



METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

Metodología para Desarrollo de Aplicaciones (MDA).

LUGAR Y FECHA DE EXPEDICION: México, D.F., Marzo de 2006

Índice.

1. Introducción.....	4
a) Elementos Centrales de Desarrollo de Software	4
b) Alcance del proyecto.....	5
c) Aplicación de la metodología a los proyectos de software.	6
d) Ventajas del uso de la metodología.....	6
e) Uso de modelado estandarizado.	6
1. Conceptos básicos del Proceso de Desarrollo de Software.	7
a) Proceso de ingeniería de software.....	7
b) Roles.....	8
<i>Analistas</i>	8
<i>Desarrolladores</i>	10
<i>Gerentes</i>	11
<i>Soporte y Producción</i>	15
<i>Equipo de pruebas</i>	16
<i>Roles adicionales</i>	18
<i>Dueños parciales del proceso (stakeholders)</i>	19
c) Disciplinas	19
d) Mejores prácticas para el desarrollo de software	20
1. Desarrollo iterativo.....	20
2. Administración de requerimientos.....	20
3. Utilización de arquitectura por componentes.....	22
4. Modelado Visual (UML).....	23
5. Verificación continua de la calidad.....	24
6. Administración del cambio.....	26
7. Implementación de las mejores prácticas.....	26
2. Descripción de flujos y actividades a realizarse dentro de la Disciplina de Modelado de Negocios.....	27
a) Propósito	28
b) Artefactos	28
c) Proceso y notación	28
d) Relación con otras disciplinas	29
3. Descripción de flujos y actividades a realizarse dentro de la Disciplina de Requerimientos.	30
a) Propósito	30
b) Artefactos	31
c) Relaciones con otras disciplinas	31
4. Descripción de flujos y actividades a realizarse dentro de la Disciplina de Análisis y Diseño.....	33
a) Propósito	33

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

b) Relaciones con otras disciplinas.....	34
c) Artefactos.....	34
5. Descripción de flujos y actividades a realizarse dentro de la Disciplina de Implementación.	35
a) Propósito:.....	36
b) Relación con otras disciplinas.....	36
6. Descripción de flujos y actividades a realizarse dentro de la Disciplina de Implementación.	36
a) Propósito:	38
a) Relación con otras disciplinas.....	39
7. Descripción de flujos y actividades a realizarse dentro de la Disciplina de Distribución.	39
a) Propósito.....	41
b) Relación con otras disciplinas.....	41
8. Descripción de flujos y actividades a realizarse dentro de la Disciplina de Administración de Cambios.	42
a) Introducción.....	42
b) Propósito.....	43
c) Relación con otras disciplinas.....	44
9. Descripción de flujos y actividades a realizarse dentro de la Disciplina de Administración de Proyectos.	45
d) Introducción:.....	46
e) Propósito.....	46
f) Relación con otras disciplinas.....	46
g) Relación con PMI / PMboK.....	47
11) Descripción de flujos y actividades a realizarse dentro de la Disciplina de Ambiente.....	48
Anexo A: Artefactos de las disciplinas.....	49
Disciplina de Modelado de Negocios.....	49
Disciplina de Requerimientos.....	49
Disciplina de Administración de Proyectos.	50
Disciplina de ambiente.....	50
Disciplina de pruebas.....	50
Disciplina de Administración de versiones.	51

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

1. Introducción.

La presente metodología de Desarrollo de Software se deriva de las prácticas de diseño iterativo y programación basada en objetos. Se busca mediante ella estandarizar el desarrollo de software en la SCT.

a) Elementos Centrales de Desarrollo de Software

- **Mejores prácticas:** Se aplicarán las mejores prácticas de desarrollo de software:
 - Desarrollo iterativo.
 - Administración de requerimientos
 - Utilización de arquitecturas basadas en componentes
 - Modelado visual basado en UML
 - Evaluación continua de la calidad
 - Administración del cambio acorde a principios ITIL.
- **Elementos esenciales de desarrollo de software:**
 - **Visión:** Se desarrollará una visión clara en cada uno de los proyectos de desarrollo de software. Esta se constituye de requerimientos de alto nivel y restricciones al alcance del proyecto, y estará íntimamente relacionada al caso de negocio. Contestará las siguientes preguntas:
 - Cuales son los términos clave? (Glosario)
 - Qué problema estamos solucionando? (Enunciado del problema)
 - Quienes son los dueños? Quienes los usuarios? Que requieren de nosotros?
 - Que características tiene el producto?
 - Cuales son los casos de uso?
 - Cuales son los requerimientos no funcionales?
 - Existe alguna restricción?
 - **Plan:** La disciplina de Administración de proyectos se encargará de resolver las siguientes cuestiones:
 - Concebir el proyecto.
 - Evaluar alcance y riesgos.
 - Monitorear y controlar el proyecto.
 - Planear y evaluar cada iteración y fase.
 - **Riesgos:** Se encarga de minimizar riesgos y rastrear problemas sin resolver, mediante listas actualizadas de los mismos. Se encargará de retroalimentar al Administrador de Proyectos para la creación de planes alternos.
 - **Caso de negocios:** Provee la información necesaria para determinar si es conveniente invertir en el proyecto, creando un

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- plan presupuestal para realizar el proyecto establecido por la Visión. Su objetivo es desarrollar la justificación para el proyecto.
- **Arquitectura:** La arquitectura de un servicio de software es la organización en estructura de los componentes significativos, los cuales interactúan a través de interfaces, los cuales se construyen en base a componentes e interfaces más básicas. Las preguntas a responderse son: Cuales son las piezas principales? Cómo se ensamblan adecuadamente?
 - **Prototipos:** La creación de prototipos involucra el desarrollo iterativo para la construcción, pruebas y ejecución de versiones del producto, detectando problemas lo antes posible.
 - **Evaluación:** Se requiere comunicación constante, con datos objetivos. Esto, par ala resolución de asuntos administrativos y técnicos durante la vida del proyecto.
- El lenguaje de definición de procesos: La base de los dos componentes anteriores es un meta-modelo de procesos. Este provee un lenguaje para definir elementos del proceso necesarios durante la ingeniería de software. Este lenguaje se basa en UML y su extensión SPEM. Los tres elementos base de la metodología son:
 - Rol: Un mecanismo de agrupamiento que define responsabilidades en términos de actividades a realizarse. Puede ser cubierto por un individuo o un grupo de personas.
 - Actividad: Es una unidad de trabajo desarrollada por un rol, definida por pasos, artefactos de entrada y de salida. Su objetivo es actualizar o crear artefactos.
 - Artefactos: Productos de un proyecto de desarrollo de software; consisten en entradas y salidas para las actividades. Se definen por tipos, identificados por estereotipos básicos:
 - Modelos.
 - Elementos de modelado.
 - Documentos.
 - Documentos de especificaciones.
 - Almacén de datos.
 - Documento de Planeación.
 - Documento de Evaluación.
 - Ejecutables.
 - Infraestructura.
 - Genéricos.

b) Alcance del proyecto.

La metodología abarca a todas aquellas personas que desarrollen software dentro de la SCT. Específicamente, deberá abarcar:

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- Al personal de desarrollo de software que trabaje como parte de un equipo de proyectos, incluyendo a los propietarios de los desarrollos.
- Personal de ingeniería de software: Ingeniería de Procesos y Gerentes.

c) Aplicación de la metodología a los proyectos de software.

La metodología de desarrollo de software propuesta es iterativa y funciona gracias al desarrollo incremental de productos. Esta práctica es conveniente cuando se considera que la metodología ha sido desarrollada tomando en cuenta las necesidades de la institución.

d) Ventajas del uso de la metodología.

El ambiente de desarrollo establecido ofrece ventajas basadas en los estándares utilizados y en el ambiente configurable de desarrollo de procesos, las cuales permiten:

- Publicar un proceso de ingeniería a la medida del proyecto y accesible a todo el equipo.
- A los procesos ingenieriles ser desarrollados en función de las necesidades del proyecto.
- Provee a cada usuario de filtros a usarse para eliminar la información redundante.

e) Uso de modelado estandarizado.

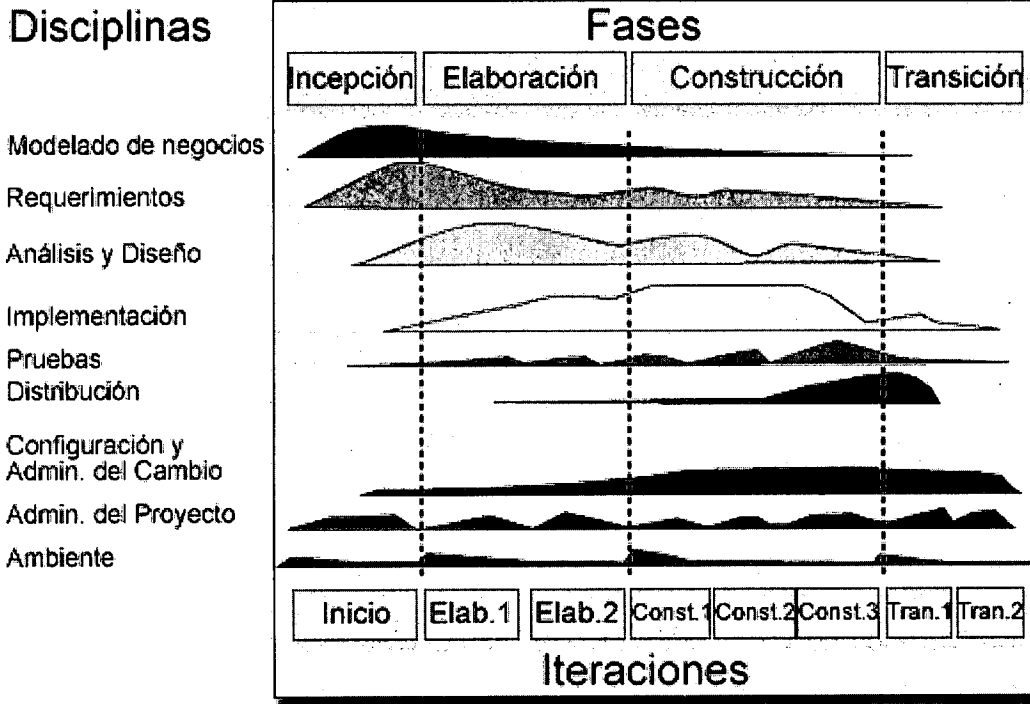
Para lograr proyectos exitosos, es necesario empoderar a los miembros de el equipo de desarrollo a comunicarse con una voz común. A menos de que todos los equipos de trabajo cuenten con un vocabulario estándar, será imposible unificar las actividades del equipo.

La función del Leguaje Unificado para el Modelado (UML) es precisamente esta. Consiste en un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema basado en software. UML es el lenguaje estándar para describir sistemas en un acercamiento. Existen 5 representaciones para desarrollo de software:

- Analistas y diseñadores: Vista lógica de la estructura.
- Integradores de sistemas: Vista de procesos (desempeño, escalabilidad, capacidades).
- Programadores: Vista de implementación (administración del software).
- Ingeniería de sistemas: Vista de distribución (topología de Sistemas, Entrega, Instalación, Comunicaciones).
- Usuario final: Vista de casos de uso (funcionalidad).

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

1. Conceptos básicos del Proceso de Desarrollo de Software.



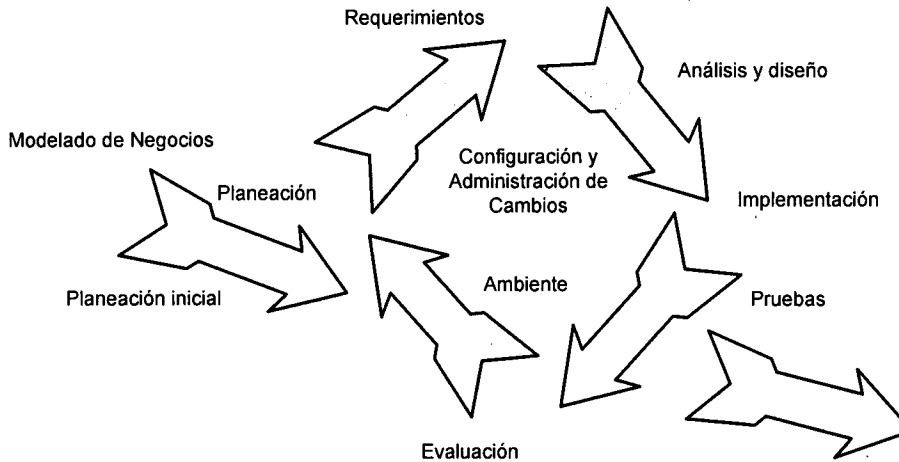
Las disciplinas serán llevadas a cabo por Roles. Un rol no representa a una persona – pueden ser una, más, o una persona puede llevar a cabo varios roles.

a) Proceso de ingeniería de software.

