

# REPORTE MADI

DIEGO ARMANDO APARICIO ÁLVAREZ  
MANUELA JIMENEZ ROMERO

UNIVERSIDAD EAFIT  
ESCUELA DE INGENIERÍAS  
DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS  
MEDELLÍN  
2009

# REPORTE MADI

DIEGO ARMANDO APARICIO ÁLVAREZ  
MANUELA JIMENEZ ROMERO

Proyecto de Grado para optar el título de  
Ingenieros de Sistemas

Asesor:

Carlos Hernando Montoya Medina  
Ingeniero de Sistemas

UNIVERSIDAD EAFIT  
ESCUELA DE INGENIERÍAS  
DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS  
MEDELLÍN

2009

Nota de aceptación:

---

---

---

---

---

Firma del presidente del jurado

---

Firma del jurado

---

Firma del jurado

Medellín 29 de Mayo 2009

A nuestras familias por estar siempre  
apoyándonos en cada etapa de  
nuestra carrera.

## **AGRADECIMIENTOS**

Agradecemos toda la colaboración que hemos recibido de cada una de las personas que hizo posible este proyecto de grado y mostraron interés en resolver cada una de las dudas que se nos presentaron, de igual forma a cada una de las personas que nos motivaron constantemente para la realización de este trabajo.

Tenemos un agradecimiento muy especial a todas las compañías desarrolladoras de nuestra ciudad que hicieron que este proyecto se hiciera realidad. También queremos darle nuestro profundo agradecimiento al ingeniero Carlos Hernando Montoya experto en gestión de desarrollo de aplicaciones, por brindarnos toda su asesoría con la mejor disponibilidad que podíamos obtener, aportándonos sus ideas, recomendaciones y experiencia.

Finalmente queremos agradecer a la Universidad EAFIT y a todas las personas que nos apoyaron durante cada una de las etapas de nuestro proyecto de grado.

# CONTENIDO

	Pág.
ANEXOS	8
INTRODUCCION	11
OBJETIVOS	12
OBJETIVO GENERAL	12
OBJETIVOS ESPECÍFICOS	12
1. CONTEXTO GENERAL	13
1.1. PROYECTOS	13
1.1.1. Planificación de un Proyecto de ingeniería de software	13
1.1.1.1. Actividades que se derivan de la planificación	14
1.1.2. Organización de un proyecto de ingeniería de software	15
1.1.2.1. Actividades que se derivan de la organización	16
1.1.3. Gestión de un proyecto de ingeniería de software	16
1.1.3.1. Actividades de Gerenciamiento de Proyectos	17
1.1.4. Equipo de Trabajo	18
1.2. CICLO DE VIDA DEL SOFTWARE	18
1.2.1. Un modelo de ciclo de vida del software	18

1.2.2. Descripción de los diferentes modelos de ciclo de vida del software	19
1.2.3. Modelo Lineal	19
1.2.3.1. Modelo Cascada.	20
1.2.3.2. Modelo de Desarrollo Incremental.	21
1.2.3.3. Modelo de Desarrollo Evolutivo.	22
1.2.3.4. Modelo de Prototipado de Requisitos.	23
1.2.3.5. Modelo Espiral.	23
1.2.4. Seleccionando Modelos de Ciclo de Vida.	25
1.3. ALGUNAS TÉCNICAS DE LA INGENIERÍA DE SOFTWARE, MÉTODOS Y MARCOS METODOLÓGICOS	26
1.3.1. Métodos de Desarrollo de Software	26
1.3.1.1. Enfoque Estructurado	26
1.3.1.2. Enfoque Orientado a objetos	26
1.3.1.3. Programación Orientada a Componentes	27
1.3.1.4. SOA	27
1.3.1.5. Desarrollo Orientado a Aspectos	28
1.3.2. Marcos Metodológicos	28
1.3.2.1. RUP	28
1.3.2.2. AUP	29

1.3.2.3. Programación extrema XP	29
1.3.2.4. CRYSTAL	30
1.3.2.5. AM –Agile Modeling	30
1.3.2.6. SCRUM	31
1.3.2.7. OSS	31
1.3.3. Técnicas	32
1.3.3.1. UML	32
1.4. REPORTE CHAOS	32
1.5. OTROS REPORTES	37
1.5.1. Historia de Otros Reportes	37
1.5.2. Reportes Encontrados a nivel mundial	37
1.5.2.1. Informes de GAO (Government Account Office)	38
1.5.2.2. Informes de FINANCIAL EXPRESS	41
1.5.2.3. Informes ESPITI (European Software Process Improvement Training Initiative)	43
2. MARCO TEÓRICO	45
2.1. ESTRUCTURA DEL REPORTE MADI	45
2.2. BENEFICIARIOS	47
2.3. PLANTEAMIENTO DE HIPÓTESIS	48



2.3.1. Mala Gestión de Usuarios	48
2.3.2. Metodologías utilizadas en el desarrollo del proyecto	48
3. DESARROLLO DE LA PROPUESTA	51
3.1. INTRODUCCIÓN ENCUESTA	51
3.2. ENCUESTA	52
EMPRESAS ENTREVISTADAS EN LA CIUDAD DE MEDELLIN	61
3.2.1. Gerencia de Desarrollo Inversura	63
3.2.2. Centro de Informática de la Universidad EAFIT	63
3.2.3. Intergrupo	63
3.2.4. Ceiba	64
3.2.5. MVM Ingeniería de Software S.A	64
3.2.6. Tecnofactor S.A.	65
3.2.7. Sonda	65
3.2.8. Cognox Ltda.	65
3.2.9. Insitu	66
3.2.10. Dicosoft	66
3.3. RESULTADO GENERAL	67
3.3.1. Resultados de Encuestas	67
Resultado General	68
3.4. ANÁLISIS DE LAS ENCUESTAS	73

