rastro de los requerimientos de cambios que se presenten a lo largo del proyecto.
- Gestión de Configuración, se incluyen actividades como por ejemplo, lo que

ITIL 3.0 denomina CMDB (“*Configuration Management Database*”), en la cual se establecen a través de “*artefactos*”, tales como documentos que detallan los modelos bajo control de versiones cuyos cambios necesitan ser visibles.

- *Manejo de proyecto*, en la que se describe como se cambia de una fase a otra y dentro de una iteración a otra, y los documentos o “*artefactos*” entregables resultan de cada etapa.

La siguiente gráfica mide las disciplinas a través de las diferentes fases del desarrollo de software, midiendo las fases como la organización a través del tiempo (eje X), y las disciplinas a través del contenido (eje Y).



Figura 2.1 Flujos de trabajo en cada fase de RUP(WillyDev)

Cada fase se visualiza como un conjunto de iteraciones y como una secuencia de “*microproyectos*”, ajustados al modelo tradicional de cascada dentro de cada iteración. Puede verse que la acción de cada rol en cada disciplina se traslapa entre sí en un

determinado momento, y esto se presta para que por ejemplo en un determinado momento del tiempo en que la codificación ha llegado hasta un cierto nivel de completitud, el modelamiento o los requerimientos funcionales del sistema puedan variar, haciendo que se haya perdido parte del tiempo empleado en la implementación del sistema basado en las anteriores especificaciones. Adicionalmente, no tiene cabida el concepto de manejo de incidentes, ni aprendizaje de los mismos por medio de una base de conocimientos (KB en ITIL).

### *CapabilityMaturityModelIntegration (CMMI)*

Desarrollado en sus primeras versiones alrededor desde 1987 (como CMM) y continuada hasta 2006 y 2007 en su versión 1.1 y 1.2. Fue creado por miembros de la industria, el gobierno de los EE.UU. y el Instituto de Ingeniería de Software (SEI).

CMMI 1.2 comprende 3 grandes áreas:

- CMMI-DEV (Desarrollo),
- CMMI-ACQ (Adquisición)
- y CMMI-SVC (Servicios en general, aún en borrador).

En CMMI-DEV existen dos modelos: CMMI-DEV y CMMI-DEV + IPPD (*"IntegratedProduct and ProcessDevelopment"*). A su vez contiene a 22 Áreas de Proceso:

- |  |   |
|--|---|
| 1. Análisis de Causas y Resolución (CAR)         | 8. Enfoque Organizacional en Procesos (OPF)                 |
| 2. Gestión de la configuración (CM)              | 9. Rendimiento de Procesos Organizacionales (OPP)           |
| 3. Análisis de Decisiones y Resolución (DAR)     | 10. Formación Organizacional (OT)                           |
| 4. Gestión Integrada de Proyectos (IPM)          | 11. Monitorización y Control de Proyecto (PMC)              |
| 5. Medición y Análisis (MA)                      | 12. Planificación de proyecto (PP)                          |
| 6. Innovación y Despliegue Organizacionales(OID) | 13. Aseguramiento de calidad de Procesos y Productos (PPQA) |
| 7. Definición de procesos organizacionales (OPD) |   |

- |   |                           |
|---|---------------------------|
| 14. Integración de Producto (PI)            | (SAM)                     |
| 15. Gestión Cuantitativa de Proyectos (QPM) | 20. Solución Técnica (TS) |
| 16. Gestión de Requerimientos (REQM)        | 21. Validación (VAL)      |
| 17. Desarrollo de Requerimientos (RD)       | 22. Verificación (VER)    |
| 18. Gestión de Riesgos (RSKM)               |                           |
| 19. Gestión de Acuerdos con Proveedores     |                           |

Como guía para efectuar las actividades respectivas se representan de dos formas, en cada una se expresa una magnitud de nivel de mejoramiento de cada área de proceso:

	<i>Representación Continua</i>	<i>Representación Escalonada</i>
	<b>NIVEL DE CAPACIDAD</b>	<b>NIVEL DE MADUREZ</b>
<b>Nivel 0</b>	Incompleto	-
<b>Nivel 1</b>	Realizado	Inicial
<b>Nivel 2</b>	Manejado	Manejado
<b>Nivel 3</b>	Definido	Definido
<b>Nivel 4</b>	Manejado cuantitativamente	Manejado cuantitativamente
<b>Nivel 5</b>	Optimizando	Optimizando

**Tabla 2.2.1 Niveles de capacidad y madurez de CMMI(Monografías, 2008)**

Cada área tiene su respectivo conjunto de niveles de madurez. La mejora consiste entonces en definir: Componentes Requeridos, que deben ser obligatoriamente satisfechos, agrupados en Objetivos Genéricos y Específicos; y Componentes Esperados, que pueden ser utilizados para obtener los componentes requeridos, y son las Prácticas Genéricas y Específicas.

La siguiente gráfica muestra las relaciones entre varias de las 22 áreas de proceso en la categoría de Ingeniería de Software (CMMI-SE):

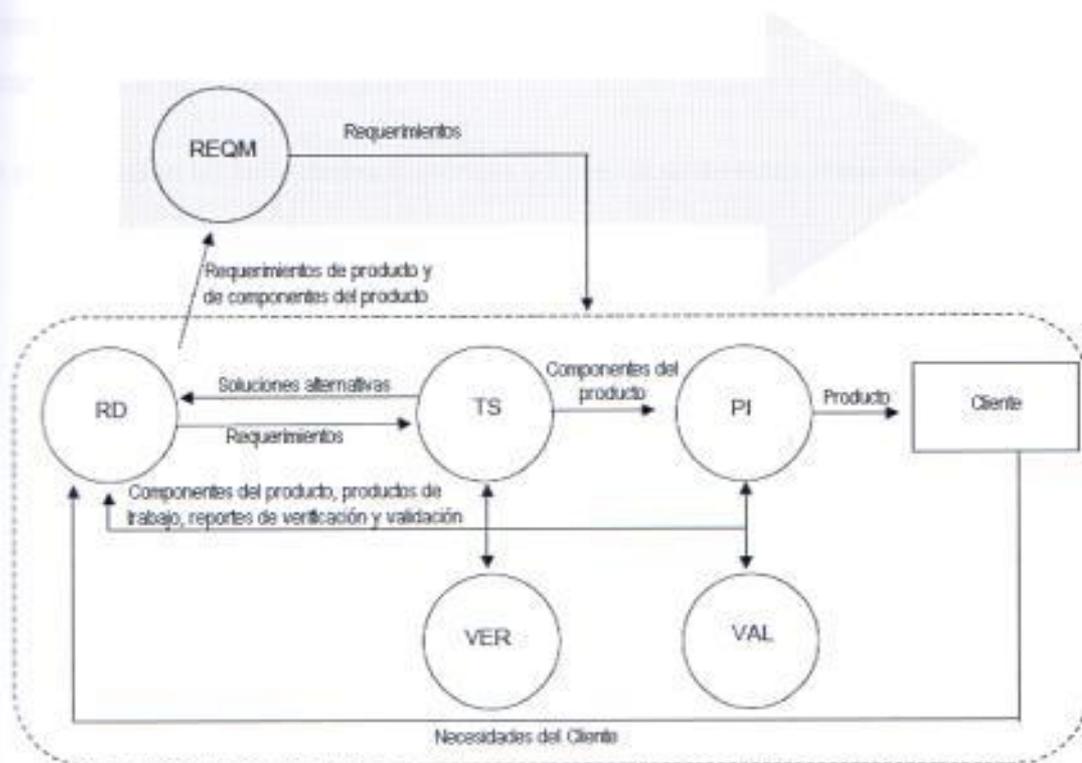


Figura 2.2 Relación entre las áreas de Proceso de CMMI – Ing. Software (Monografías, 2008)

### Relación entre Áreas de Proceso de Ingeniería

La siguiente gráfica muestra las relaciones entre varias de las 22 áreas de proceso en la categoría de Ingeniería de Software (CMMI-SE):

Desarrollo de Requerimientos (RD) identifica las necesidades del cliente y las transforma en "requerimientos del producto" para ser suministrados a Solución Técnica (TS), quien produce la arquitectura del producto, el diseño del producto en sus componentes, y el diseño de cada componente. TS utiliza a Verificación (VER) para la verificación de ese diseño, mismo que será entregado a Integración del Producto (PI) para su implementación. Validación (VAL) es un área incremental que valida el producto, sus componentes, los "artefactos" intermedios y los procesos respecto a las necesidades de los clientes. Los conflictos que son descubiertos son usualmente resueltos entre RD y TS. Los requerimientos son cubiertos por REQM, que describe las actividades para obtener y controlar los cambios, dados por los clientes o incluso por

otras áreas del proceso de ingeniería. Esta última área es recursiva, dinámica y transversal a las demás categorías.

A continuación las actividades Genéricas y Específicas de estas áreas involucradas:

- Administración de Requerimientos
  - SG1. Administrar Requerimientos, e Identificar Inconsistencias con el plan de proyecto.
    - SP1. Obtener una comprensión común de los requerimientos, por fuentes oficiales.
    - SP2. Crear compromisos en los involucrados en el cumplimiento de los mismos.
    - SP3. Administrar cambios en los requerimientos, aplicando métricas apropiadas de volatilidad de requerimientos para juzgar si se requieren nuevos controles o modificar los actuales.
    - SP4. Mantener trazabilidad bidireccional entre requerimientos y artefactos.
    - SP5. Identificar inconsistencias entre artefactos del proyecto y los requerimientos.
- Desarrollo de Requerimientos
  - SG1. Desarrollar requerimientos del cliente, necesidades, expectativas, restricciones e interfaces son recogidas y traducidas en requerimientos del cliente
    - SP1. Obtener necesidades de todas las partes interesadas para todas las fases del ciclo de vida del producto.
    - SP2. Desarrollar los requerimientos del cliente.
  - SG2. Desarrollar requerimientos de productos, en los que los requerimientos del cliente son refinados y elaborados.
    - SP1. Establecer requerimientos del producto y de componentes del producto.
    - SP2. Destinar requerimientos de componentes del producto.
    - SP3. Identificar requerimientos de la interfaz, en el caso de

integración entre productos existentes.

- SG3. Analizar y validar requerimientos, desarrollando una definición de funcionalidad requerida.
  - SP1. Establecer conceptos operacionales y escenarios asociados.
  - SP2. Establecer una definición de funcionalidad requerida.
  - SP3. Analizar requerimientos, jerarquizándolos, y si ellos cumplen con los objetivos de los niveles más altos de la jerarquía del producto, y a su vez proveen mayor definición de niveles más bajos y detallados.
  - SP4. Analizar requerimientos para lograr balance. Costos, cronogramas, funcionalidades, componentes reutilizables, mantenimiento o riesgos.
  - Validar requerimientos tempranamente que dé certeza de una validación final exitosa.

- Solución Técnica

- SG1. Seleccionar soluciones para componentes del producto.
  - SP1. Desarrollar alternativas de solución y criterios de selección.
  - SP2. Seleccionar soluciones para componentes del producto.
- SG2. Desarrollar el diseño.
  - SP1. Diseñar el producto o los componentes del producto.
  - SP2. Establecer un paquete de datos técnicos.
  - SP3. Diseñar interfaces de componentes del producto utilizando los criterios establecidos.
  - SP4. Ejecutar análisis de hacer, comprar o reutilizar (make-or-buyanalysis).
- SG3. Implementar el diseño del producto.
  - SP1. Implementar el diseño del producto en el siguiente nivel de la jerarquía: asignación, refinamiento y verificación.
  - SP2. Desarrollar la documentación de apoyo del producto, para instalar, operar y mantener el producto.

- Integración de Productos

- SG1. Preparación para la Integración del producto.
  - SP1. Determinar la Secuencia de Integración
  - SP2. Establecer el Ambiente de Integración del Producto.
  - SP3. Establecer Procedimientos y Criterios de Integración del Producto.
- SG2. Asegurar la compatibilidad de la interfaces.
  - SP1. Revisar descripción de interfaces para completitud.
  - SP2. Gestionar Interfaces.
- SG3. Ensamblar los componentes del producto y liberarlo.
  - SP1. Confirmar componentes para integración preparados.
  - SP2. Ensamblar los componentes del producto.
  - SP3. Evaluar Componentes del Producto ensamblados
  - SP4. Empaquetar y Entregar Productos o Componentes del Producto.
- Verificación
  - SG1. Preparar la verificación.
    - SP1. Seleccionar artefactos para verificación.
    - SP2. Establecer ambiente para verificación.
    - SP3. Establecer procedimientos y criterios de verificación.
  - SG2. Realizar revisión de pares de artefactos.
    - SP1. Preparar la revisión de pares.
    - SP2. Conducir la revisión de pares.
    - SP3. Analizar los datos de la revisión de pares.
  - SG3. Verificar artefactos seleccionados, contra sus requerimientos específicos.
    - SP1. Realizar la verificación
    - SP2. Analizar los resultados de la verificación, identificando acciones correctivas.
- Validación
  - SG1. Preparar la validación
    - SP1. Seleccionar productos para la validación

- SP2. Establecer el ambiente para la validación
- SP3. Establecer procedimientos y criterios de validación.
- SG2. Validar Productos o Componentes del Producto.
  - SP1. Realizar la validación.
  - SP2. Analizar los resultados de la validación.

Puede darse cuenta que se establece una fuerte separación y distinción entre las Areas de Proceso, definiendo claramente sus roles, alcances, competencias y limitaciones. Cabe resaltar la diferenciación entre VER y VAL, como áreas que se entienden respectivamente con TS y PI, efectuando verificaciones y validaciones sobre componentes o "artefactos" que tienen que ver con un mismo proyecto pero que comprenden diferentes elementos. Lo mismo ocurre con REQM y RD, teniendo a REQM como un facilitador que interactúa entre áreas, las cuales actúan tomando la posta de un "artefacto" terminado.

Lamentablemente, hasta este momento, este modelo no contempla ni Control de Configuraciones ni Base de conocimientos, a menos que estos sean adjuntados al modelo de manera implícita a alguna(s) de las áreas del proceso.

Adicionalmente, este modelo recomienda con tanta especificidad los procesos que llega a ser un poco complejo y rígido en cierto modo ponerlo en práctica. Para proyectos pequeños presenta mayores dificultades y trabas que para proyectos grandes, en los cuales es ideal.

#### *Six Sigma (6 $\sigma$ ) (Mercadeo.com)*

Es una metodología implementada en 1982 por Bill Smith, como una estrategia de negocios y mejora de la calidad, y posteriormente mejorado y popularizado por General Electric y Motorola.

Six Sigma (6 $\Sigma$ )			
Sigma	% Good	% Defects	DPMO
1	30,9%	69,1%	691.462
2	69,1%	30,9%	308.538
3	93,3%	6,7%	66.807
4	99,38%	0,62%	6.210
5	99,977%	0,023%	233
6	99,9997%	0,00034%	3,4

Figura 2.3 Tabla de valores Sigma de defectos permisibles (Mercadeo.com)

Se centra en la eliminación de defectos y fallas en la entrega de un producto o servicio al cliente. Se basa en un estudio estadístico de un valor o conjunto de valores cuantificables que representen a las "fallas" o "defectos", tales como histogramas o diagramas de Pareto. La meta de 6 $\sigma$  es llevar el control de un proceso para llevarlo desde el punto de partida hasta el punto Six Sigma (o desviaciones estándar), es decir, a un nivel de 3.4 defectos por millón de productos producidos. Esto se da mediante un proceso de: definición, medición o muestreo, análisis, mejoramiento y control.



mayor responsabilidad del proyecto.

- El Modelo de Procesos, el cual organiza los procesos necesarios para crear y entregar una solución, a través del ordenamiento del tiempo, dividiéndolo en distintas fases marcadas como hitos.

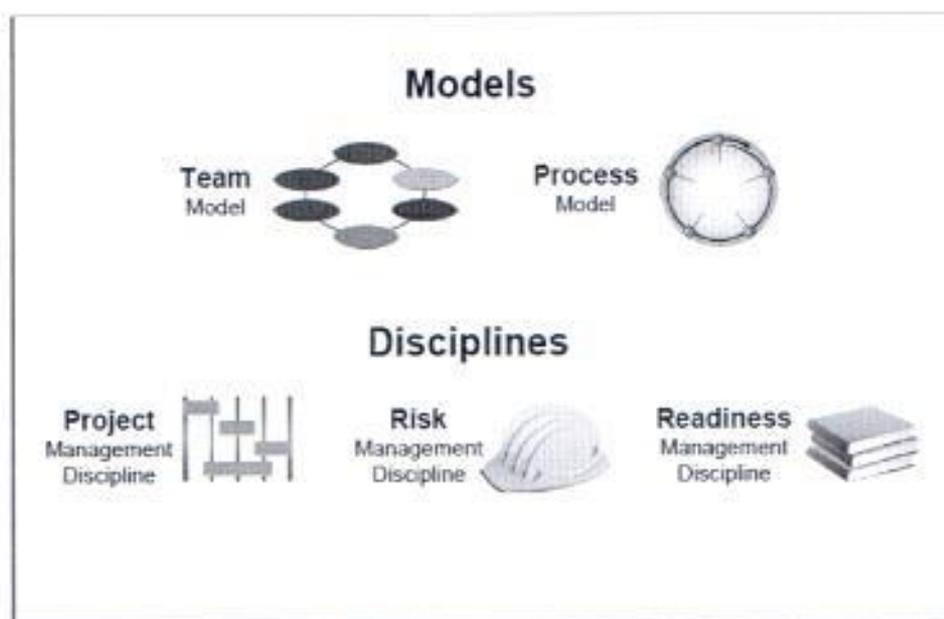


Figura 2.5 Vista general de MSF (Microsoft Corporation - MSF Estructura del Equipo, 2009)

Las disciplinas son las siguientes:

- Disciplina de Administración de Proyectos, que asegura que las actividades de gestión de proyectos sean encaminadas, y que ellas ayuden en lugar de entorpecer el éxito del equipo.
- Disciplina de Administración de Riesgos, que es usada para minimizar sorpresas, "apagado de incendios", y otras actividades caras para proactivamente manejar riesgos.
- Disciplina de Administración de Predisposición y Disponibilidad (Readiness), que es usada para identificar las destrezas requeridas por cada equipo para cada proyecto, y para usar los proyectos como oportunidades para aprender.

## El Modelo de Equipo

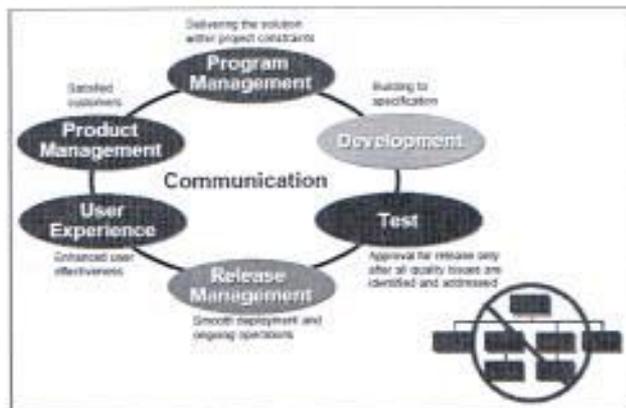


Figura 2.6 Modelo de Equipos de MSF (Microsoft Corporation - MSF Estructura del Equipo, 2009)

Se definen inicialmente los siguientes participantes externos al equipo:

- Socios externos: Individuos o grupos que están interesados o preocupados en el resultado de un proyecto.
- Auspiciante del proyecto. Quien inicia o aprueba un proyecto y su resultado.
- Cliente. Individuo que espera ganar valor de negocio de la solución.
- Usuario Final, quien usa o interactúa con la solución.
- Operaciones, que es la organización de TI responsable de la operación pendiente de la solución después que sea entregada.

## Modelo del proceso



Figura 2.7 Modelo de Procesos de MSF (Microsoft Corporation - MSF Ciclo de Vida, 2009)

En la figura puede verse las diferentes etapas en las que el modelo de proceso se fundamenta. Inicia con un visionado, en el que se definen las metas a alto nivel del proyecto, y básicamente qué deberá hacer la solución. Una fase de planeación en la

que se definen como se construirá la solución y cómo será desplegada. Una fase de desarrollo, en la que se construye y prueba la solución hasta que el alcance haya sido completado. Una fase de estabilización, en la que la solución es probada, todos sus asuntos de calidad hayan sido identificados y resueltos, y luego preparada para su lanzamiento a producción, cuyas tareas se establecen en la última fase de Liberación.

Empieza nuevamente el ciclo si se revisan los objetivos estratégicos, en cuyo caso la funcionalidad del sistema aumentará a través del tiempo por medio del versionamiento.

Se establece una relación directa y de intercambio o trueque mutuo entre tres elementos. Recursos, Cronograma, y Características de la solución, bajo las cuales debe existir el respectivo equilibrio.

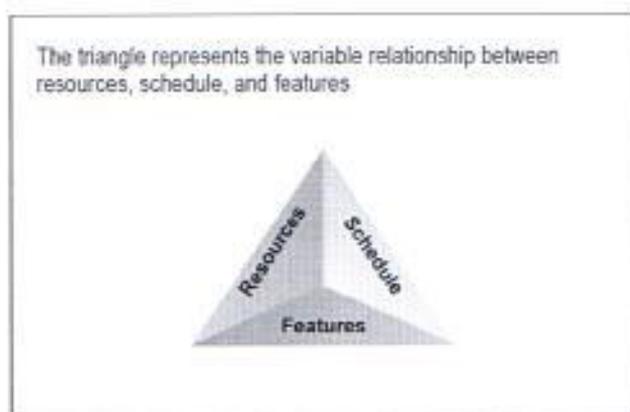


Figura 2.8 Triángulo de Negociación de MSF (Microsoft Corporation - MSF Visionamiento, 2009)

### Disciplina de Manejo del Riesgo

Su función inicial es diferenciar entre riesgo e inconvenientes y problemas. Se encarga de identificar, declarar el riesgo, analizar y priorizar, planificar y calendarizar, reportar y rastrear y finalmente controlar. Este ciclo lleva luego al aprendizaje registrado en una Base de Conocimientos para Riesgos (Microsoft Corporation - MSF Manejo de Riesgos, 2009). Cada riesgo es identificado, su causa y consecuencia, estableciendo someramente la probabilidad de ocurrencia de la causa, y midiendo el costo de oportunidad y pérdida total; es decir el impacto causado una vez que en el proyecto,

una vez que la causa haya sido confirmada (cuando la probabilidad de la causa sea igual a 100%).

MSF tiene la particularidad de ser una metodología perfectamente adaptable a proyectos grandes y pequeños; es decir, es escalable. Adicionalmente, dado su flexibilidad y adaptabilidad, puede convertirse en una metametodología, al permitir utilizarla en conjunción con otras metodologías afines a su esencia, tales como DSL.

Expuestas las características principales de cada una de las metodologías descritas, se puede dar un menor valor a RUP dado el hecho que se da cabida a cambios inesperados en los requerimientos funcionales, convirtiendo al modelo en un modelo parecido en cierto modo a cascada, cambiante constantemente. El modelo Six Sigma, es un modelo reactivo y correctivo que opera en base a problemas suscitados con aplicaciones ya puestas a producción, lo cual más allá de ser una metodología puede ser una forma de medir la calidad del software desplegado.

CMMI es una excelente metodología de desarrollo ordenado de sistemas, definiendo con explícita claridad los roles que ocupa cada área de proceso dentro del ciclo de vida del proyecto; sin embargo, su excesiva rigidez no permite ser usado para proyectos pequeños, o en los que los recursos humanos no son abundantes.

MSF, en cambio ofrece gran flexibilidad al cubrir todos los aspectos del desarrollo de sistemas; tales como riesgo, control de configuraciones y aprendizajes constante; afines claramente a las recomendaciones de ITIL, y que no contradice a lo que especifica COBIT en sus cuatro dominios principales (Planificación y organización, adquisición e implantación, soporte y servicios, y monitoreo).

La propuesta final consiste en utilizar MSF como metodología base para el desarrollo de software, implementando en el Modelo de Equipo, una asignación que apunte al esquema bien diferenciado de CMMI-SE; definiendo en base a esta manera, las tareas que los roles que tengan los roles de Desarrollo a TS y PI, y el rol de Pruebas a VER y VAL.

## Cronograma de Trabajo

Este proyecto, pues en sí mismo es un proyecto, consta de dos fases: una capacitación y el desarrollo de un proyecto piloto. Cuando empieza la segunda fase, arrancan dos en paralelo.

Las capacitaciones se realizarán al personal del área de Sistemas en primera instancia, pues ellos deberán convertirse en los promotores y "evangelizadores" al interno de la

organización, guiando a sus contrapartes a través del desarrollo de la metodología.

Se conformarán para el caso presente los 5 grupos que se describen en la figura a continuación.