El proceso de ingeniería de software de la SCT persigue la realización de un conjunto de pasos para lograr un resultado. Se persigue la construcción de un producto de software o la mejora de otro ya creado.

Para lograr este fin, se han creado las disciplinas que definen los flujos de trabajo entre elementos de proceso.

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)



En términos de modelado de negocios, el proceso de desarrollo es un proceso de negocios. Se describe un conjunto de procesos de ingeniería de software que comparten una estructura y arquitectura común.

El proceso de ingeniería deberá asignar tareas y responsabilidades al personal indicado (los roles).

b) Roles.

Los Roles son los perfiles que adoptará cada uno de los miembros del grupo de trabajo durante el plan de trabajo. Una persona o un grupo de personas pueden tomar un rol. Una persona puede llevar a cabo más de un rol. Si un proyecto no requiere un rol, éste se desecha.

Los siguientes son los roles que se utilizarán a lo largo de las actividades. Cabe mencionar que no necesariamente todos los proyectos requieren todos los roles aquí mencionados:

Analistas.

Los analistas son aquellas personas que investigan y recopilan requerimientos para las demás disciplinas.

- Analista de Procesos de Negocio.
 - Persona responsable de la definición de la arquitectura del negocio, casos de uso, actores y su interacción.
 - El analista de procesos de negocio se encargará de crear los siguientes artefactos:
 - Visión del negocio
 - Documento de Arquitectura del Negocio
 - Glosario del negocio.

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- Sus funciones serán:
 - Identificar las metas del negocio
 - Crear el glosario de términos comunes
 - Definir la arquitectura del negocio
 - Encontrar los actores de negocio y casos de uso
 - Evaluar la organización
 - Establecer objetivos
 - Mantener operativas las reglas del negocio
 - Estructurar el modelo de casos de uso.
- Diseñador del negocio.
 - Detalla las especificaciones de parte de una organización.
 - Sus funciones son las siguientes:
 - Detallar un caso de negocio en particular
 - Determinar los trabajadores y entidades de negocio
 - Detallar a los trabajadores requeridos
 - Detallar entidades de negocio
 - Definir requerimientos de automatización.
- Analista de Sistemas
 - Coordina los requerimientos y el modelado de casos de uso, describiendo funcionalidad y limitando el alcance del proyecto; determina los actores que existen y que casos de uso requieren al interactuar con el sistema.
 - Se encargará de crear los siguientes artefactos:
 - Plan de Administración de Requerimientos.
 - Visión.
 - Especificaciones.
 - Glosario
 - Solicitudes de los dueños del negocio.
 - Sus funciones serán:
 - Desarrollar un plan de requerimientos
 - Desarrollar la visión
 - Involucrar a los dueños del negocio
 - Administrar dependencias
 - Crear un vocabulario de uso común
 - Definir a los actores y casos de uso
 - Estructurar el modelo de casos de uso.
- Analista de Requerimientos
 - Especifica y mantiene los requerimientos del sistema a detalle.
 - Se encargará del artefacto:
 - Requerimientos de software
 - Sus funciones serán las siguientes:
 - Detallar casos de uso
 - Detallar requerimientos de software.

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

Desarrolladores.

Los desarrolladores organizan los roles que están involucrados en el diseño e implementación de software.

- Arquitecto de Software
 - Es responsable de la arquitectura del sistema, la cual incluye las decisiones técnicas que limitan el diseño e implementación del proyecto.
 - Se encargará de crear los siguientes artefactos:
 - Documento de arquitectura de software
 - Arquitectura de referencia
 - Formará parte de desarrollo de los siguientes modelos.
 - Modelo de distribución
 - Modelo de análisis
 - Modelo de implementación
 - Sus funciones serán:
 - Priorización de casos de uso
 - Análisis de la arquitectura
 - Construcción de la prueba de concepto
 - Establecer viabilidad de la prueba de conceptos
 - Identificar mecanismos de diseño
 - Estructurar el modelo de implementación
 - Incorporar elementos de diseño existentes
 - Describir la distribución
 - Identificar los diseños de elementos.
 - Describir la arquitectura final.
- Diseñador
 - Es responsable del diseño de parte del sistema, limitado por los requerimientos, arquitectura y proceso de desarrollo del proyecto.
 - Sus funciones son:
 - Análisis del caso de uso
 - Diseño del caso de uso
 - Diseño de subsistemas
 - Diseño de clases
 - Diseño de elementos de pruebas
- Diseñador de la interfaz de usuario.
 - Coordina el diseño de la interfaz de usuario, investiga requerimientos de usabilidad y crea los prototipos necesarios.
 - Sus funciones son:
 - Diseñar la interfaz de usuario
 - Realizar prototipos
- Diseñador del encapsulado

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- Su única función es asegurarse que el sistema tendrá el desempeño adecuado una vez que los niveles de concurrencia planeados se alcancen.
- Diseñador de Base de Datos
 - Es responsable de diseñar el modelo de almacenamiento persistente de datos que el sistema utilizará.
- Implementador
 - Es responsable de desarrollar y probar los componentes de acuerdo a los estándares adoptados de proyectos, para su integración en sistemas. Cuando elementos de pruebas son creados para el soporte a pruebas, el implementador también debe desarrollar y probar los componentes y subsistemas.
 - Sus funciones son:
 - Implementar las pruebas de desarrollo.
 - Ejecutar las pruebas.
 - Analizar el comportamiento en tiempo de ejecución
 - Implementar elementos de diseño
 - Implementar elementos para pruebas
 - Desarrollar artefactos para la instalación
 - Los artefactos que desarrolla son:
 - Artefactos de instalación
- Integrador
 - Responsables de planear la integración del proyecto, y de desarrollar la integración de los elementos implementados, creando versiones para prueba (Builds).
 - Artefactos:
 - Plan de creación de versiones de prueba
 - Sus funciones son:
 - Creación de espacios de trabajo de integración.
 - Planear la integración de los subsistemas
 - Planear la integración de los sistemas
 - Crear las líneas bases
 - Integrar los subsistemas
 - Integrar los sistemas
 - Promover líneas bases
- Roles adicionales
 - Revisor
 - Coordinador de revisiones
 - Revisor técnico
 - Dueño del proceso

Gerentes

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

Los gerentes se encargan de administrar y configurar el proceso de ingeniería de software.

- Administrador de Proyectos
 - El administrador del proyecto planea, administra y asigna recursos, ordena prioridades, coordina instrucciones con los clientes y mantiene el proyecto enfocado. El Administrador del Proyecto establece un juego de prácticas que aseguran la integridad y calidad de los artefactos del proyecto.
 - Es responsable de los siguientes artefactos:
 - Evaluación de la iteración.
 - Plan de iteraciones.
 - Caso de negocios.
 - Plan de aseguramiento de la calidad.
 - Plan de desarrollo de software
 - Plan de métricas
 - Plan de administración de riesgos
 - Lista de riesgos
 - Plan de resolución de problemas
 - Plan de aceptación de productos
 - Métricas de proyecto
 - Evaluación del estatus
 - Ordenes de trabajo
 - Sus funciones son las siguientes:
 - Inicio del proyecto
 - Desarrollo del caso de negocios
 - Identificar riesgos
 - Iniciar el proyecto
 - Planeación y desarrollo
 - Definir, monitorear y controlar procesos
 - Planear fases e iteraciones
 - Compilar el plan de desarrollo de software
 - Definir el personal y la organización del proyecto
 - Principio y fin de iteraciones
 - Desarrollar el plan de iteraciones
 - Conseguir el personal requerido
 - Iniciar la iteración
 - Evaluar la iteración
 - Preparar los fines de las fases
 - Preparar el fin del proyecto
 - Administración Operativa
 - Monitorear el estatus del proyecto
 - Asignar y priorizar los trabajos
 - Reportar el estatus del proyecto

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- Manejar excepciones y problemas (Producto no conforme)
 - Otros planes
 - Desarrollar un plan de aceptación de productos
 - Desarrollar un plan de aseguramiento de calidad
 - Desarrollar un plan de solución de problemas
 - Desarrollar un plan de administración de riesgos
 - Desarrollar un plan de mediciones
- Administrador del Control de Cambios
 - El administrador del Control de Cambios supervisa que el proceso de control de cambios funcione de manera adecuada.
 - Es responsable de los siguientes artefactos:
 - Solicitud de cambios
 - Sus funciones son:
 - Establecer el proceso de control de cambios
 - Revisar la solicitud del control de cambios
 - Confirmar una solicitud de cambios duplicada o rechazada
- Administrador de configuraciones
 - El rol de Administración de Configuraciones es responsable para proveer la infraestructura para la Administración de Configuraciones a los miembros del equipo de desarrollo y producción.
 - Sus artefactos son:
 - Plan de Administración de configuraciones
 - Resultados de la auditoría de configuraciones
 - Sus funciones son:
 - Establecer políticas de Administración de la Configuración
 - Realizar auditoría de la configuración
 - Configuración del ambiente de administración de configuraciones
 - Documentación del plan de administración de configuraciones
 - Reportar en el estatus de la configuración
 - Crear la unidad de trabajo
- Administrador de pruebas
 - El rol del administrador de pruebas tiene la responsabilidad del éxito del ambiente de pruebas. El rol requiere calidad y dedicación, planeación de recursos y administración, y la solución de problemas que impiden el esfuerzo de las pruebas.
 - Sus funciones son:
 - Aceptar la misión
 - Identificar los motivadores de las pruebas

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- Obtener un compromiso para las pruebas
 - Evaluar y promover la calidad
 - Evaluar y mejorar el esfuerzo de pruebas
- Sus artefactos son:
 - Plan de pruebas
 - Sumario de la evaluación de pruebas
- Administrador de implantación
 - Es el responsable de la planeación de la transición de producto la comunidad de los usuarios, asegurándose que los planes son realizados de manera apropiada, administrando actividades y monitoreando el progreso.
 - Sus actividades son:
 - Desarrollar el plan de implantación
 - Administrar las pruebas de aceptación
 - Proveer acceso al sitio de descarga
 - Verificar el producto creado
 - Definir los insumos requeridos
 - Administrar las pruebas beta
 - Liberar a manufactura
 - Documentar las notas de release.
 - Sus artefactos son:
 - Plan de implantación
 - Resumen de insumos
 - Notas de liberación
 - Producto final
- Ingeniero de procesos.
 - Juegan un papel importante en la administración de un proyecto de software, ya que soporta a los demás roles en RUP. Le da las herramientas al equipo de manera eficiente y veloz, y se asegura que los miembros no tienen problemas para realizar sus labores.
 - Sus funciones son:
 - Adecuar el proceso al negocio
 - Desarrollar el caso
 - Preparar las plantillas de los artefactos
 - Preparar guías para el proyecto
 - Lanzar el proceso
 - Sus artefactos son:
 - Análisis de la organización
 - Caso de desarrollo
 - Plantillas específicas
 - Guías específicas por proyecto
- Gerente de revisiones

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- Es responsable de evaluar la planeación del proyecto y los artefactos derivados de la evaluación del proyecto en los puntos definidos para revisión durante el ciclo de vida del producto.
- Sus funciones son:
 - Evaluación de la iteración
 - Aceptación de la iteración
 - Revisión del ciclo de vida
 - Revisión de la aceptación del proyecto
 - Revisión de Planeación del Proyecto
 - Revisión del plan de iteración
- Sus funciones son
 - Revisión de registros
- Roles adicionales
 - Revisor
 - Coordinador de revisiones
 - Dueño del proyecto

Soporte y Producción

Estos roles no están relacionados directamente con la definición, administración, desarrollo o pruebas de software. Sin embargo, su apoyo se requiere para el proceso – ya sea durante el proyecto, o para proveer materiales adicionales relacionados con el producto final.