3.4.1. Proyectos	74
3.4.1.1. Proyectos Exitosos y Proyectos Fracasados	74
3.4.1.2. Proyectos Cancelados	76
3.4.1.3. Estimación de Tiempos	78
3.4.2. Requisitos y usuarios	79
3.4.2.1. Usuarios	79
3.4.2.2. Requisitos	83
3.4.3. Metodología, Planeación y Equipos de trabajo	85
3.4.3.1. Metodología	85
3.4.3.2. Planeación	86
3.4.3.3. Equipo de Trabajo	88
ANEXO A	90
ANEXO B	98
BIBLIOGRAFÍA	92

## LISTA DE IMÁGENES

	Pág.
<b>Figura 1.</b> Ciclo de <i>Vida</i> Lineal .....	19
<b>Figura 2.</b> Modelo de ciclo de vida Cascada .....	20
<b>Figura 3.</b> Modelo de desarrollo Incremental.....	21
<b>Figura 4.</b> Modelo Evolutivo.....	22
<b>Figura 5.</b> Modelo prototipado de Requisitos, basado en el modelo de desarrollo incremental. 23	
<b>Figura 6.</b> Modelo Espiral .....	24
<b>Figura 7.</b> Los reportes “The Standish Group - Chaos” desde 1994 hasta el 2004	35
<b>Figura 8.</b> Standish Group Chaos Report - 2006.....	36
<b>Figura 9.</b> Resultados del Informe de GAO .....	39
<b>Figura 10.</b> Resultados del informe CHAOS .....	40
<b>Figura 11.</b> Resultados del informe ESPITI .....	44
<b>Figura 12.</b> Resultados del Reporte MADI.....	74
<b>Figura 13.</b> Resultados del Reporte MADI.....	75
<b>Figura 14.</b> Resultados del Reporte MADI.....	77
<b>Figura 15.</b> Resultados del Reporte MADI.....	78
<b>Figura 16.</b> Resultados del Reporte MADI.....	79

<b>Figura 17.</b>	Resultados del Reporte MADI.....	80
<b>Figura 18.</b>	Resultados del Reporte MADI.....	82
<b>Figura 19.</b>	Resultados del Reporte MADI.....	83
<b>Figura 20.</b>	Resultados del Reporte MADI.....	84
<b>Figura 21.</b>	Resultados del Reporte MADI.....	85
<b>Figura 22.</b>	Resultados del Reporte MADI .....	87
<b>Figura 23.</b>	Resultados del Reporte MADI.....	88
<b>Figura 24.</b>	Resultados del Reporte MADI.....	89
<b>Figura 25.</b>	Resultados del Reporte MADI.....	89

## **Anexos**

Pág.

Anexo A The standish Group Report 1995

90

Anexo B Reporte GAO – OMB Can more effectively Use Its Investment Reviews

98

## INTRODUCCIÓN

El sector informático de nuestra ciudad se caracteriza por estar muy orientado a la industria de desarrollo de software y este hecho nos ha motivado a plantear un estudio que ayude a determinar cuales pueden ser las fortalezas y debilidades de este sector.

Es así como hemos creado el Reporte MADi, este es un reporte que cuenta con una metodología similar a la adoptada por el Standish Group en su reconocido estudio CHAOS, el cual nos ayudo a tener una visión global de los problemas en distintos puntos durante la ejecución de proyectos de desarrollo de software. El reporte MADi es un informe en donde se puede hallar información relevante en del sector, cumpliendo unos estándares de presentación e investigación que tiene además del CHAOS, otro tipo de reporte como lo son GAO y FINANCIAL EXPRESS REPORTS.

Lo que tiene de particular el reporte MADi es que el estudio realizado es sobre la industria de desarrollo de software en nuestra ciudad y lo que pretende es lograr una mejora en el sector y sembrar la semilla, de estar siempre preguntándonos como fue la actuación durante la ejecución de un proyecto y como puedo mejorar en de futuros proyectos.

# OBJETIVOS

## OBJETIVO GENERAL

Investigar y documentar en las empresas de desarrollo de nuestra ciudad, las causas generales por las que un proyecto informático puede llegar a fracasar o ser un éxito.

## OBJETIVOS ESPECÍFICOS

- Identificar la importancia que tienen los usuarios en los proyectos, debido a que estos son los expertos en los temas del negocio.
- Descubrir si realmente la alta gerencia de la empresa usuaria es la que le da el soporte a los proyectos de software.
- Destacar el valor que tiene una adecuada elicitación de requisitos, dado que son estos los que definen el software que se debe construir por lo que se convierten en nuestra guía durante todo el desarrollo del proyecto.
- Estudiar la forma como están realizando la etapa de planeación de las empresas de desarrollo de nuestra ciudad.
- Indagar la gestión de los equipos de trabajo en las empresas desarrolladoras de Medellín y como estos respondieron a las exigencias del proyecto.

- Mostrar la alineación actual que existe entre las tecnologías de información y la visión estratégica de las empresas usuarias.
- Cuestionar a las empresas desarrolladoras acerca de las actuales metodologías comparadas frente a las metodologías ágiles.
- Identificar oportunidades de mejora en los procesos de desarrollo de estas compañías y reducir el riesgo de fracaso.



# 1. CONTEXTO GENERAL

## 1.1. PROYECTOS

Para referirnos al término proyectos vamos a dar una definición muy breve pero que nos ayuda a entender este término de manera fácil. La palabra proyecto viene del latín *projectus* y es contemplada como un conjunto de actividades coordinadas e interrelacionadas para cumplir con una meta u objetivo específico [1].

En ingeniería de software un proyecto es un conjunto de actividades interrelacionadas con el fin de obtener un producto software. Los proyectos de software tienen varios componentes y existen muchas formas y técnicas para realizarlos, a continuación vamos a presentar algunos elementos relacionados con los proyectos de ingeniería de software y algunas de las hipótesis que hemos planteado.