Id		Nombre de tarea	Duración
4	<b>0</b>	<b>Cursos de Capacitación Personal Sistemas en MSF</b>	<b>40 días</b>
5		Grupo 1	10 días
6		Grupo 2	10 días
7		Grupo 3	10 días
8		Grupo 4	10 días
9		Grupo 5	10 días
10		<b>Proyecto Piloto</b>	<b>64 días</b>
11		<b>Fase de Visionamiento</b>	<b>64 días</b>
12		Sesiones de Coaching Proyecto Piloto Noviembre	23 días
13		Sesiones de Coaching Proyecto Piloto Diciembre	30 días
14		Reuniones de Presentación y Kick-Off	1 día
15		Elaboración del Documento de Visión y Alcance	30 días
16		Elaboración de la Propuesta de Características	7 días
17		Elaboración de la Plantilla de Riesgo	7 días
18		<b>Implementación MSF en BG</b>	<b>252 días</b>
19		Recopilación de Propuestas	32 días
20		<b>Elaboración de Procedimientos y Políticas</b>	<b>115 días</b>
21		Entrega de la Versión 1.0	85 días
22		Entrega de la Versión 2.1	9 días
23		Entrega de la Versión 2.3	3 días
24		Entrega de la Versión 2.4	2 días
25		Entrega de la Versión 2.6	18 días
26		<b>Ingeniería reversa de programas existentes para Host</b>	<b>118 días</b>
27		Revisión de Documentación existente	6 días
28		Revisión de Documentación disponible para desarrollar	6 días
29		Elaboración de Documentación para programas de Host - Ing. Reversa	90 días
30		Definición de Diseño Lógico y Físico para programas de Host	90 días

El Proyecto Piloto consiste del acompañamiento del tutor en MSF en el desarrollo de un servicio, aplicando cada uno de sus lineamientos

## Capítulo 3: Ejecución del Proceso de Implementación de la Metodología

### Capacitación del Personal Actual de Sistemas en MSF

Después de completar este curso (40 h), los alumnos estarán en capacidad de:

- Explicar cómo es que una estrategia de solución puede simplificar los retos que enfrentamos cuando estamos desarrollando soluciones tecnológicas.
- Describir cómo es el modelo de equipos de MSF y cómo se aplica a un proyecto MSF.
- Discutir los conceptos de grupos de roles en MSF.
- Explicar cómo se escalan los equipos de MSF, dependiendo del tamaño y la complejidad del proyecto.
- Explicar la administración de proyectos en equipos MSF.
- Explicar cómo se aplica la disciplina de Administración de Riesgos a los proyectos MSF.
- Explicar la fase de Visión de la solución.
- Explicar cómo se asegura que la solución cumple con los requerimientos y metas iniciales del negocio.
- Explicar cómo es que los equipos planean qué es lo que van a desarrollar en la solución.
- Explicar la fase de Construcción de la solución.
- Explicar la fase de Estabilización de la solución.
- Describir las tareas involucradas en las pruebas y el piloto de la solución.
- Explicar la fase de Instalación de la solución.
- Describir las tareas involucradas en la implementación de la solución en un ambiente de producción.
- Describir las labores involucradas en la terminación del proyecto.

Contenido del curso

#### ***Módulo 1: Introducción a Microsoft Solutions Framework.***

- Explica las características del ambiente organizacional actual que afectan a TI.
- Sugiere algunas maneras en las que TI puede ayudar a las organizaciones a alcanzar sus metas.
- Explica algunos de los factores que se interponen en el camino de los proyectos exitosos de TI.
- Describe a dónde se debe dirigir la atención de TI para lograr que sus proyectos sean más efectivos.
- Demuestra cómo MSF interactúa con el Microsoft Operations Framework (MOF).

Al completar este módulo el alumno estará en capacidad de:

- Explicar el propósito de las características claves de MSF.
- Explicar cuándo y por qué se debe usar MSF.
- Definir la terminología clave de MSF.
- Explicar cómo una estrategia de solución puede aliviar los retos de desarrollar soluciones de tecnología.
- Identificar cómo se hubieran podido resolver los problemas de los proyectos de TI que han fallado si se hubiera aplicado MSF.

## **Módulo 2: Construyendo un equipo MSF.**

- Discute algunos de los problemas típicos de los proyectos en problemas o fallidos.
- Demuestra cómo las metas básicas para los proyectos exitosos contienen los remedios para los problemas más comunes.
- Explica cómo los principios fundamentales de MSF guían el funcionamiento del equipo.
- Presenta el Modelo de Equipos de MSF.
- Discute los grupos de roles y las áreas funcionales.
- Identifica las formas de escalar los equipos.
- Introduce el concepto de Equipos de Características y Equipos de Funciones.
- Define la administración del proyecto y sus múltiples áreas de responsabilidad.

### **Lecciones:**

- El Modelo de Equipos.
- Los Grupos de Roles de MSF.
- Escalando los equipos para eficiencia del proyecto.
- Una aproximación escalable a la administración de los equipos.

### **Actividades:**

- Comunicándonos en los equipos.

### **Al completar este módulo el alumno estará en capacidad de:**

- Describir el propósito y las metas de cada miembro del equipo en un proyecto MSF.
- Identificar las tareas, las responsabilidades y las áreas funcionales de un Grupo de Roles.
- Explicar cómo organizar un equipo MSF con un número específico de participantes.
- Identificar las seis metas de calidad que impulsan al equipo y definen el Modelo de Equipos.
- Listar las características de los equipos efectivos.
- Explicar cómo los equipos MSF pueden ser escalados hacia arriba y hacia abajo dependiendo de las necesidades.
- Explicar cómo la administración de proyectos se distribuye a lo largo de todos los líderes del equipo en un proyecto MSF.
- Describir los beneficios entregados al equipo por medio de la disciplina de administración de proyectos.

## **Módulo 3: Administrando los riesgos del proyecto.**

- Este módulo explica cómo se aplica la disciplina de administración de riesgos en un proyecto MSF y describe los seis pasos del proceso de administración de riesgos MSF.
- Lecciones
- Administrando los riesgos en todo el proyecto.
- Pasos para una administración efectiva de los riesgos.
- Actividades:
- Creando planes de riesgos.
- Identificación y análisis de riesgos.
- Al completar este módulo el alumno estará en capacidad de:

- Explicar cómo se aplica la disciplina de administración de riesgos en un proyecto MSF.
- Describir los seis pasos del proceso de administración de riesgos MSF.

#### ***Módulo 4: Estableciendo el ciclo de vida de sus proyectos.***

- Este módulo introduce el Modelo de Procesos de MSF. Adicionalmente explica cómo se relaciona el Modelo de Equipos de MSF con el Modelo de Procesos de MSF.

##### **Lecciones:**

- Conceptos básicos del Modelo de Procesos de MSF.
- Aclarando las responsabilidades de cada área.

##### **Al completar este módulo el alumno estará en capacidad de:**

- Describir las cinco fases del Modelo de Procesos de MSF y sus hitos principales.
- Describir cómo se relaciona el Modelo de Equipos de MSF con el Modelo de Procesos de MSF.

#### ***Módulo 5: Iniciando su proyecto.***

- Este módulo se enfoca en la fase de Visión, la primera de las cinco fases del Modelo de Procesos.

##### **Lecciones:**

- La fase de Visión de MSF.
- Seleccionando y preparando el Equipo del Proyecto.
- Definiendo la solución.
- Definiendo el alcance de la solución.
- Estableciendo las bases para Revisiones y Cambios.

##### **Actividades:**

- Haciendo el borrador del documento de Visión/Alcance.

##### **Al completar este módulo el alumno estará en capacidad de:**

- Explicar la meta, los hitos y los entregables de la fase de Visión de MSF
- Discutir las consideraciones para seleccionar y preparar un Equipo del Proyecto.
- Identificar y describir los elementos recomendados para el documento de Visión y Alcance.
- Explicar las técnicas de MSF para gestionar el alcance de la solución.
- Explicar cómo el control de cambios y la administración de configuraciones pueden usarse para mantener la trazabilidad hasta las metas del negocio.

### ***Módulo 6: Planeando el proyecto.***

Este módulo se enfoca en la fase de planeación, la segunda de las cinco fases del Modelo de Procesos.

#### **Lecciones:**

- La fase de Planeación de MSF.
- Planeando qué se va a construir.
- Planeando cómo se va a construir.
- Planeando cuándo se va a construir.
- Preparando el ambiente.

#### **Actividades:**

- Asociando elementos de trabajo a los planes.

Al completar este módulo el alumno estará en capacidad de:

- Explicar las metas, los hitos y los entregables de la fase de Planeación de MSF.
- Describir cómo el MSF utiliza tres diseños de vista diferentes para soportar una solución.
- Identificar cómo el plan maestro del proyecto y sus respectivos planes guían al equipo en la manera de construir la solución.
- Describir los criterios de éxito de la fase de Planeación de MSF.

### ***Módulo 7: Desarrollando la solución.***

Este módulo se enfoca en la fase de desarrollo, la tercera de las cinco fases del Modelo de Procesos.

#### **Lecciones:**

- La fase de Desarrollo de MSF.
- Probando la solución.

Al completar este módulo el alumno estará en capacidad de:

- Explicar las metas, los hitos y los entregables de la fase de Planeación de MSF.
- Explicar las metas, los procesos y el resultado de las pruebas durante la fase de desarrollo de un proyecto MSF.

### ***Módulo 8: Estabilizando su solución.***

Este módulo se enfoca en la fase de estabilización, la cuarta de las cinco fases del Modelo de Procesos.

#### **Lecciones:**

- La fase de Estabilización de MSF.
- Pruebas y Pilotos para la estabilización.

Al completar este módulo el alumno estará en capacidad de:

- Explicar las metas, los hitos y los entregables de la fase de Estabilización de MSF.
- Describir las tareas involucradas en las pruebas y el piloto de la solución.

### ***Módulo 9: Instalando su solución.***

Este módulo se enfoca en la fase de Instalación, la quinta de las cinco fases del Modelo de Procesos. Esta es la fase en donde el equipo lleva la solución al ambiente de producción.

#### **Lecciones:**

- La fase de Instalación de MSF.
- Instalando en un ambiente de producción.
- Completando su proyecto.

Al completar este módulo el alumno estará en capacidad de:

- Explicar las metas, los hitos y los entregables de la fase de Instalación de MSF.
- Describir los pasos para instalar una solución en un ambiente de producción.
- Describir las actividades recomendadas involucradas en la terminación del proyecto.

## **Coaching del Proyecto Piloto**

Mediante la ejecución de esta acción, la cual será en paralelo a la anterior, se planifica el acompañamiento y tutoría al equipo de trabajo de un proyecto institucional. Las pre-condiciones necesarias para empezar este acompañamiento son las siguientes:

- Se requiere de la aprobación de un proyecto institucional, de amplia importancia para la empresa, cuya atención del gobierno corporativo sea fácilmente ganable.
- El proyecto fue sometido por algún comité institucional, y priorizado debidamente, de tal manera que la disponibilidad de los recursos no sea un impedimento sino un punto a favor del éxito del proyecto.
- El equipo de trabajo haya sido conformado en base a las destrezas y capacidades propias del entorno de trabajo correspondiente al ámbito del proyecto.
- Se recomienda inclusive que el equipo físicamente se encuentre en la misma sala de trabajo, durante el tiempo que dure el proyecto, eso facilita las

comunicaciones. De manera general esta recomendación se sugiere para todos los proyectos, no solo para este piloto.

- El proyecto debe contar con el apoyo y colaboración de todos los interesados y auspiciantes dentro de la organización. Esto facilita la gestión del tutor o coach.
- La labor de este tutor es recomendar en todo momento, o en común acuerdo con la organización en cuanto al número de horas de acompañamiento diaria, semanal, etc., cualquier acción o gestión que deba realizar el equipo de trabajo del proyecto, siguiendo los lineamientos que sugiere MSF y con esto ayudarlos a cumplir con éxito los objetivos del proyecto y a la vez tener la experiencia de realizarlo siguiendo MSF.

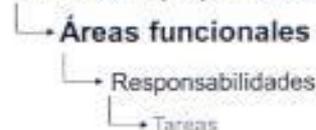
## Modelo de Equipos de MSF

Un rol en MSF identifica un conjunto de áreas funcionales relacionadas y las responsabilidades que están relacionadas con estas áreas. Cada uno de estos roles contiene de tres a seis áreas que son necesarias para ayudar su meta de calidad. Las áreas funcionales casi siempre pero no siempre requieren similares conjuntos de destrezas. Las responsabilidades definidas para cada área necesitan ser llevadas a cabo para cumplir con la meta de calidad para la solución

## Roles del Equipo MSF y sus Áreas Funcionales



### Rol de Equipo MSF



### Ejemplo

#### Program management



Figura 3.1 Roles del Modelo de Equipos de MSF (Microsoft Corporation - MSF Estructura del Equipo, 2009)

El trabajo de los roles en el modelo de equipos de MSF está estructurado como sigue:

- Cada uno de los seis roles contiene varias áreas funcionales
- Cada área funcional tiene varias responsabilidades principales asociadas con ella.
- En cambio, cada responsabilidad puede ser descompuesta en tareas requeridas para llevarla a cabo.

## ***Rol de Program Management***

La meta principal de este rol es **Entregar la solución dentro de las restricciones del proyecto**. Cuatro áreas funcionales soportan esta meta. Ellas están listadas a lo largo de sus responsabilidades asociadas como sigue:

### **Administración del Proyecto**

- Llevar rastro y administrar el presupuesto y el cronograma maestro del proyecto.
- Conducir el proceso de la administración de riesgos.
- Administrar la asignación de recursos y facilitar la comunicación dentro del equipo.
- Rastrear el progreso y administrar los reportes de estado.

### **Arquitectura de la solución**

- Conducir el diseño global de la solución, y administrar la especificación funcional.
- Administrar el alcance de la solución y decisiones críticas de negociación.

### **Aseguramiento del proceso**

- Conducir el aseguramiento de calidad del proceso, y definir y recomendar mejoras.

### **Servicios administrativos**

- Implementar procesos y ayudar a los líderes de equipo a usarlos.

## ***Rol de Development***

Este rol usa la arquitectura de la solución, los diseños de la solución y la especificación funcional para crear la solución.

Su meta principal es construir la solución de acuerdo a esas especificaciones.

Cuatro áreas funcionales soportan esta meta:

#### Consultoría tecnológica

- Servir como consultor tecnológico; evaluando y validando tecnologías.

#### Arquitectura y diseño de la implementación

- Mapear la arquitectura de la empresa y la implementación de la solución.
- Apropiarse e implementar los diseños físicos de la solución.

#### Desarrollo de la aplicación

- Codificar características para cumplir con especificaciones de diseño.
- Conducir revisiones de código; conducir pruebas de unidad con la ayuda del rol de Test.

#### Desarrollo de la infraestructura

- Desarrollar características para cumplir especificaciones de diseño.
- Desarrollar documentación de instalación; scripts para instalación automática.
- Conducir revisiones de código; llevar a cabo pruebas unitarias con la ayuda del rol de Test.

#### ***Rol de Test***

Todo software se entrega con defectos. La meta principal del rol de test es aprobar una solución para liberación o instalación solo cuando todos los asuntos o incidentes de calidad hayan sido identificados y resueltos. "Resolver" puede significar arreglar el defecto en cuestión o documentar soluciones temporales (work-around).

Tres áreas funcionales soportan esta meta. Se enlistan junto con sus responsabilidades asociadas como sigue:

### Planificación de Pruebas

- Desarrollar una aproximación de las pruebas y plan.
- Participar en la configuración de la Barra de Calidad.
- Desarrollar especificaciones de las pruebas.

### Ingeniería de Pruebas

- Desarrollar y mantener casos de pruebas automatizados, herramientas y scripts
- Conducir pruebas para determinar exactamente el estado del desarrollo de la solución.
- Manejar el proceso de construcción.

### Reportería de Pruebas

- Proveer al equipo de información de la calidad del producto.
- Rastrear todos los errores y asuntos de comunicación para asegurarse su resolución antes de la liberación.

### ***Rol de Release Management***

La meta principal de este rol es una instalación y operaciones subsiguientes sin inconvenientes. Cinco áreas funcionales soportan esta meta. Estas están listadas junto con las responsabilidades asociadas como sigue:

#### Infraestructura

- Desarrollar una planificación de infraestructura, y políticas y procedimientos.
- Coordinar uso y planificación del ambiente físico a través de las geografías.
- Administrar adquisiciones de hardware y software.
- Construcción de ambientes de pruebas y etapas que exactamente reflejen producción.

#### Soporte

- Proveer enlace primario y servicio al cliente a los usuarios de IT.

- Administrar los acuerdos de nivel de servicio con el cliente
- Proveer resolución de problemas e incidentes.

#### Operaciones

- Manejar cuentas, controles de configuración del sistema, mensajería, bases de datos, etc.

#### Logística

- Proveer deberes de administración de logística al equipo

#### Administración del lanzamiento comercial

- Manejar todos los aspectos de obtener el producto en el canal

#### ***Rol de UserExperience***

La meta principal de este rol es mejorar la efectividad de la solución para los usuarios. Seis áreas funcionales soportan esta meta. Estas están listadas junto con las responsabilidades asociadas como sigue:

#### Accesibilidad

- Conducir conceptos de accesibilidad y requerimientos dentro del diseño.

#### Internacionalización

- Mejorar la calidad y la usabilidad de la solución en los mercados internacionales.

#### Abogacía por el usuario

- Actuar como un abogado del usuario ante el equipo del proyecto.

#### Material de soporte y entrenamiento

- Desarrollar y ejecutar estrategias de aprendizaje.

- Diseñar y desarrollar sistemas de soporte y documentación de ayuda.

### Usabilidad

- Recopilar, analizar y priorizar requerimientos del usuario; desarrollar escenarios de uso, casos de uso; proveer retroalimentación en el diseño de la solución.

### Diseño de interfaces de usuario

- Conducir el diseño de las interfaces del usuario.

### ***Rol de Product Management***

El rol de Product Manager se enfoca en los clientes y su satisfacción. Nótese que es muy importante que el cliente para cada proyecto sea claramente identificado y entendido (el cliente del proyecto requiere de una característica particular que puede no ser el auspiciante del proyecto). Cuatro áreas funcionales soportan esta meta. Estas están listadas junto con las responsabilidades asociadas como sigue:

### Valor de negocio

- Definición y mantenimiento de la justificación del negocio para este proyecto.
- Definición y medida de la realización valor de negocio y métricas.

### Marketing

- Conducir mercadeo y mensajes de relaciones públicas.

### Abogacía por el cliente

- Conducir una visión compartida del proyecto y la solución, mientras se manejan las expectativas y comunicaciones del cliente.

### Planificación del Producto

- Recopilación, análisis y priorización de los requerimientos del cliente y del

negocio.

- Determinación de las métricas del negocio y criterios de éxito.
- Identificación de los planes de liberación multi-versión.

## Escalamiento hacia abajo – Combinación de roles para equipos más pequeños

Es claro que las metas de los roles pueden tener niveles de conflicto. Estas tensiones hacen al modelo dinámico pero incrementa las posibilidades de problemas cuando se trata de combinar roles. Las combinaciones de roles no son raras –y si el equipo elige combinaciones de roles y maneja activamente los riesgos asociados, los problemas que ocurren deberían ser mínimos.

Roles *pueden* ser combinados, pero algunas combinaciones poseen *riesgos*

	Product Management	Program Management	Development	Test	User Experience	Release Management
Product Management		<b>N</b>	<b>N</b>	<b>P</b>	<b>P</b>	<b>I</b>
Program Management	<b>N</b>		<b>N</b>	<b>I</b>	<b>I</b>	<b>P</b>
Development	<b>N</b>	<b>N</b>		<b>N</b>	<b>N</b>	<b>N</b>
Test	<b>P</b>	<b>I</b>	<b>N</b>		<b>P</b>	<b>P</b>
User Experience	<b>P</b>	<b>I</b>	<b>N</b>	<b>P</b>		<b>I</b>
Release Management	<b>I</b>	<b>P</b>	<b>N</b>	<b>P</b>	<b>I</b>	

**P** Posible    **I** Improbable    **N** No Recomendado

Figura 3.2 Roles combinables para equipos pequeños (Microsoft Corporation - MSF Estructura del Equipo, 2009)

Este ejemplo muestra una permutación de las posibilidades de combinar roles para un equipo pequeño de proyecto. Estas tres personas consisten de dos persona tomando múltiples que cuando son unidos tienen la más baja cantidad de riesgo asociado con su combinación; tanto como mantener al individuo con el rol de Developer aislado de otros roles adicionales.

## Desarrollo de Procedimientos y Políticas acordes a MSF

El tercer paso dentro de esta estrategia de ejecución, en paralelo a las otras dos, consiste en documentar y validar en función de las lecciones aprendidas del proyecto piloto y de la capacitación, un proceso lo más detallado posible, pero a la vez, lo más flexible sobre los pasos a seguir, según los lineamientos de MSF. Recordemos que MSF no es en sí misma una metodología, sino una disciplina o un marco de trabajo. La metodología la realiza cada empresa de acuerdo a sus necesidades, realidades y expectativas. MSF se basa no solamente en un modelo de procesos y de equipos, sino también en principios fundamentales y conceptos clave o prácticas probadas.

### Principios fundamentales aplicados al Modelo de Equipos y Conceptos clave o prácticas probadas

#### *Principios fundamentales*

- **Trabajar hacia una visión compartida:** Una visión compartida para un proyecto es fundamental para el trabajo del equipo. El proceso de crear una visión ayuda a clarificar metas y traer conflictos y asunciones incorrectas a la luz, de tal manera que puedan ser resueltas. Una vez que se lleguen a un acuerdo, la visión motiva al equipo y ayuda a asegurar que todos los esfuerzos estén alineados en servicio de la meta del proyecto. Esto también provee una manera de medir su éxito.
- **Enfoque en el valor de negocio:** Mantener el foco en el valor de negocio es importante porque proveer valor de negocio es el mandato más básico de IT. Mantener la realidad del negocio y las metas de negocio específicas de un proyecto claramente en mente ayuda al equipo a tomar decisiones y negociar en función a características de la solución o de otros aspectos del proyecto, si llegasen a ser necesarios.
- **Permanecer ágil y esperar cambios:** Un equipo que está mentalmente preparado para el cambio y tiene la agilidad de estar hábil para capitalizar en oportunidades y esquivar problemas potenciales mejora en grado superlativo

las oportunidades de éxito. Incluso los proyectos más cuidadosamente planificados están sujetos a cambios, los cuales pueden venir de un sinnúmero de fuentes diferentes, tales como presiones de negocio y desarrollos tecnológicos.

- **Facultar a los miembros del equipo:** Significa darles los recursos y autoridad para cumplir sus responsabilidades asociadas con sus roles. No solo ayuda a los individuos a hacer su trabajo, sino que conduce la habilidad de confiar en compañeros de equipo para hacer y entregar sus compromisos.
- **Fomentar comunicaciones abiertas:** La comunicación es el centro del modelo de equipos porque es crítico para la habilidad del equipo para trabajar juntos y cumplir con las metas del negocio. Los problemas de comunicación, conducen a la falta de entendimiento o conciencia, son frecuentemente citadas como una causa raíz de fallas en los proyectos; el modelo de equipos es específicamente estructurada para eliminar las barreras de comunicación que existen en estructuras de equipos más jerárquicas. Comunicaciones abiertas significan comunicaciones en dos vías –diciendo y escuchando. MSF aboga por el fomento de las comunicaciones abiertas dentro del equipo y con los interesados.
- **Establecer rendición clara de cuentas, responsabilidad compartida:** Este principio tiene una cara interna y externa. Internamente, el equipo comparte la responsabilidad equitativamente para el éxito del proyecto en reconocimiento de la idea que el proyecto no puede ser considerado exitoso si uno de sus metas no han sido alcanzadas. Clara rendición de cuentas se refiere a un requerimiento usual por los clientes del proyecto y/o los auspiciantes para tener un único, punto explícitamente asignado de rendición de cuentas para el ulterior éxito o falla de un proyecto. Adicionalmente, otros interesados externos pueden requerir rendición de cuentas con respecto a metas definidas.

### ***Conceptos clave y prácticas probadas***

**Equipo de iguales:** Este concepto ve a un equipo como un grupo de roles claramente definidos, cada uno apoderándose de una meta definida que es necesaria para el éxito del proyecto. El equipo es no jerárquico. MSF ha encontrado que ciertas actitudes

mentales o mentalidades (mindsets), son clave para su funcionamiento. A continuación se mencionan.

**Mentalidad enfocada al cliente:** Significa que los miembros del equipo están continuamente mentalizados que los clientes satisfechos son prioridad número uno. Este concepto está estrechamente relacionado con el principio de mantener el foco en el valor de negocio, porque los clientes son aquellos que reciben el valor del negocio.

**Mentalidad de producto:** Representa una concepción del trabajo. Ver el trabajo de uno como parte del proyecto que está apuntando hacia la entrega de un producto, tangible o intangible, ayuda a los miembros del equipo a entender el significado de una tarea en particular que estén llevando a cabo en el contexto de una meta real del equipo.

**Mentalidad cero defectos:** Representa un compromiso en que cada parte de cada miembro del equipo a alcanzar una Barra de Calidad predefinida en su trabajo a través del proyecto. Esto no significa literalmente cero defectos. Significa construir Calidad en el trabajo durante el transcurso de todo el proyecto como oposición a meramente verificarla al final.

**Voluntad de aprender:** Es otra actitud prerrequisito para el óptimo funcionamiento del equipo. Ayuda al aprendizaje de nuevas destrezas y conocimiento que son necesarias para el trabajo, aprendizaje que trabaja para éxitos sucesivos, y aprendizaje de errores para evitar repetirlos. Ayuda a los equipos a romper paradigmas de hacer las cosas con su tolerancia a errores como parte del costo del progreso. Está estrechamente relacionado con el principio "Permanecer ágil y esperar cambios".

MSF ha adoptado tres prácticas que han sido probadas como efectivas para mejorar las tasas de éxito de los equipos en entregar soluciones. La primera, usar equipos pequeños e interdisciplinarios, forma parte de un equipo MSF. Las razones que estos equipos son exitosos están estrechamente relacionados con las comunicaciones y los principios de agilidad. Trabajar juntos en un sitio común es otra manera de eliminar las barreras de comunicaciones y construir un sentido de identidad y unidad de equipo. La

participación total en el diseño de la solución asegura que la solución tiene información de entrada de todas las principales perspectivas y refuerza el sentido de responsabilidad que cada miembro de equipo tiene para la solución.

### **Visionamiento (Microsoft Corporation - MSF Visionamiento, 2009)**

Lo primero que MSF indica luego de categorizar (planeado o no planeado) y tipificar un proyecto (alto potencial, estratégico, operacional clave o de soporte), es construir el equipo de trabajo con el que se trabajará. El equipo de trabajo conformado será capaz de definir las respectivas secciones de los siguientes documentos (artefactos o entregables):

- Documento de Estructura del Proyecto
- Documento de Visión/Alcance
  - Propuesta de Características
- Herramienta de Evaluación Inicial de Riesgos

Dados estos entregables, podemos referir hitos internos en esta fase, los cuales son los siguientes:

- Equipo central organizado.
- Visión/Alcance esbozado.

El hito final será en esta fase: tener la Visión/Alcance aprobados por el equipo y por el Cliente.