- Documentador técnico
 - Es responsable de proveer a los usuarios con material como guías, archivos de ayuda, etc.
 - Sus funciones son:
 - Crear el material de soporte
 - Desarrollar el manual de estilos
 - Artefactos:
 - Material de soporte al usuario final
 - Manual de guías de estilo
- Administrador de Sistemas
 - Mantiene el ambiente de desarrollo, en hardware, software, administración de sistemas, respaldos, etc.
 - Sus funciones son:
 - Soporte al desarrollo
 - Responsable de la infraestructura

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- Especialista en herramientas
 - Es el responsable de soportar las herramientas utilizadas durante el proyecto. Incluye seleccionar y adquirir herramientas. También realiza la configuración y verifica que las mismas funcionen.
 - Sus funciones son:
 - Seleccionar y adquirir herramientas
 - Configurar herramientas
 - Verificar e instalar herramientas

- Desarrollador de cursos
 - Es el responsable del desarrollo de material de entrenamiento que permita a los usuarios entender el uso del producto.
 - Sus funciones son:
 - Crear los materiales de entrenamiento
 - Es responsable de:
 - Los materiales de entrenamiento

- Diseñador gráfico
 - Produce la imagen gráfica del proyecto final
 - Sus funciones son:
 - Crear la imagen gráfica
 - Su artefacto es:
 - Imagen gráfica final

Equipo de pruebas.

El equipo de pruebas organiza los roles que tratan con el desarrollo de la evaluación del producto final. Existen roles adicionales en la disciplina que extienden los requerimientos de otros roles. Estos roles se encuentran en las descripciones anteriores.

- Evaluador.
 - Sus funciones varían dependiendo del proyecto y las habilidades requeridas para el rol.
 - Las siguientes habilidades son requeridas.
 - Conocimiento de técnicas para pruebas.
 - Habilidades para solución y diagnóstico de problemas
 - Conocimiento del sistema o aplicación evaluados
 - Conocimiento de redes y arquitectura de sistemas
 - Si se requieren pruebas automatizadas:
 - Entrenamiento en el uso de la herramienta.
 - Experiencia en el uso de herramientas automatizadas
 - Habilidades de programación
 - Habilidades de diagnóstico

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- Sus responsabilidades primarias son:
 - Identificar la implementación adecuada para las pruebas.
 - Implementar pruebas individuales
 - Configurar y ejecutar las pruebas
 - Verificar los resultados de las pruebas
 - Analizar y recuperar ambientes de errores de ejecución.
- El rol se asigna de dos formas:
 - Se asigna a uno o más miembros del personal para que desempeñen roles de analista y evaluador. Esta es una forma estándar de operar, buena para equipos pequeños de trabajo.
 - Asignar uno o más personas al rol de evaluador. Esto funciona bien en equipos grandes y es útil para separar responsabilidades si algunos miembros tienen más experiencia que otros.
- Analista de evaluaciones.
 - Es responsable de identificar y definir las pruebas requeridas, de monitorear el progreso de pruebas y resultados, y de evaluar la calidad del producto. Usualmente tiene la responsabilidad de representar los intereses de los dueños del proyecto que no tienen representación directa en el proceso.
 - Sus actividades son:
 - Identificar los objetivos de las pruebas
 - Identificar ideas para pruebas
 - Definir detalles de pruebas
 - Definir los requerimientos de los dueños del negocio
 - Determinar los resultados de las pruebas
 - Verificar los cambios.
 - Los artefactos generados son:
 - Plan de pruebas
 - Resumen de evaluación de pruebas
 - Solicitudes de cambio
 - Guías para pruebas
 - Ideas para pruebas
 - Resultados de pruebas.
- Diseñador de pruebas.
 - El diseñador de pruebas define el enfoque de las pruebas y es responsable de su implementación. El rol requiere identificar las técnicas, herramientas y guías adecuadas para implementar las pruebas requeridas, y el proponer guías para los requerimientos del plan de pruebas.

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- Sus actividades son:
 - Definir las pruebas a realizarse.
 - Definir las configuraciones del ambiente.
 - Identificar mecanismos de prueba
 - Estructurar la implementación de pruebas
 - Definir elementos de pruebas.
- Genera los siguientes artefactos.
 - Plan de pruebas
 - Arquitectura de automatización de pruebas
 - Especificación de interfaces de pruebas
 - Diseño y estrategia de pruebas

Roles adicionales.

Se encuentran separados de los otros por no tener cabida en las descripciones anteriores.

- Revisor.
 - El revisor provee retroalimentación a los miembros del equipo acerca de sus artefactos.
 - Descripción: Cualquier miembro puede ser revisor, siempre y cuando tenga los conocimientos requeridos.
- Coordinador de revisiones.
 - Es responsable de realizar revisiones e inspecciones formales, y asegurarse de que al momento que se realicen tengan como resultado un estándar satisfactorio.
 - Descripción: Administra el proceso de revisión de dos tipos: administrativas y técnicas. Se asegura que:
 - Las revisiones correspondientes se realicen.
 - Se incluyan al personal correspondiente.
 - Las revisiones se realicen de manera eficiente.
 - Las actividades que resulten de la revisión se administren correctamente.
- Revisor técnico.
 - Es responsable de proveer retroalimentación al proceso de revisión.
 - Descripción: Se debe considerar tanto las habilidades requeridas para el rol como los acercamientos que se pueden tomar al asignar estos proyectos a un rol en particular.
 - Debe:
 - Tener la capacidad de producir el artefacto revisado
 - Tomar responsabilidad por otros artefactos, los cuales dependen del revisado.

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- Responsabilizarse de actividades siguientes en las cuales se requiere el artefacto.

Dueños parciales del proceso (stakeholders)

Son los responsables de representar los intereses de un grupo que deben quedar cubiertos una vez que el proyecto termine. El rol se realiza por cualquiera que sea afectado por el proyecto.

c) Disciplinas.

Las disciplinas son un conjunto de actividades relacionadas a un "área delicada" dentro del proyecto. El agrupamiento de actividades en disciplinas ayuda a entender el proyecto desde una perspectiva tradicional de tipo "cascada". Típicamente, por ejemplo, es más común realizar ciertas actividades de requerimientos en coordinación con otras de análisis y diseño.

Las disciplinas del proyecto son:

- Modelado de Negocios
- Requerimientos
- Análisis y diseño
- Implementación
- Pruebas
- Despliegue
- Configuración y Administración de Cambios
- Administración de Proyectos (como parte de PMI)
- Ambiente

El flujo de trabajo de las disciplinas es una secuencia estructurada de actividades que se realizan para lograr un resultado en particular. Sin embargo, no se puede considerar que esta secuencia representa la realidad del trabajo cotidiano, pues no representan la opcionalidad de las actividades o la naturaleza iterativa de proyectos reales. Su valor está en la manera en que nos ayudan a comprender el proceso al dividirlo en "áreas de interés".

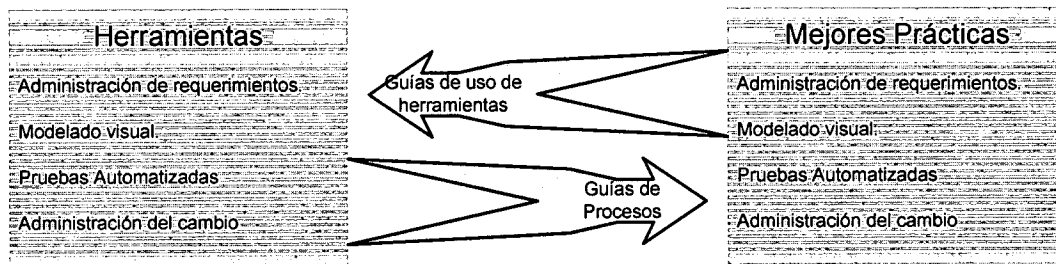
Cada área está asociada con uno o más modelos. Estos se componen de artefactos asociados. Los artefactos más importantes son los modelos que genera cada disciplina:

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

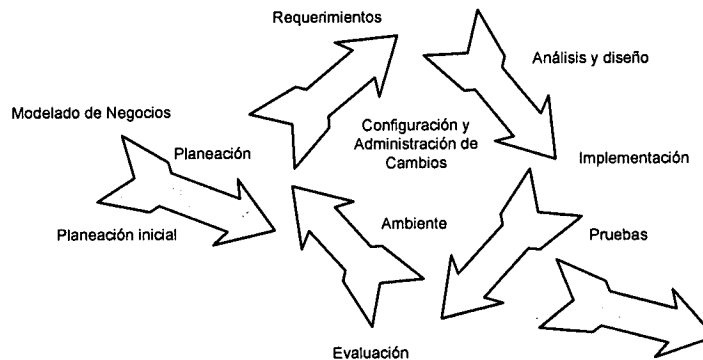
- Modelo de casos de uso.
- Modelo del diseño.
- Modelo de implementación.
- Conjunto de pruebas.

d) Mejores prácticas para el desarrollo de software.

En el desarrollo de software institucional deberán aplicarse las siguientes mejores prácticas:



1. Desarrollo Iterativo.



El desarrollo iterativo es la base del proceso de ingeniería de software para la SCT.

2. Administración de requerimientos.

Consiste en un acercamiento sistémico para encontrar, documentar, organizar y mantener control sobre los requerimientos cambiantes de cualquier proyecto.

Considerando que un requerimiento consiste en: "una condición o capacidad que el sistema debe cumplir", se define la Administración de Requerimientos como:

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- Organizar, solicitar y documentar requerimientos del sistema.
- Establecer y mantener un acuerdo entre el cliente y el equipo de proyectos con relación a los requerimientos cambiantes del sistema.

Las claves para una correcta administración de requerimientos incluyen mantener una política clara de los mismos, junto con los atributos apropiados de otros requerimientos y artefactos del proyecto.

Considerando que los problemas a los que se enfrentan los administradores de proyecto son variados, es imposible declarar que la recolección de requerimientos es una tarea sencilla:

- Los requerimientos no son siempre obvios, pues proveen de fuentes distintas.
- Los requerimientos no son fácilmente descritos por escrito.
- Existen muchos tipos de requerimiento que requieren distintos tipos de detalle.
- El número de requerimientos puede fácilmente ser excesivamente grande si se pierde el control.
- Los requerimientos se relacionan unos con otros y con entregables del proceso de ingeniería de software.
- Los requerimientos tienen propiedades y valores únicos; no son necesariamente igual de importantes o fáciles de cumplir.
- Los requerimientos del personal interesado son variados.
- Los requerimientos suelen cambiar con el tiempo.

Existirá siempre el riesgo de cambio durante la administración de requerimientos. La complejidad está en que un requerimiento que ha cambiado implica que se deberá dedicar tiempo para implementar una nueva característica, y que esta seguramente impactará en otros requerimientos.

La administración de requerimientos implica:

- Establecer una línea base.
- Determinar las dependencias a rastrear.
- Establecer parámetros de rastreo entre elementos relacionados.
- Implementar el control de cambios.

Un método recomendado para organizar los requerimientos funcionales es el utilizar casos de uso. En lugar de una lista de requerimientos, pueden ordenarse de manera tal que establezcan una historia del uso del sistema. Esto ayuda a completar el panorama, y provee un mejor entendimiento de los requerimientos por parte del usuario.

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

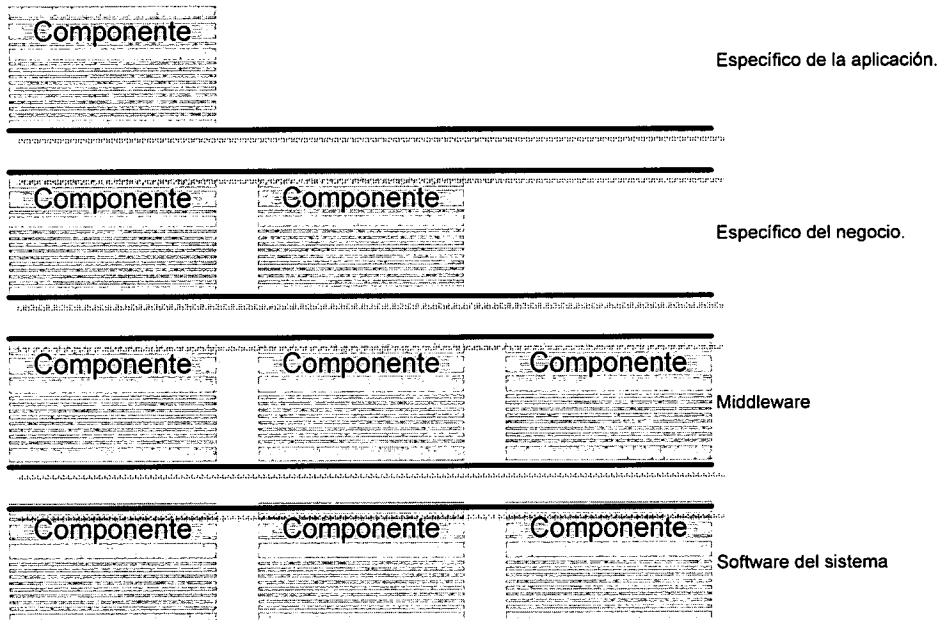
Desde una perspectiva tradicional, es difícil determinar como funciona un sistema de la forma en que lo hace. La dificultad proviene por la falta de un “hilo negro” en el sistema. A falta de este “hilo”, se utilizan los casos de uso.