### 1.1.1. Planificación de un Proyecto de ingeniería de software

El inicio de los proyectos de ingeniería de software debe ser un buen plan. La planificación se basa en definir unas metas iniciales para cumplirlas a medida que vamos desarrollando el proyecto, teniendo en cuenta los riesgos potenciales y los posibles problemas que se van a presentar y afecten conseguir dichas metas. En este punto inicialmente hemos encontrado en fuentes secundarias los posibles problemas que se dan en la planificación de los proyectos, los cuales vamos a enunciar como hipótesis a lo que queremos realizar en nuestro estudio, los problemas son [2]:

- Requisitos incorrectos e incompletos.

- Muchas especificaciones de requisitos son variables y sujetas a cambios mayores.
- La planificación no se lleva a cabo por la creencia errónea de que es una pérdida de tiempo y que los planes cambiarán de todos modos.
- La planificación de costos y plazos no es actualizada y se basa en necesidades de mercadeo y no en los requisitos del sistema.
- Es difícil estimar el tamaño y complejidad de un proyecto de software de manera que sean estimaciones realistas.
- Los costos y plazos no son recalculados cuando los requisitos del sistema o el ambiente de desarrollo cambia.
- No se manejan factores de riesgo.
- La mayoría de las organizaciones de desarrollo de software no recolectan datos de proyectos pasados para inferir comportamientos futuros.
- Las compañías no establecen políticas o procesos de desarrollo de software.

#### **1.1.1.1. Actividades que se derivan de la planificación**

Ahora vamos a revisar las principales actividades que se deben de tener en cuenta en la etapa de planificación de los proyectos [3]:

- Fijar objetivos y metas.
- Desarrollar estrategias para el desarrollo del proyecto.
- Desarrollar políticas.
- Anticipar futuras situaciones.
- Construir una bitácora de riesgos.
- Determinar posibles cursos de acción ante imprevistos.
- Tomar decisiones de planificación.
- Fijar procedimientos y reglas.
- Desarrollar los planes del proyecto.

- Elaborar presupuestos.
- Documentar los planes del proyecto.

### **1.1.2. Organización de un proyecto de ingeniería de software**

Al inicio del proyecto es necesario definir una estructura organizacional que nos permita ser eficientes para completar todas las tareas del proyecto a tiempo y también para poder establecer roles en cada una de las tareas.

Según nuestra percepción, en nuestra ciudad hay varios problemas identificados en las empresas de desarrollo y en los proyectos de ingeniería de software. Queremos validar si de verdad estos problemas afectan en gran medida el desarrollo de un proyecto:

- Nuestra hipótesis es que en las empresas de la ciudad las estructuras organizacionales son muy complejas, ya sea porque el operar de las mismas no permite un cambio en la estructuración o tenemos muy arraigado el tipo de empresa jerárquica.
- Muchos líderes de los proyectos quieren estar tanto en la parte técnica como en la parte de la gestión de su equipo, si bien esto al principio puede ser visto con buenos ojos, el líder debe tratar de tener una visión mucho más global que el equipo de trabajo, por lo que es conveniente que sólo cumpla su rol durante la ejecución del proyecto.
- Los equipos de trabajo están saturados con tareas de diversos proyectos, lo que no les permite poder centrar su conocimiento a un proyecto específico y comienzan a trabajar como apaga incendios.
- Las tareas que están mal definidas o indefinidas desde el inicio del proyecto en ocasiones sólo se cargan sobre una persona, en la mayoría de los casos estas personas son los desarrolladores o los usuarios finales, según lo dicta nuestra

experiencia. Este es uno de los puntos más difíciles de detectar con nuestra investigación pero vamos a tratar de averiguarlo.

#### **1.1.2.1. Actividades que se derivan de la organización**

Las organizaciones antes de comenzar un desarrollo deben definir una serie de actividades que son el insumo esencial de los proyectos de ingeniería de software, veamos a continuación que actividades hemos encontrado [4]:

- Identificar y agrupar las funciones, actividades y tareas del proyecto.
- Seleccionar una estructura organizacional.
- Definir responsabilidades y autoridades en el equipo de trabajo.
- Establecer el perfil de cada puesto.
- Documentar las decisiones organizacionales.

#### **1.1.3. Gestión de un proyecto de ingeniería de software**

Los coordinadores y gerentes de proyectos son personas designadas para estar constantemente apoyando, animando y evaluando el equipo de trabajo en el desarrollo del proyecto. El jefe debe orientar el proyecto y tener claro las responsabilidades y asignaciones de cada uno de los integrantes del equipo de trabajo, para poder guiarlos hacia la mayor productividad en Pro del proyecto. Esta no es una tarea fácil y tiene algunos problemas que queremos revisar para conocer el rol actual de nuestros jefes de proyectos en la ciudad:

- Fallas en la comunicación efectiva entre los integrantes del proyecto y los usuarios.
- El dinero no es un motivador suficiente para los desarrolladores de software, aunque es un factor necesario. En la ciudad cada vez más se han desmejorado los sueldos de

los ingenieros de sistemas [5]. Los ingenieros de sistemas también están ávidos de nuevo conocimiento y de mejorar las técnicas que conocen.

- Las compañías y los jefes no poseen las técnicas y herramientas apropiadas para motivar a los ingenieros de software. No necesariamente tiene que ser con gasto de la compañía, solo el reconocimiento es una de las técnicas que se pueden usar.
- Los clientes, usuarios y gerentes no reconocen el impacto potencial que causa en el software un aparentemente cambio trivial, por ejemplo, ellos creen que es “sólo un problema simple de programación” [6].

#### **1.1.3.1. Actividades de Gerenciamiento de Proyectos**

Para el buen desarrollo del proyecto, es necesario que el gerente tenga claras sus funciones. A continuación mencionaremos las principales actividades del rol Gerente de Proyecto [7]:

- Proveer liderazgo.
- Supervisar personal.
- Delegar autoridad.
- Motivar personal.
- Construir equipos.
- Coordinar actividades.
- Facilitar comunicaciones.
- Resolver conflictos.
- Manejar cambios.
- Explicar las decisiones de dirección.