### ***Estructura del Proyecto (Microsoft Corporation - MSF Visionamiento, 2009)***

Este documento contiene entre las principales secciones: las Metas, Objetivos, Supuestos y Restricciones del proyecto; así como definir el alcance del mismo en base a herramientas como el triángulo y la matriz de negociación; y las siguientes aproximaciones o estimaciones de administración:

- Aproximación a la Administración de Riesgos y Problemas
- " " de Configuración
- " " de Cambio
- " " de Liberación
- " Aseguramiento de Calidad del Proyecto
- " Comunicación del Proyecto
- " Ambiente de Equipo

### Definir el Alcance de la Solución

El alcance comprende cada una de las partes de la visión para la solución que pueda ser cumplida dentro de las restricciones de una versión dada. Hay que diferenciar dos conceptos:

1. Alcance de la Solución: Es la suma de los productos y servicios a ser provistos en una versión de la solución.
2. Alcance del Proyecto: El trabajo desarrollado por el equipo para entregar ciertos ítems en el alcance de la solución.

Bajo esta diferenciación, múltiples proyectos pueden ser requeridos para completar la solución entera. Para que estos proyectos cumplan con el alcance de la solución, se restringe su alcance por medio del versionamiento; esto es, definir a la solución dividiéndola en una serie de liberaciones versionadas, determinando el contexto de la solución actual y las siguientes, se crea un plan multi-versión, y se define el alcance de la versión 1.0.

Para evitar una expansión no controlada del alcance, se manejan herramientas como el Triángulo de Negociación y la Matriz de Negociación.

#### *Triángulo de Negociación (véase Figura 2.2.8)*

En proyectos, una relación existe entre las variables de proyecto de los recursos (gente y dinero), cronograma (tiempo), y características (alcance). Después que el equipo establece los valores para cada lado del triángulo, cualquier cambio en uno de sus lados requiere un ajuste en uno o en ambos lados restantes para mantener el balance

del proyecto. Esto incluye, potencialmente, el mismo lado en que el cambio primero ocurrió.

El nivel fijo de calidad es presumido, y no es negociable en la mayoría de los casos.

La clave para liberar una solución que se ajuste a todas las necesidades del cliente es encontrar el correcto balance entre los recursos, fecha de liberación y características.

Los clientes prefieren no cortar con características favoritas y no pueden entender cómo mantenerlas afecta el cronograma del proyecto y los recursos. Se usa el triángulo de negociación para explicar estas restricciones y presentar opciones de negociación a los clientes.

### *Matriz de Negociación*

Otra herramienta poderosa para manejar negociaciones. Este es un acuerdo entre el equipo y el cliente, realizado desde un comienzo del proyecto, que implica las prioridades por defecto cuando se trate de decisiones de negociación. Pueden haber excepciones sobre estas prioridades, si fuese necesario. Pero el beneficio principal de establecer prioridades por defecto es ayudar a hacer estas decisiones menos polémicas.

	Fijos	Elegibles	Ajustables
Recursos	✓		
Cronograma		✓	
Características			✓

Figura 3.3 Matriz de Negociación (Microsoft Corporation - MSF Visionamiento, 2009)

La matriz usa estos términos:

<b>Fijo</b>	Describe restricciones del proyecto que son esencialmente invariables
-------------	---

<b>Elegible</b>	Describe aquellas que son identificadas como prioridades deseadas.
<b>Ajustable</b>	Describe las que pueden ser ajustadas para acomodar las otras dos.

Para entender como la matriz de negociación funciona, las variables de **recursos**, **cronograma** y **características** pueden ser insertados en los blancos de la siguiente oración:

*Dadas las \_\_\_\_\_ Fijas, nosotros escogeremos un \_\_\_\_\_, y ajustaremos el \_\_\_\_\_ como sea necesario.*

### ***Visión y Alcance(Microsoft Corporation - MSF Visionamiento, 2009)***

#### Definición del Problema u Oportunidad

Las tareas a efectuarse son: recolectar información a través de entrevistas individuales o en grupo de interesados (*stakeholders*—por ejemplo un vicepresidente), detectar asuntos pendientes conocidos, consolidar y analizar la información, y documentar el problema u oportunidad de manera clara.

Esta sección reviste la siguiente importancia: Establece la motivación para iniciar el proyecto; direcciona qué y por qué se lo quiere hacer; está enfocado a nivel de negocio; y puede incluir información relevante adicional, tales como información de mercado, análisis competitivo y retroalimentación del cliente.

#### Creación de una Visión<sup>1</sup> Compartida

Las tareas a efectuarse son: Elaborar una visión en reuniones del equipo central, o aceptar una visión dada al equipo por auspiciantes como parte de una asignación de proyecto.

Como importancia: orienta al equipo en una dirección común, simplifica y asegura consistencia en la toma de decisiones, motiva al equipo, refuerza las metas de la solución y mantiene el enfoque en la calidad de la solución.

<sup>1</sup>Visión: Una vista sin límites de la solución. (Microsoft Corporation - MSF Visionamiento, 2009)

## Recopilar Requerimientos<sup>2</sup> de Alto Nivel

Las tareas necesarias son: Manejar el proceso, desde el punto de vista del *Product Manager*; usar técnicas como entrevistas, encuestas, prototipos, observación y revisión de documentación existente; y enfocarse en el “qué” –no en el “como”.

## Crear perfiles de usuario<sup>3</sup>

En este paso se identifican las categorías de usuarios, determinan que necesitan hacer, evaluar los niveles de destrezas de los usuarios, manejar las expectativas y necesidades de los usuarios y facilitar decisiones de diseño.

Es importante este paso para proveer información desde todos los roles para crear el Concepto de Solución, forma parte de los criterios para evaluar la visión y el alcance, y da pie a desarrollar requerimientos detallados en una fase posterior.

## Concepto de Solución

Comprende una serie de aproximaciones a alto nivel manejados por varios roles. Un Concepto de Solución es una descripción a alto nivel de cómo la solución resolverá el problema. Provee una base para la planificación y desarrollo del trabajo requerido para construir una especificación, así como sus prioridades.

## ***Evaluación Inicial de Riesgos(Microsoft Corporation - MSF Manejo de Riesgos, 2009)***

Un **Riesgo en un Proyecto** es una posibilidad de un resultado negativo, que es asumido para perseguir una oportunidad para ganar en el proyecto. La disciplina de *Administración de Riesgos* de MSF:

- Distingue riesgos de problemas que existen YA (“problemas conocidos”)
- Define un proceso de administración de riesgos para proactivamente identificar,

<sup>2</sup> **Requerimiento:** Una condición que la solución debe enfrentar. (Microsoft Corporation - MSF Visionamiento, 2009)

<sup>3</sup>**Perfiles de usuario:** Descripciones de usuarios eventuales de la solución en términos de su geografía, estructuras organizacionales y de comunicación, funciones de usuario, disponibilidad de recursos y otros descriptores relevantes. (Microsoft Corporation - MSF Visionamiento, 2009)

analizar y direccionar riesgos.

- Incrementa la probabilidad de éxito en un proyecto a través de minimizar la potencialidad de fallo.

El **Proceso de Administración de Riesgos** consiste de los siguientes pasos:

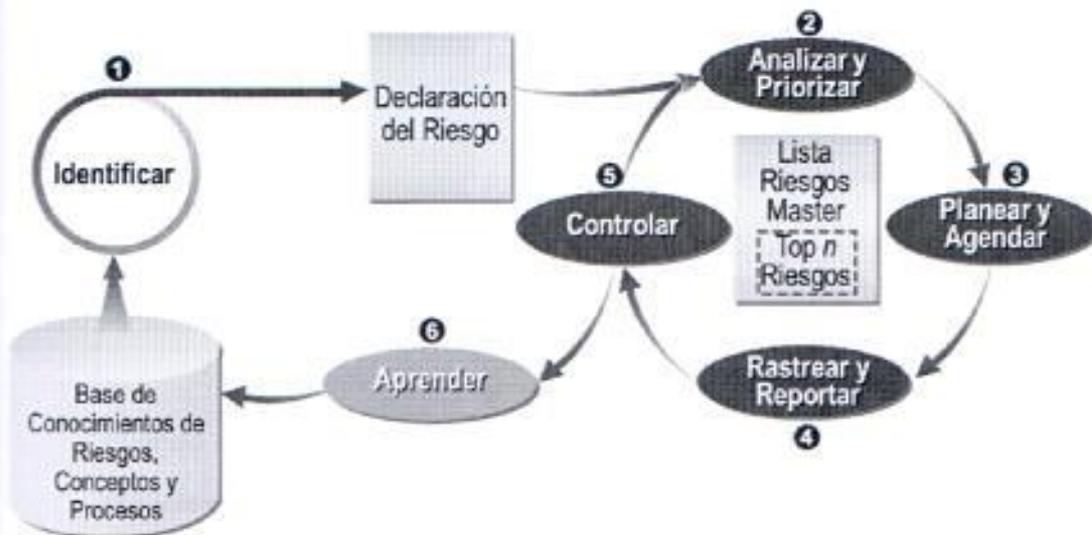


Figura 3.4 Disciplina de Administración del Riesgo (Microsoft Corporation - MSF Manejo de Riesgos, 2009)

1. **Identificar:** Todo el equipo puede expresar posibles riesgos a través de tormentas de ideas.

Para declarar correctamente un riesgo se deben identificar los componentes del mismo: El estado observado de la situación (condición) dentro del proyecto tanto como el estado observado de la situación que podría ocurrir (consecuencia). Las dos declaraciones están enlazadas por la expresión "por lo tanto", o "como resultado", que implica una incerteza (en otras palabras, menos de 100%) pero una relación causal. No se trata de una declaración "Si-Entonces", sino más bien una declaración de incertidumbre: "No sabemos aún acerca de X, por lo tanto..."

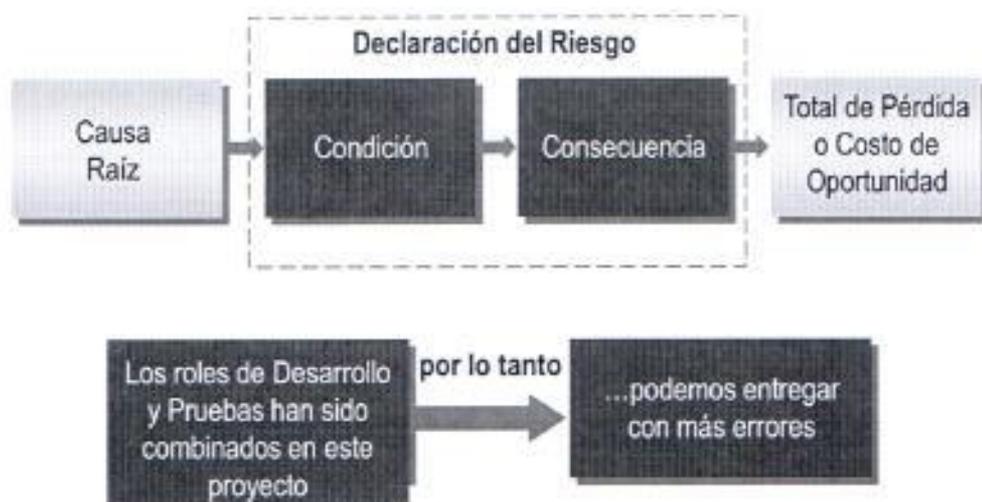


Figura 3.5 Declaración del Riesgo (Microsoft Corporation - MSF Manejo de Riesgos, 2009)

Cuando el equipo formule un riesgo, debe considerar la causa del potencial, no realizado y menos deseable resultado, y el resultado en sí. Entender las causas raíces pueden ayudar a identificar riesgos adicionales y relacionados, y entender los efectos que le corresponden (pérdida total o costo de oportunidad) puede ayudar a evaluar correctamente el impacto de la consecuencia en la organización.

2. **Analizar y Priorizar:** El equipo examina la lista de ítems de riesgo producidos en la etapa anterior y priorizarlos para tomar acción, registrándolos en la Lista Maestra de Riesgos. A partir de esta lista, se puede determinar los "top" por los cuales se comprometerán recursos para planificar y ejecutar una estrategia específica.

La *Probabilidad de un Riesgo* es la medida de la oportunidad de ocurrencia que tenga la condición. Usando un valor numérico mayor que cero (pues igual a cero el riesgo no posee una amenaza) y menor a 100% (puesto que de ser así el riesgo es una certeza; es decir, un problema conocido). Este valor es consensuado no importa la técnica que se use para cuantificarlo.

El *Impacto de un Riesgo* es un estimado de la severidad de los efectos adversos, o la magnitud de una pérdida, o el costo potencial de oportunidad. Puede ser

en términos financieros o con una medida subjetiva, en el primer caso, puede serle familiar a los auspiciadores de negocios. Se recomendaría una escala de 1 a 5 o de 1 a 10.

La *Exposición del Riesgo* mide el tratamiento del riesgo, en su forma más simple de análisis cuantitativo, es calculada multiplicando la probabilidad del riesgo por el impacto.

Prioridad	Condición	Consecuencia	Probabilidad	Impacto	Exposición
1	Desarrolladores van a trabajar con nueva tecnología	Desarrollo tomará más tiempo debido a la necesidad de ellos de aprender	30%	2	.6
2	El equipo de Desarrollo está dividido entre Londres y LA	La comunicación entre ellos será difícil	20%	2	.4

Figura 3.6 Herramienta para análisis cualitativo de Riesgos (Microsoft Corporation - MSF Manejo de Riesgos, 2009)

El paso termina con la ordenación de mayor a menor en función de su exposición. Cobrará mayor importancia para el equipo de trabajo los Top-N (5, 10 o 20 por ejemplo) que resulten de esta ordenación.

3. **Planificar y Calendarizar:** Las actividades de planificación llevadas a cabo por el equipo traducen la lista priorizada de riesgos en planes de acción. La planificación involucra estrategias y acciones para cada una de los riesgos-top, priorizando las acciones y creando un plan de administración del riesgo. La calendarización involucra la integración de las tareas requeridas para implementar los planes de acción de riesgos en un cronograma de proyecto asignándolos a individuos y activamente siguiendo su estado. Calendarizar la administración del riesgo y controlar las actividades no difiere de lo recomendado por MSF para calendarizar actividades de proyecto. Es importante que el equipo entienda que las actividades de control de riesgo son

una parte esperada del proyecto y no un conjunto adicional de responsabilidades a ser realizadas en una base voluntaria. Todas las actividades deberían ser contadas dentro de la elaboración del cronograma del proyecto y proceso de reporte del estado.

Los planes de acción incluyen:

- *Investigación*: ¿Podemos adquirir más información sobre el riesgo antes de tomar acción?
- *Aceptar*: ¿Podemos vivir con las consecuencias si el riesgo fuera a ocurrir? Hay que documentar el por qué el equipo ha elegido aceptarlo.
- *Evitar*: ¿Podemos evitar el riesgo cambiando el alcance del proyecto?
- *Transferir*: ¿Podemos transferir este riesgo a otros proyectos, equipos, organizaciones o individuos?
- *Mitigar*: ¿Puede el equipo hacer algo para reducir la probabilidad o el impacto del riesgo? Esto se transforma en un **Plan de Mitigación**.
- *Aplicar Contingencia*: ¿Puede el impacto reducirse a través de una reacción planeada? Esto se transforma en un **Plan de Contingencia**.

Los Planes de Contingencia se invocan a través de *triggers* o gatillos (disparadores), los cuales son de distintos tipos:

- *Apuntando a tiempo*: Construidos en base a fechas. Ej.: Hito completado tarde.
- *Umbral*: Basado en factores que pueden ser medidos o contados. Ej.: Conteo de errores alcanza 75.
- *Eventos*: Basado en eventos específicos a ocurrir. Ej.: Miembros del equipo son reasignados o renuncian.

El paso termina documentando los planes de acción, resumiendo:

Condición	Consecuencia	Mitigación	Contingencia	Disparador	Responsable
Desarrolladores van a trabajar con nueva tecnología	Desarrollo tomará más tiempo debido a la necesidad de ellos de aprender	Proveer entrenamiento técnico a los desarrolladores	Reversar a una versión anterior	Desarrolladores no han pasado examen relacionado a tecnología	Brenda Diaz
El equipo de Desarrollo está dividido entre Londres y Los Angeles	La comunicación entre ellos será difícil	Mantener una reunión semanal via teleconferencia entre Londres y Los Angeles	Establecer un portal de comunicación basada en internet para publicar información importante del proyecto	La falta de comunicación resulta en un atraso en el cronograma	Erik Ismert

Figura 3.7 Herramienta para la gestión de Riesgos (Microsoft Corporation - MSF Visiónamiento, 2009)

**Rastrear y Reportar:** Un reporte detallado cubre el estado de todos los riesgos que debieron haberse dados al equipo. Este reporte generalmente asigna a uno de los cuatro posibles alternativas de estado para cada uno:

- En resolución
- En progreso de acciones de Riesgos de acuerdo al plan.
- En progreso de acciones de Riesgos en varianza con el plan y por tanto en necesidad de acción correctiva.
- Un cambio en la situación, requiriendo análisis de riesgos y reconsiderar los planes de acción.

4. **Controlar:** En este paso se debería:

- Controlar los planes de acción de los riesgos.
- Corregir los planes debido a variaciones.
- Responder a eventos incidentes.

5. **Aprender:** Se enfoca en tres objetivos clave:

- Proveer aseguramiento de calidad en las actividades en curso de manejo de riesgo de tal manera que el equipo se pueda retroalimentar.
- Capturar lecciones aprendidas, especialmente acerca de la identificación y exitosas estrategias de mitigación, para el beneficio de otros equipos; esto contribuye a la base de conocimiento de riesgos.

## Planificación (Microsoft Corporation - MSF Planificación, 2009)

En la fase de Visionamiento el equipo recopiló requerimientos iniciales y escenarios de uso desde el negocio para crear el concepto de solución, la cual es el punto de partida de la fase de Planificación; misma que es usada para traer claridad y especificidad al concepto de solución.

### Validación de la Tecnología/Creación de Prototipos

En este paso se busca poder realizar predicciones con un grado de confianza, eliminar fallas de diseño y errores de arquitectura, la tecnología utilizada en la solución debe ser validada. Esto se hace para determinar si las características y funciones se ejecuten como fueron declaradas en la documentación. Si no, cualquier ajuste o arreglo debe ser anotado para asistir en la planificación y manejo de riesgos. Si la tecnología ya es conocida, se opta en su lugar por la creación de prototipos de cualquier índole: papel, VB6, etc.

### Diseño de la Solución

El proceso de diseño de MSF construye vistas conceptuales, lógicas y físicas del diseño. El trabajo en cada una de las vistas puede parcialmente traslaparse. Por ejemplo, la lógica puede empezar antes que la vista conceptual esté completa.

El proceso de diseño es evolucionario. Mientras el equipo progresa a través del Diseño Conceptual, Lógico y Físico, adiciona nuevas capas de información al diseño.

#### *Diseño Conceptual*

El propósito de la vista de Diseño Conceptual es capturar y entender las necesidades del negocio y requerimientos del usuario en su contexto apropiado, y entonces crear el diseño conceptual basado en ellos, involucrando auspiciantes del negocio, usuarios, administradores y demás.

Aquí se identifican las “-dades”:

- Seguridad
- Disponibilidad
- Confiabilidad
- Administrabilidad
- Escalabilidad
- Soportabilidad

### *Diseño Lógico*

Durante la fase de diseño lógico, una vista total de la solución está comenzando a tomar forma. Esta vista está más cerca de la declaración final de la solución (y como trabajarán juntos todos los elementos) que la vista durante la vista conceptual, pero aún no es suficiente para comenzar a construir.

El diseño lógico describe la solución en términos amplios de la organización, estructura, sintaxis e interacción de sus partes. Cualquier miembro del equipo debería poder ver el diseño lógico e identificar las partes importantes de la solución, y como estas partes interactúan para resolver el problema.

### *Diseño Físico*

El diseño físico es el proceso de describir componentes, servicios y tecnologías de la solución desde la perspectiva de los requerimientos de desarrollo. El diseño físico descompone los requerimientos para un sistema en sus partes constitutivas para simplificar y estimar el trabajo que se necesita para crearlo.

Durante el trabajo del diseño físico, el equipo crea diseños y arquitectura que esté lista para implementarse. Estos diseños muestran exactamente donde estará ubicado todo, y toma en cuenta las restricciones de las configuraciones específicas.

### Especificaciones Funcionales

Las especificaciones funcionales describen la funcionalidad de la solución a ser construida y alinea los artefactos creados durante la fase de diseño.

Este entregable importante:

- Consolida en un entendimiento común de los requerimientos de negocio y del usuario.
- Sirve como un contrato entre el equipo y el cliente en qué será entregado.
- Articula una manera lógica para descomponer el problema y modularizar la solución.
- Provee un camino y estructura para usarse en la planificación, calendarización y construcción de la solución.

#### Recomendaciones:

- Entender las necesidades de las audiencias.
- Mapea las características del usuario final y procesos de negocio.
- Especifica concisamente y a un nivel consistente de detalle.
- Cuantificar donde sea necesario.
- Incluir restricciones como información de entrada.
- Iterar casi siempre para fomentar retroalimentación.
- Esboce lo más temprano posible, congele el detalle lo más tarde posible.

#### *Partes de la Especificación Funcional*

- **Resumen de la Visión/Alcance:** Tal como fue definida y acordada.
- **Información de Respaldo:** Ubica la solución en un contexto de negocio.
- **Metas de diseño:** Especifica las metas de diseño clave que el desarrollador usa para tomar decisiones.
- **Escenarios de uso:** Describen los problemas de negocio de los usuarios en el contexto de su ambiente.
- **Características y servicios:** Define la funcionalidad que la solución debe entregar.
- **Especificaciones de los componentes:** Define los productos que serán usados para entregar.
- **Dependencias:** Identifica dependencias en otros proyectos o recursos que están detrás del control del equipo de trabajo.

- **Apéndices:** Incluye documentos de arquitectura de la empresa que definen la arquitectura global para la solución. También incluye detalle del diseño conceptual adicional (por ejemplo, encuestas de campo y perfiles de usuario) y detalles de diseño físico. Muchas soluciones de infraestructura requieren diseños a nivel de empresa para definir como los múltiples componentes de servidor interactúan.

### Convertir Aproximaciones en Planes

Un plan es una descripción de *cómo* la solución será completada.

Cada rol afina estas aproximaciones en planes:

Planes Típicos	Rol líder
Plan de Comunicaciones	Product Manager
Plan de Desarrollo	Desarrollador
Plan de Entrenamiento	UserExperience
Plan de Seguridad	Desarrollador Release Manager
Plan de Pruebas	Tester
Plan de Presupuesto	Program Manager
Plan de despliegue, liberación o instalación	Release Manager
Plan de Adquisiciones y Facilidades	Release Manager Program Manager
Plan Piloto	Program Manager

### Preparar Estimaciones a partir de Planes

Usando tareas discretas (visiblemente entregables) sobre intervalos cortos de tiempo, es una manera de llevar rastro de lo que todo el equipo está haciendo y qué progreso están llevando. Es más fácil claramente definir tareas más pequeñas, lo que permite estimaciones más precisas del trabajo involucrado. Cuando se usan tareas de duración más cortas, el margen de error en la estimación de duraciones y compleción de tareas dentro de ellas es menor. Por ejemplo, es difícil para una tarea de 1 semana desfasarse a 6. El resultado de este desglose de tareas se llama *WBS (WorkBreakdownStructure – Estructura de Desglose de Trabajo o EDT)*.

El EDT es un grupo de elementos de proyectos orientados a entregables que organiza y define el alcance de trabajo total del proyecto. Cada nivel descendiente representa

una definición crecientemente detallada del trabajo del proyecto.

MSF recomienda las siguientes técnicas de estimación:

- Use estimación de abajo hacia arriba, lo que significa que aquellos quienes ejecutan las tareas proveen la estimación.
- Use prototipos para ganar experiencia y ayudar en la estimación.
- Ubique tiempo de holgura (*buffer time*) como una tarea discreta, lo que significa que no se adhiere a cada tarea. Esta es una labor del Program Manager, acordada por el equipo.

Primero, los roles del equipo contribuyeron a la creación de varios planes. Ahora, esos varios planes necesitan ser integrados y sincronizados en un Plan Maestro del Proyecto. La sincronización consiste en direccionar dependencias e ítems críticos para proveer una vista completa del proyecto. Este proceso es conducido por el Program Manager.

#### Preparación de los Ambientes de Desarrollo, Pruebas y Producción

Los ambientes de desarrollo y pruebas deben ser representativos del ambiente de producción, en equilibrio con los costos asociados. La separación entre estos ambientes es crítica.

#### Acuerdo entre Equipo y Cliente/Stakeholders al final de la fase

El trabajo de la fase de Planificación, que intenta responder a preguntas básicas acerca de la solución a través de la creación de la arquitectura y diseño de la solución, planes del proyecto, y cronogramas, culmina con la culminación, entrega y acuerdo de los siguientes entregables:

- Las especificaciones funcionales abarcan la arquitectura y diseño de la solución en sus descripciones detalladas de **qué** está planeado a construir.
- El plan maestro del proyecto ensambla en un solo plan, varios planes de proyectos que describen **cómo**, desde un número de perspectivas diferentes, la

solución será construida.

- El cronograma maestro del proyecto que especifica **cuándo** cada aspecto del trabajo será realizado.

Los hitos internos representan fechas de vencimiento para versiones de línea base de los entregables, un recordatorio de naturaleza iterativa de planificación. Ofrecen una forma de moverse hacia adelante sin toda la información necesaria, por medio de la construcción de las expectativas que cada plan revestirá muchas veces. El último hito interno marca una configuración exitosa de los ambientes de desarrollo y pruebas, lo cual es la mayor actividad anterior a empezar el desarrollo.

## Desarrollo(Microsoft Corporation - MSF Desarrollo, 2009)

La meta de esta fase es construir varias características y entregables de la solución. Esto incluye los componentes de código, la infraestructura (software, hardware, redes, y demás facilidades), y entregables de documentación para los usuarios y los operadores. Esto incluye también escribir el lanzamiento de comunicaciones.

### ***Construcción de la Funcionalidad Núcleo***

El desarrollo comienza con la construcción de la funcionalidad núcleo de la solución, y adiciona características incrementalmente. Las características son construidas de acuerdo a la secuencia configurada en los planes y cronogramas que fueron preparadas en la fase de planificación. Construir y probar están integrados a lo largo de la fase de desarrollo.