Los casos de uso se utilizan en las siguientes situaciones:

- En procesos de negocio; se les llama “caso de uso de negocio” y se cubre por la disciplina de modelado de negocios.
- En requerimientos de software: se cubren en la disciplina de requerimientos.
- En la disciplina de Administración de Proyectos se utilizan como base para planear el desarrollo iterativo.
- Forman parte del modelo de diseño como parte de la disciplina de Análisis y Diseño.
- Se implementan en escenarios de prueba por la disciplina de Implementación y pruebas.
- En la disciplina de Distribución, los casos de uso forman la columna vertebral del manual de uso.

3. Utilización de arquitectura por componentes.

La utilización de arquitectura por componentes simula una construcción por capas:



Una arquitectura basada en componentes constituye en un desarrollo basado en componentes reutilizables. Al componerse de elementos independientes,

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

reutilizables, y modulares, ayudan a disminuir la complejidad y a ser reutilizados.

Al basar el desarrollo de software en casos de uso, las actividades de diseño deberán centrarse en la noción de arquitectura de sistemas, y de software del sistema. El enfoque principal de las iteraciones primarias es el de producir y validar una arquitectura de software, que al principio toma la forma de un ejecutable. Este termina correspondiendo al desarrollo final.

La arquitectura ejecutable implica una implementación parcial del sistema, construida para demostrar algunas funciones y propiedades. El propósito es el de mitigar los riesgos de desempeño, capacidad, confiabilidad, etc., para que esta sea implementada durante la fase de construcción.

El diseño arquitectónico es importante por las siguientes razones:

- Permite adquirir y retener el control intelectual del proyecto, administrando su complejidad y manteniendo su integridad.
 - Un sistema complejo es un conjunto de elementos tácticos, que requiere estructura para organizar estas partes sistémicamente. La arquitectura establece los medios para mejorar la comunicación y para entender el proyecto como un conjunto de referencias.
- Es efectivo para la reutilización a gran escala.
 - Gracias a la identificación de partes comunes permite compartir funcionalidad en sistemas que se encuentran en un dominio común.
- Provee la base de administración de proyectos.

4. Modelado Visual (UML)

El modelado visual es el uso de notaciones gráficas para la captura de diseño de software. Esta permite elevar el nivel de abstracción, manteniendo un estándar de sintaxis y semántica, mejorando la comunicación en el equipo de trabajo y evitando ambigüedad durante la implementación.

El modelo visual es una vista simplificada del sistema, mostrando las partes esenciales de un sistema desde una perspectiva particular, ocultando las secciones no esenciales. Los modelos ayudan de las siguientes maneras:

- Ayudan al entendimiento de sistemas complejos.

La importancia del modelado se incrementa con la complejidad de los sistemas. Algunos ejemplos de modelado correcto son:

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- a) Usar casos de uso para especificar el comportamiento de un sistema.
- b) Capturar el diseño en diagramas de clase y diagramas de modelado de datos.
- c) Diagramas de transición para modelar comportamiento dinámico.

- Facilitan la comparación de alternativas a un costo menor

Es posible construir modelos sencillos a bajo costo para explorar alternativas de diseño. Todo tipo de ideas se capturan y analizan por desarrolladores antes de invertir en desarrollo de código. Al combinarse con desarrollo iterativo, el modelado visual ayuda a los desarrolladores a analizar cambios en el diseño y a comunicar dichos cambios al equipo.

- Forman la base de la implementación

Con las herramientas correctas, un modelo puede ser usado para generar código inicial. Esto es conocido como "generación de código". Los modelos pueden también ser mejorados para incluir la información suficiente para construir el sistema...

- Capturan los requerimientos de manera precisa

El requerimiento fundamental es el de capturar los requerimientos críticos del proceso. Se requiere utilizar un modelado preciso y sin ambigüedad para que todos los involucrados entiendan y acepten estos. Un modelo que separa el comportamiento externo del sistema de la implementación ayuda a usar el sistema de manera adecuada, sin perder tiempo en los detalles de la implementación.

- Comunican las decisiones finales sin ambigüedad

El lenguaje obedece las siguientes reglas:

- Sirve como un lenguaje para comunicar decisiones que no son obvias o que no pueden ser inferidas a partir del código.
- Provee semántica lo suficientemente rica como para capturar las decisiones estratégicas y tácticas.
- Ofrece una forma lo suficientemente concreta para el entendimiento, y para manipulación mediante herramientas.

5. Verificación continua de la calidad.

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

La calidad de los artefactos generados durante el proceso debe ser verificada en varios puntos del proyecto. Los artefactos deben ser evaluados a la vez que las actividades que los generan sean completadas, y al final de cada iteración. Cuando se produce software ejecutable, este debe estar sujeto a pruebas durante las iteraciones, lo cual provee de información suficiente para eliminar defectos en la arquitectura. Esto contrasta con el acercamiento tradicional que deja las pruebas del software integrado hasta el final del ciclo de vida del producto.

La Calidad, como se entiende dentro del proceso, es un conjunto multidimensional de características que debe poseer un producto para ser admisible por el usuario final. La calidad del producto implica construir el producto correcto, mientras que la calidad del proceso implica construir el producto de manera adecuada.

La Calidad se entiende como un proceso cuya responsabilidad es de todos los involucrados, y debe ser parte fundamental de todas las actividades en el proceso, responsabilizando a todos los integrantes de el éxito del producto. Cada rol o disciplina deberá promover los siguientes atributos de calidad:

- Calidad del producto: contribución de calidad de cada artefacto producido.
- Calidad del proceso: el alcance de calidad en las actividades del proceso.

Existe un rol que tiene la responsabilidad de la calidad: supervisa las actividades para asegurarse que la calidad se administra, mide y logra. El rol es el del Administrador de Proyectos.

Deberán evitarse las siguientes afirmaciones acerca de la Calidad:

- La calidad puede ser añadida o “probada” en el producto: la calidad no puede ser alcanzada si no es descrita, medida y parte del proceso de creación del producto.
- La calidad es dimensión única, y significa lo mismo para todos: La calidad implica un gran número de métricas y criterios, los cuales son necesarios para brindar un buen servicio a los dueños del negocio. La calidad se puede evaluar mediante:
 - Progreso
 - Varianza entre métricas establecidas
 - Resistencia a fallas
 - Funcionalidad
 - Desempeño

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- La calidad sucede por sí misma: Para que exista la calidad debe haber un proceso que la soporte. El propósito del proyecto ENSIS es crear el soporte de calidad requerido para el desarrollo de software.

6. Administración del cambio.

Un Administrador de Proyectos debe manejar los siguientes recursos a su disposición:

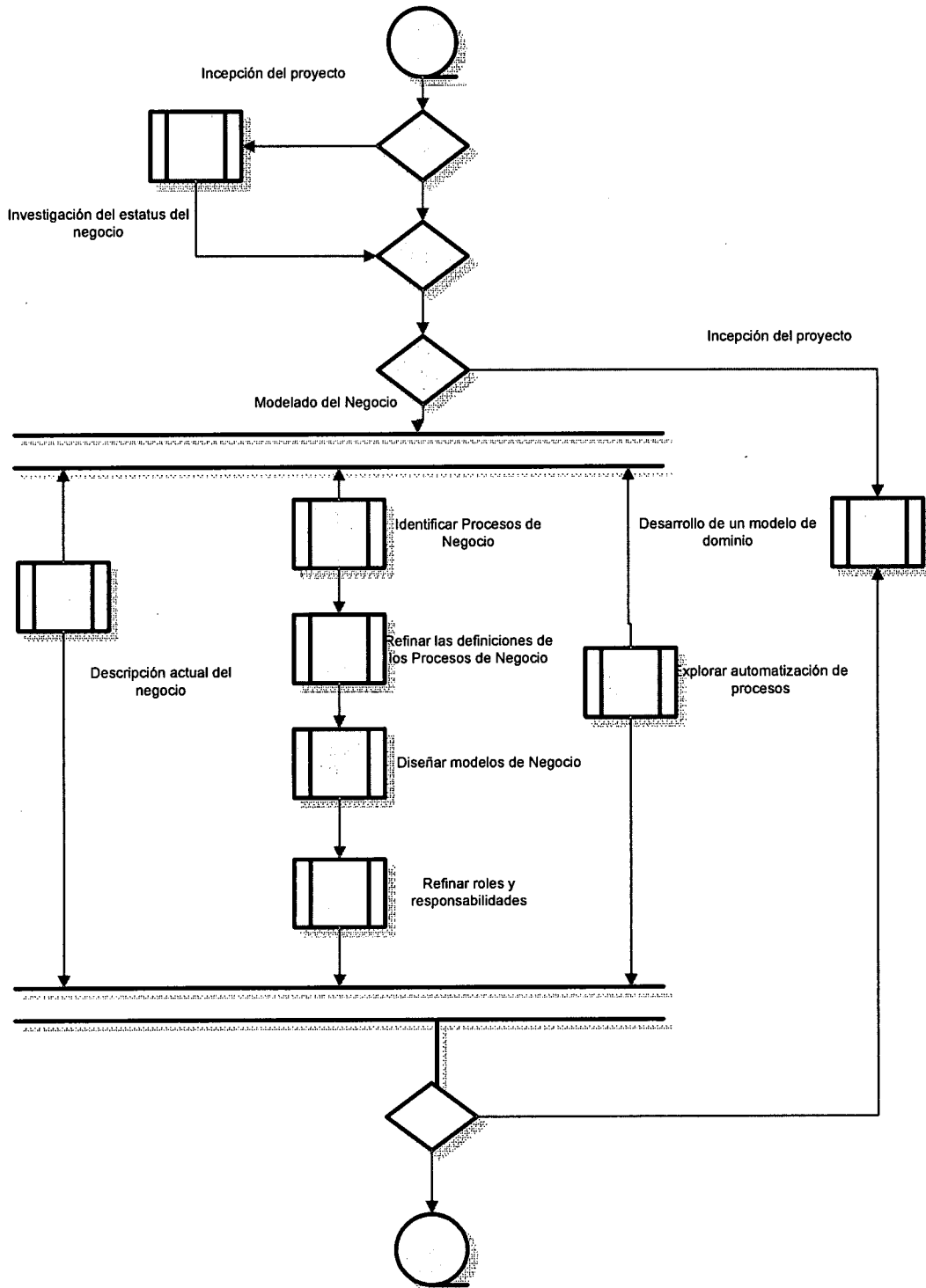
- Desarrolladores
- Equipos de trabajo
- Sitios de trabajo
- Iteraciones de proceso
- Distribuciones beta
- Productos
- Plataformas

La ausencia de control degenerará rápidamente en caos. La Administración del cambio se lleva a cabo mediante la disciplina de Configuración y Administración del Cambio.

7. Implementación de las mejores prácticas.

Las seis actividades mencionadas anteriormente deberán realizarse a lo largo de los proyectos.

2. Descripción de flujos y actividades a realizarse dentro de la Disciplina de Modelado de Negocios.



METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

a) Propósito

El propósito del modelado de negocios es:

- Entender los problemas actuales en la organización para identificar el potencial de mejora.
- Entender el impacto del cambio organizacional.
- Asegurar que clientes, usuarios, desarrolladores y otras partes tienen un entendimiento común de la organización
- Comprender los requerimientos de software para soportar la organización.
- Entender como un sistema de software a ser implementado apoya los procesos.

No es suficiente con contar con un organigrama; es necesaria una visión dinámica del negocio. Los negocios cambian de acuerdo a los factores que los controlan y mantienen sano. Pueden ser por ejemplo factores para reducción de costos, mejora de calidad, o disminución en tiempos.

El mejor soporte a la organización que se puede dar es el proveer información precisa, en tiempo, suficiente y relevante.

b) Artefactos

Los siguientes artefactos se han definido para el Desarrollo de Software, y se encuentran disponibles en el servidor UI-RATIONAL.sct.gob.mx. Asimismo, se presentan al final del documento, junto con sus instructivos de llenado.

- Administración de Requerimientos
- Especificación de Casos de Uso del negocio
- Especificación Casos de Uso del Sistema
- Especificaciones suplementarias al negocio
- Especificaciones suplementarias al sistema
- Evaluación organizacional
- Glosario
- Reglas de negocio
- Planteamiento de la visión del negocio
- Planteamiento de la visión de los sistemas

c) Proceso y notación

La notación para estos casos de negocios se realizan mediante el Lenguaje Unificado para Modelado, o UML. Este se utilizará para los procesos de negocio y tecnológicos.