#### **1.1.4. Equipo de Trabajo**

El equipo de trabajo debe ser conformado por personas idóneas para el desarrollo del proyecto. Fuera de las capacidades técnicas que deben tener, es de gran importancia que cada uno de los integrantes del grupo tenga una buena relación con los demás, por que comparten los mismos valores, como la responsabilidad, el compañerismo, la colaboración y el respeto. También es importante que todos los integrantes del equipo de trabajo estén comprometidos y quieran participar en el proyecto.

### **1.2. CICLO DE VIDA DEL SOFTWARE**

Debido a la falta de gestión que se presentaba cuando la industria comenzó a involucrar programas informáticos en sus actividades diarias, los ingenieros informáticos y de sistemas comenzaron a buscar una alternativa que les ayudara a gestionar la creación de nuevos sistemas. En ese entonces todo se realizaba por prueba y error y el control de cambios no se tenía contemplado, por lo que los usuarios nunca paraban de hacer cambios.

Un punto de quiebre a estas practicas surgió con un modelo llamado Cascada, que nació a finales de los 70, este marco fue realizado por Winston Royce [8].

#### **1.2.1. Un modelo de ciclo de vida del software**

Un modelo de un ciclo de vida es un marco de referencia utilizado por los ingenieros de software para ordenar cada una de las actividades que deben realizar en el proyecto [9], así como también son un marco para la administración del proyecto, porque permite validar el avance de cada una de las actividades. Entre sus principales características se encuentran las siguientes:

- Describir las fases que se deben realizar durante el desarrollo de software.
- Definir las actividades en cada una de las fases a ejecutarse.

### 1.2.2. Descripción de los diferentes modelos de ciclo de vida del software

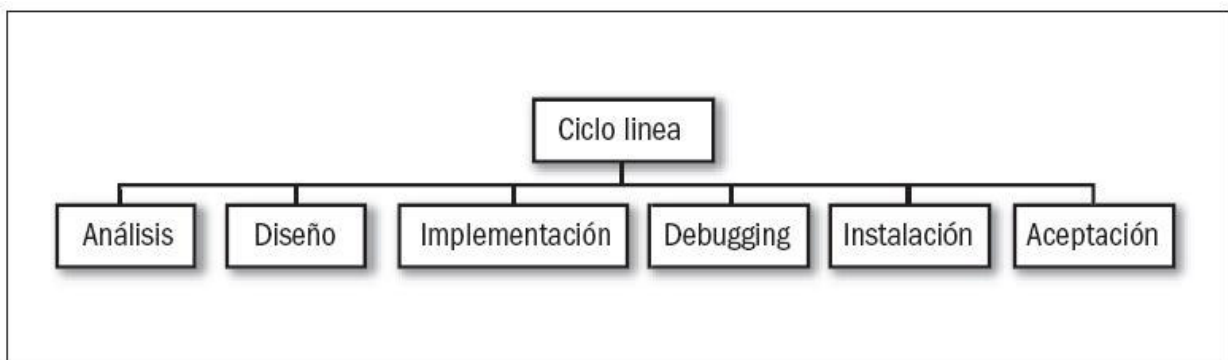
Vamos a dar una breve descripción de los ciclos de vida más relevantes y que han sido utilizados durante los últimos años para el desarrollo de software:

### 1.2.3. Modelo Lineal

Como lo indica su nombre lo que se hace en este modelo es separar las actividades para luego realizarlas de forma lineal. Esto quiere decir que primero hacemos una actividad y a continuación la siguiente si y sólo si ya acabamos la primera de las actividades. Se deben tener en cuenta que las actividades no se retroalimentan unas a otras. Es un buen marco para desarrollos pequeños en los que los tiempos son mínimos.

Veamos la figura para este modelo.

**Figura 1.** Ciclo de *Vida* Lineal



**Fuente:** USR.CODE. "Implementación y debugging.", "Ciclo de Vida del Software" [10].

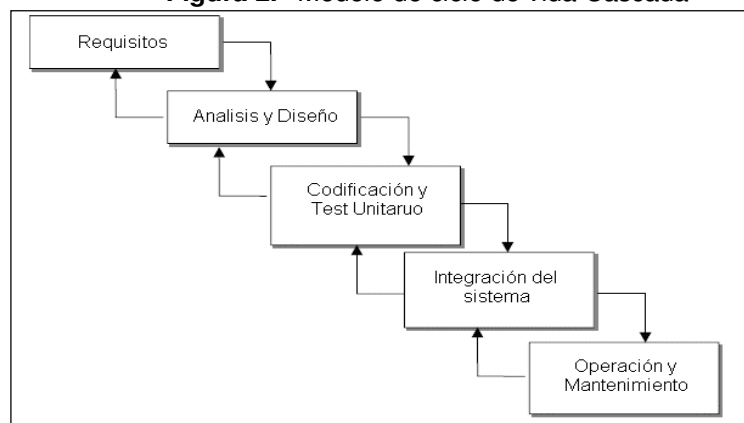
### 1.2.3.1. Modelo Cascada.

Es el más básico de todos los modelos, y sirve como bloque de construcción para otros modelos de ciclo de vida. La visión del modelo cascada del desarrollo de software es muy simple; dice que el desarrollo de software se puede lograr a través de una secuencia simple de fases [11], pero no necesariamente fases lineales debido a que este si permite iteraciones entre cada una de las fases [12].

Cada fase tiene un conjunto de metas bien definidas. El modelo de ciclo de vida cascada, captura algunos principios básicos:

- Planear un proyecto antes de embarcarse en él.
- Definir el comportamiento externo deseado del sistema antes de diseñar su arquitectura interna.
- Documentar los resultados de cada actividad y/o fases.
- Modelar y diseñar un sistema antes de codificarlo o construirlo.
- Realizar las pruebas al sistema después de construirlo.