Cada rol del equipo está trabajando concurrentemente en actividades diferentes y específicas:

- El rol de Desarrollador se enfoca en la creación de la funcionalidad principal (o núcleo) de la solución, y luego añadir características.
- El rol de Tester prepara el conjunto de casos de pruebas, basados parcialmente en los casos de uso desarrollados durante la planificación.
- El rol de Release Manager desarrolla procedimientos para el personal de

operaciones, planes de liberación y encuestas en sitio.

- El rol de UserExperience comienza a probar la usabilidad<sup>4</sup> de la interfaz con el usuario, típicamente usando modelos (a escala o maquetas). La documentación del usuario y entrenamiento también se preparan durante esta fase.

### ***Desarrollo de la Infraestructura***

Una solución exitosa requiere el desarrollo de la infraestructura tanto como el desarrollo de código. El developer de MSF tiene la responsabilidad de ambos.

- Desarrollar la infraestructura refiere a la implementación, pruebas y documentación:
- Configuraciones de hardware y software.
- Scripts y procesos de liberación
- Herramientas de liberación automatizada y checklists de instalación.
- Guía para resolución de problemas (*Troubleshooting*)
- Procedimientos de respaldo y recuperación

### ***Prueba de conceptos***

La prueba de conceptos es una manera de mitigar el riesgo de invertir profundamente en una solución solo para encontrar que sus elementos de infraestructura no trabajarán bien en producción. Una prueba de conceptos de MSF no es la misma que un prototipo o validación de tecnología.

- Los **prototipos** pueden ser contruidos rápidamente en cualquier momento cuando se necesiten (típicamente durante visionamiento o planificación) para explorar una característica o arquitectura propuesta. Ellos están para demostración solamente y no tienen calidad de producción.
- La **validación de tecnología** prueba que los productos y tecnologías de la

---

<sup>4</sup>**Pruebas de usabilidad:** Pruebas que miden la usabilidad de la solución desde la perspectiva del usuario. Los usuarios ordinarios desarrollan tareas en un ambiente de laboratorio mientras son observados por los testers. Ellos prueban cosas tales como cuántos clics son requeridos para cumplir una tarea, o cómo diferentes diseños de páginas web ayudan o entorpecen los intentos de encontrar información.

solución en un ambiente simple y aislado.

- Una **prueba de conceptos** prueba la infraestructura de la solución en un ambiente de producción simulado.

El hecho de completar esta prueba y asegurarse que la infraestructura de la solución coexistirá con las aplicaciones existentes es un hito interno.

### ***Construcciones (Builds)***

El propósito principal de crear construcciones o *builds* es delinear el estado de la solución entera en un momento específico del tiempo, de tal manera que pueda ser probado. Se trata no solamente de componentes de código, sino también de estructuras de directorio, elementos de infraestructura, documentación y algunas veces scripts de instalación automatizada. Se define como un ensamblado periódico de todos los elementos de la solución que son suficientemente completos. Cada *build* debería tener un número y bajo control de versiones.

### ***Desarrollo de la solución usando Liberaciones Internas (InternalReleases)***

Es mejor dividir el conjunto completo de características de la solución en diversos grupos, entonces construirlos y probar cada grupo antes de moverse al siguiente. Cada grupo de características se construye iterativamente, usando un ciclo de liberaciones internas.

El proceso usualmente es el siguiente:

- El número de ciclos de liberaciones internas y el conjunto de características para cada uno deberían ser definidos en el plan de desarrollo en la fase de planificación.
- Las características para una liberación interna están desarrolladas sobre el curso de *builds* "diarios".
- Después que todas las características de una liberación interna han sido construidas, son probadas juntas. Los errores son rastreados y resueltos

(depurados).

- Cada liberación interna debe alcanzar una barra de calidad definida para aceptación, documentada en el plan de pruebas.
- El plan de pruebas también es revisado a través de iteraciones (bajo control de cambios).



Figura 3.8 El Ciclo de una Liberación Interna (Microsoft Corporation - MSF Desarrollo, 2009)

Note que si la liberación interna no es una liberación a producción. Es liberado a otros equipos que están trabajando en proyectos concurrentes que requieren pruebas de compatibilidad con la solución. La liberación interna está también hecha disponible al cliente y a los auspiciantes para evaluación. Esto frecuentemente provee oportunidades para ellos de asegurarse que la solución está en el camino correcto.

### ***Construir entregables de UserExperience***

Este rol es responsable de los entregables siguientes:

- Materiales de referencia del usuario (manuales y archivos de ayuda)
- Elementos gráficos de interfaz de usuario
- Entrenamiento a usuarios finales
- Escenarios de pruebas de usabilidad

## Construir documentación de Operaciones

Esta documentación es conducida por el rol de *Release Manager*. Se debe construir concurrentemente con otros entregables de la solución; desarrollar iterativamente, y sincronizar el desarrollo con las liberaciones internas correspondientes.

Los entregables incluyen: Guías de operaciones, y procedimientos estándar de operación; procedimientos de soporte y mesa de ayuda; base de conocimientos; y, entrenamiento del personal de soporte.

## Pruebas

Las pruebas se hacen para asegurar la calidad de una solución. Debido a que la calidad refiere a muchas metas, que van al grado al cual la solución resuelve un problema de negocio, con relación también al rendimiento y confiabilidad, se deberán hacer varios tipos de pruebas. En el nivel más alto, la meta es validar la solución contra la especificación funcional. Esta es la parte del hilo de trazabilidad que ayuda a asegurar que una solución cumple con las metas originales. Se deben exponer todos los errores que el equipo debe resolver, las fallas de diseño, fallas de comportamientos no contemplados de usuario, y probar todos los elementos de la solución.



Figura 3.9 Las pruebas no se limitan a una única fase (Microsoft Corporation - MSF Desarrollo, 2009)

Es común pensar en las pruebas como una actividad al final de un proyecto. En MSF, las pruebas son una actividad continua que se expande a varias fases.

### ***Pruebas de Cubrimiento***

Las pruebas de cubrimiento son típicas de las fases de desarrollo. Algunos de los tipos de pruebas de cubrimientos son los siguientes:

- Las pruebas unitarias son pruebas pre construidas para descubrir errores tempranamente.
- Las pruebas funcionales verifican características y su funcionalidad.
- Pruebas de verificación de construcción validan la compilación de código después de cada construcción.
- Pruebas de regresión también se ejecutan luego de una construcción (*build*) diario para asegurarse que la calidad ha mejorado en lugar de deteriorarse.

### ***Pruebas de Usabilidad***

Las pruebas de usabilidad intentan asegurar que la solución trabaja en un ambiente para el cual está diseñado y se enfoca en probar la solución como los usuarios y el personal de operaciones lo usarían. Diferentes pruebas verifican la compatibilidad con otros elementos del ambiente, la habilidad de rendir bajo condiciones normales y de estrés, y los errores en la documentación.

Los tipos de pruebas de usabilidad son: De configuración, de compatibilidad, de estrés, de rendimiento, de documentación y archivos de ayuda, y demás de usabilidad propiamente dicha.

### ***La naturaleza de los errores<sup>5</sup> (bugs)***

No todos los errores (bugs) son defectos, pero todos los defectos observados son errores. Todos los errores necesitan ser revisados y resueltos antes de la instalación o

---

<sup>5</sup> **Error (Bug):** Cualquier asunto originado por el uso de la solución.

liberación.



**Error** – Cualquier asunto debido al uso de la solución.

Figura 3.10 Subconjuntos representativos de errores, defectos, etc. (Microsoft Corporation - MSF Desarrollo, 2009)

### ***Manejar errores efectivamente***

El ciclo de rastreo de errores depende de un buen repositorio de errores, generalmente una base de datos. Los reportes de errores entran en la base de datos a medida como los errores se vayan encontrando. Por defecto, un nuevo error es identificado como activo.

Los reportes de errores son priorizados y asignados para resolución al equipo de desarrollo. Esto también se conoce como clasificación. Un error es resuelto o arreglándolo o tomando la decisión de reclasificarlo o no arreglándolo.

Pruebas de Regresión confirman que el arreglo resolvió en error original. Si lo hizo, el error se cierra. Si no, el error se reclasifica como activo.

Clasificar los errores los hace sujetos de acción. Se clasifican por: Severidad (Cuánto el error afectará a la solución) y Prioridad (Cuán importante es arreglar el error). Por ejemplo, una clara directiva basada en los niveles de prioridad podría ser "Arreglar

todos los errores de prioridad 1 antes de direccionar los de prioridad 2".

### ***Resolver Errores***

Se deben indicar cómo los errores serán resueltos a través de etiquetas adjuntas tales como:

- Para arreglar
- Duplicado (luego de haber sido corregido)
- Pospuesto
- No se puede reproducir
- Por falla de diseño
- Decidido no arreglarlo
- Para solicitar característica

Resolver los errores es un paso intermedio antes de cerrarlos.

### **Estabilización (Microsoft Corporation - MSF Estabilización, 2009)**

La meta es mejorar la calidad de la solución para alcanzar los criterios de aceptación para liberación a producción. El enfoque del equipo está en mejorar la calidad de la solución, direccionar asuntos pendientes para prepararla para liberación, hacer que la transición entre las características construidas se enfoque en la calidad, hacer estable la solución y prepararla para la instalación.

Los entregables de esta fase son los siguientes:

- Revisión de Piloto
- Versiones listas para instalación de:
  - Código fuente y ejecutables
  - Scripts y documentación de la instalación
  - Ayuda del usuario final y materiales de entrenamiento
  - Documentación de operaciones
  - Notas de instalación
- Reportes de Pruebas y Errores
- Documentos del Proyecto

Es imposible conocer cuántos errores habrá y cuán severo su impacto será. Por esta razón, es difícil determinar si el proyecto está a tiempo. Las técnicas de Convergencia de Errores y Rebote de Cero Errores ayudan a administrar estas incertidumbres. Un buen rastreo de errores es un prerrequisito para usar estas técnicas.

Hay que recordar que los errores de documentación y de scripts de instalación están rastreadas y administradas juntas con los errores de código.

### ***Adoptar una mentalidad de fecha de entrega fija***

Tratar la fecha de entrega planificada de un proyecto como una variable fija compromete a convertirse en una fecha realista, estimula la creatividad removiendo la opción de moverla, usarla como fija es un factor clave en la toma de decisiones, y motiva al equipo a empujar el proceso hacia adelante.

### ***Convergencia de Errores***

La convergencia de errores es el punto en el cual la tasa de errores resueltos excede la tasa de errores encontrados cada día.

Las tendencias que refleja la convergencia de errores son: ayudar a predecir cuando la solución estará lista para liberación y proveer un indicador objetivo de progreso.

No es posible conocer bien por adelantado cuándo se alcanzará la convergencia de errores. Sin embargo, destinos razonables deberían ser configurados para calendarizar este hito basado en una estimación de abajo hacia arriba. Cuanta más información de tendencias se recolecte, la convergencia de errores puede ser proyectada y usada para reconfigurar la fecha de hito.

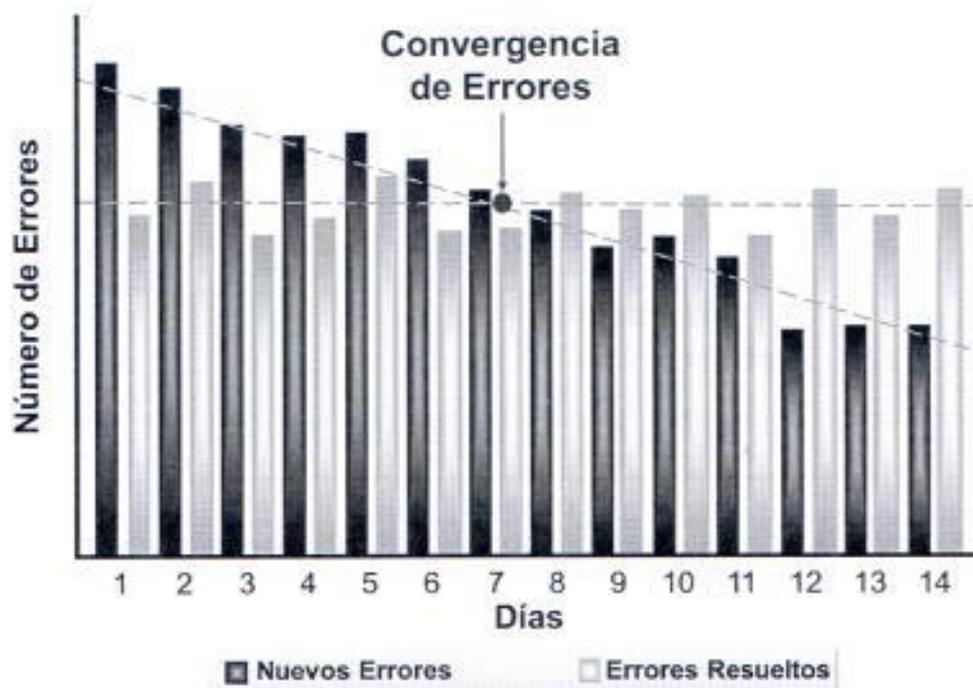


Figura 3.11 Convergencia de Errores (Microsoft Corporation - MSF Estabilización, 2009)

### ***Rebote de Cero Errores***

El rebote de cero errores es un indicador positivo para un proyecto, indicando que el desarrollo ha capturado con el remanente de los errores activos necesitando resolución. Normalmente es un signo de que la calidad de las construcciones va mejorando. Nótese que un número de pequeños rebotes de cero errores seguirán al primero.



◆ Hito interno – Rebote de cero errores

Figura 3.12 Rebote de Cero Errores.

### ***Consideraciones generales***

Como puede ser evidente, una condición necesaria para tener rebote de cero errores es haber llegado previamente a convergencia de errores. Una vez que se ha alcanzado convergencia de errores, se considera a la solución en su **Versión Alfa**, y una vez que se ha alcanzado el primer rebote de cero errores, la solución se considera en su **Versión Beta**. Los siguientes rebotes catalogan a la solución como Beta2, Beta3, etc., a criterio del equipo y el cliente decidir este número antes de continuar con el siguiente paso dentro del proceso.

Si no se ha alcanzado convergencia de errores después de 14 días seguidos de pruebas diarias y listado de errores para rastreo, la solución debe regresarse a la fase de Desarrollo.

### ***Pruebas de Aceptación del Usuario***

Una vez logrado lo anterior, las pruebas de aceptación del usuario aseguran el acuerdo con el usuario, en un ambiente de no-producción (laboratorio), de que la solución

cumplirá con las necesidades de los usuarios.

El proceso de estas pruebas es conducido por el rol de UserExperience, y no debería ser confundido con la aceptación del Cliente y fin del proyecto. El hito interno después de este proceso es haber completado estas pruebas de aceptación.

### ***Designando un Candidato a Liberación (ReleaseCandidate)***

Un Candidato a Liberación (RC) es una construcción que ha sido testada y está lista para liberarse a un piloto.

Contiene todos los elementos (tales como código, documentación, soporte y entrenamiento) requerido para liberación. Se ajusta para liberación de acuerdo a criterios o barras de calidad, y es validado por pruebas de pre-producción. Puede haber múltiples candidatos a liberación (RC1, RC2, etc.) Si el candidato a liberación pasa las pruebas de pre-producción, se mueve a la siguiente etapa –una prueba piloto. Si falla, estos asuntos son reportados y resueltos, resultando en otro candidato a liberación. Conducir Pruebas en Pre-Producción

Durante las pruebas de pre-producción, el candidato a liberación está evaluada por el personal de operaciones, y los procedimientos y procesos necesitados para ejecutar un piloto están en su lugar y testados.

Tener un proceso de reverso listo es una parte importante del manejo del riesgo que forma parte de los pilotos. Cuando un piloto ha sido ejecutado y los asuntos pendientes descubiertos, el piloto necesitará ser reversado si los pendientes son demasiado severos. Las máquinas de usuario o servidores necesitarán ser restaurados para sus estados pre-piloto. Un buen manejo de la configuración es clave para tal efecto. Estos registros de configuración aseguran que cualquier cambio pueda ser reversado para restaurar un estado previo del ambiente.

El personal de operaciones de IT "gobiernan" y son responsables del ambiente de producción, así que la aprobación de operaciones es requerida para acceder al ambiente de producción, incluso para un piloto.

Al final de este hito de prueba de pre-producción completada, un candidato a liberación ha sido aceptado para el piloto y todo está listo para la liberación del piloto.

### ***Pruebas bajo condiciones en vivo***

Un **Piloto** es una prueba de la solución bajo condiciones en vivo; es decir, en ambiente de producción, tales como: un subconjunto de servidores de producción, un subconjunto de usuarios en un grupo, y un periodo de prueba para el ambiente entero de producción, con la habilidad de reversar.

La meta del piloto es tomar el siguiente paso en el proceso de estabilización en curso y reducir los riesgos asociados con la instalación o liberación.

No todo proyecto requiere de un piloto. Esto va a depender del grado de complejidad del mismo, de la innovación implicada en el mismo (si es una solución nueva), o de la relevancia o criticidad que implica el hecho de implementar dicha solución. De no cumplirse estas condiciones, el hecho de efectuar un piloto puede obviarse.

Antes de empezar un piloto, los participantes del mismo clarifican los criterios de éxito para este, los procesos de retroalimentación deben estar en su sitio, los usuarios del piloto deberán ser notificados, y listas de chequeo usadas para verificar procesos y demás aspectos del piloto.

Durante el piloto se minimiza el riesgo de interrupción para el grupo piloto, y se ejecutan chequeos de último minuto para asegurarse que nada haya cambiado desde las pruebas de pre-producción.

Hay que evitar el error de creer que conducir el piloto es demasiado fácil. Tan pronto como sea posible, trate cada caso de uso y situación que se presente en producción, incluyendo tareas de operación. También se recomienda intencionalmente hacer caer el piloto con el fin de probar el procedimiento de reverso, recuperación de desastres, y plan de continuidad de negocio. Se deben probar todos los elementos de la solución, no solo la tecnología y hacer los pilotos representativos, si es necesario conducir más de un piloto, hacerlo.

Después de que hayan sido evaluados los resultados del piloto, hay que seleccionar una de varias estrategias para continuar:

- **Dar un paso adelante:** Preparar otro candidato para liberación y libere al grupo original y otros adicionales de un modo en serie. La liberación de uno o más grupos puede haber sido parte de un plan original o puede haber sido una contingencia disparada por un primer piloto no aceptable.
- **Reverso:** Regresar el grupo piloto a su estado previo.
- **Suspender el piloto:** Poner el piloto en espera o cancelado.
- **Parchar y continuar:** Arreglar la construcción o *build* que representa al piloto y continuar.
- **Proceder a la fase de Liberación:** Avanzar a la instalación del *build* del piloto al ambiente de producción completo.

Los entregables al finalizar esta fase son los siguientes:

- Revisión del piloto
- Versiones listas para instalación de:
  - Código fuente y ejecutables
  - Scripts y documentación de instalación
  - Ayuda de usuario final y materiales de entrenamiento.
  - Documentación de operaciones
  - Notas de la instalación
- Reportes de pruebas y errores
- Documentos del proyecto.

## Instalación/Liberación (Microsoft Corporation - MSF Instalación, 2009)

Los entregables al finalizar esta fase son los siguientes:

- Sistemas de información de soporte y operaciones.
  - Procedimientos y procesos

- Base de conocimientos, reportes y libros de bitácoras.
- Repositorio de todas las versiones de documentos, conjuntos de carga, configuraciones, scripts y código.
- Reporte de cierre del proyecto
  - Documentos del proyecto
  - Encuestas al cliente
  - Pasos siguientes

En algunas organizaciones el equipo del proyecto es responsable de la completa liberación. En otras organizaciones puede haber un grupo independiente del equipo del proyecto que es responsable de la instalación, tal como un grupo de ingeniería central u operaciones.

El plan de instalación o liberación debería ser actualizado como sea requerido a través de las fases de desarrollo y estabilización.

### ***Agrupar componentes para una eficiente instalación***

Se debe identificar componentes núcleo versus los específicos del sitio.

Los componentes **núcleo (core)** son los que están ubicados en una ubicación central o clave que permite la interoperabilidad de la solución completa. Son componente que están habilitando la tecnología de la solución empresarial; tales como, controladores de dominio, ruteadores de mail, servidores de acceso remoto, o servidores de bases de datos.

Los componentes **específicos del sitio** son componentes ubicados en sitios individuales que habilitan a los usuarios a utilizar la solución. Las instalaciones en sitio dependen de esta tecnología.

Las principales estrategias para instalación de la tecnología núcleo y específicas en sitio son (podrían usarse ambas a la vez, dependiendo de la solución):

- **Serial:** Se instalan los componentes núcleo antes que los componentes en sitio, lo cual tiene menos riesgo y se adecua para instalaciones más cortas y en ambientes más pequeños.
- **En Paralelo:** Se instalan a la vez ambos componentes. Recomendado para ambientes grandes o instalaciones que se extenderán sobre un largo periodo.

### ***Instalando tecnología núcleo***

La instalación de estos componentes comprende:

- Configurar sistemas.
- Cargar y configurar software
- Preparar ubicaciones físicas
- Movilizar hardware configurado.

Después que estos componentes sean preparados, la operación en vivo es iniciada, y luego la funcionalidad de los componentes núcleo es validada.

### ***Ejecutar instalaciones en sitio***

La instalación en sitio representa un proceso dentro de un proceso. Esto involucra la ejecución de un plan bien pensado para instalar la solución.

Los sitios pueden ser instalados serialmente por menos equipos o en paralelo por más equipos. Las instalaciones en sitio en paralelo requieren mayor coordinación y provee menos oportunidad de lidiar con un incremento de uso. Sin embargo, una instalación más serializada puede introducir más confusión a los usuarios, especialmente cuando la nueva solución debe coexistir con un sistema *legacy*.

### ***Entrenamiento durante la instalación en sitio***

Es la responsabilidad del equipo de instalación asegurarse que todos los usuarios reciban entrenamiento útil. Durante la fase de planificación, el equipo del proyecto habrá desarrollado un plan de entrenamiento que prescriba el nivel apropiado y tipo de entrenamiento para todos los usuarios. Es necesario evitar la tentación de emplear una estrategia de entrenamiento de "talla única". Si la estrategia de entrenamiento requiere que todos los usuarios experimenten el mismo proceso de entrenamiento, los usuarios novatos serán forzados a aprender demasiado rápido y los usuarios más experimentados se sentirán reprimidos. Hay que considerar alternativas para conducir el mismo proceso de entrenamiento, tales como las sesiones de entrenamiento y un programa de mentoría.

El entrenamiento puede ser hecho antes, durante o después de la instalación.

### ***Estabilización de la instalación***

Estabilizar asegura que el sistema es estable mientras los recursos aún están en sitio. Mueve la administración del sistema a operaciones y personal de soporte.

Después de que cada instalación en sitio sea instalada y completada, la adición de cada nuevo conjunto de usuarios puede impactar en la solución completa. Es importante reconocer que nuevos problemas pueden aparecer durante este tiempo. Estabilizar la solución completa es crítico antes de que el equipo del proyecto sea reasignado.

### ***El periodo de quietud***

El periodo de quietud comienza cuando las instalaciones estén completas. Se necesita un periodo de quietud de entre 15 a 30 días para generar estadísticamente información útil.

El proyecto está ahora moviéndose fuera de MSF y empezando a moverse hacia MOF. El plan de operaciones y los criterios de aceptación establecidos por el equipo de MSF durante la fase de planificación definieron los niveles de servicio para la solución. Estos sirven como una línea de base para el Acuerdo de Nivel de Servicio <sup>6</sup>(SLA), el cual operaciones empezará a monitorear cuando la responsabilidad sea transferida a ellos.

Parte del desenganche del proyecto incluyen las operaciones de transición y funciones de soporte al personal permanente (operaciones). En muchos casos, puede ser necesario diseñar nuevos sistemas de soporte; dado el alcance en el último de los casos, puede ser prudente considerar aquello como un proyecto separado.

### ***Encuestar satisfacción al cliente***

Esto provee una oportunidad para el equipo de recibir retroalimentación al cliente. Es un componente necesario del Reporte de Cierre y representa validación del proyecto entero. Esto provee una perspectiva del cliente en la calidad de la solución, puede proveer entrada para futuras mejoras de procesos, y puede tomar la forma de entrevistas y/o cuestionarios.

Las personas encuestadas deberían incluir auspiciantes del proyecto y demás interesados clave como mínimo.

### ***El Reporte de Cierre***

Es el entregable físico de la instalación. Esto incluye las versiones finales de todos los mayores entregables: el documento de visión/alcance, las especificaciones funcionales, y así sucesivamente.

Típicamente, el reporte de cierre del proyecto también incluye un resumen de la información solicitada a los clientes y usuarios y un resumen de los siguientes pasos

---

<sup>6</sup>**Acuerdo de Nivel de Servicio (ServiceLevelAgreement – SLA):** Un acuerdo entre IT y la comunidad de usuarios que define las responsabilidades de todos los participantes y que vincula la administración de IT para proveer un servicio particular de una calidad y cantidad convenidas. Restringe las demandas de los usuarios que pudieran ubicarse contra esos límites definidos por el acuerdo.

conocidos, esencialmente respondiendo la pregunta "¿Hacia dónde vamos a partir de aquí?".

### ***Conducción de Revisiones del Proyecto***

Típicamente el equipo conduce dos tipos de revisiones:

Del equipo: El equipo principal refleja su revisión en el proceso y los resultados de la implementación. Además documenta su reflejo a manera de ítems de acción en el próximo plan de proyecto.

Del cliente: Es conducida con los auspiciantes e interesados clave, identifica asuntos pendientes, niveles de satisfacción al cliente y calidad, determina siguientes pasos.

### ***Obtener la firma del cliente***

Una vez que se cierra el proyecto, el Product Manager obtiene la firma final de cierre por parte del cliente, señalando la aprobación del cliente de la solución y permiso para desengancharse del proyecto.