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

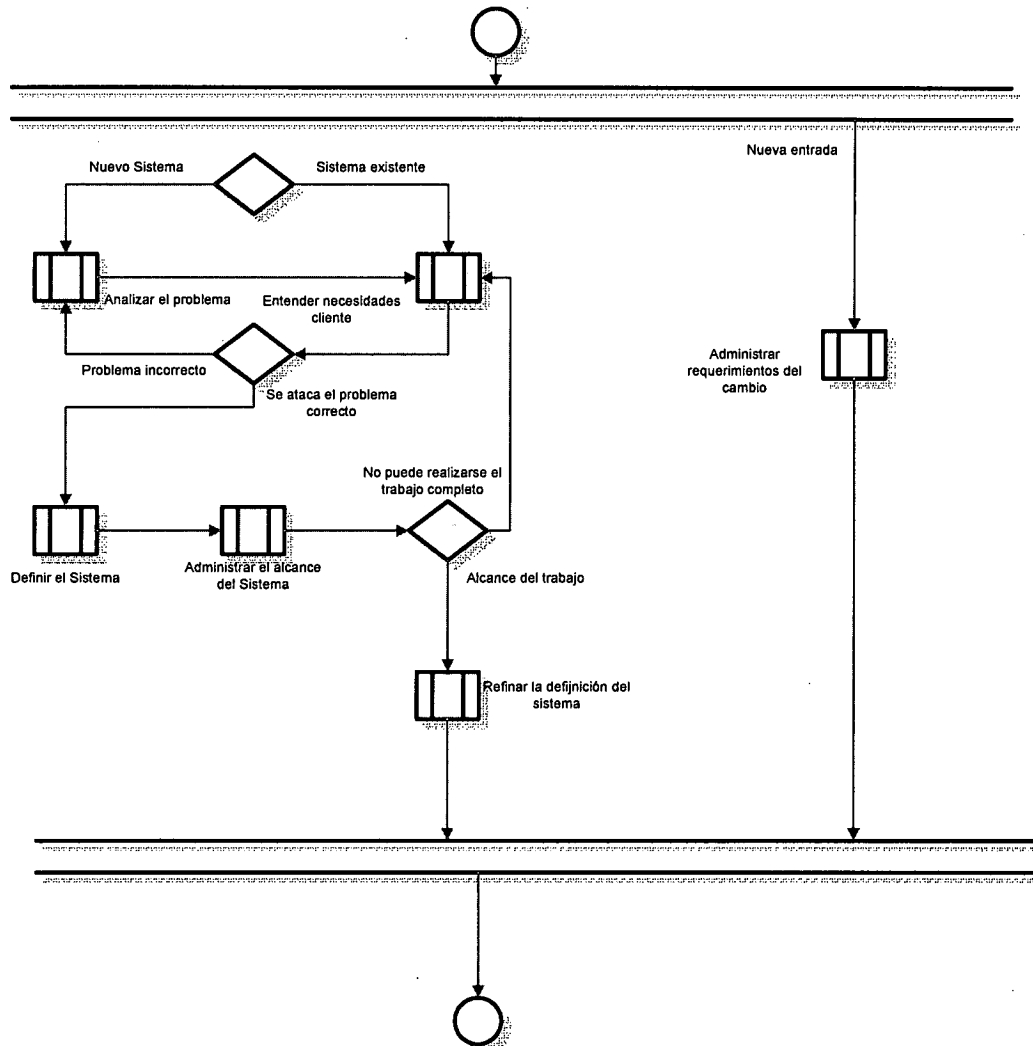
De manera similar, el modelado se llevará a cabo de manera iterativa, comenzando con una visión general y definiendo detalles paso a paso. En cada iteración se realizan los ajustes necesarios, y se verifica el trabajo realizado (factores importantes para la siguiente iteración).

d) Relación con otras disciplinas

La relación con otras disciplinas queda como sigue:

- **Requerimientos:** Utiliza los modelos de negocio como entradas para entender los requerimientos del sistema.
- **Análisis y diseño:** Usa los modelos de negocio como entrada para definir sistemas de software que funcionan adecuadamente en la organización.
- **Distribución:** Usa los modelos de negocio como ayuda para la planeación de la distribución de un sistema de software.
- **Ambiente:** Desarrolla y mantiene artefactos, como las encontradas en el Modelado de Negocios.

3. Descripción de flujos y actividades a realizarse dentro de la Disciplina de Requerimientos.



a) Propósito.

El propósito de la disciplina de requerimientos es:

- Establecer y mantener acuerdos con los clientes y dueños de procesos acerca de las actividades que debe realizar el sistema.
- Proveer a los desarrolladores del sistema con un mejor entendimiento de los requerimientos del sistema.
- Definir los límites del sistema.
- Proveer una base para planificar los contenidos técnicos de la iteración.
- Proveer la base para estimar los costos y el tiempo para desarrollar un sistema.

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- Definir una interfaz para el usuario dentro del sistema, enfocada en las necesidades y metas de los usuarios.

Para lograr estas metas es importante entender la definición y el alcance del problema que se tratará de solucionar con el sistema. Para esto, se requerirá de las siguientes entradas por parte de Modelado de Negocios:

- Reglas de Negocio
- Modelo de casos de Uso
- Análisis del Modelo de Negocio.

Además, se definirán los:

- Dueños
- Peticiones de dueños.

b) Artefactos.

Durante esta fase, de manera general, se crearán los siguientes artefactos:

- Plan de Administración de Requerimientos
- Especificación de Casos de Uso de sistemas
- Especificaciones suplementales para sistemas
- Visión del Sistema

La descripción del sistema se encontrará en la Visión. Para mayor profundidad, los Casos de Uso serán utilizados.

Las especificaciones suplementales se formarán por las peticiones de los dueños del proceso.

El modelo de caso de uso será el medio de comunicación y contacto entre el cliente, usuarios y desarrolladores de la funcionalidad del sistema, lo que permitirá:

- Que los clientes y usuarios validen que el sistema funciona como así lo esperan.
- Que los desarrolladores de sistemas construyan lo que se espera de ellos.

En caso necesario, deberá crearse un glosario con información relevante acerca de la terminología utilizada durante esta disciplina.

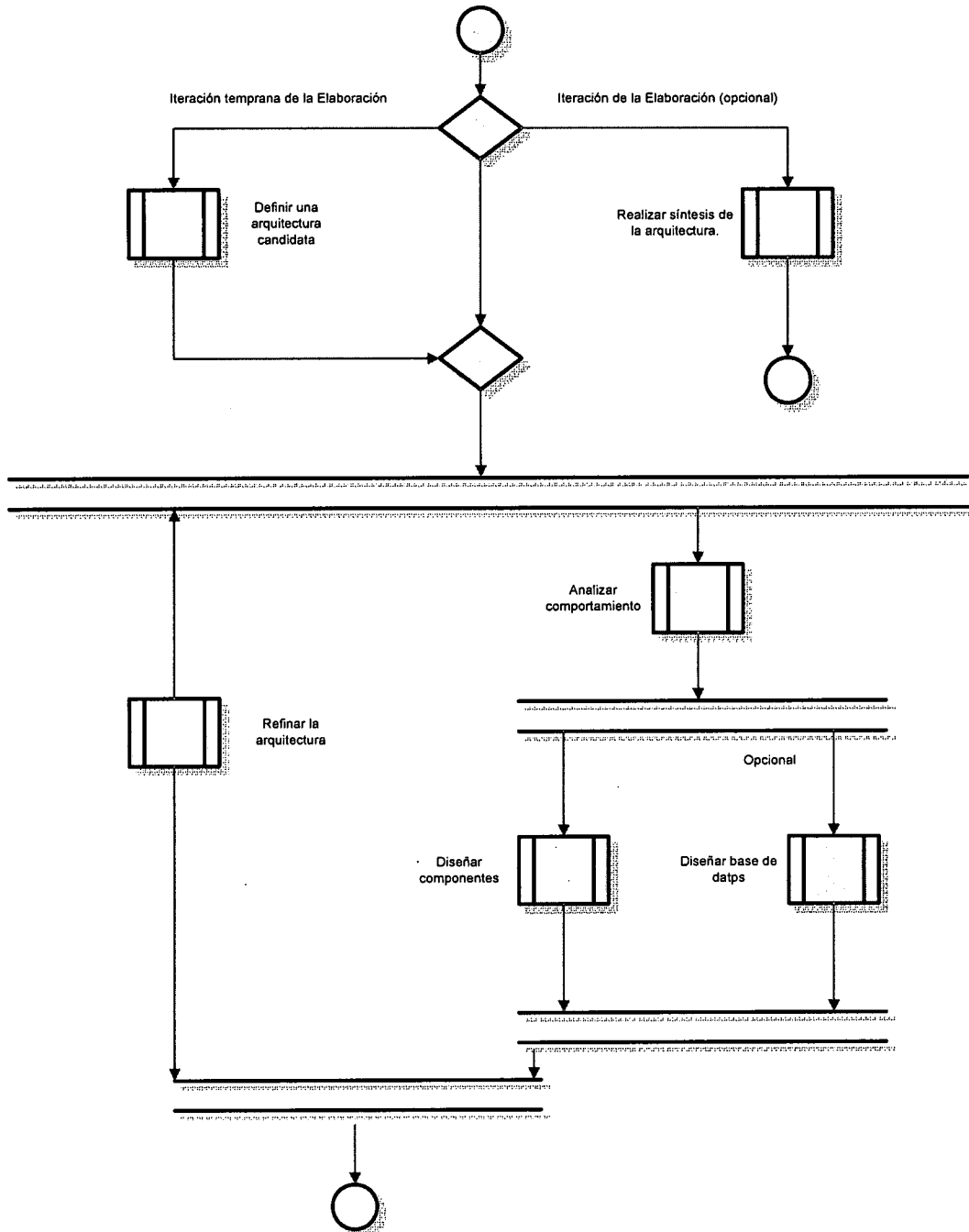
c) Relaciones con otras disciplinas.

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- Modelado de Negocios provee las Reglas de Negocio, el Modelo de Casos de Uso y el Modelo de Análisis de negocio, con un modelo de dominio y un contexto organizacional.
- Análisis y diseño obtiene su entrada (el modelo de casos de uso y Glosario) de Requerimientos. Los errores en el modelo de casos de uso se descubren en el Análisis y Diseño; se generan solicitudes de cambio que son aplicadas.
- Pruebas valida el sistema contra el Modelo de Casos de Uso. Los casos de uso y las especificaciones suplementarias proveen información de requerimientos usados en la definición de la misión de la evaluación, y en las pruebas subsecuentes.
- Configuración y Administración de cambios proveen el mecanismo de control de cambios para los requerimientos. El mecanismo para proponer un cambio es enviarlo como una solicitud de cambio, la cual es revisada por el Comité de Control de Cambios.
- Administración de Proyectos planea el proyecto y cada iteración (descrita en el plan de iteraciones). El modelo de caso de uso y Plan de Administración de Requerimientos son entradas importantes para la planeación de la iteración.
- Ambiente: Desarrolla y mantiene los artefactos de soporte utilizados durante la administración de requerimientos y modelado de casos de uso.

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

4. Descripción de flujos y actividades a realizarse dentro de la Disciplina de Análisis y Diseño.



a) Propósito.

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

El propósito de la disciplina de análisis y diseño es:

- Transformar los requerimientos en un diseño del sistema propuesto.
- Crear una arquitectura robusta para el sistema
- Adaptar el diseño para que concuerde con el ambiente de implementación, enfocándose en el desempeño.

b) Relaciones con otras disciplinas.