**Figura 2.** Modelo de ciclo de vida Cascada



**Fuente:** VARAS C., Marcela. Gestión de proyectos de desarrollo de software [13].



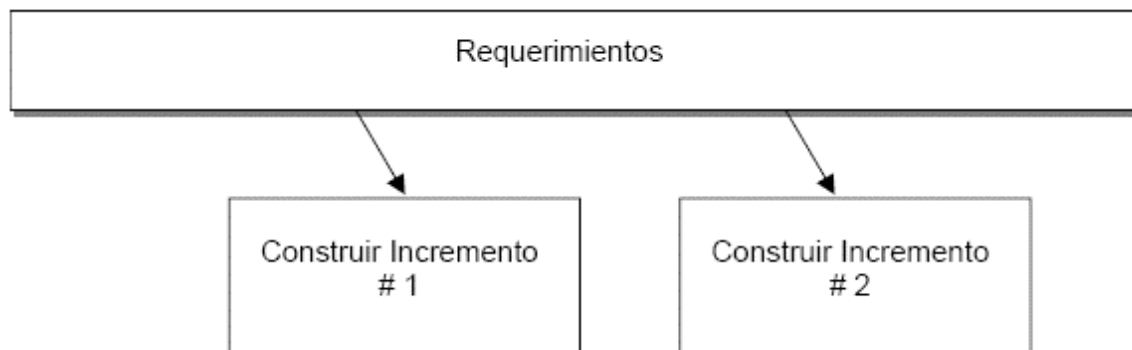
Como lo podemos apreciar en la figura 2 este modelo nos permite la retroalimentación a diferencia del modelo lineal, aunque esta complejidad lo hace ser un modelo muy lento.

### 1.2.3.2. Modelo de Desarrollo Incremental.

Los riesgos asociados con el desarrollo de sistemas largos y complejos son enormes. Una forma de reducir los riesgos es construir sólo una parte del sistema, reservando otros aspectos para niveles posteriores [14].

Este modelo nos permite realizar una serie de entregas a medida que se va desarrollando cada módulo para que el usuario o cliente vaya viendo la funcionalidad del sistema antes de terminar el proyecto.

**Figura 3.** Modelo de desarrollo Incremental



**Fuente:** VARAS C., Marcela. Gestión de proyectos de desarrollo de software [15]

El modelo de desarrollo incremental provee algunos beneficios significativos para los proyectos:

- Construir un sistema pequeño es siempre menos riesgoso que construir un sistema grande.
- Al ir desarrollando por partes las funcionalidades, es más fácil determinar si los requisitos planeados para los niveles subsiguientes son correctos.

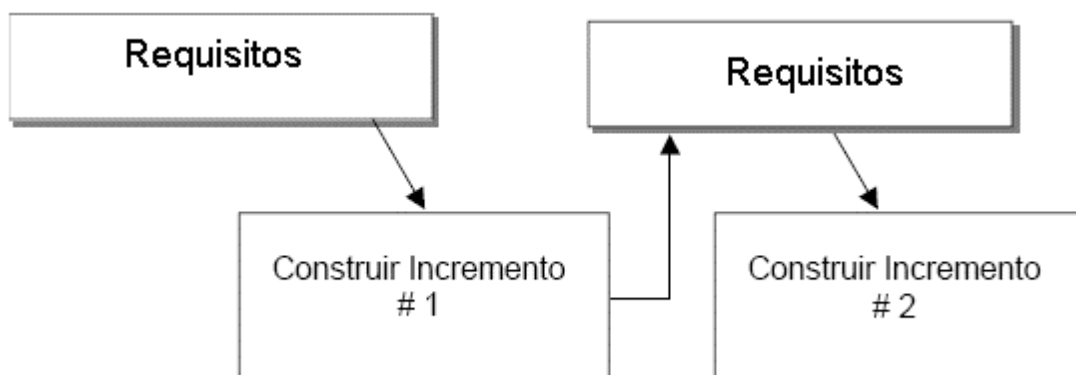
- Los errores de desarrollo realizados en un modelo incremental, pueden ser solucionados antes del comienzo del próximo incremento.

Este modelo se basa en el modelo cascada debido a que en cada uno de los módulos construidos se aplica el modelo cascada para resolver el requerimiento del usuario.

### 1.2.3.3. Modelo de Desarrollo Evolutivo.

Como el modelo de desarrollo incremental, el modelo de desarrollo evolutivo (algunas veces denominado como prototipado evolutivo) construye una serie de versiones sucesivas de un producto. Sin embargo, mientras que la aproximación incremental presupone que el conjunto completo de requisitos es conocido al comenzar, el modelo evolutivo asume que los requisitos no son completamente conocidos al inicio del proyecto [16]. Este modelo supone que los requisitos provienen en cualquier momento del desarrollo, debido a los pocos conocimientos de los usuarios en tecnología y adicionalmente porque hay requisitos concurrentes que pueden afectarse unos a otros. Se hace un análisis inicial con los requisitos iniciales y dependiendo del resultado del análisis se implementan solo algunos de estos requisitos.

**Figura 4.** Modelo Evolutivo



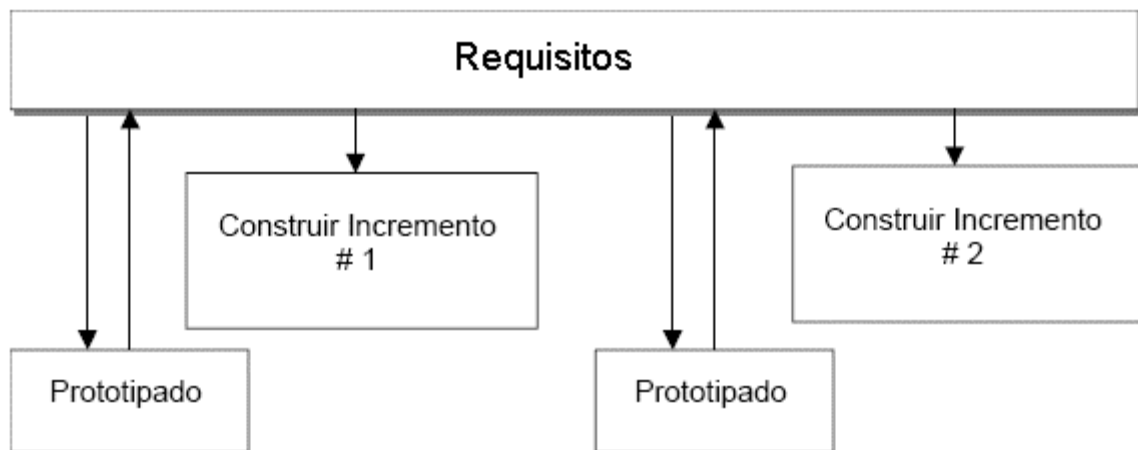
**Fuente:** VARAS C., Marcela. Gestión de proyectos de desarrollo de software [17]