Los entregables en esta fase, entonces, son los siguientes:

- Sistemas de información de operaciones y soporte
  - Procedimientos y procesos
  - Base de conocimientos, reportes y bitácoras.
- Repositorio de todas las versiones de documentación, carga de datos, configuraciones y código.
- Reporte de cierre del proyecto
  - Encuesta de satisfacción del cliente
  - Documentos del proyecto
  - Pasos siguientes

### ***Factores de éxito para la fase de liberación***

- La solución está ejecutándose establemente y en un ambiente de producción
- El cliente ha firmado y cerrado el proyecto
- El equipo se ha desenganchado del proyecto
- El conocimiento del proyecto ha sido capturado.

## **Capítulo 4: Estabilización del Proceso de Implementación**

### **Evaluación de la Capacitación en MSF**

Cómo se había indicado en el capítulo anterior, se efectuó una capacitación al personal existente de Sistemas sobre todos los aspectos que cubre el marco de trabajo de MSF tal como consta en el programa descrito anteriormente. Esta capacitación tuvo la acogida esperada por los estudiantes, los cuales vieron como coherente las buenas prácticas planteadas por esta disciplina y como esperanzadora la posibilidad de aplicarlas dentro de la empresa. Por ende, su aceptación de acoger como suyas estas prácticas dentro de la jurisdicción de sus tareas fue inmediata.

## Puntaje (sobre 10)

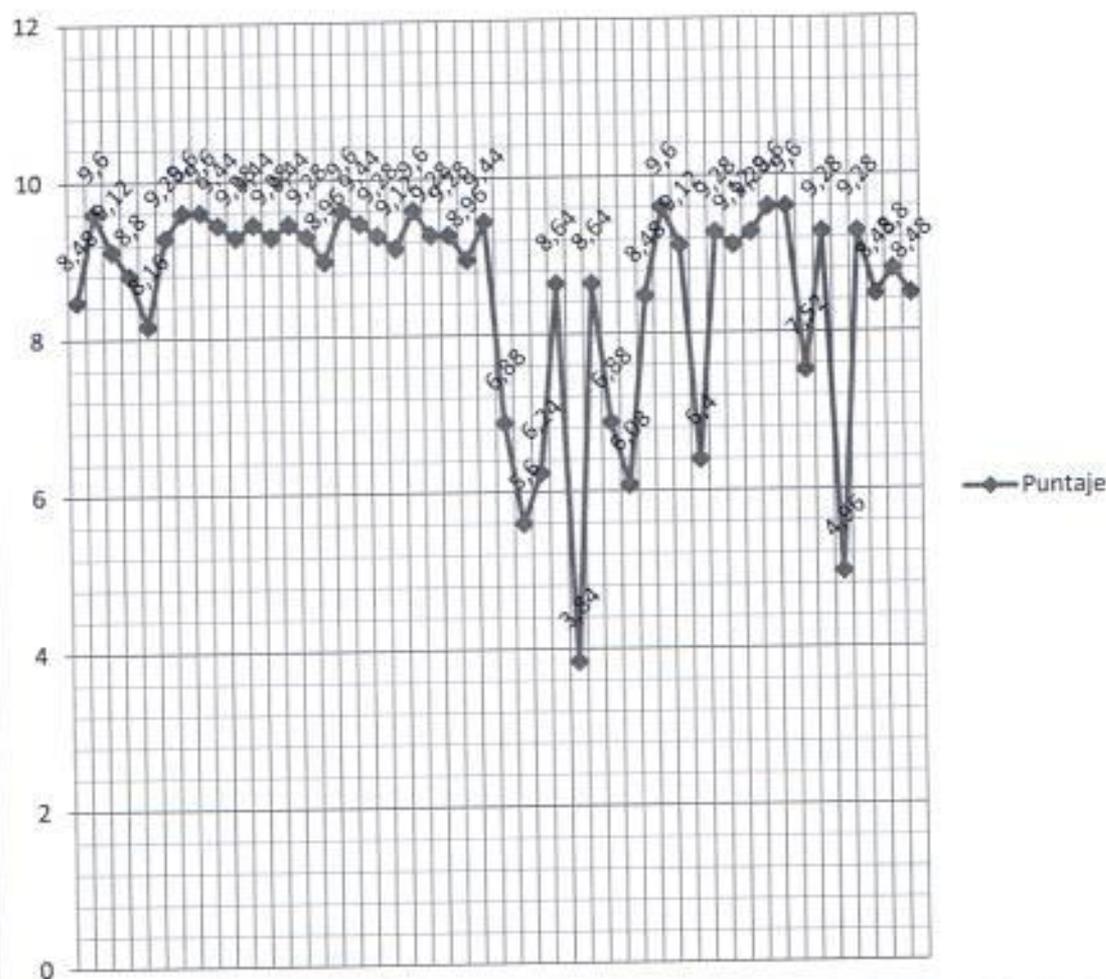


Figura 4.1 Evaluación de Capacitación Formal en MSF

Se realizó una evaluación al personal inscrito en el curso buscando evaluar los conocimientos recibidos, y los resultados obtenidos son resumidos en la figura anterior en un puntaje sobre 10. Esto refleja el entendimiento suficiente del personal de los conceptos impartidos casi uniforme en todos. Si se aprecian bajos en la gráfica básicamente radica en el siguiente hecho: El conjunto de estudiantes, todos de Sistemas, está compuesto por personas de variadas habilidades y destrezas, con antigüedades en la empresa muy diferentes, teniendo como factor común que los más antiguos evidentemente tienen menor apertura de recepción de nuevas formas de trabajar, debido a su experiencia ganada a través de los años de realizar su trabajo en base a lo que su sentido común le ha dictado.

La retroalimentación recibida de parte de los alumnos del curso; es decir, sus opiniones se resumen en los siguientes puntos:

- Son muy buenas las prácticas impartidas en el curso, muy positivas, pero no son aplicables a la realidad de la empresa.
- Puede que funcione implementar lo enseñado en el curso, pero acompañado de eso debería haber una reevaluación de cargos, responsabilidades y salarios.
- Fue muy positiva la capacitación recibida siempre y cuando los tomadores de decisiones, incluidos los usuarios y clientes tengan también en claro desde y hasta donde llegan sus responsabilidades y exista un grado de compromiso de su parte.

## Resultados e Influencia de MSF en el Proyecto Piloto

El proyecto piloto arrancó como fue previsto y requerido: un proyecto institucional priorizado debidamente, cuyo equipo fue conformado en función de sus destrezas para el contexto, y reunidos hasta donde fue posible en una misma área común junto con el tutor. El tutor por su parte se reunió presencial y virtualmente con el equipo de trabajo impartiendo el acompañamiento debido, y brindando las recomendaciones del caso.

Lamentablemente si bien es cierto empezó la implantación de la metodología como un proyecto en sí mismo, y el piloto fue auspiciado debidamente en un principio por los interesados y patrocinadores del mismo, sobre la marcha perdió el interés y el entusiasmo tanto de ellos como del equipo de trabajo en llevar a cabo dicho proyecto siguiendo los lineamientos que indica MSF. El proyecto, como era de esperarse avanzó en cada paso que dicta el modelo de procesos de una manera lenta, puesto que no es fácil desaprender un modo de trabajar para adoptar otro. Pero a este proyecto fueron afectados otros factores que se evidenciaron a lo largo del desarrollo del proyecto. Los cuales entre otros fueron los siguientes:

- Las reuniones del equipo de trabajo junto con el tutor (coach) que se pactaron con cierta periodicidad empezaron a no cumplirse. El día a día de cada miembro de equipo, y la falta de compromiso y seguridad en el proceso que

estaban comenzando a construir, hizo que muchos miembros del equipo de trabajo comenzaran a faltar a las reuniones e incumplir con sus responsabilidades en cada etapa.

- El proyecto comenzó a perder auspicio y compromiso de parte de los interesados y patrocinadores, puesto que por una parte no veían resultados tangibles a corto plazo, retroalimentado por la falta de colaboración de los mismos clientes y auspiciantes a proporcionar la información adecuada; y muchos administradores de los recursos involucrados veían hasta cierto punto como inútil o distractor de sus tareas comunes, asignar las horas definidas al trabajo en este proyecto piloto.
- Las presiones del cliente y los auspiciantes, debido a reacciones de mercado y a factores de competitividad y diferenciación, comenzaron a afectar al proyecto en sí mismo y a ponerlo en riesgo, puesto que la velocidad a la que iba avanzando no cubría sus expectativas.

Sin embargo, y a pesar de estas malas noticias, el proyecto piloto terminó exitosamente, dadas las expectativas previstas por el tutor por ser la primera experiencia en un proyecto utilizando MSF como forma de trabajo, el llenado de plantillas y lo que ello implica, se completó de elaborar, revisar, aprobar y firmar los entregables de la fase de Visionamiento de este proyecto, con lo cual fue concluida exitosamente. El equipo de trabajo entendió perfectamente lo que había que hacer.

Sin embargo, dados los puntos anteriores, se decidió por detener el proyecto al finalizar esta fase y comenzar a desarrollar el producto del modo que se lo ha venido realizando hasta el momento, y a la velocidad que esto implicaba. Puede consultarse el Capítulo 1 de Antecedentes para mayores detalles sobre esta operativa.

## **Barreras Administrativas y Organizacionales de la Implementación de la Metodología**

Las siguientes barreras a nivel de la organización van a ser enlistadas y ordenadas desde las operativas hasta las estratégicas y ejecutivas.

- Los programadores, los cuales estaban acostumbrados a realizar toda tarea de análisis de requerimientos, preparación de ambientes y programación propiamente dicha, ahora pretenden despojarse de aquellas tareas totalmente, descuidando una de sus áreas funcionales: la consultoría tecnológica, que es vital para dar una idea clara del alcance de la solución durante las 2 primeras fases del proceso.
- No existe el personal apropiado y especializado que cuente con las destrezas requeridas para cubrir los roles promulgados por el marco de trabajo. Se cuentan con programadores con destrezas de análisis, pero al ocupar estos nuevos roles dejan de programar y en ellos se encuentra todo el conocimiento técnico de las aplicaciones puesto que ellos fueron quienes los programaron. No se cuenta actualmente con la cantidad de recursos requerida para tal efecto.
- Deben existir administradores de recursos que gestionen, supervisen, monitoreen la disponibilidad y asignación de cada uno de los programadores y analistas a las tareas de los proyectos en curso. Eso implica también un cambio de responsabilidades de los actuales supervisores de estos recursos, puesto que ellos también realizaban tareas de análisis de requerimientos y proyectos, puesto que sus perfiles rinden para tal efecto. Las buenas prácticas promulgan el hecho de que ellos solo se dediquen a las tareas antes mencionadas. Actualmente no se da esta separación de asignaciones, y cuesta mucho desprenderse de esas tareas, si no se adquieren nuevos recursos que suplan estas necesidades.
- El Gobierno de IT debe ser el primero en promulgar de una manera categórica, y explícita contra toda adversidad la defensa y auspicio constante de dicha implementación, de lo contrario, no habrá fuerza que permita que se sostenga al interno con los colaboradores a su cargo, ni tampoco ante las demás áreas quienes no se sentirán comprometidas, ni peor aún identificadas con el propósito principal y motor de la misma, como una herramienta para la estandarización, mejora de la calidad, y sentido de cumplimiento de los compromisos adquiridos.
- Las demás áreas que se convierten en clientes internos y usuarios de la

metodología de trabajo, deben más allá de estar de acuerdo con las políticas y lineamientos del Gobierno de IT con respecto al manejo y gestión de Proyectos de Tecnología, deben estar convencidos de los beneficios y ventajas que para ellos será el hecho de tener una clara rendición de cuentas, fiel entrega a fecha fija y obtención de calidad en las soluciones entregadas. Pero también deben estar conscientes de que tales beneficios conllevan también un esfuerzo así como una responsabilidad grande de ellos en cuanto a su participación activa dentro del desarrollo de los mismos, incorporando a personal de sus áreas como parte del equipo de trabajo, así como el hecho de brindar toda la información necesaria al proyecto para el levantamiento de información, misma que su rol demanda que sea llenada y presentada por todo el equipo, incluyendo ellos mismos, en el rol del "Product Manager". Actualmente los clientes y usuarios simplemente entregan una iniciativa al área de Sistemas, y esperan que sean ellos quienes evolucionen, expandan, diluciden dudas, asuman en muchas ocasiones, y levanten ellos mismos exclusivamente la información requerida. En muchas ocasiones deben jugar el papel de usuarios de la aplicación creada.

- El Gobierno corporativo y Comité Ejecutivo debe ser un vigilante exhaustivo del progreso y avance de la implementación de esta metodología, convencido de los réditos de la misma, reflejados en las numerosas empresas en todo el mundo beneficiadas con el uso de esta herramienta. Deben constantemente realizar "benchmarking" con dichas empresas para rescatar elementos positivos encontrados en los resultados de aplicar estas buenas prácticas en sus organizaciones, a fin de incorporarlas como suyas al interno. Si la empresa a nivel ejecutivo, no auspicia de manera constante y pide reportes de su avance al Gobierno de IT; y se mantiene con ese ímpetu durante el tiempo de madurez de esta implantación, el ritmo se pierde y es fácil caer en las viejas prácticas, y hacer que las excepciones en cuanto al no uso del proceso en construcción se conviertan en la regla y se retroceda en este proceso. Cuando se inició este cambio radical, tuvo efectivamente el impulso necesario para arrancar, pero con el paso del tiempo, las presiones del día a día, así como una falta del sentido de

inversión necesario para justificar algún costo incurrido en el mismo, fueron ahogando esta labor, haciéndola caer de lleno o estancarla hasta donde se pudo llegar con ella. Finalmente, el proceso se queda en espera de que sea retomado nuevamente.

- Es necesario que como vigilante, monitor y evaluador de la implantación de esta metodología, participe una institución auditora certificada y de renombre que conozca completamente los riesgos, barreras posibles de implementación de la misma, a fin de que mantenga la “velocidad de crucero” en cuanto al avance de dicha implementación, y periódicamente informe al Comité Ejecutivo métricas del grado de madurez alcanzado y eleve las alertas apropiadas y oportunidad cuando no se avance lo esperado en ese periodo de tiempo. Una consultoría adicional que acompañe durante el proceso sería recomendable. Actualmente no existe ninguno de estos dos ítems.

## Capítulo 5: Procedimientos y Políticas. Metodología MSF adaptada

### Versión Final de los Procedimientos a Implementarse

Teniendo en cuenta lo expuesto y revisado en el Capítulo 3, sobre las mejores prácticas, Modelo de Procesos de MSF y algunos pocos criterios basados en el entorno organizacional de la empresa, se muestra gráficamente expresado en notación ORM (Visio) y se describe el flujo bajado al mayor nivel de detalle posible, permitido por la flexibilidad y facilidad de usar cualquier criterio basado en sentido común o juicio de expertos. Como todo proceso cada paso tiene un inicio y un fin.

Para facilidad de lectura e identificación, se numeraron los procesos en las cinco fases de MSF (1 – Visionamiento; 2 – Planificación; 3 – Desarrollo; 4 – Estabilización; 5 – Instalación). Dentro de cada uno se numeran los pasos a un segundo nivel, y así sucesivamente mientras más se entra a detalles de cada uno de los pasos de este flujo. En los procesos de mayor detalle (al más bajo nivel posible), se definen los cuatro roles

del modelo RACI que promulga COBIT: el Responsable, el "Accountable" (el último responsable, o quien rinde cuentas), el Consultado y el Informado. Esto a efectos de determinar responsabilidades sobre cada una de las tareas y asignaciones dentro de cada fase. Se deberá notar que en toda ocasión se define una sola persona o rol de MSF para cada elemento del cuadro RACI, salvo en uno: el Comité de Aprobación de Cambios (CAB), el cual representa una entidad autónoma dentro de la empresa que se encarga de aprobar o denegar todo control de cambios de las soluciones y servicios instalados en la empresa, además de todo control de cambios a nivel de la triple restricción a lo largo de la ejecución del proyecto: de tiempo, de alcance o de recursos. Típicamente esta entidad está conformada por representantes de todas las áreas (o por lo menos la gran mayoría o las más representativas) de la empresa, junto con el supervisor general, gerente o vicepresidente del área de IT de la empresa. Estos elementos diversos hacen que se confronten entre sí en pro de un consenso común y las decisiones que tomen satisfagan a los intereses institucionales de la empresa. Por tanto se requiere un nivel alto de los integrantes de este comité. Típicamente puede ser conformado por el Comité de Tecnología de Información, o una Oficina de Administración de Proyectos (PMO).

## Visionamiento

Los dos primeros pasos en esta fase (1.1 y 1.2, Figura 5.1 Fase de Visionamiento) apuntan a tipificar y categorizar el Proyecto, a efectos que el equipo de trabajo y demás interesados estén conscientes desde un comienzo del contexto en el que se comienza a trabajar. Luego se construye el equipo de trabajo quien será el encargado de este proyecto. Con estos elementos, es posible ya crear la estructura del proyecto y poder llenar el primer artefacto o entregable de esta fase: el Documento de Estructura del Proyecto (véase Estructura del Proyecto (Microsoft Corporation - MSF Visionamiento, 2009)); el desarrollo de este paso se verá más adelante.

Como siguiente paso se describirá la visión y el alcance del proyecto según promulga la disciplina y generará como entregable el documento de Visión y Alcance (véase Visión y Alcance (Microsoft Corporation - MSF Visionamiento, 2009)); cuya profundización de

este paso se detalla más adelante.

Como un paso adicional que completa esta aproximación a la planificación se elabora la Propuesta de Características para este proyecto.

Sin grado de dependencia es posible crear en paralelo, y con esto no quiere decir necesariamente que deba hacerse de esta manera (dependiendo de la disponibilidad de tiempo de los recursos de realizar varias tareas a la vez), los documentos de Evaluación del Estado Actual de la Infraestructura (véase Evaluación del Estado Actual de la Infraestructura), e iniciar con la Gestión de Riesgos, elaborando por primera vez la Herramienta de Gestión de Riesgos tal como fue descrito en el Capítulo 3 (véase Evaluación Inicial de Riesgos (Microsoft Corporation - MSF Manejo de Riesgos, 2009)).

Opcionalmente, el líder de esta fase (el Product Manager en Visionamiento), puede emitir un Reporte como Líder de Equipo, y/o cada miembro emitir a los interesados el suyo propio.

Finalmente, cuando los documentos mandatorios han sido llenados, son revisados ya aprobados por el equipo de trabajo, para luego ser revisados y aprobados por el Cliente y opcionalmente por algún otro interesado según sea el caso (Paso 1.17, Figura 5.1 Fase de Visionamiento). El entregable de esta fase es el conjunto de los documentos: Estructura del Proyecto y Visión/Alcance (junto con la Propuesta de Características) aprobados por el equipo y el cliente. Una vez que ocurra eso, automáticamente pasamos a la siguiente fase: Planificación.

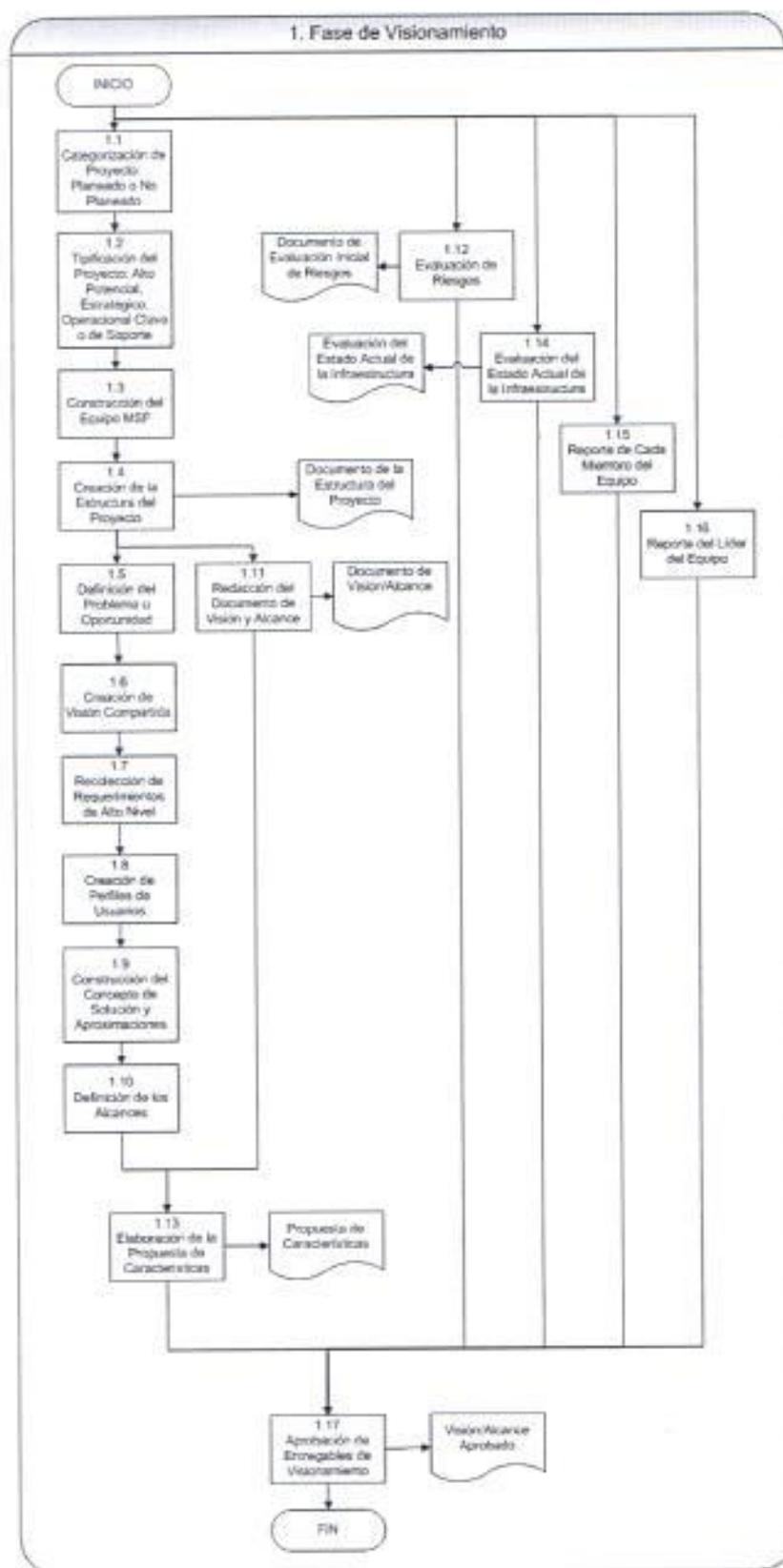


Figura 5.1 Fase de Visionamiento

### ***Construcción del Equipo MSF***

Lo primero que se debe hacer al tener un proyecto a cargo es identificar quién es el Cliente, y demás interesados y auspiciantes, y se busca su compromiso de participación a lo largo de todo de todo el proyecto, el medio de hacerlo puede ser formal o informal, puesto que una de las formas de asegurar el compromiso del Cliente es poner su nombre, área y departamento en cada documento emitido a lo largo de este proyecto (1.3.1 y 1.3.2, Figura 5.2 Construcción del Equipo MSF). Enseguida se debe identificar aquella persona que hable por el Cliente ante el equipo y viceversa: el Product Manager. Este es el idóneo para determinar la aproximación de las características generales de lo que se busca en el proyecto y su magnitud en términos de tamaño o complejidad. Luego es posible Seleccionar los que participarán en los demás roles. Si este comportamiento se debe replicar por cada una de las características del proyecto, medida con la magnitud determinada anteriormente, tenemos el equipo conformado.

El responsable principal, así como el que rinde cuentas de la ejecución de este paso es el Comité de Aprobación de Cambios (CAB). En este caso debe gestionar con la Gerencia de Desarrollo, o con el Administrador de Recursos que serán los candidatos de los demás roles, que pueden ser de IT o de otras áreas, tales como sus supervisores o jefes inmediatos. El informado de este proceso es el cliente.

La primera tarea es identificar estos posibles candidatos. Se promulga o recomienda el hecho de contar con una Base de Datos de Habilidades y Destrezas; dentro de la cual, a criterio de la empresa básicamente contiene objetivamente todas las habilidades técnicas (*hardskills*) o aptitudes que están disponibles para realizar un determinado tipo de tarea, y habilidades o destrezas humanas (*softskills*) que incluyen por ejemplo: facilidad de palabras, habilidad para la negociación, e incluso buena ortografía (1.3.X.1, Figura 5.2 Construcción del Equipo MSF)

Una vez identificado el o los recursos candidatos, se procede a adquirirlos con el administrador respectivo, mediante un acuerdo verbal o escrito, mientras se asegure que el recurso de capital humano va a dedicar parte o todo su día laboral al desarrollo

del proyecto durante el periodo que dure el mismo (1.3.X.2, Figura 5.2 Construcción del Equipo MSF)

Si se logra obtener dicha obtención, se registra el recurso adquirido, que al final quedará inscrito dentro del Documento de Estructura del Proyecto (1.3.X.3, Figura 5.2 Construcción del Equipo MSF). Si el rol es tan complejo que se necesitan más personas para que lo asuman (1.3.X.4, Figura 5.2 Construcción del Equipo MSF), se procede de igual manera para su obtención (1.3.X.5, Figura 5.2 Construcción del Equipo MSF), cada uno de los cuales cumplirán con cada una de las áreas funcionales de cada rol (véase Modelo de Equipos de MSF).

El administrador de este recurso (por ejemplo el Gerente de Desarrollo), debe considerar los siguientes elementos para evaluar la disponibilidad del recurso (1.3.X.2, Figura 5.2 Construcción del Equipo MSF): Proyectos en curso y pendientes, además de tareas en curso y pendientes, así como vacaciones planificadas, permisos. (como maternidad) y demás. Cuando esto se dé se procede a la negociación entre el que solicita el recurso y su administrador (1.3.X.2.2, Figura 5.2 Construcción del Equipo MSF). Si la negociación no es exitosa por algún motivo, en responsabilidad compartida de los dos se busca evaluar la contratación de recursos externos para este proyecto (1.3.X.2.3, Figura 5.2 Construcción del Equipo MSF). El paso termina con la determinación si el recurso está o no disponible para este proyecto.