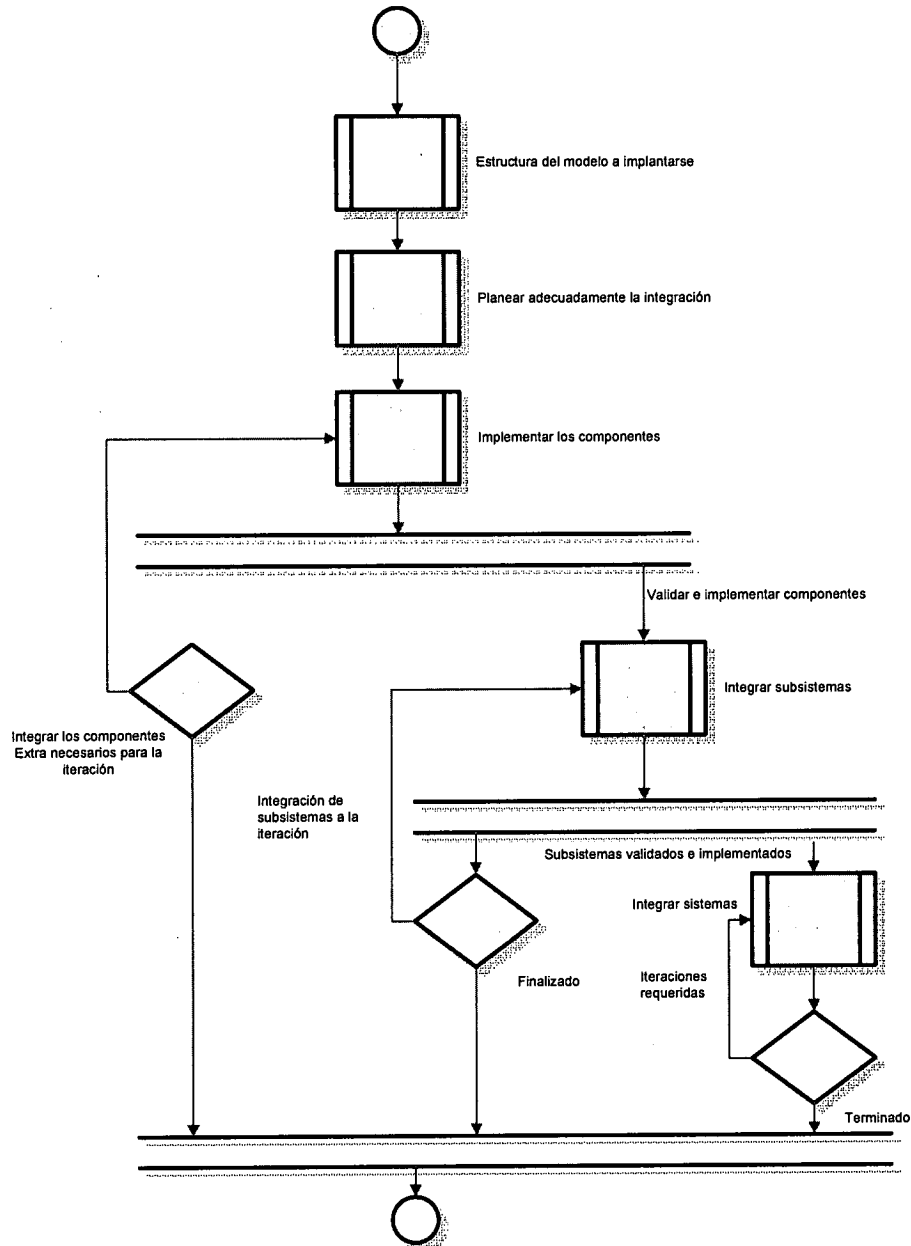
- Modelado de Negocios: Provee un contexto organizacional para el sistema.
- Requerimientos: Provee la entrada primaria para el análisis y diseño.
- Implementación: La disciplina implementa el diseño.
- Pruebas: Realiza la batería de pruebas diseñada durante el análisis y el diseño.
- Ambiente: Desarrolla y mantiene los artefactos que son utilizados durante el análisis y diseño.
- Administración de Proyectos: Planea el proyecto, y cada una de las iteraciones.

c) Artefactos.

Podrán utilizarse los siguientes artefactos, en caso de ser requeridos:

- Prueba de concepto de la arquitectura
- Modelo de distribución
- Arquitectura de referencia
- Modelo de análisis
- Modelo de diseño
- Modelos de datos.

5. Descripción de flujos y actividades a realizarse dentro de la Disciplina de Implementación.



METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

a) Propósito:

El propósito de la implementación es:

- Definir la organización del código en términos de los subsistemas a implementarse.
- Implementar los elementos de diseño en términos de elementos a implementarse (archivos binarios, ejecutables y otros)
- Probar los componentes desarrollados como unidades.
- Integrar los resultados producidos por implementadores en el sistema ejecutable.

La disciplina limita su alcance a como las clases individuales se evalúan de forma independiente. La integración de los subsistemas queda a cargo de la disciplina de Pruebas.

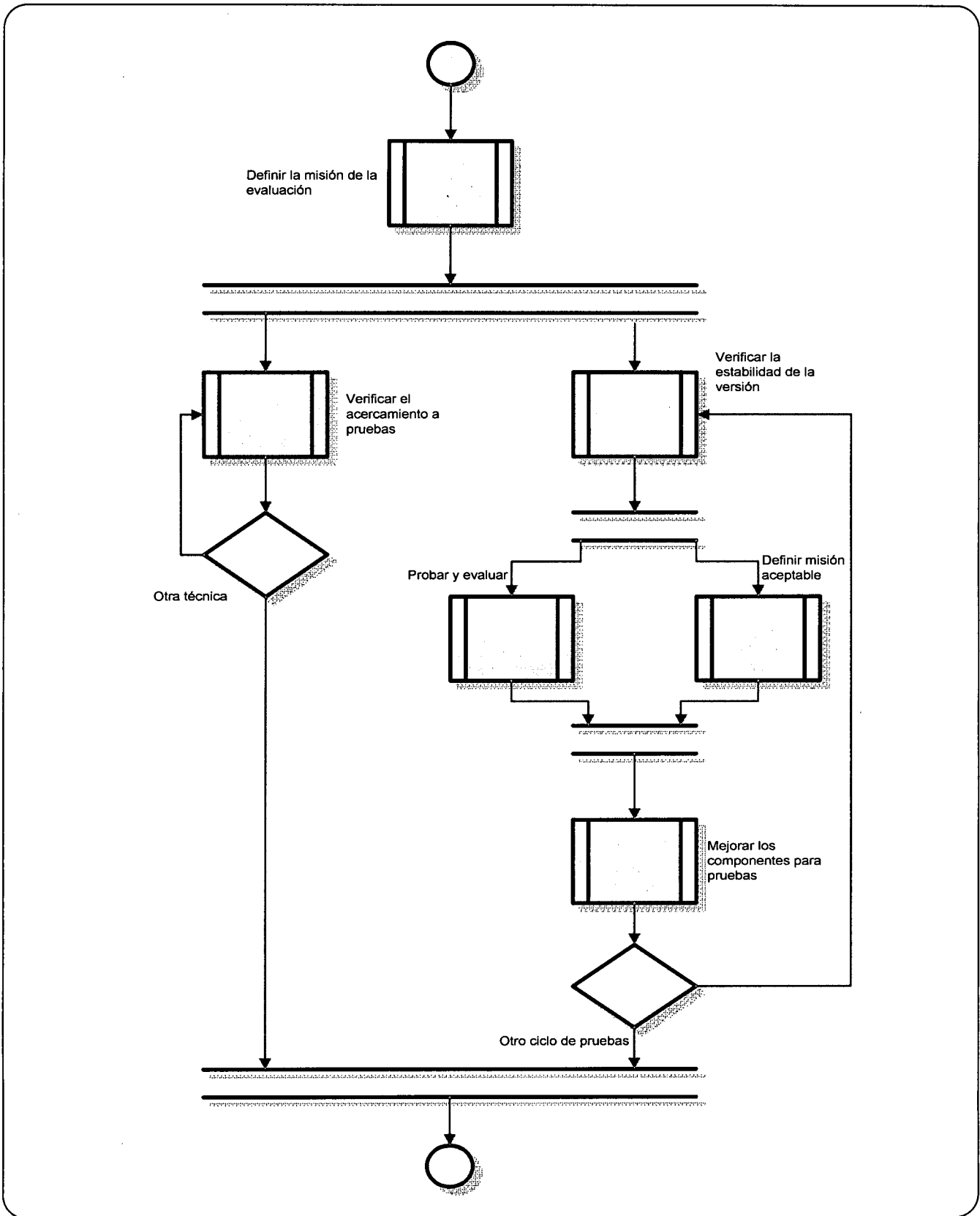
b) Relación con otras disciplinas.

Se relaciona con las siguientes disciplinas:

- La disciplina de requerimientos describe como, en un modelo de casos de uso, se capturan los requerimientos que la implementación debe cumplir.
- La disciplina de análisis y diseño describe el desarrollo del modelo a diseñarse. Este representa la implementación, y pasa información a la disciplina de Implementación.
- La disciplina de pruebas describe como probar la integración en cada sistema construido durante la integración del sistema. Describe también como probar el sistema para verificar que los requerimientos han sido cumplidos, y como deberán enviarse defectos.
- La disciplina de ambiente describe como desarrollar y mantener artefactos a utilizarse durante la implementación: el proceso, las guías de diseño y programación.
- La disciplina de despliegue describe como usar el modelo de implementación para producir y entregar el código al cliente.
- La disciplina de Administración de Proyectos describe como planear el proyecto, con procesos como el plan de iteración, administración de cambios y defectos.

6. Descripción de flujos y actividades a realizarse dentro de la Disciplina de Implementación.

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)



METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

a) Propósito:

La disciplina de pruebas es un proveedor de servicios para otras disciplinas en cierto sentido. Las pruebas se enfocan en evaluar la calidad del producto.

La calidad del producto se logra a través de las siguientes mejores prácticas:

- Encontrar y documentar defectos en la calidad del software desarrollado.
- Hacer recomendaciones respecto a la calidad percibida del software.
- Validar y probar los supuestos realizados durante el diseño y la especificación de requerimientos a través de demostraciones concretas.
- Validar que el producto de software funciona como es requerido.
- Validar que los requerimientos se implementen de manera adecuada.

Existe una diferencia fundamental entre Pruebas y otras disciplinas. Las pruebas deben encontrar y exponer debilidades dentro del producto de software. Para lograr el mayor beneficio posible se necesita una filosofía diferente a la establecida en los Requerimientos, Análisis y Diseño e Implementación. Mientras que éstas se distinguen por su enfoque a la integridad, Pruebas se enfoca a la no integridad de los proyectos.

Las preguntas básicas a realizarse son:

- ¿Cómo puedo destruir esta implementación?
- ¿En qué situaciones se presentará un comportamiento anómalo?

La disciplina pone a prueba los supuestos, riesgos, y la inseguridad inherente en el trabajo de las demás disciplinas, y estudia estos problemas usando demostraciones y evaluación imparcial. Se deben evitar las siguientes acciones:

- Establecer un acercamiento que no exhiba las deficiencias del software, sus problemas y debilidades.
- Un acercamiento que es exageradamente negativo o destructivo, ya que puede terminar por considerarse que el software es inaceptable, alejando la disciplina de Pruebas de las demás.

Deberá considerarse que los costos por la disciplina de Pruebas constituirá del 30 al 50% del costo total de desarrollo del software. Esto se deriva de las siguientes razones:

- Las pruebas al software son muy difíciles.

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- Las pruebas generalmente se realizan sin una metodología específica. El éxito depende de la calidad y cualidades específicas del personal con el que se cuenta.
- No se utilizan herramientas automatizadas que permitan realizar pruebas extensivas de los datos y resultados. Debe utilizarse una metodología adecuada y herramientas de última generación.