#### 1.2.3.4. Modelo de Prototipado de Requisitos.

Este modelo plantea realizar un prototipo del sistema en el que los usuarios tienen la posibilidad de dar sugerencias y a partir de estas ir tomando las ideas que se van a dejar como versión final en el sistema real.

Su fin principal es una demostración o evaluación [18]. Veamos la grafica el modelo

**Figura 5.** Modelo prototipado de Requisitos, basado en el modelo de desarrollo incremental.



**Fuente:** VARAS C., Marcela. Gestión de proyectos de desarrollo de software [19]

#### 1.2.3.5. Modelo Espiral.

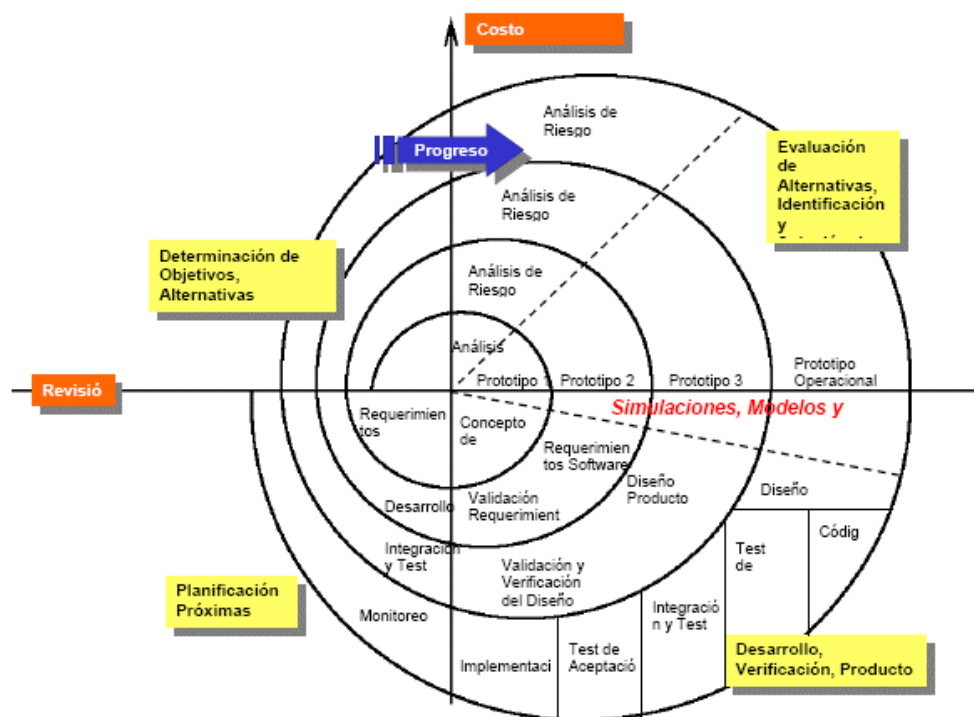
El modelo espiral es un modelo que mezcla conceptos del modelo incremental y el prototipado. Se realizan varios ciclos y cada vez que se hace un ciclo se hace un prototipo cada vez mejor, por lo que el concepto de riesgos y requisitos no levantados desde el principio son manejados como constante.

Los cuatro pasos a seguir son [20]:

1. Determinar qué se quiere lograr.

2. Determinar las rutas alternativas que se pueden tomar para lograr estas metas.  
Por cada ruta, analizar riesgos y resultados finales, y seleccionar la mejor.
3. Seguir la alternativa seleccionada en el paso 2.
4. Establecer lo que se tiene terminado.

**Figura 6.** Modelo Espiral



**Fuente:** VARAS C., Marcela. Gestión de proyectos de desarrollo de software [21]

Es el enfoque más realista actualmente, aunque no sea el más usado y adicionalmente se encuentra como uno de los modelos menos usados en la industria del software por su falta de madurez.

#### **1.2.4. Seleccionando Modelos de Ciclo de Vida.**

Cuando se inicia un proyecto, según los principios generales de la ingeniería del software, se debe seleccionar el Modelo de Ciclo de Vida a seguir. Los diferentes modelos se pueden utilizar en forma combinada, de manera que puedan utilizarse las ventajas de cada uno en un único proyecto.

El modelo ó la combinación de modelos se deben elegir en función de las características del proyecto, ya que no hay un modelo que aplique 100% a todos los proyectos de una organización. [22]

Los criterios que hay que considerar antes de la elección de uno o varios de los modelos a implementar en un proyecto son:

- Las necesidades de los usuarios: “Requisitos”.
- Complejidad del proyecto y de la solución del mismo.
- El dinero disponible para el proyecto.
- El tiempo que se tiene planeado para el proyecto.
- Disponibilidad de los usuarios.

Adicionalmente para hacer una buena selección del modelo a adoptar habrá que hacerse una serie de cuestionamientos [23]:

- ¿Qué tantos son los riesgos del proyecto?
- ¿Qué tan claros están los requisitos?
- ¿Se conoce bien la tecnología ha utilizar?

- ¿Qué visibilidad requiere el proyecto?
- ¿Qué tanta planeación hacia adelante es requerida?
- ¿Qué restricciones se tienen?