Si el recurso no está disponible, se tiene la alternativa de unir roles, al incorporar a una persona en dos roles, siempre y cuando se lo permitan las recomendaciones de MSF acerca de la unión de roles (véase Escalamiento hacia abajo – Combinación de roles para equipos más pequeños). Un caso especial ocurre con el rol de Program Manager, nuevamente si el proyecto es de gran magnitud, es posible dividirlo en los roles de Arquitecto de Solución y Project Manager.

Cuando a pesar de intentar todas las alternativas anteriores, el recurso no fue adquirido, se eleva esta novedad como un riesgo y se gestiona como tal (véase Evaluación Inicial de Riesgos (Microsoft Corporation - MSF Manejo de Riesgos, 2009)).

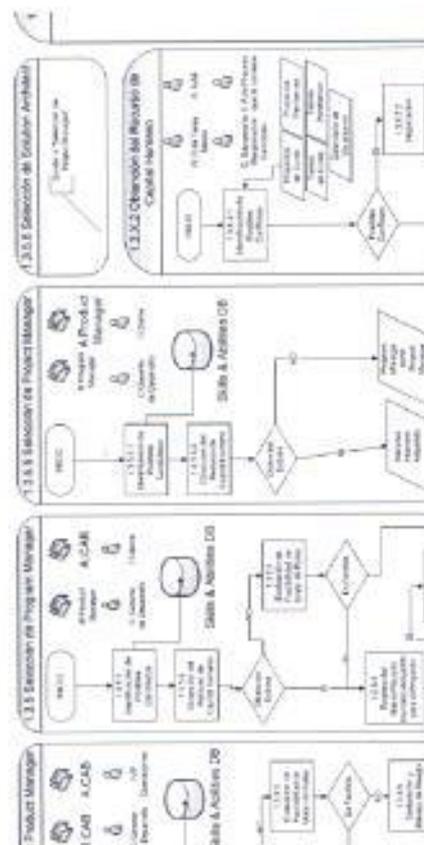


Figura 5.2 Construcción del Equipo MSF

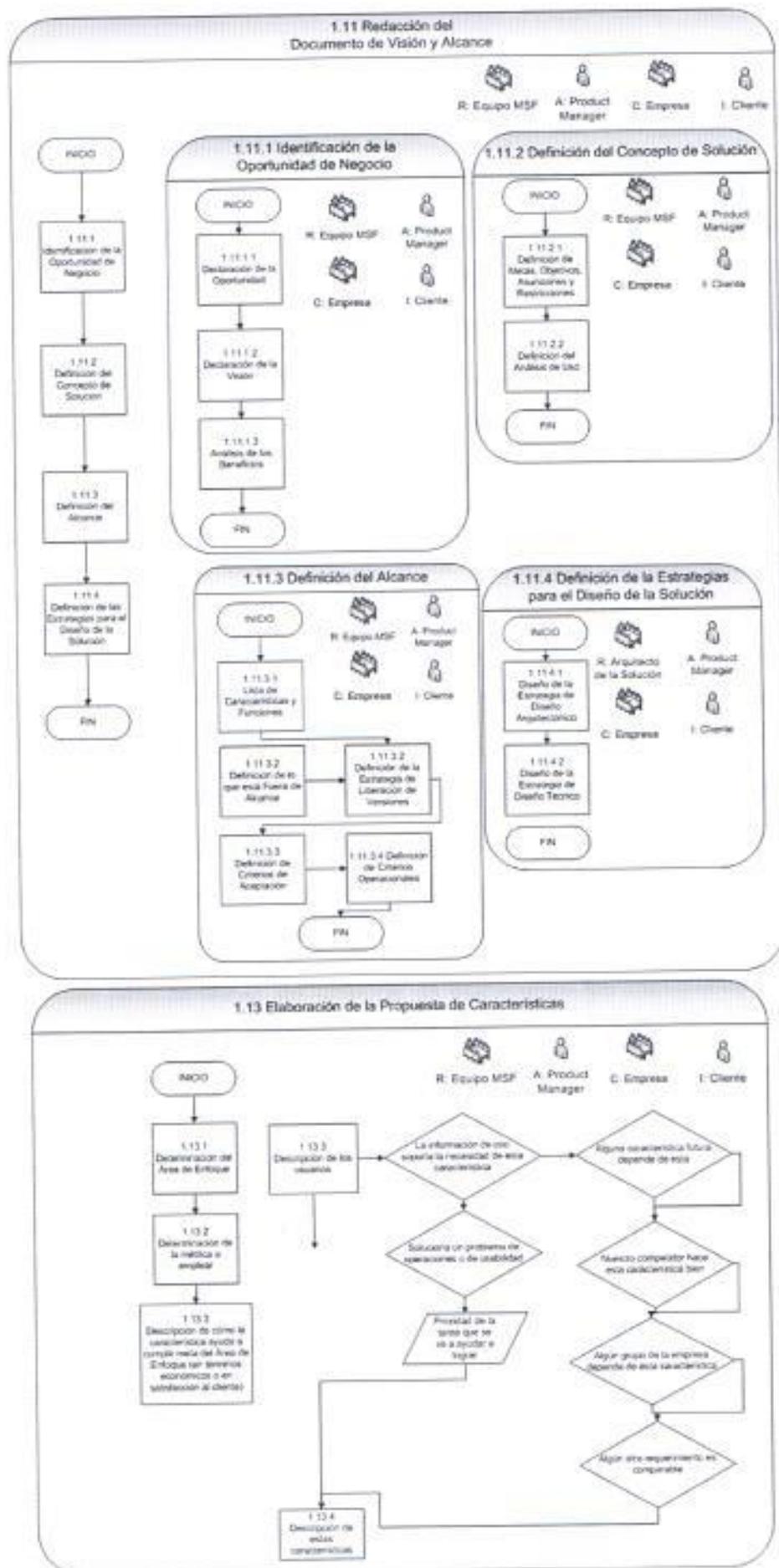
### ***Redacción del Documento de Visión y Alcance***

Cada uno de los capítulos y secciones (1.11, Figura 5.3 Redacción del Documento de Visión y Alcance) que comprende un documento de Visión y Alcance se detalla a continuación, siguiendo lo revisado en el Capítulo 3 (véase Visión y Alcance (Microsoft Corporation - MSF Visionamiento, 2009)). En la redacción de este como en cualquier documento o entregable de fase, el responsable es todo el equipo de trabajo, pero el que rinde cuentas ante el cliente es el líder de esta Fase: el Product Manager. Para tal efecto, la fuente primordial de consulta puede estar en cualquier colaborador de la empresa que dentro del contexto participe en alguna instancia en la operativa en cuestión. El cliente también debe estar informado de estos avances, mediante reportes de Miembros de Equipo y/o de Líder del Proyecto.

En el caso de la sección de Definición del Concepto de Solución (1.11.2, Figura 5.3 Redacción del Documento de Visión y Alcance) es una tarea primordialmente activa del

Arquitecto de la Solución como parte del rol de Program Manager, pero nuevamente como forma parte del Equipo MSF también es responsable de esta tarea.

La **Propuesta de Características** es un anexo al Documento de Visión y Alcance que define el conjunto de funciones y características que deberá tener la solución, así como se definen los criterios de aceptación, operacionales y la estrategia de versiones.



## Planificación

Recordemos que los tres entregables de la fase de Planificación son: El Documento de Especificaciones Funcionales, el Plan Maestro del Proyecto y el Cronograma Maestro del Proyecto. Se empieza esta fase verificando estableciendo una validación; es decir, una pequeña (o varias) prueba(s) de la tecnología a utilizarse en la solución, si esta es nueva (2.1, Figura 5.4 Fase de Planificación), o (inclusivo), creando un prototipo (2.2, Figura 5.4 Fase de Planificación) que dará una idea al usuario de cómo quedaría la solución una vez terminada. Nuevamente, las herramientas pueden ser muy variadas: desde papel hasta Visual Basic, o cualquier otra herramienta de prototipado (véase Validación de la Tecnología/Creación de Prototipos).

Luego se comienzan a elaborar en este orden los siguientes entregables muy necesarios en esta fase para continuar con lo demás (2.2, 2.4 y 2.7, Figura 5.4 Fase de Planificación): el Diseño Conceptual, el Diseño Lógico y el Diseño Físico (véase Diseño de la Solución). Después de realizado el diseño conceptual, se está apto para identificar y elaborar los Escenarios de Uso (2.5, Figura 5.4 Fase de Planificación), como una evolución de los perfiles de usuario en la fase de Visionamiento (véase Crear perfiles de usuario), así como los Requerimientos de Negocio, del Sistema y de Operación (2.6, 2.8 y 2.10, Figura 5.4 Fase de Planificación). Estos últimos, junto con los otros tres documentos, habilitan para terminar de elaborar el documento de Especificaciones Funcionales (2.9, véase Especificaciones Funcionales).

Para crear el Plan Maestro del Proyecto, se evolucionan las aproximaciones generadas en Visionamiento y se convierten en planes de acción (2.11, véase Convertir Aproximaciones en Planes), para luego unirlos en el Plan Maestro (2.12, Figura 5.4 Fase de Planificación), que no es más que resúmenes o referencias a los documentos individuales de los planes.

A raíz de estos planes, se desprenden paquetes de trabajo (2.13, Figura 5.4 Fase de Planificación) que servirán para construir el EDT (WBS) respectivo (véase Preparar Estimaciones a partir de Planes). Finalmente se unen todos los cronogramas con las

dependencias respectivas, para convertirse en el Cronograma Maestro del Proyecto.

Como parte de dichas tareas deberán constar aquellas que permitan preparar los ambientes de desarrollo, pruebas y producción (véase Preparación de los Ambientes de Desarrollo, Pruebas y Producción).

Estos tres entregables son revisados y aprobados por el equipo de trabajo y sometidos a revisión y aprobación del Cliente y demás interesados (véase Acuerdo entre Equipo y Cliente/Stakeholders al final de la fase). Una vez realizado, automáticamente se pasa a la siguiente fase.

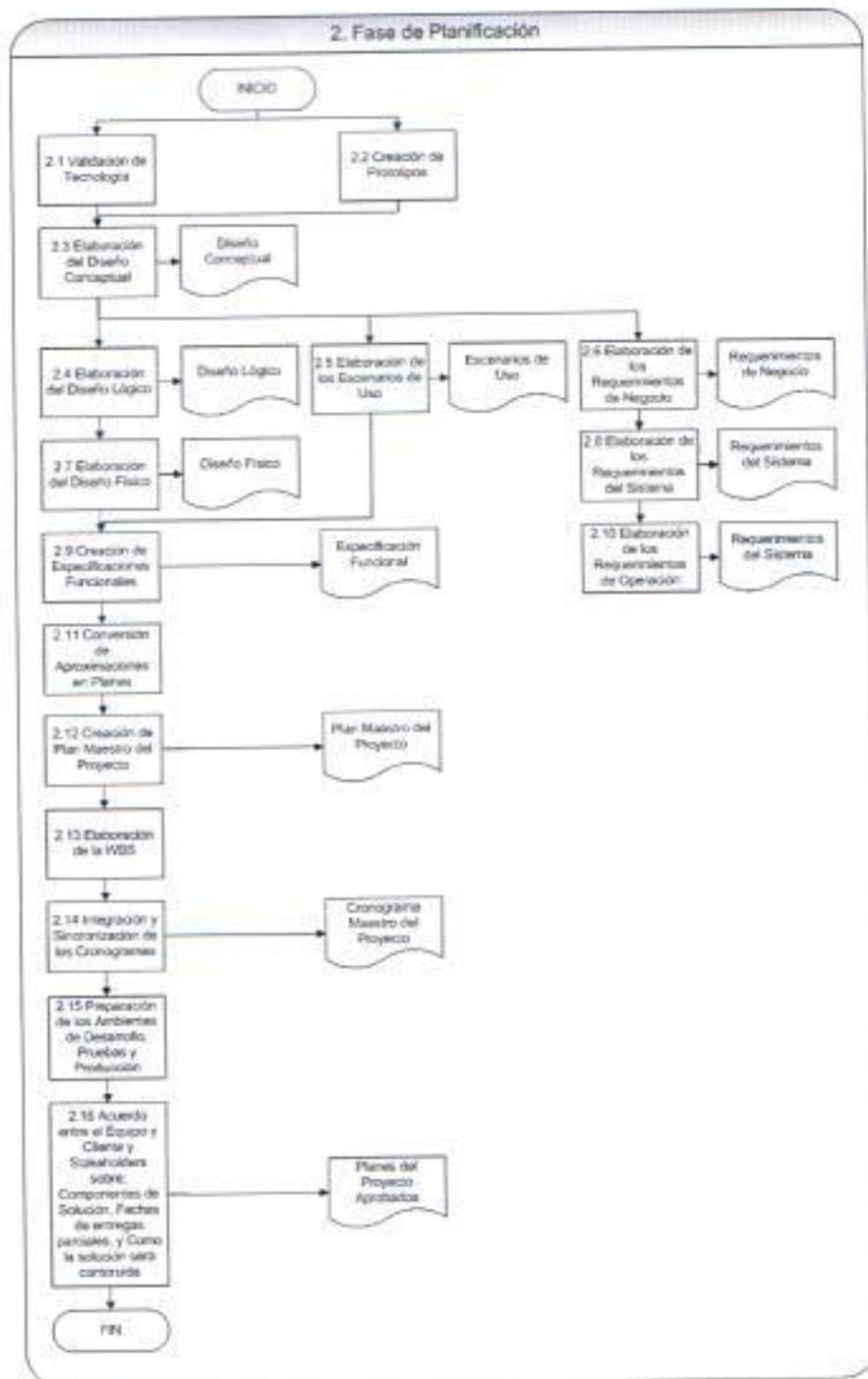


Figura 5.4 Fase de Planificación

### ***Diseño Conceptual***

A continuación se detallan los pasos para detallar lo referente al diseño conceptual, en cuatro pasos, y a la derecha los responsables de las secciones que este documento comprende, según el modelo RACI.

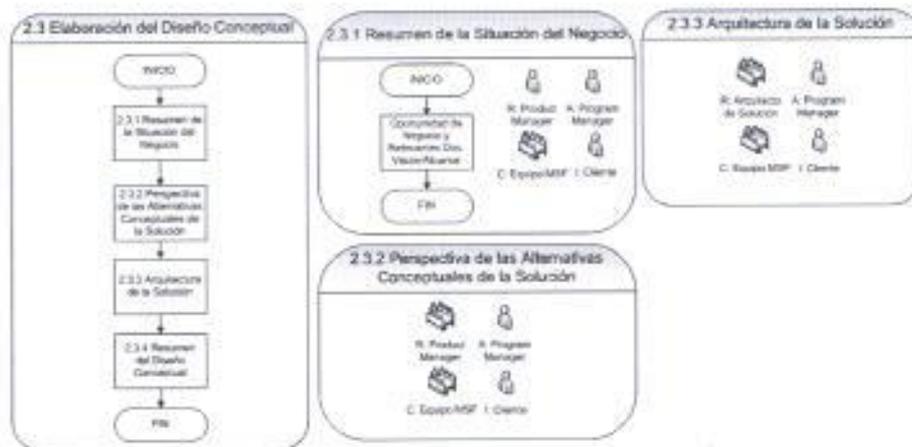
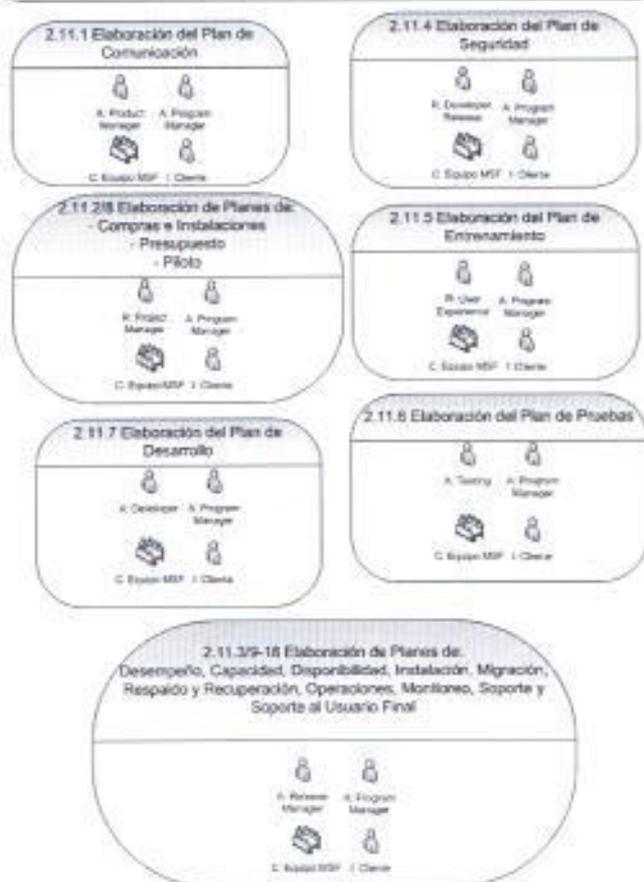


Figura 5.5 Diseño Conceptual

### **Conversión de Aproximaciones en Planes**

Se expone a continuación la siguiente recomendación del orden y bajo qué dependencias deberán ejecutarse todas las tareas que implican creación de planes individuales por el equipo de trabajo del proyecto; es decir, que planes se pueden crear después de realizar qué planes (*véase Convertir Aproximaciones en Planes*). Más abajo se detalla quienes son los responsables en modelo RACI para la realización de cada uno de los planes. Nótese que el último responsable o *Accountable* de cada plan es el líder de esta fase de Planificación: el Program Manager. En este caso se consulta mediante reuniones de trabajo a todos el equipo para recabar dicha información, y el cliente siempre será informado de dichos avances; con esto se asegura el principio de fomentar comunicaciones abiertas (*véase Principios fundamentales*).



## Desarrollo

A lo largo de esta fase el líder del equipo es tanto el rol de Developer como UserExperience. Esta fase comienza con lo más elemental: la construcción de la funcionalidad núcleo (3.1, véase Construcción de la Funcionalidad Núcleo), y sin necesidad de depender de ninguna otra actividad, se pueden empezar a preparar el Conjunto de Casos de Prueba (3.2, Figura 5.7 Fase de Desarrollo), los procedimientos generados por la solución para el personal de Operaciones (3.3, Figura 5.7 Fase de Desarrollo), y la revisión de los prototipos creados en la fase anterior (3.4, Figura 5.7 Fase de Desarrollo). Luego se desarrolla la infraestructura (3.7, véase Desarrollo de la Infraestructura) y desarrolla pruebas de conceptos (3.10, véase Prueba de conceptos) y son validados (3.11, Figura 5.7 Fase de Desarrollo). Por otro lado después de desarrollar los procedimientos para operaciones, se ejecutan los planes de instalación definidos en la fase anterior (3.5, véase Construir documentación de Operaciones), y si aplicase se desarrollan encuestas en sitio, en el caso de proyectos que requieren instalación distribuida, a efectos de obtener información detallada acerca de la misma. Los prototipos creados sirven para ser validados por el rol de UserExperience mediante el desarrollo y ejecución de pruebas de usabilidad (3.6, véase Pruebas de Usabilidad); esto le sirve luego a este rol para crear y/o actualizar los documentos del usuario y de entrenamiento (3.9, véase Construir entregables de UserExperience).

Luego se comienza el gran lazo de construcción de la solución, tal como se describe en el capítulo 3, en Desarrollo de la solución usando Liberaciones Internas (InternalReleases).

En cada iteración de desarrollo de una liberación interna, existe un lazo interno para desarrollar cada build (3.12.1, véase Construcciones (Builds)), junto con su documentación y scripts de instalación (3.12.2 y 3.12.3, véanse Construir entregables de UserExperience y Construir documentación de Operaciones), misma que debe ser revisada constantemente. Se ejecutan las pruebas de cubrimiento correspondientes (3.12.4, véase Pruebas de Cubrimiento), y dependiendo de si se pasan esas pruebas se

continúa con el siguiente build, de lo contrario se efectúan las correcciones (3.12.5, Figura 5.7 Fase de Desarrollo). Cuando no existen más builds que construir, se efectúan las revisiones correspondientes de la documentación hasta el momento elaboradas (3.13.6 y 3.12.7, Figura 5.7 Fase de Desarrollo). Los entregables fueron definidos previamente (véase Desarrollo (Microsoft Corporation - MSF Desarrollo, 2009)).

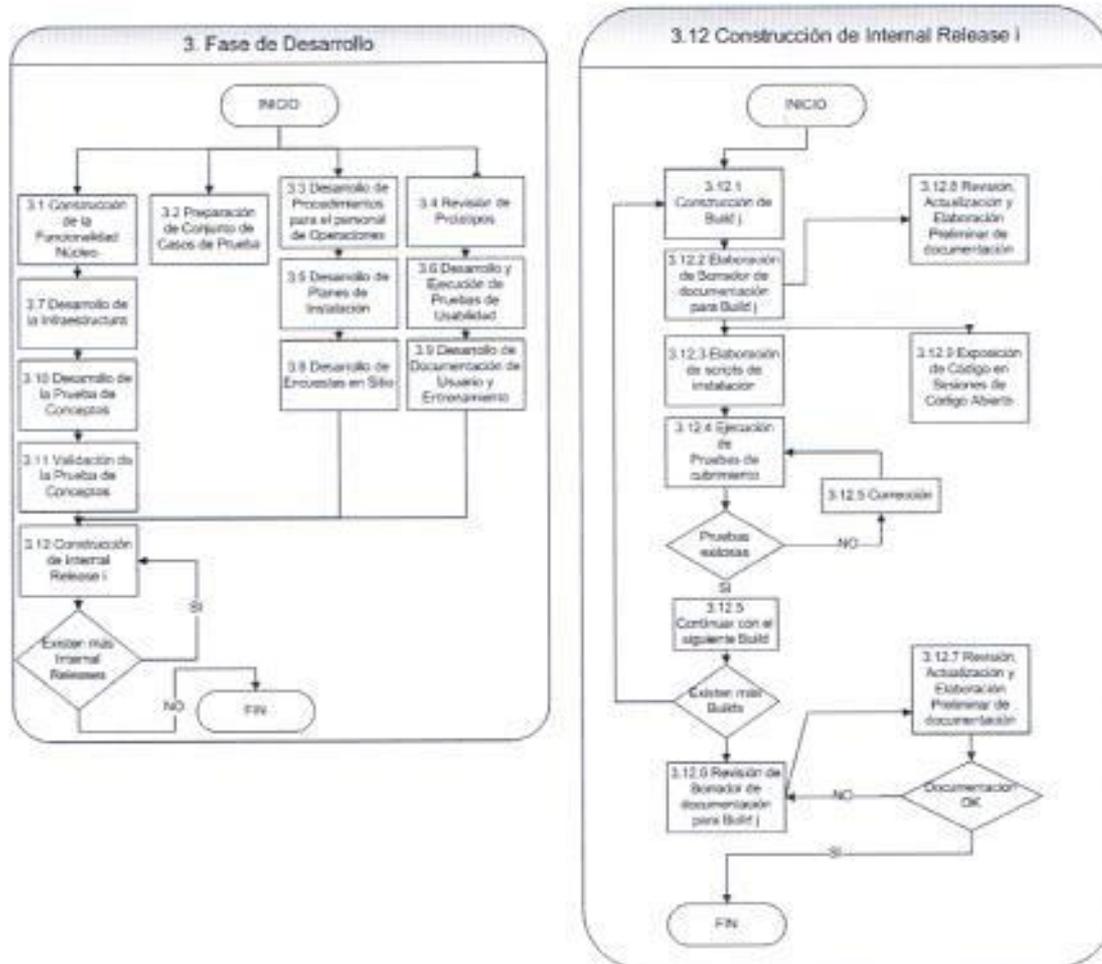


Figura 5.7 Fase de Desarrollo

## Estabilización

A lo largo de esta fase el líder del equipo es tanto el rol de Developer como Testing. A continuación se extiende el proceso de revisión, identificación y resolución de errores, a nivel de liberación interna, la cual se recomienda que sea diaria (4.1, Figura 5.8 Fase de Estabilización). Se deben identificar y categorizar (4.2, véase Manejar errores efectivamente) los asuntos pendientes de calidad encontrados (issues). Se identifican

aquellos que son errores (véase La naturaleza de los errores (bugs)), y se acumulan los del día junto con los pendientes de días anteriores (4.3 y 4.4, Figura 5.8 Fase de Estabilización). Se resuelven los errores (4.5, véase Resolver Errores); y se aplican los conceptos de Convergencia de Errores (véase Convergencia de Errores) y Rebote de Cero Errores (véase Rebote de Cero Errores), y políticas como el hecho de que si no se encuentra convergencia de errores en un periodo mayor de 14 días (2 semanas), la solución se regresa a fase de Desarrollo. En caso de encontrarse ambos conceptos, y no haber más errores identificados (teniendo en cuenta que la solución se encuentra en versión beta, con un número de beta definido previamente como política a nivel de proyecto), se puede optar por salir del lazo.

En este punto se pueden efectuar pruebas de certificación con el usuario (4.8, véase Pruebas de Aceptación del Usuario); y a completarlas exitosamente (nombrándolo RC, véase Designando un Candidato a Liberación (ReleaseCandidate)), se tarja la documentación existente en sus última versiones (4.9, Figura 5.8 Fase de Estabilización). Se ejecutan opcionalmente un piloto, de acuerdo a las condiciones del proyecto (4.10, véase Pruebas bajo condiciones en vivo). Cuando se considere el piloto como exitoso, cumpliendo los criterios de aceptación definidos en la fase de visionamiento y de acuerdo al plan de desarrollo, pruebas y demás, se sale de esta fase con los entregables ya descritos (véase Estabilización (Microsoft Corporation - MSF Estabilización, 2009)) aprobados por el equipo y por el cliente.