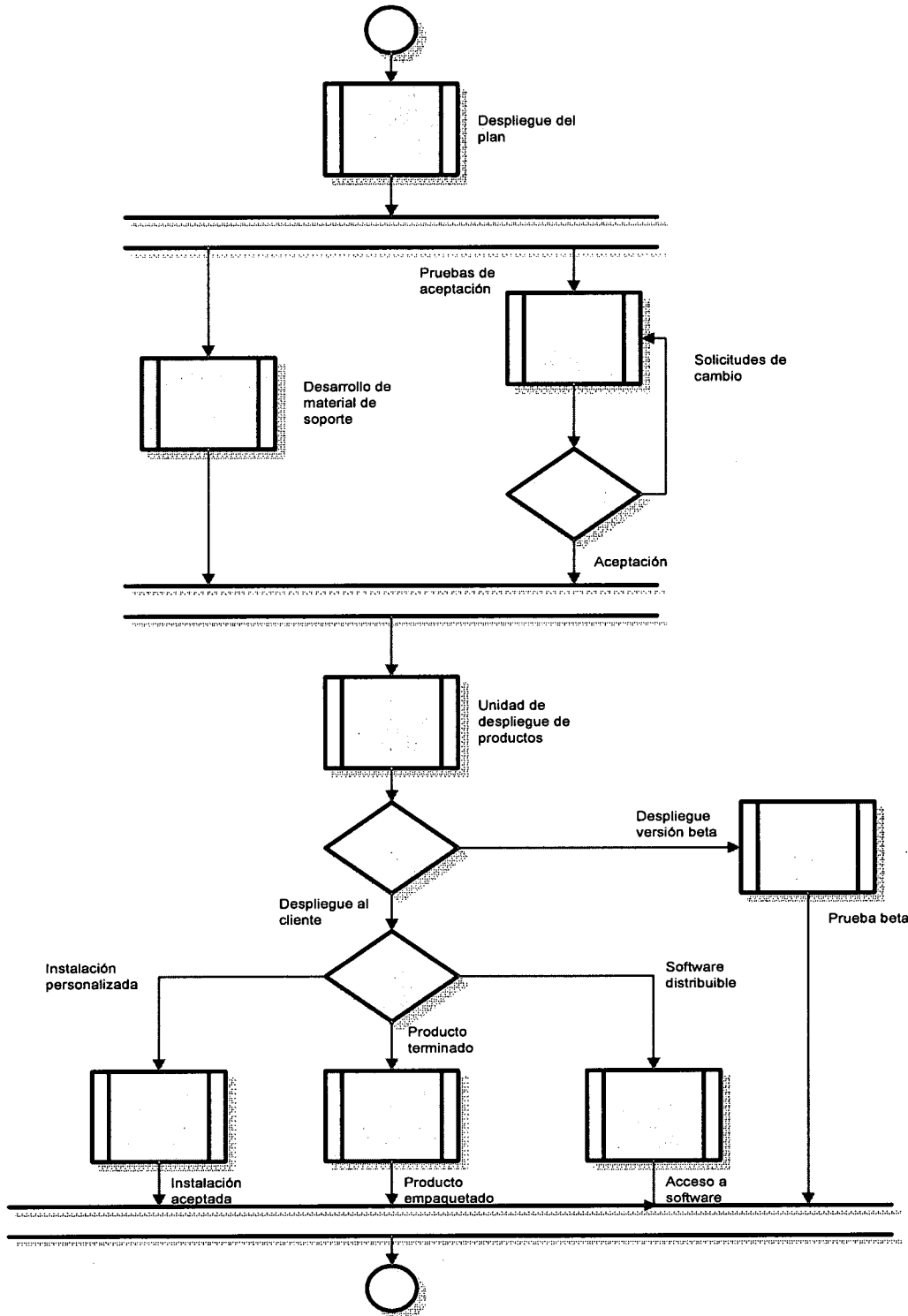
El desarrollo de software de calidad es esencial en el éxito del software de Misión Crítica – específicamente las aplicaciones de Software Institucional para la SCT, en las cuales una falla puede traer como consecuencia problemas en la operación.

Debe considerarse un acercamiento iterativo y continuo enfocado a la calidad. Si se consideran estos costos durante el inicio del proyecto, el riesgo asociado con la implementación de software de pobre calidad se minimizará.

- a) Relación con otras disciplinas.
- a) Requerimientos: Captura los requerimientos del producto de software, que es una de las entradas más importantes para determinar las pruebas a realizarse.
- b) Análisis y diseño: determina el diseño apropiado para el software. Esta es una entrada importante para las pruebas a desarrollarse.
- c) Implementación: Produce las versiones de software que son validadas por la disciplina de pruebas. En una iteración, múltiples versiones se probarán, una por ciclo de pruebas.
- d) Despliegue: Entrega el producto de software al usuario final. Cuando el software ha sido validado por la disciplina de pruebas, se conducen las pruebas beta y y de aceptación como parte de la disciplina.
- e) Ambiente: Desarrolla y mantiene artefactos de soporte utilizados durante las pruebas.
- f) Administración de Proyectos: Planea el proyecto y trabajo por cada iteración. Define la misión adecuada para la evaluación.
- g) Configuración y Administración de Cambios: Controla el cambio dentro del equipo de proyecto. El esfuerzo verifica que cada cambio ha sido realizado apropiadamente.

7. Descripción de flujos y actividades a realizarse dentro de la Disciplina de Distribución.

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)



METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

a) Propósito

La disciplina provee los siguientes modelos de distribución de software.

- Instalación personalizada
- Producto terminado
- Software distribuible

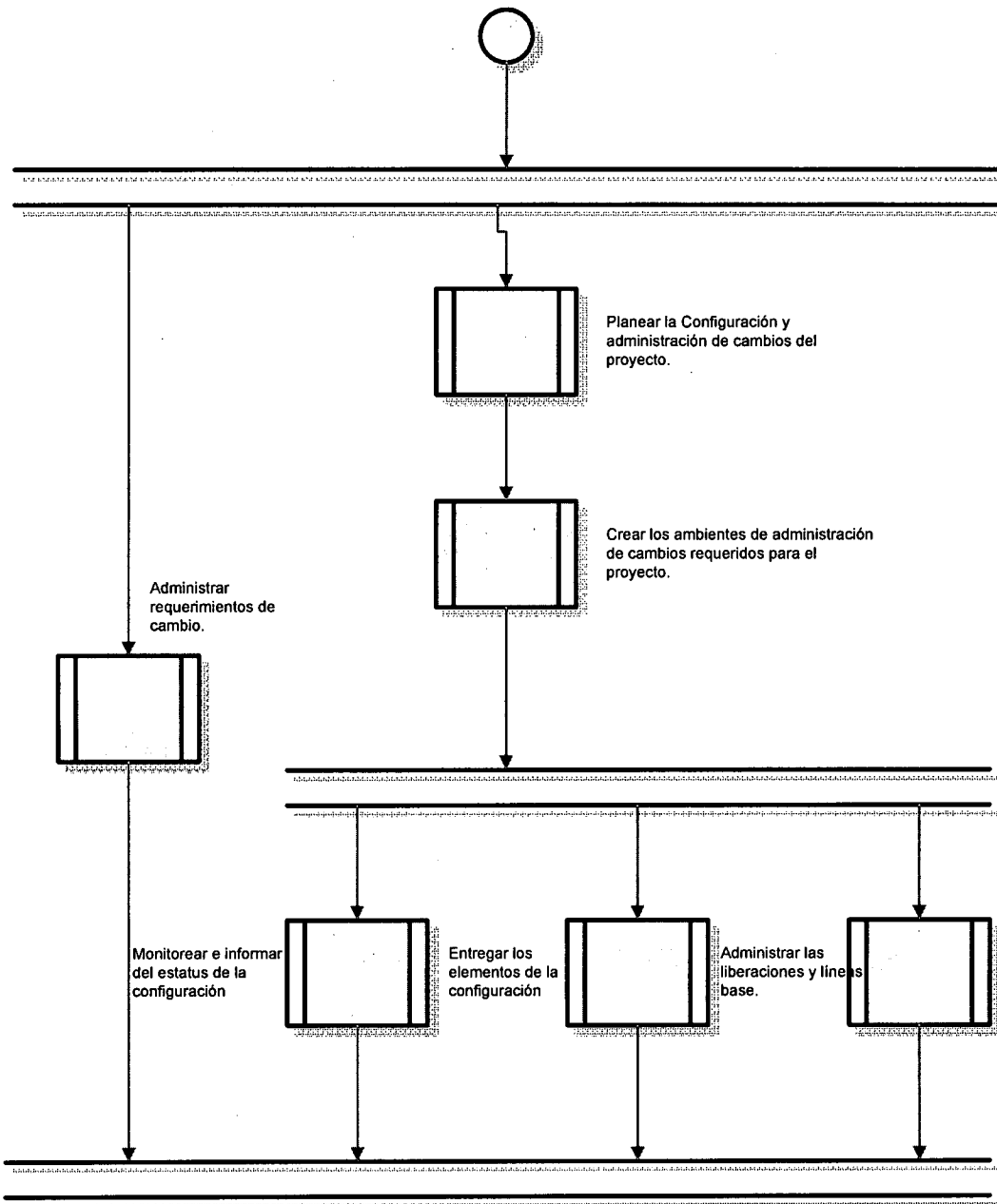
En esta disciplina el énfasis está en las pruebas del producto antes de que el mismo sea entregado al cliente.

b) Relación con otras disciplinas.

- **Requerimientos:** Produce las especificaciones que consisten en el modelo de casos de uso y requerimientos no funcionales. Junto con la interfaz de usuario (en prototipo), la especificación de requerimientos de software es una de las entradas más importantes para desarrollar materiales de soporte y entrenamiento.
- **Pruebas:** Es la contraparte de la implementación, los elementos esenciales que proveen son el sumario de evaluación de pruebas y las actividades para implementar, ejecutar y manejar las pruebas.
- **Configuración y Administración de Cambios:** Provee el build de línea base, y provee los mecanismos para manejar Administración de Cambios que se generan como parte de las pruebas beta y de aceptación.
- **La Administración de Proyectos** desarrolla un Plan de Iteración y Desarrollo de software, fundamentales para desarrollar el plan de implementación.
- **La Disciplina de Ambiente** provee las herramientas requeridas para el ambiente de pruebas.

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

8. Descripción de flujos y actividades a realizarse dentro de la Disciplina de Administración de Cambios.



a) Introducción.

Esta disciplina controla el cambio, y mantiene la integridad de, los artefactos de un proyecto.

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

Tiene las siguientes funciones:

- Identificar los elementos de la configuración.
- Restringir los cambios a dichos elementos.
- Auditar los cambios realizados, y
- Definir y mantener la configuración de dichos elementos.

Los métodos, procesos, herramientas que son utilizados para proveer la administración del cambio y configuración se consideran el Sistema CM.

Este sistema debe contener información acerca de:

- Desarrollo de productos
- Promoción de productos
- Implementación
- Mantenimiento

Y retiene aquellos elementos que pueden ser útiles posteriormente.

b) Propósito.

Es esencial para controlar artefactos producidos por la gente que trabaja en un proyecto común. El Control evita la confusión y asegura a la gente que los artefactos generados no entran en conflicto debido a las siguientes situaciones.

- Actualización simultánea.
 - Cuando dos o más miembros del grupo trabajan en el mismo artefacto, el último en actualizar destruye el trabajo del primero. El problema básico es que si un sistema no soporta la actualización simultánea esto implica que los cambios se destruyen. El reto es detectar los cambios inmediatos para resolver todos los problemas de integración cuando los cambios son incorporados.
- Notificación limitada.
 - Un problema se arregla en un artefacto compartido, y algunos de ellos no son notificados del cambio.
- Versiones múltiples.
 - Si existen múltiples versiones de una aplicación, la solución debe ser aplicada a todas las versiones existentes.

El sistema CM provee los siguientes beneficios.

- Soporte a los métodos de desarrollo.
- Mantiene la integridad del producto.
- Asegura que el producto se desarrolle correctamente.
- Provee un ambiente estable en el cual se pueda desarrollar el producto.
- ~~Restringe los cambios a los artefactos basados en políticas del proyecto.~~

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- Provee un registro de auditoría de por qué se realizaron los cambios.

c) Relación con otras disciplinas.