## **1.3. ALGUNAS TÉCNICAS DE LA INGENIERÍA DE SOFTWARE, MÉTODOS Y MARCOS METODOLÓGICOS**

### **1.3.1. Métodos de Desarrollo de Software**

Los métodos de desarrollo de software son un “enfoque sistematizado para el desarrollo de software cuyo propósito es establecer formas de trabajo estandarizadas y compartidas por los miembros del equipo de desarrollo con el objetivo de facilitar la producción de software de alta calidad con características predecibles y medibles” [24].

Los métodos de desarrollo de software son originados a partir de los modelos de programación y son soportados en gran medida por principios de ingeniería, cada vez que surge un nuevo método toma todas las bondades de los métodos anteriores [25].

#### **1.3.1.1. Enfoque Estructurado**

Este es el primer enfoque que surgió, acá comenzamos a hablar de funciones y procedimientos que se conectan con datos para poder realizar la ejecución de los programas. Los procedimientos o funciones se comunican entre si y la característica más especial de este método es que todo el código se hace de manera secuencial.

#### **1.3.1.2. Enfoque Orientado a objetos**

En este enfoque hablamos de objetos, a los cuales se les da una caracterización muy parecida a la realidad, los objetos se comunican por medio de mensajes y su

programación no es secuencial. En este enfoque comenzamos a hablar de hilos y procesos.

### **1.3.1.3. Programación Orientada a Componentes**

En este enfoque se comienza a hablar de contenedores y componentes dentro de estos, es un método que facilita la separación entre la interfaz y la implementación. Se caracteriza por su base en el enfoque orientado a objetos y por tener una integración de muchos de los elementos de esta metodología dentro de los componentes.

### **1.3.1.4. SOA**

La Arquitectura Orientada a Servicios (en inglés Service Oriented Architecture), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio.

Permite la creación de sistemas altamente escalables que reflejan el negocio de la organización, a su vez brinda una forma estándar de exposición e invocación de servicios (comúnmente pero no exclusivamente servicios Web), lo cual facilita la interacción entre diferentes sistemas propios o de terceros.

SOA define las siguientes capas de software:

- De aplicaciones básicas - Sistemas desarrollados bajo cualquier arquitectura o tecnología, geográficamente dispersos y bajo cualquier figura de propiedad.
- De exposición de funcionalidades - Donde las funcionalidades de la capa aplicativas son expuestas en forma de servicios (servicios Web).
- De integración de servicios - Facilitan el intercambio de datos entre elementos de la capa aplicativa orientada a procesos empresariales internos o en colaboración.
- De composición de procesos - Define el proceso en términos del negocio y sus necesidades, y que varía en función del negocio.
- De entrega - Los servicios son desplegados a los usuarios finales.

SOA proporciona una metodología y un marco de trabajo para documentar las capacidades de negocio y puede dar soporte a las actividades de integración y consolidación [26].

### **1.3.1.5. Desarrollo Orientado a Aspectos**

Es la última metodología que surgió y su entendimiento es el más complejo. Para definirlo vamos a usar una definición de la Doctora Raquel Anaya “El desarrollo orientado a aspectos es una aproximación que permite analizar los asuntos de interés de un sistema (concerns) y detectar la funcionalidad transversal (crosscutting concerns) en más de un punto del sistema” [27].

### **1.3.2. Marcos Metodológicos**

El marco metodológico es la referencia de las actividades que se van a realizar durante el desarrollo de una aplicación y se usa junto un modelo de ciclo de vida de software y un método de desarrollo. Algunos marcos metodológicos son:

#### **1.3.2.1. RUP**

El Proceso Unificado de Rational (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP no es un marco metodológico con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

El RUP está basado en 5 principios clave que son [28]:

- Adaptar el proceso.
- Equilibrar prioridades.
- Demostrar valor iterativamente.
- Elevar el nivel de abstracción.



- Enfocarse en la calidad.

### **1.3.2.2. AUP**

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Describe de una manera simple y fácil la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas (test driven development - TDD), Modelado Ágil, Gestión de Cambios Ágil, y Refactorización de Base de Datos para mejorar la productividad [29].

### **1.3.2.3. Programación extrema XP**

Es un enfoque de la ingeniería de software formulado por Kent Beck, Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto del ciclo de vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos [30].

Principios de la programación extrema:

- Simplicidad.
- Comunicación.
- Retroalimentación.
- Coraje o Valentía.

- Respeto.

#### **1.3.2.4. CRYSTAL**

Crystal es una familia de métodos que comparten principios en común. Estos principios comunes, o “código genético” (como lo llama su creador Alistair Cockburn), se enfocan en las entregas frecuentes, la comunicación cercana con el usuario/cliente y la mejora a través de la reflexión. Existen diferentes métodos Crystal para diferentes tipos de proyectos, y las organizaciones pueden personalizar un proceso específico para cada proyecto.

Las prioridades que comparten todos los métodos Crystal son:

- Seguridad en el desenlace del proyecto.
- Eficiencia en el desarrollo.
- Habitabilidad de las reglas (el equipo se siente cómodo con ellas).

El énfasis de las metodologías Crystal está en la comunicación y la cooperación entre las personas. De hecho, a diferencia de otros métodos que se centran en arquitectura, en los procesos, o incluso en las herramientas, podríamos decir que Crystal está “centrado en personas” (people-centric) [31].

#### **1.3.2.5. AM –Agile Modeling**

Modelado Ágil (AM) es una metodología basada en la práctica para modelado efectivo de sistemas de software. La metodología AM es una colección de prácticas, guiadas por principios y valores que pueden ser aplicados por profesionales de software en el día a día. AM no es un proceso prescriptivo, ni define procedimientos detallados de como crear un tipo de modelo dado. En lugar de eso, sugiere prácticas para ser un modelador efectivo. La idea es pensar en AM como un arte, no como una ciencia.