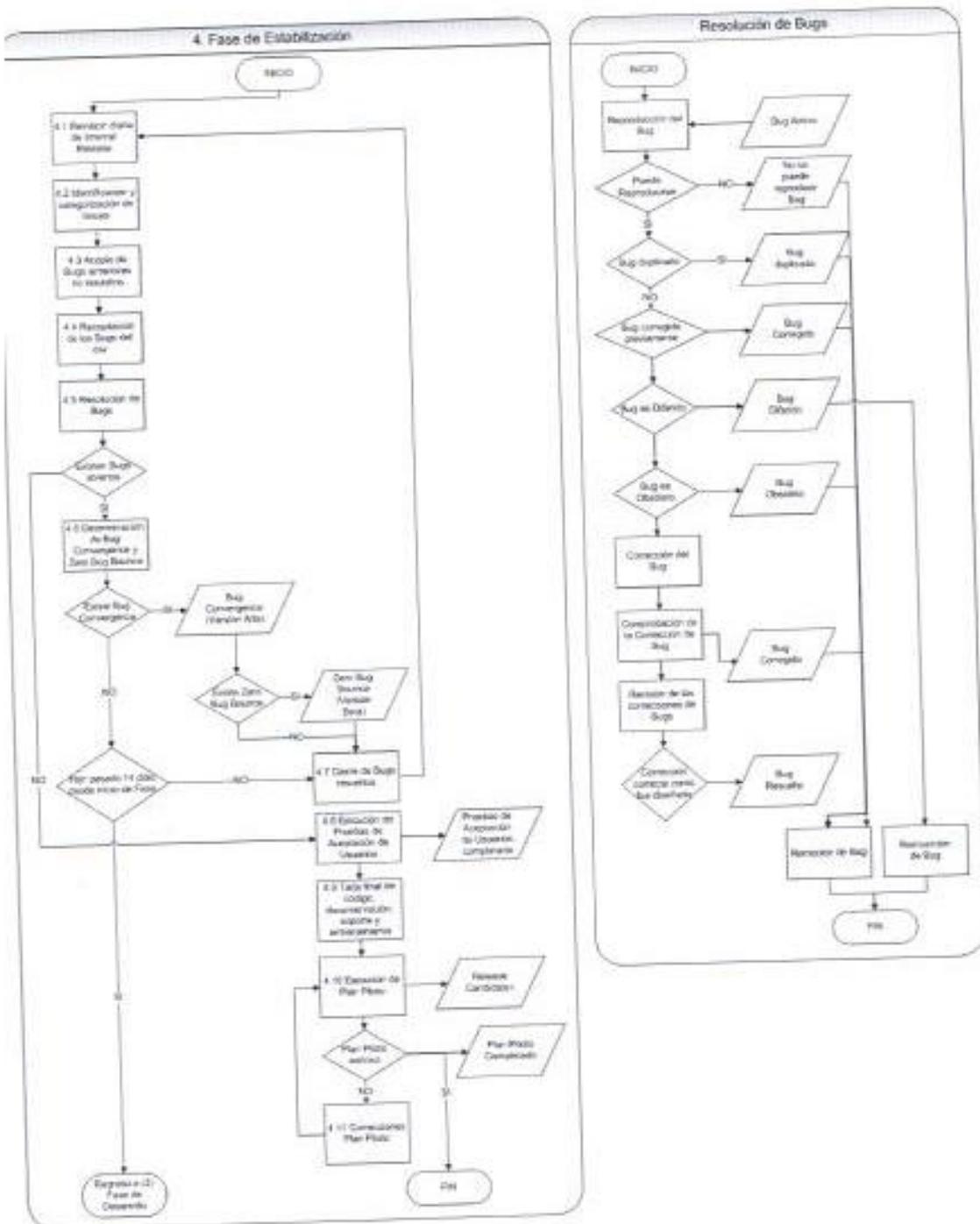


Figura 5.8 Fase de Estabilización

## Instalación

En esta fase el líder del equipo es el Release Manager. Se recomienda que lo primero que se instalen los componentes núcleo de la solución (5.1, véase *Instalando tecnología núcleo*), para luego los componentes adicionales (5.2, Figura 5.9 Fase de Instalación) y luego aquellas instalaciones en sitio (5.3, véase *Ejecutar instalaciones en sitio*), mientras

se puede avanzar con el entrenamiento en sitio (5.4, véase Entrenamiento durante la instalación en sitio). Luego se realizan las tareas de estabilización de la instalación (5.5, véase Estabilización de la instalación). Mientras se ingresa al periodo de quietud (5.6, véase El periodo de quietud), se definen y valoran en vivo los SLAs esbozados en la fase de visionamiento y pulidos en la de planificación (5.7, Figura 5.9 Fase de Instalación). Luego, se debe emitir el Reporte de Cierre del Proyecto (5.8, Figura 5.9 Fase de Instalación), y una encuesta de satisfacción del cliente (5.9, véase Encuestar satisfacción al cliente), la cual debe evaluarse (5.10, Figura 5.9 Fase de Instalación). La fase ni el proyecto no debe terminarse hasta que no se obtenga la firma del cliente (5.11, véase Obtener la firma del cliente). Considérense los Factores de éxito para la fase de liberación al terminar esta fase.

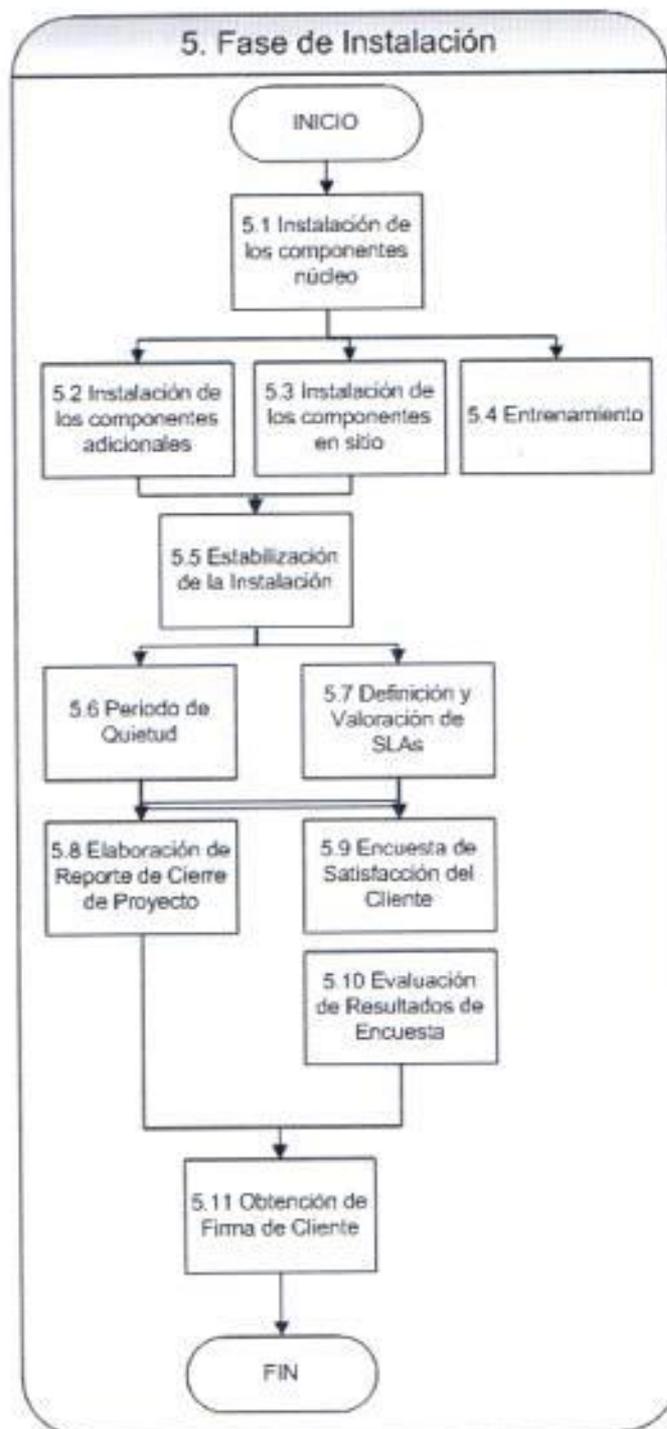


Figura 5.9 Fase de Instalación

## Políticas de Soporte a la Metodología MSF

Los siguientes son ejemplos de políticas relacionadas al flujo de procedimientos

propuestos, los cuales lo fomentan y promulgan.

## Directivas Generales

1. El desconocimiento de las presentes políticas no exime de su cumplimiento.
2. Cualquier requerimiento, petición o solicitud, que esté relacionado con algún programa, aplicación o sistema de uso de la Empresa, será considerado un Proyecto MSF; en virtud del cual, se ceñirá a los procedimientos establecidos para la Gestión del Ciclo de Vida de las Aplicaciones basados en MSF.

## De la Cultura Organizacional en el Ciclo de Vida de Desarrollo

3. En todo Proyecto MSF se deberá trabajar buscando, procurando, fomentando y consiguiendo resultados con mentalidad basada en los siguientes preceptos y los siguientes principios:

### ***Mentalidades***

- 1) Enfocarse en el Valor del Negocio
- 2) Abogar por su circunscripción
- 3) Tomar Orgullo del Trabajo
- 4) Cumplir sus Compromisos
- 5) Mirar la Imagen Completa
- 6) Fomentar un Equipo de Iguales
- 7) Practicar Buena Comunión
- 8) Aprender Continuamente
- 9) Internalizar Calidad de Servicio

### ***Principios***

- 1) Servir de compañero del Cliente
- 2) Trabajar hacia una Visión Compartida
- 3) Entregar Valor Incremental
- 4) Invertir en Calidad
- 5) Empoderar a los Miembros del Equipo
- 6) Establecer Clara Contabilidad
- 7) Aprender de Todas las Experiencias
- 8) Fomentar Comunicaciones Abiertas
- 9) Permanecer Ágil, Adaptarse al Cambio

## De los Proyectos MSF

4. Ningún miembro de algún equipo MSF podrá recibir requerimientos de ninguna índole pasado las 17:00.
5. Los jefes inmediatos de cada colaborador elaborarán dentro de los primeros 15 días del mes de Enero de cada año, el calendario de vacaciones y permisos maternales de cada uno de ellos, en función del día de ingreso a la Institución, y del periodo de gestación según sea el caso. Finalmente, se elaborará un Calendario Unificado de Vacaciones y será considerado como criterio de validación en la asignación de los recursos para cualquier Proyecto.

## De las Reuniones

6. Toda reunión será clasificada en los siguientes:
  - 1) Reunión de definición y aclaración de requerimientos.
  - 2) Reunión de revisión de avances.
  - 3) Reunión de formulación de propuestas.
  - 4) Reunión informativa a los asistentes.
  - 5) Reunión de elaboración de trabajo.
7. Una reunión puede ser convocada por cualquier colaborador, el convocante se convierte en el moderador de la reunión en el momento que la reunión se desarrolle.
8. Se recomienda no convocar a reunión entre las horas 12:00 a 14:00, ni pasadas las 18:00. Se recomienda también establecer reuniones de más de 2 horas de duración, a excepción de reuniones de elaboración de trabajo. Tampoco se recomienda convocar a más de 16 personas a una misma reunión. Si se llevan a cabo reuniones de más de 2 horas, se debe hacer un receso de 15 a 20 min. cada 2 horas.
9. El convocante a reunión debe averiguar la disponibilidad de cada participante a convocar. Si durante el tiempo proyectado de reunión, el participante se encuentra en alguna otra, el convocante no podrá convocarlo.
10. El convocante a una reunión deberá tener una razón justificada para convocar a cada uno de los participantes a la reunión. Algunas de estas razones pueden ser las siguientes:
  - Tiene capacidad para tomar decisiones,
  - Posee amplia experiencia,
  - Son conocedores del tema en cuestión,
  - Tomarán la responsabilidad,
  - Conocer parte de la información,
  - Pueden aportar soluciones,

- Se encuentran involucrados en el asunto,
  - Son quienes tienen la responsabilidad relacionada con el tema a tratar.
11. El participante convocado a reunión tendrá un tiempo de media hora para aceptar, rechazar o proponer una nueva hora.
  12. Al inicio de la reunión el moderador
  13. Si el participante convocado a reunión considera rechazar su convocatoria, debe primero consultar con el convocante el motivo por el que fue convocado, y si luego de eso aún considera rechazar dicha reunión, debe justificar convincentemente el motivo del rechazo. Si el motivo por el cual el convocado desea rechazar la convocatoria es la indisponibilidad de tiempo, no deberá rechazar la convocatoria, sino proponer una hora nueva.

## Aprobación de la Solución desarrollada

Con el fin de que este flujo de procesos basado en un marco de trabajo; el cual, a partir de este momento se convierte en una metodología de la empresa, de manera general se debe realizar los siguientes pasos antes de instaurarla como procedimiento institucional de la empresa:

1. Someterla a exposición, revisión y aprobación de los interesados y auspiciantes de este proyecto, así como del Vicepresidente de IT o su equivalente como encargado de esta área.
2. Este último deberá exponerla en el Comité de IT o Comité Ejecutivo (o ambos) en la próxima sesión o asamblea ordinaria; a efectos de aprobarla por este, y tener su auspicio para comenzar a difundirla a las demás áreas como una política institucional.
3. La decisión tomada por el Comité Ejecutivo será difundida a las cabezas (tales como Vicepresidencias o Gerencias) de todas las áreas de la empresa, a efectos de informarles de cómo deberán trabajar, haciendo que sus recursos apoyen y fomenten esta metodología, tomando en cuenta que serán potenciales clientes, auspiciantes, Product Managers, UserExperiences o incluso Testers. Si los proyectos no son de índole tecnológico, y la empresa así lo decidiera, también estas áreas tendrían potenciales Developers, Program Managers e incluso Release Managers para cierto tipo de proyectos.

4. Es necesario conformar un Comité de Aprobación de Cambios (ChangeAdvisoryBoard – CAB), para desempeñar las funciones y atribuciones expuestas en los capítulos 3 y 5. Se recomienda que esta entidad tenga un grado de autonomía y autoridad más allá de la mera coordinación de los recursos de la empresa. Por tanto, se sugiere incluir entre sus miembros a representantes de la mayoría, sino las más importantes (en base a criterios internos) áreas de la empresa. Un buen ejemplo de una CAB sería una Oficina de Administración de Proyectos (PMO).
5. En paralelo, este flujo de procesos, debe ser asentado y traducido al lenguaje conocido por la empresa por el área correspondiente (Organización y Métodos por ejemplo) junto con los demás procesos de la empresa, apalancando la Cadena de Valor, puesto que es un proceso institucional aprobado.
6. Iniciar un proceso de entrenamiento y difusión a los distintos tipos de audiencias que necesitarán aprender y familiarizarse con la metodología: Ejecutivos, Administradores de Recursos, Supervisores, y demás posibles miembros de equipo de la empresa; cada uno de ellos con un enfoque particular acorde a las tareas que deberán realizar una vez instaurada.

## Herramientas de Software de Apoyo a la Metodología

A continuación algunas herramientas utilizadas a lo largo de este proyecto, y que han sido útiles o son recomendadas para diversas circunstancias.

### Microsoft™ Project Server®

*URL:* <http://www.microsoft.com/project/en/us/project-server-2010.aspx>

*Tipo de licencia:* **Adquirida**

Herramienta tipo cliente-servidor que provee capacidades de manejo de portafolio de proyectos de la organización para ayudarla a iniciar, seleccionar, planificar y entregar proyectos. Se acopla perfectamente con Microsoft™ Office® Project® Professional, como herramienta cliente, puesto que todos los proyectos que constan en el servidor pueden ser descargados para editarse usando esta herramienta.

## Mantis Bug Tracker

*URL:*<http://www.mantisbt.org/>

*Tipo de licencia:* **Freeware GPL**

Herramienta tipo cliente-servidor creada en PHP sobre MySQL que sirve para llevar un muy completo rastro y seguimiento de todos los errores (bugs) encontrados en un determinado proyecto. Una vez instalada la herramienta en el servidor, el ambiente web permite al cliente crear proyectos, generar incidentes, y asignarlos a usuarios en la herramienta, para hacerlos pasar por un mapa de estados configurables (véase [Resolver Errores](#)), y cada uno guardar una bitácora, misma que sirve para generar reportes, estadísticas que pueden dar una idea de cuando se ha alcanzado Convergencia de Errores o Rebote de Cero Errores.

## WBS Chart Pro

*URL:*[www.criticaltools.com/wbsmain.htm](http://www.criticaltools.com/wbsmain.htm)

*Tipo de licencia:* **Freeware**

Herramienta de escritorio que se puede acoplar a Project Professional 2007 con un ambiente visual para crear paquetes de trabajo con los elementos comunes: fecha de inicio y fin, duración, costo y jerarquía de los paquetes de trabajo. Al bajar al nivel de detalle de tareas, es posible exportarlo a Project Professional como un archivo de Project en blanco.

## PERT Chart Expert

*URL:*<http://www.criticaltools.com/pertchartexpertsoftware.htm>

*Tipo de licencia:* **Freeware**

Herramienta de escritorio que se puede acoplar a Project Professional 2007 con un ambiente visual para crear diagramas de PERT (también conocidos como diagramas de red, diagramas de precedencia o diagramas lógicos) para mostrar dependencias entre los paquetes de trabajo o tareas. Es posible exportarlo a Project Professional como un archivo de Project en blanco.

## Conclusiones y Recomendaciones

1. En la presente tesis se ha presentado la situación actual de una empresa, cuyas características son los factores comunes de cualquier empresa grande que está iniciando sus procesos de apego a estándares internacionales de gestión de proyectos.
2. Se investigó y eligió de entre varias alternativas cuál es el marco de trabajo que mejor se sujete a las características, expectativas y flexibilidad necesarias en función a la cultura organizacional de la empresa, a aquella que guarda un mejor equilibrio entre estos factores.
3. Se expuso los más relevantes lineamientos de esta disciplina que justificaran las decisiones que se tomarán en el último capítulo, exponiendo lo que establece el Modelo de Procesos, Modelo de Equipos y el proceso de Gestión de Riesgos.
4. Se remarcaron las barreras y limitaciones organizacionales que ocurrirían en casi cualquier empresa, enfocados tanto al área operativa como a la administrativa y ejecutiva.
5. Finalmente, se expone el trabajo resultante en forma de un diagrama de flujo de procesos, con los entregables y responsables de cada paso, amparados en las buenas prácticas y sugerencias del marco de trabajo elegido, excluyendo aquellas que no agregarían mucho valor, sino más bien aquellas mandatorias.
6. El resultado de este trabajo es un procedimiento bastante flexible y ajustable a la gran mayoría de las estructuras organizacionales de la empresa, puesto que en este esquema, no se toman en cuenta cargos sino roles.
1. Se recomienda, más allá de encontrar una o varias maneras de sobrellevar las barreras organizacionales expuestas en el capítulo 4, llevar un mensaje de compromiso y predisposición a cambiar un modo de trabajar que si bien es cierto puede haber conducido a resultados excelentes a través del tiempo; pudo haber conllevado también a altos costos tanto de esfuerzo por parte el equipo de trabajo, como económicos, o de tiempo empleado; trayendo como resultado un producto de no tan alta calidad, o desarrollado en una fecha más allá de lo

prevista, o simplemente, que no cumple las expectativas que el cliente o el usuario inicialmente esperaba de ella, y que finalmente, va en desmedro de la imagen de un departamento y una institución.

- II. Los programadores, si bien es cierto, al inicio de la fase de Planificación reciben solamente lo inscrito en las Especificaciones Funcionales como entrada para construir la solución, según lo que la disciplina recomienda, también es cierto que en las fases previas deben ser consultores tecnológicos y recomendar al resto del equipo lo que dentro de su perspectiva técnica es viable y aplicable.
- III. Se debe empezar a construir y mantener periódicamente una Base de Habilidades y Destrezas, que cuente como mínimo a todo el personal de tecnología involucrado, junto con los de otras áreas que intervengan frecuentemente en proyectos de tecnología. El conocimiento adquirido por una persona debe ser compartido a los demás, mediante sesiones semanales de "WarRoom", en las que una persona expone sus lecciones aprendidas al resto, quienes criticarán constructivamente y/o asimilarán lo aprendido para futuras referencias. Se debe fomentar la capacitación constante y el plan de carrera individual de cada recurso. De ser necesario contratar personal, debe gestionarse con el área de RRHH correspondiente tal necesidad.
- IV. Se debe asignar la responsabilidad de la Administración de Recursos a una persona o conjunto de personas (dependiendo del tamaño de la organización), que se encarguen exclusivamente de la reserva, control y monitoreo de actividades de ellos, sobreasignación y sub-asignación, y ser intermediario de gestión de Recursos entre ellos y los Program Managers.
- V. La promoción constante y ferviente del Gobierno de IT es imprescindible y decisiva para que la implantación sea exitosa.
- VI. Las áreas solicitantes, deben estar de acuerdo e impulsar respectivamente sus voceros ante los proyectos nacientes como Product Managers de los mismos.
- VII. El Gobierno de IT y el Comité Ejecutivo deben mantener una estrecha comunicación del "pulso" del proceso en sí mismo, y del avance de todos los proyectos en términos de la triple restricción (alcance, tiempo o avance, y costos), y apoyar en consenso cualquier decisión a favor del cumplimiento de la

metodología adoptada. Constantemente compararse con empresas que adoptaron la misma disciplina a fin de recabar lecciones aprendidas y tratar en lo posible de adoptarlas con celeridad y buen criterio.

- VIII. La empresa una vez llegada a una fase temprana de madurez de adopción de la metodología debe inmediatamente, gestionar con alguna autoridad calificada la aplicación de una certificación en este marco de trabajo, y corregir o "pulir" ciertos detalles que puedan ser observados por ellos. Siempre es positivo para una empresa que sus procesos sean avalados por alguna entidad de renombre.

## Apéndices

### Apéndice 1. Metodologías de desarrollo y programación (Wikipedia, 2010)

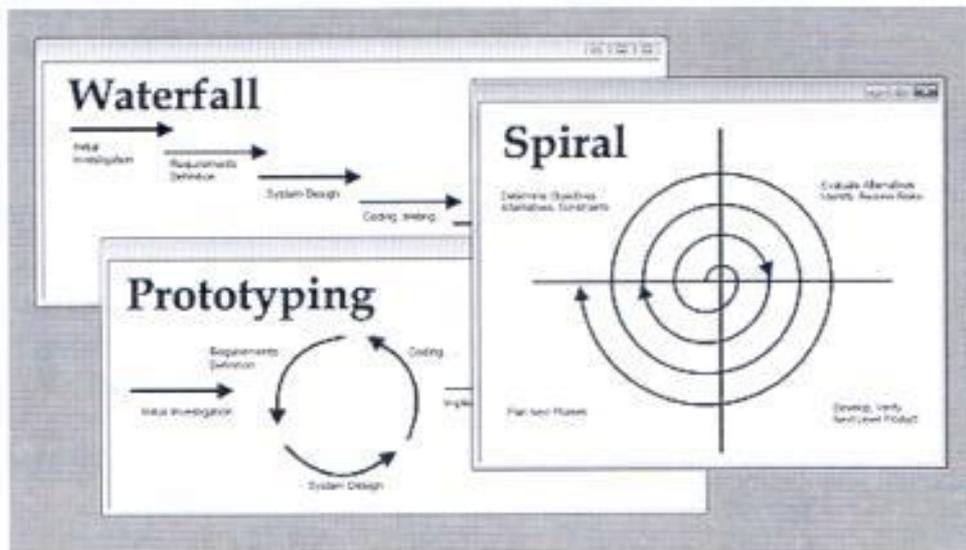


Figura 0.1 Metodologías de desarrollo

## Metodologías de desarrollo de software

1970s

- Programación estructurada desde 1969
- Programación estructurada Jackson desde 1975

## 1980s

- Structured Systems Analysis and Design Methodology (SSADM) desde 1980
- Structured Analysis and Design Technique (SADT) desde 1980
- Ingeniería de la información (IE/IEM) desde 1981

## 1990s

- Rapid application development (RAD) desde 1991.
- Programación orientada a objetos (OOP) a lo largo de la década de los 90's
- Virtual finite state machine (VFSM) desde 1990s
- Dynamic Systems Development Method desarrollado en UK desde 1995.
- Scrum (desarrollo), en la última parte de los 90's
- Rational Unified Process (RUP) desde 1999.

## Nuevo milenio

- Extreme Programming (XP) desde 1999
- Enterprise Unified Process (EUP) extensiones RUP desde 2002
- Constructionist design methodology (CDM) desde 2004 por Kristinn R. Thórisson
- Agile Unified Process (AUP) desde 2005 por Scott Ambler

## Enfoques Generales de desarrollo de software

- Modelo en cascada: Framework lineal.
- Prototipado: Framework iterativo.
- Incremental: Combinación de framework lineal e iterativo.
- Espiral: Combinación de framework lineal e iterativo.
- RAD: Rapid Application Development, framework iterativo.

Modelo en cascada

Es un proceso secuencial de desarrollo en el que los pasos de desarrollo son vistos hacia abajo (como en una cascada de agua) a través de las fases de análisis de las necesidades, el diseño, implementación, pruebas (validación), la integración, y mantenimiento. La primera descripción formal del modelo de cascada se cita a menudo a un artículo publicado por Winston Royce W.<sup>2</sup> en 1970, aunque Royce no utiliza el término "cascada" de este artículo.

Los principios básicos del modelo de cascada son los siguientes:<sup>1</sup>

- El proyecto está dividido en fases secuenciales, con cierta superposición y splashback aceptable entre fases.

- Se hace hincapié en la planificación, los horarios, fechas, presupuestos y ejecución de todo un sistema de una sola vez.
- Un estricto control se mantiene durante la vida del proyecto a través de la utilización de una amplia documentación escrita, así como a través de comentarios y aprobación / signoff por el usuario y la tecnología de la información de gestión al final de la mayoría de las fases antes de comenzar la próxima fase.

## Prototipado

El prototipado es el framework de actividades dedicada al desarrollo de software prototipo, es decir, versiones incompletas del software a desarrollar.

## Incremental

Provee una estrategia para controlar la complejidad y los riesgos, desarrollando una parte del producto software reservando el resto de aspectos para el futuro.

Los principios básicos son:

- Una serie de mini-Cascadas se llevan a cabo, donde todas las fases de la cascada modelo de desarrollo se han completado para una pequeña parte de los sistemas, antes de proceder a la próxima incremental
- Se definen los requisitos antes de proceder con la evolutivo, se realiza un mini-Cascada de desarrollo de cada uno de los incrementos del sistema
- El concepto inicial de software, análisis de las necesidades, y el diseño de la arquitectura y colectiva básicas se definen utilizando el enfoque de cascada, seguida por iterativo de prototipos, que culmina en la instalación del prototipo final.