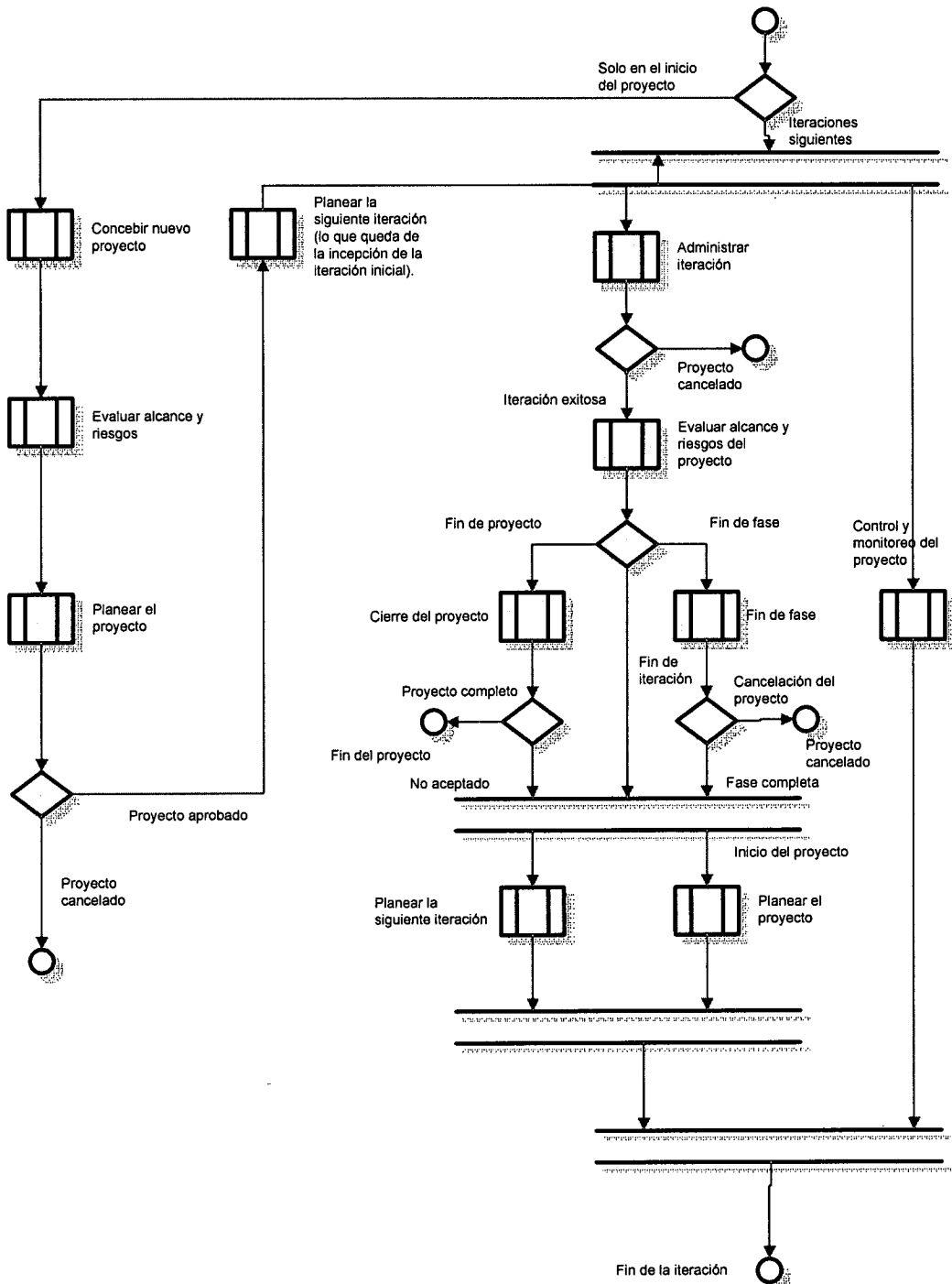
El sistema de CM de una organización se usa a lo largo del ciclo de producto, desde su inicio hasta la instalación. El sistema contiene las versiones de requerimientos, diseño y artefactos creadas por medio de artefactos que definen un sistema.

Esta disciplina constituye, por lo tanto, un sistema de almacenamiento para:

- Modelado de negocios
- Requerimientos
- Análisis y diseño
- Implementación
- Pruebas
- Distribución
- Administración de la Configuración
- Administración de Proyecto
- Ambiente

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

9. Descripción de flujos y actividades a realizarse dentro de la Disciplina de Administración de Proyectos.
(Nota: Esta disciplina se sustituye en proyectos no exclusivos a software por la aplicable de PMI).



METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

d) Introducción:

La disciplina de administración de proyectos se enfoca a balancear objetivos de proyectos, la administración de riesgos, y de superar obstáculos para entregar un producto que sirva a los clientes y usuarios.

e) Propósito

El siguiente es un acercamiento de administración de proyectos que mejora las posibilidades de contar con un proyecto exitoso.

El propósito de la Administración de Proyectos es:

- Proveer una guía para manejar proyectos de software.
- Proveer un conjunto de mejores prácticas para la planeación, ejecución y monitoreo de proyectos.
- Proveer una guía para la administración de riesgos.

Se requieren metodologías como PMI para manejar los siguientes puntos:

- Administración del personal: Contrataciones, entrenamiento, etc.
- Administración del presupuesto: definir, otorgar, etc.
- Administración de contratos con proveedores y clientes.

La disciplina se enfoca en:

- * Administración de riesgo
- * Planeación para proyectos iterativos
- * Monitoreo de proyectos iterativos.

f) Relación con otras disciplinas.

La disciplina de Administración de Proyectos provee las guías para crear y administrar el proyecto, por lo que se usan todas las disciplinas existentes.

- Modelado de negocios
- Requerimientos
- Análisis y diseño
- Implementación
- Pruebas
- Instalación

Se considera a Administración de Proyectos como una disciplina de soporte, junto con:

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- Configuración y cambios
- Ambiente

g) Relación con PMI / PMBoK.

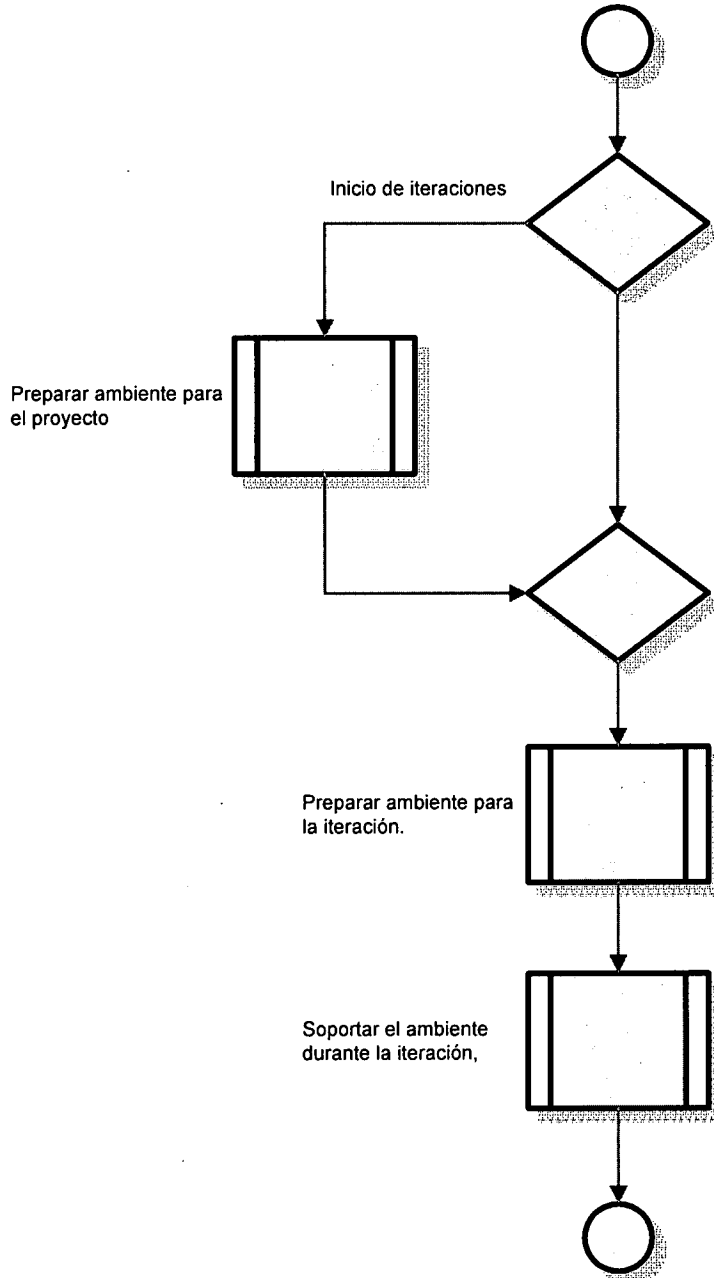
Esta no es una metodología que se enfoque a la administración de proyectos. Se describe únicamente aquella parte que está directamente relacionada con el desarrollo de software, y de esa parte, algunos tópicos se han ignorado o se tratan de manera muy superficial.

El acercamiento provisto en esta sección se desprende del PMI PMBoK (Project Management Institute's Project Management Body of Knowledge).

Deberán utilizarse todas las prácticas establecidas por el PMI en cualquier proyecto realizado.

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

11) Descripción de flujos y actividades a realizarse dentro de la Disciplina de Ambiente.



METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

a) Propósito.

Se enfoca en las actividades necesarias para configurar el proceso para un proyecto. Describe las actividades necesarias para desarrollar las guías de soporte de un proyecto. El propósito de las actividades es el de proveer a la organización con los procesos y herramientas necesarias para el equipo de desarrollo de software.

b) Soporte a disciplinas.

Soporta a todas las disciplinas.

Anexo A: Artefactos de las disciplinas

Disciplina de Modelado de Negocios.

- Administración de Requerimientos
- Especificación de Casos de Uso de Negocio.
- Especificación de Casos de Uso de Sistema
- Especificaciones Suplementarias del Negocio
- Especificaciones Suplementarias del Sistema
- Evaluación de la Organización
- Glosario
- Reglas de Negocio
- Visión del Negocio
- Visión de Sistemas

Disciplina de Requerimientos.

- Plan de Administración de Requerimientos
- Plan de Especificación de Casos de Uso de Sistema
- Plan de visión del Sistema
- Plan de Especificaciones Suplementarias de Sistema

Disciplina de Administración de Proyectos.

- Evaluación del estado del proyecto
- Lista de asuntos
- Minutas
- Plan de aceptación del producto
- Plan de Iteración
- Evaluación de Iteración
- Lista de riesgos
- Plan de cierre
- Plan de Desarrollo de Software
- Plan de Métricas

Disciplina de ambiente

- Ambiente

Disciplina de pruebas

- Estrategia de pruebas
- Plan de Iteraciones maestro
- Resumen de evaluación

Disciplina de Administración de versiones.

- Guía de la auditoría funcional.
- Verificación de la auditoría física
- Hallazgos de las auditorías

Anexo B: Descripción y tutorial de UML.

Introducción.

El Modelado Visual de procesos provee una estructura para la resolución de problemas, la cual permite re-agrupar conceptos de manera gráfica.

Las abstracciones que permite lograr ayudan al manejo de la complejidad, y reducen el tiempo de desarrollo de aplicaciones, minimizando costos de desarrollo, y permitiendo contar con una metodología adecuada para administrar el riesgo.

UML es un lenguaje para:

- Especificar
- Visualizar
- Construir
- Documentar

Los artefactos generados por los sistemas de software.

Las metas que busca UML son:

- Definir un lenguaje sencillo de usar y semánticamente completo.
- Unificar los lenguajes de Booch, OMT, y Objectory.
- Incluir las mejores prácticas de otros lenguajes

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

- Proveer flexibilidad para los procesos requeridos
- Permitir el intercambio de modelos e interfaces con repositorios

Conceptos básicos de UML:

UML se compone de:

- Bloques de construcción
- Reglas de construcción

Bloques de construcción:

- Elementos del modelo
 - Clases, interfaces, componentes, casos de uso, etc.
- Relaciones
 - Asociaciones, generalización, dependencias, etc.
- Diagramas
 - Diagramas de clase, diagramas de caso de uso, diagramas de interacción, etc.

Reglas de construcción.

Indican que un modelo o el fragmento de un modelo sigue las reglas semánticas y sintácticas establecidas.

Se establecen reglas para:

- Nombramientos
- Alcance
- Visibilidad
- Integridad
- Ejecución

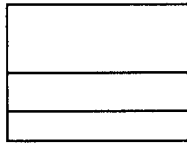
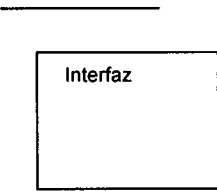
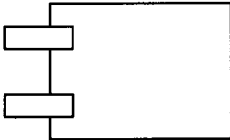
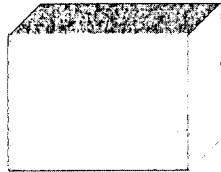
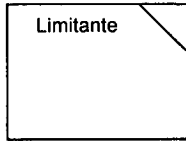
METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

Se puede esperar, sin embargo, que durante el período de construcción los modelos sean inconsistentes e incompletos. Al final de una iteración deberán presentarse entregables concretos.

Modelado estructurado:




El modelado estructurado es una vista del sistema que enfatiza la estructura de los objetos, sus clasificadores, relaciones, atributos y operaciones.

Sus elementos primarios son:

CLASE	Una descripción de un conjunto de objetos que comparten atributos, métodos, operaciones, relaciones y semántica.	
INTERFAZ	Un conjunto de operadores nombrados que caracterizan el comportamiento de un elemento.	
COMPONENTES	Un componente físico, reemplazable de un sistema que empaqueta la implementación y provee un juego de interfaz.	
NODOS	Objeto físico que representa un recurso computacional.	
LIMITANTES	Una condición semántica, o restricción.	

METODOLOGÍA PARA EL DESARROLLO DE APLICACIONES (MDA)

Las asociaciones son:

ASOCIACIÓN	Relación entre dos o más clasificadores que involucra conexiones entre instancias.	
AGREGACIÓN	Forma especial de asociación que especifica una relación entre el agregado y el componente.	
GENERALIZACIÓN	Relación taxonómica entre un elemento general y otro más específico	
DEPENDENCIA	Una relación entre dos elementos modelados, en el cual el cambio de un elemento afectará al otro.	