## Espiral

Los principios básicos son:

- La atención se centra en la evaluación y reducción del riesgo del proyecto dividiendo el proyecto en segmentos más pequeños y proporcionar más facilidad de cambio durante el proceso de desarrollo, así como ofrecer la oportunidad de evaluar los riesgos y con un peso de la consideración de la continuación del proyecto durante todo el ciclo de vida.
- Cada viaje alrededor de la espiral atraviesa cuatro cuadrantes básicos: (1) determinar objetivos, alternativas, y desencadenantes de la iteración; (2) Evaluar alternativas; Identificar y resolver los riesgos; (3) desarrollar y verificar los resultados de la iteración, y (4) plan de la próxima iteración.<sup>3</sup>
- Cada ciclo comienza con la identificación de los interesados y sus condiciones de ganancia, y termina con la revisión y examinación.<sup>3</sup>

## Rapid Application Development (RAD)

El *desarrollo rápido de aplicaciones* (RAD) es una metodología de desarrollo de software, que implica el desarrollo iterativo y la construcción de prototipos. El desarrollo rápido de aplicaciones es un término originalmente utilizado para describir un proceso de desarrollo de software introducido por James Martin en 1991.

Principios básicos:

- Objetivo clave es para un rápido desarrollo y entrega de una alta calidad en un sistema de relativamente bajo coste de inversión.
- Intenta reducir los riesgos inherentes del proyecto partiéndolo en segmentos más pequeños y proporcionar más facilidad de cambio durante el proceso de desarrollo.
- Orientación dedicada a producir sistemas de alta calidad con rapidez, principalmente mediante el uso de iteración por prototipos (en cualquier etapa de desarrollo), promueve la participación de los usuarios y el uso de herramientas de desarrollo computarizadas. Estas herramientas pueden incluir constructores de Interfaz gráfica de usuario (GUI), ComputerAided Software Engineering (CASE) las herramientas, los sistemas de gestión de bases de datos (DBMS), lenguajes de programación de cuarta generación, generadores de código, y técnicas orientada a objetos.
- Hace especial hincapié en el cumplimiento de la necesidad comercial, mientras que la ingeniería tecnológica o la excelencia es de menor importancia.
- Control de proyecto implica el desarrollo de prioridades y la definición de los plazos de entrega. Si el proyecto empieza a aplazarse, se hace hincapié en la reducción de requisitos para el ajuste, no en el aumento de la fecha límite.
- En general incluye Jointapplicationdevelopment (JAD), donde los usuarios están intensamente participando en el diseño del sistema, ya sea a través de la creación de consenso estructurado en talleres, o por vía electrónica.
- La participación activa de los usuarios es imprescindible.
- Iterativamente realiza la producción de software, en lugar de colgarse de un prototipo.
- Produce la documentación necesaria para facilitar el futuro desarrollo y mantenimiento.

## Otros enfoques de desarrollo de software

- Metodologías de desarrollo Orientado a objetos, Diseño orientado a objetos (OOD) de Grady Booch, también conocido como Análisis y Diseño Orientado a Objetos (OOAD). El modelo incluye seis diagramas: de clase, objeto, estado de transición, la interacción, módulo, y el proceso.
- Top-downprogramming, evolucionado en la década de 1970 por el investigador de IBMHarlan Mills (y NiklausWirth) en Desarrollo Estructurado.

- Proceso Unificado, es una metodología de desarrollo de software, basado en UML. Organiza el desarrollo de software en cuatro fases, cada una de ellas con la ejecución de una o más iteraciones de desarrollo de software: creación, elaboración, construcción, y las directrices. Hay una serie de herramientas y productos diseñados para facilitar la aplicación. Una de las versiones más populares es la de RationalUnifiedProcess.

## Apéndice 2. Descripción de las plantillas de MSF y su contenido

### Visionamiento

#### *Estructura del Proyecto*

Aproximaciones al Proyecto .....	
Metas del Proyecto, Objetivos, Supuestos y Restricciones .....	
Alcance Del Proyecto .....	
Matriz de Negociación del Proyecto.....	
Aproximación al Proyecto Maestro .....	
Aproximación a los Hitos .....	
Estimados del Proyecto.....	
Resumen del Cronograma.....	
Roles y Responsabilidades .....	
Conocimiento, Habilidades y Capacidades .....	
Estructura del Equipo .....	
Protocolos del Proyecto .....	
Aproximación a la Administración de Riesgos y Problemas .....	
Aproximación a la Administración de Configuración.....	
Aproximación a la Administración de Cambio .....	
Aproximación a la Administración de Liberación.....	
Aproximación al Aseguramiento de la Calidad del Proyecto.....	
Aproximación a la Comunicación del Proyecto .....	
Aproximación al Ambiente del Equipo .....	
Evaluación de riesgos y problemas.....	
Glosario del Proyecto .....	

**Visión y Alcance**

Oportunidad de Negocio .....	
Declaración de Oportunidad .....	
Declaración de la Visión .....	
Análisis de Beneficios .....	
Concepto de la Solución .....	
Metas, Objetivos, Supuestos y Restricciones .....	
Análisis de Uso .....	
Perfiles de Usuario .....	
Escenarios de Uso .....	
Requerimientos .....	
Requerimientos del Negocio .....	
Requerimientos del Usuario .....	
Requerimientos Operacionales .....	
Requerimientos del Sistema .....	
Alcance .....	
Lista de Características/Funciones .....	
Fuera del Alcance .....	
Estrategia de Liberación de Versiones .....	
Criterios de Aceptación .....	
Criterios Operacionales .....	
Estrategias para el diseño de la Solución .....	
Estrategia de Diseño Arquitectónico .....	
Estrategia de Diseño Técnico .....	

## **Evaluación del Estado Actual de la Infraestructura**

Resumen .....	
Objetivos.....	
Alcance del Proyecto .....	
Recomendación.....	
Visitas al Sitio .....	
Estado de la Suficiencia de Datos.....	
Grupos de Hardware.....	
Inventarios .....	
Arquitectura Tecnológica .....	
Arquitecturas de Red.....	
Arquitecturas del Dominio de Windows .....	
Seguridad de Windows.....	
Servicios de la Infraestructura de Red.....	
Servicios e Infraestructura Informática .....	
Uso del Sistema Actual.....	
Métricas de Uso .....	
Análisis de Uso del Estado Futuro del Sistema .....	
Métricas Proyectadas .....	
Análisis del Ambiente Operativo.....	
SLAs (ANSs).....	
Análisis de Seguridad.....	
Estado Actual.....	
Requerimientos del Estado Futuro.....	

## **Planificación**

### ***Diseño Conceptual***

Resumen del Diseño Conceptual.....	
Situación del Negocio.....	
Perspectiva de las Alternativas Conceptuales de la Solución.....	
Arquitectura de la Solución .....	

### ***Diseño Lógico***

Resumen del Diseño Lógico.....	
Objetos.....	
Comportamientos.....	
Atributos .....	
Relaciones .....	
Apéndice: Creación del Diseño Lógico .....	

**Diseño Físico**

Resumen del Diseño Físico.....	
Restricciones Ambientales y Supuestos .....	
Dependencias .....	
Dependencias del Proyecto .....	
Dependencias del Ambiente de Hardware .....	
Dependencias del Ambiente de Software .....	
Desarrollo de Aplicación .....	
Servicios del Usuario (UI – Interface de Usuario).....	
Diseño del Componente <<Nombre del Componente>> .....	
Servicios del Negocio (Lógica del Negocio en Niveles de Capa Intermedia) .....	
Diseño del Componente<<Nombre del Componente>> .....	
Capa de Servicio de Datos .....	
Diseño del Componente<<Nombre del Componente>> .....	
Diseño de la Base de Datos .....	
Instalación de la Infraestructura .....	
Topologías de la Solución .....	
Comunicación de Intranet .....	
Servicios .....	
Protocolos .....	
Seguridad .....	
Comunicación de Internet .....	
Servicios .....	
Protocolos .....	
Seguridad .....	
Comunicación de Extranet .....	
Servicios .....	
Protocolos .....	
Seguridad .....	
Impacto de los Diagnósticos de la Red .....	
Estrategia de Autenticación/ Directorio Activo/ Dominio NT .....	
Direccionamiento .....	
Capa del Cliente .....	
Capa de Servidor .....	
Configuraciones DHCP .....	
Resolución de Nombres .....	
Configuraciones WINS .....	
Configuraciones DNS .....	

Acceso Remoto.....	.....
Estandares de Nombre.....	.....
Ubicación del Servidor.....	.....
Tamaño del Servidor.....	.....
Modelo Administrativo.....	.....
Características de los Productos/Servicios Individuales a ser Implementadas.....	.....
Configuración del Servicio<<Nombre del Servicio>>.....	.....
Configuración del Ambiente.....	.....
Estrategia de Seguridad.....	.....
Implicaciones del Diseño de Seguridad.....	.....
Problemas de la Implementación de Seguridad.....	.....
Consideraciones del Soporte operacional de Seguridad.....	.....
Instalación y Configuración.....	.....
Requerimientos de Hardware.....	.....
Requerimientos de Software.....	.....
Desinstalación.....	.....
Diseño para la Instalación.....	.....
Diseño para la Migración.....	.....
Diseño para la Integración.....	.....
Soporte de Accesibilidad.....	.....
Soporte Multi-Lenguaje.....	.....
Diseño para la Soportabilidad.....	.....
Registro/Eventos.....	.....
Mensajes de Error.....	.....
Herramientas de Diagnóstico.....	.....
Recuperación desde mensajes de Corrupción/ Error.....	.....
Problemas.....	.....
Apéndice.....	.....

## ***Escenarios de Uso***

Resumen de los Escenarios de Uso.....	.....
<Caso de Uso 1: Nombre>.....	.....
<Caso de Uso 2: Nombre>.....	.....
Aproximación General.....	.....
Preguntas para hacer una tormenta de Ideas.....	.....
Desarrollo de Escenarios.....	.....
Paso 1: Comience con Escenarios Principales.....	.....
Paso 2: Defina los Escenarios Secundarios.....	.....
Paso 3: Documentar los Escenarios.....	.....
Ejemplo de Escenario de Uso.....	.....
>Escenario de Uso de Control de vinculación de Tarjeta Inteligente.....	.....

## ***Requerimientos del Negocio***

Resumen de los Requerimientos del Negocio .....	
Análisis Costo Beneficio – ROI .....	
Escalabilidad para las Necesidades del Negocio .....	
Disponibilidad/Confiabilidad para las Necesidades del Negocio – Acceso a los Servicios y Soluciones .....	
Seguridad Fuerte .....	
Soporte para Múltiples Tipos de Dispositivos .....	
Requerimientos de Desempeño .....	
Responsabilidad y Auditoría hacia los Patrocinadores del Negocio .....	

## ***Requerimientos del Usuario***

Resumen .....	
Experiencia del Usuario .....	
Facilidad de Uso .....	
Confiabilidad .....	
Desempeño .....	
Requerimientos de multi-lenguajes .....	
Accesibilidad .....	
Requerimientos del Entrenamiento al Usuario Final .....	

## ***Requerimientos del Sistema***

Resumen de los Requerimientos del Sistema .....	
Dependencia de los Sistemas y de los Servicios .....	
Requerimientos de Interoperabilidad .....	
Evaluación del Impacto en la Red .....	
Impacto del Versionamiento/Ciclo de Vida a los sistemas y al Ambiente .....	

## ***Requerimientos de Operación***

Resumen de los Requerimientos de Operación .....	
Escalabilidad .....	
Seguridad Fuerte .....	
Administrabilidad .....	
Soportabilidad .....	
Disponibilidad/Confiabilidad .....	
Requerimientos de Personal .....	
Requerimientos de Entrenamiento del Personal de Operaciones .....	
Resumen de los Acuerdos de Nivel de Servicio y Requerimientos 7 x 24 x 365 .....	

**Especificación Funcional**

Resumen de la Visión/Alcance del Proyecto.....	
Historia del Proyecto .....	
Resumen Ejecutivo de la Especificación Funcional .....	
Justificación del Proyecto y Metas del Diseño .....	
Resumen de los Requerimientos del Negocio .....	
Resumen de los Requerimientos del Usuario .....	
Resumen de los requerimientos del Sistema .....	
Resumen de los Requerimientos de operaciones .....	
Resumen de los Escenarios de Uso/ Estudio de Casos de Uso.....	
Recorte de Características y Escenarios No Soportados .....	
Supuestos y Dependencias.....	
Diseño de la Solución .....	
Resumen del Diseño Conceptual .....	
Resumen del Diseño Lógico .....	
Resumen de Diseño Físico.....	
Resumen de la Estrategia de Seguridad.....	
Resumen de los Requerimientos de Instalación/Configuración .....	
Resumen de los Requerimientos de Des-Instalación.....	
Resumen de los Requerimientos de Integración.....	
Resumen de Soportabilidad.....	
Resumen de los Requerimientos Legales .....	
<<Comience el texto aquí>> .....	
Resumen del Riesgo .....	
Referencias .....	
Apéndice.....	

*Plan de Capacidad*

*Plan de Compra e Instalaciones*

*Plan de Comunicaciones*

*Plan de Desarrollo*

*Plan de Desempeño*

*Plan de Disponibilidad*

*Plan de Entrenamiento*

*Plan de Instalación*

*Plan de Migración*

*Plan de Monitoreo*

*Plan de Operaciones*

*Plan de Presupuesto*

*Plan de Pruebas*

*Plan de Respaldo y Recuperación*

*Plan de Seguridad*

*Plan de Soporte al Usuario Final*

*Plan de Soporte*

*Plan Piloto*

**Plan Maestro del Proyecto**

Resumen del Plan Maestro del Proyecto.....	
Estructura de la división del trabajo .....	
Planes Individuales .....	
Plan de Desarrollo .....	
Plan de Pruebas .....	
Plan de Comunicaciones.....	
Plan de Soporte .....	
Plan de Operaciones .....	
Plan de Seguridad .....	
Plan de Disponibilidad .....	
Plan de Capacidad .....	
Plan de Monitoreo.....	
Plan de Desempeño .....	
Plan de Soporte para el Usuario-Final.....	
Plan de Instalación .....	
Plan de Entrenamiento .....	
Plan de Adquisiciones e Instalaciones locativas .....	
Plan Piloto .....	
Plan de Presupuesto .....	
Herramientas .....	

## Desarrollo

**Auditoria de Trazabilidad**

Resumen .....	
Dimensiones y Alcance de la Auditoria .....	
Auditoria de la Especificación Funcional .....	
Auditoria de Análisis de Requerimientos .....	
Requerimientos del Negocio .....	
Requerimientos del Sistema .....	
Requerimientos del Usuario .....	
Requerimientos de Operaciones .....	
Auditoria del Diseño Conceptual .....	
Auditoria del Diseño Lógico .....	
Auditoria el Diseño Físico .....	
Auditoria del Plan Maestro del Proyecto .....	
Auditoria del Plan de Desarrollo .....	
Auditoria del Plan de Prueba .....	
Auditoria del Plan de Comunicación .....	
Auditoria del Plan de Soporte de Microsoft .....	
Auditoria del Plan de Operaciones .....	
Seguridad .....	
Disponibilidad .....	
Capacidad .....	
Monitoreo .....	
Desempeño .....	
Soporte .....	
Respaldo y Recuperación .....	
Auditoria del Plan de Soporte al Usuario Final .....	
Auditoria del Plan de Instalación .....	
Auditoria del Plan de Entrenamiento .....	
Auditoria del Plan de Compras e Instalaciones Físicas .....	
Auditoria del Plan de Migración .....	
Auditoria del Plan Piloto .....	
Auditoria del Plan de Presupuesto .....	
Auditoria del Plan de Herramientas .....	

## **Reporte de Pruebas y Errores**

Resumen de Reporte de Pruebas y Errores.....	
Métodos de Pruebas y Herramientas.....	
Resumen de Métodos de Pruebas.....	
Resumen de Herramientas de Pruebas.....	
Reporte de Pruebas.....	
Resumen de los Resultados de las Pruebas – Variaciones del Último Reporte.....	
Discrepancias en la Ejecución del Plan de Prueba.....	
Revisiones de Pruebas Faltantes y Plan de Pruebas.....	
Áreas de pruebas.....	
Área 1 de Producto.....	
Meta (s) de Pruebas.....	
Criterio de Evaluación.....	
Resultados.....	
Recomendaciones.....	
Área 2 de Producto.....	
Meta (s) de Pruebas.....	
Criterio de Evaluación.....	
Resultados.....	
Recomendaciones.....	
Reporte de Errores.....	
Estado de la Construcción.....	
Reporte de Errores.....	
Análisis de Convergencia de Errores.....	
Plan de Pruebas Revisado.....	

## **Estabilización**

### **Especificación de Pruebas**

Resumen de la Especificación de Prueba.....	
Especificaciones de Entrada.....	
Especificaciones de Salida.....	
Ambiente de Pruebas.....	
Requerimientos Procedimentales Especiales.....	
Dependencias de Inter-caso.....	
Casos de Prueba.....	
Elemento de Caso de Prueba – 1.....	
Característica A.....	
Característica B.....	
Referencias del Caso de Prueba 1.....	
Caso de Prueba 2.....	
Característica B.....	
Referencias del Caso de Prueba 2.....	

*Revisión del Piloto*

Resumen .....	
Objetivos .....	
Factores de Éxito y Métricas .....	
Síntesis de la Retroalimentación del Piloto .....	
Sesiones con la Comunidad de Usuarios .....	
Reporte de Problemas .....	
Encuestas al Usuario Final .....	
Observaciones del Equipo del proyecto .....	
Reportes de las Operaciones Estadísticas .....	
Retroalimentación del Equipo del Proyecto .....	
Retroalimentación del Entrenamiento .....	
Retroalimentación de la Instalación .....	
Retroalimentación de Soporte .....	
Retroalimentación de la Comunicación .....	
Problemas Encontrados .....	
Sugerencias para Mejoras .....	
Resultados del Piloto y Recomendaciones .....	

## Instalación

### *Analisis Post Proyecto*

Resumen.....	.....
Objetivos.....	.....
Planeación.....	.....
Logros.....	.....
Retos.....	.....
Lecciones Aprendidas.....	.....
Recursos.....	.....
Logros.....	.....
Retos.....	.....
Lecciones Aprendidas.....	.....
Administración del Proyecto/Cronogramas.....	.....
Logros.....	.....
Retos.....	.....
Lecciones Aprendidas.....	.....
Desarrollo/Diseño/Especificaciones.....	.....
Logros.....	.....
Retos.....	.....
Lecciones Aprendidas.....	.....
Prueba.....	.....
Logros.....	.....
Retos.....	.....
Lecciones Aprendidas.....	.....
Comunicación.....	.....
Logros.....	.....
Retos.....	.....
Lecciones Aprendidas.....	.....
Equipo/Organización.....	.....
Logros.....	.....
Retos.....	.....
Lecciones aprendidas.....	.....
Solución.....	.....
Logros.....	.....
Retos.....	.....
Lecciones Aprendidas.....	.....
Herramientas.....	.....
Logros.....	.....
Retos.....	.....
Lecciones Aprendidas.....	.....
Apéndice: Guía para una reunión exitosa de análisis Post Proyecto.....	.....

## Reporte de Terminación del Proyecto

Resumen del Reporte del Cierre del Proyecto.....	
Entrega de la Visión.....	
Cambios que Impactaron la Visión.....	
Cambios en el Negocio/Organización .....	
Cambios en la visión.....	
Cambios en el Equipo .....	
Cambios en el Proceso del Cliente.....	
Cambios en el Proceso del Proyecto.....	
Cambios en los Planes .....	
Cambios en las Especificaciones .....	
Cambios en las Líneas de Tiempo .....	
Pasos Sigüientes: Visión de Continuar con los Esfuerzos del Proyecto, Sigüiente Versión.....	

## Bibliografía

Cascada, W. -M. (s.f.). *Modelo de Cascada*. Obtenido de [http://en.wikipedia.org/wiki/Waterfall\\_model](http://en.wikipedia.org/wiki/Waterfall_model)

Espiral, W. -M. (s.f.). Obtenido de Wikipedia - Modelo Espiral: [http://en.wikipedia.org/wiki/Spiral\\_model](http://en.wikipedia.org/wiki/Spiral_model)

Kruchten, P. (s.f.). *The Rational Unified Process. An Introduction*.

Mercadeo.com. (s.f.). *Six Sigma*. Obtenido de <http://www.mercadeo.com/archivos/six-sigma.pdf>

Microsoft Corporation - MSF Ciclo de Vida. (2009). 1846A Student Guide - Cap. 4. *Microsoft Solutions Framework* (pág. 22). Guayaquil: Microsoft Corporation.

Microsoft Corporation - MSF Desarrollo. (2009). 1846A Student Guide - Cap. 6. *Microsoft Solutions Framework* (pág. 37). Guayaquil: Microsoft Corporation.

Microsoft Corporation - MSF Estabilización. (2009). 1846A Student Guide - Cap. 8. *Microsoft Solutions Framework* (pág. 25). Guayaquil: Microsoft Corporation.

Microsoft Corporation - MSF Estructura del Equipo. (2009). 1846A Student Guide - Cap.

2. *Microsoft Solutions Framework* (pág. 45). Guayaquil: Microsoft Corporation.

Microsoft Corporation - MSF Instalación. (2009). 1846A Student Guide - Cap. 9. *Microsoft Solutions Framework* (pág. 31). Guayaquil: Microsoft Corporation.

Microsoft Corporation - MSF Manejo de Riesgos. (2009). 1846A Student Guide - Cap. 3. *Microsoft Solutions Framework* (pág. 34). Guayaquil: Microsoft Corporation.

Microsoft Corporation - MSF Planificación. (2009). 1846A Student Guide - Cap. 6. *Microsoft Solutions Framework* (pág. 54). Guayaquil: Microsoft Corporation.

Microsoft Corporation - MSF Visionamiento. (2009). 1846A Student Guide - Cap. 5. *Microsoft Solutions Framework* (pág. 48). Guayaquil: Microsoft Corporation.

Monografías. (07 de 2008). *Modelo CMMI*. Recuperado el 27 de 06 de 2011, de Modelo CMMI: <http://www.monografias.com/trabajos56/modelo-cmmi/modelo-cmmi2.shtml>

RAD, W. . (s.f.). Obtenido de [http://en.wikipedia.org/wiki/Rapid\\_application\\_development](http://en.wikipedia.org/wiki/Rapid_application_development)

Reynoso, C. (04 de 2004). *Métodos Heterodoxos en Desarrollo de Software*. Recuperado el 27 de 06 de 2011, de Métodos Heterodoxos en Desarrollo de Software: [http://docs.google.com/viewer?a=v&q=cache:Ai0yz42t2f8J:carlosreynoso.com.ar/wp-content/plugins/download-monitor/download.php%3Fid%3D156+cascada+espiral+joint+lean+development&hl=es&pid=bl&srcid=ADGEEsJpjCk1pRUor9My3XOu2fXmAq7uJg\\_7\\_Kt1Slg5FzScQcoXq2rbsV7NB](http://docs.google.com/viewer?a=v&q=cache:Ai0yz42t2f8J:carlosreynoso.com.ar/wp-content/plugins/download-monitor/download.php%3Fid%3D156+cascada+espiral+joint+lean+development&hl=es&pid=bl&srcid=ADGEEsJpjCk1pRUor9My3XOu2fXmAq7uJg_7_Kt1Slg5FzScQcoXq2rbsV7NB)

Royce, W. D. (s.f.). *Managing the Development of Large Software Systems*. Recuperado el 21 de 03 de 2011, de Managing the Development of Large Software Systems: <http://www.cs.umd.edu/class/spring2003/cmssc838p/Process/waterfall.pdf>

Wikipedia. (8 de Diciembre de 2010). *Metodología de desarrollo de software*.

Recuperado el 19 de Febrero de 2011, de Metodología de desarrollo de software:

[http://es.wikipedia.org/wiki/Metodolog%C3%ADa\\_de\\_desarrollo\\_de\\_software](http://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software)

Wikipedia. (12 de Marzo de 2011). *Rapid application development*. Recuperado el 21 de 03 de 2011, de Rapid application development:

[http://en.wikipedia.org/wiki/Rapid\\_application\\_development#cite\\_ref-RAD1\\_1-0](http://en.wikipedia.org/wiki/Rapid_application_development#cite_ref-RAD1_1-0)

WillyDev, J. y. (s.f.). Obtenido de <http://www.willydev.net/insiteCreation/ElFlecha.aspx?q=RUP&start=10>

WillyDev, J. y. (s.f.). Obtenido de <http://www.willydev.net/insiteCreation/ElFlecha.aspx?q=RUP&start=10>

## Glosario de términos

**Microsoft Solutions Framework (MSF)**<sup>®</sup>: Marco de trabajo para Soluciones, creada por Microsoft. Refiérase al término como tal en inglés.

**MOF (Microsoft Operations Framework)**<sup>®</sup>: Marco de trabajo para Operaciones, creada por Microsoft. Refiérase al término como tal en inglés.

**Product Manager**: Administrador del Producto, Gestor del Producto, Gerente del Producto. Quién se encarga de hablar por el Cliente ante el Equipo, y por Equipo ante el Cliente. No confundir este rol con un cargo jerárquico.

**Program Manager**: Administrador del Programa, Gestor del Programa, Gerente del Programa. Quién se encarga de hacer realidad lo transmitido por el Cliente al Equipo por medio de su Product Manager. No confundir este rol con un cargo jerárquico.

**Developer:** Desarrollador. Quien(es) en efecto son los ejecutores o constructores de la solución, ya sea tecnológica o de cualquier índole.

**Tester:** Testeador. Quien(es) se encargan de probar o evaluar la solución para evaluar si cumple con todos los criterios de calidad definidos previamente.

**UserExperience:** Experiencia del Usuario. Quienes se encargan de asegurarse que los usuarios de la solución encuentren una ventaja de productividad en el uso de la solución.

**Release Manager:** Administrador Gestor o Gerente del Lanzamiento. Se encargan de asegurarse que la solución sea fácilmente entregada, puesta en producción y administrada por el área de Operaciones.