AM tiene tres objetivos [32]:

- Definir y mostrar como implementar una colección de valores, principios y prácticas que conlleven a un modelado ligero efectivo.
- Explorar la aplicación de técnicas de modelado en proyectos de software a través de un enfoque ágil, tal como XP, DSDM o SCRUM.
- Explorar como mejorar el modelado bajo procesos prescriptivos, tales como el Proceso Rational Unificado (RUP) o el Proceso Unificado Empresarial (EUP)

#### **1.3.2.6. SCRUM**

Es un proceso de desarrollo de software iterativo y creciente utilizado comúnmente en entornos basado en el desarrollo ágil de software.

Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en Scrum son el ScrumMaster, que mantiene los procesos y trabaja de forma similar al director de proyecto, el ProductOwner, que representa a los stakeholders (clientes externos o internos), y el Team que incluye a los desarrolladores [33].

#### **1.3.2.7. OSS**

En términos generales, se habla de software de código abierto o de código fuente abierto (open source software, o OSS por sus siglas en inglés) cuando el código fuente del software (objetos de código computacional que dan origen a los programas de ordenador, escritos en un lenguaje de programación como C, C++, Java y PHP, que son ininteligibles para los humanos) se encuentra disponible para ser consultados y/o modificados por cualquier persona interesada; el código fuente está “abierto” o “a la vista” [34].

### **1.3.3. Técnicas**

Ahora veamos algunas técnicas utilizadas en el desarrollo de software.

#### **1.3.3.1. UML**

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentarlo y construirlo. En otras palabras, es el lenguaje en el que está descrito el modelo [35].

## **1.4. REPORTE CHAOS**

En 1986, el presidente de la Corporación Transarc participo en la publicación de un artículo. El objetivo de este era hacer una analogía entra la construcción de puentes y el desarrollo de software.

La hipótesis consistía en que los puentes se construyen generalmente en el tiempo establecido, el presupuesto asignado, y generalmente no se caen. Todo lo contrario pasa en el desarrollo de software, ya que en el desarrollo de software la mayoría de las veces no se desarrolla en el tiempo establecido, no se acierta en el presupuestado planeado y por lo tanto se fracasa. Algunas de las principales razones para que exista una diferencia tan marcada, es que para construir un puente se hace un diseño

detallado, adicionalmente cuando un puente se cae se analizan los posibles errores cometidos, aprendiendo de estos para futuros proyectos. Desafortunadamente en el desarrollo de software pasa todo lo contrario, se ocultan los fracasos y errores cometidos, y el peor error que cometemos es que no aprendemos de los problemas o dificultades de proyectos pasados. [36] Basado en lo anterior, The Standish Group ha venido realizando estudios desde 1994. Este estudio es sobre las características que hacen que un proyecto tenga éxito o fracase. Para el caso de los proyectos que fracasan, el objetivo es identificar las posibles causas y definir los elementos que pueden llevar a que el proyecto sea exitoso. Este informe detalla los resultados de más de 50.000 proyectos terminados de TI, además de 450 talleres y otros grupos de interés. Esta investigación ha sido la más grande en la historia de la industria de software.

Hay que resaltar que hoy en día las empresas de desarrollo de software están haciendo un gran esfuerzo en mejorar la productividad y calidad en el desarrollo de software. Basándose en las buenas prácticas, la estandarización y la utilización de aplicaciones de modelos de mejoramiento de procesos, Como:

- Capability Maturity Model Integration (CMMI) [37], es un modelo para la mejora de procesos que proporciona a las organizaciones los elementos esenciales para procesos eficaces.

Las mejores prácticas CMMI se publican en los documentos llamados modelos. En la actualidad hay dos áreas de interés cubiertas por los modelos de CMMI: Desarrollo y Adquisición.

- La Biblioteca de Infraestructura de Tecnologías de Información, frecuentemente abreviada ITIL [38] (del inglés Information Technology Infrastructure Library), es un marco de referencia de las mejores prácticas destinadas a facilitar la entrega de servicios de tecnologías de la información (TI). ITIL resume un extenso conjunto de

procedimientos de gestión ideados para ayudar a las organizaciones a lograr calidad y eficiencia en las operaciones de TI. Estos procedimientos son independientes del proveedor y han sido desarrollados para servir como guía que abarque toda infraestructura, desarrollo y operaciones de TI.

- ISO/IEC 15504 [<sup>39</sup>], modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software.

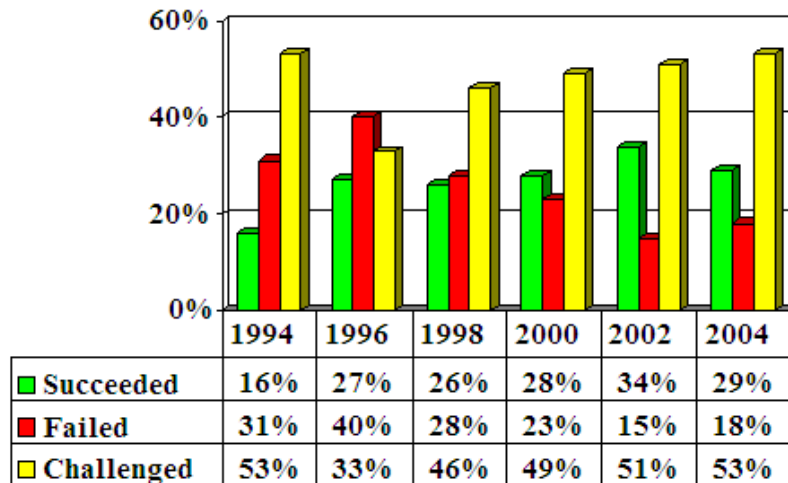
The Standish Group clasifica los proyectos en tres categorías: [<sup>40</sup>]

- Proyectos Exitosos (Successful): Aquellos que son completados en el tiempo estimado, con el presupuesto, con todas las funcionalidades originalmente requeridas.
- Proyectos Retadores (Challenged): Aquellos completados y en operación pero con retrasos de implementación, por encima del presupuesto y/o con menos funcionalidades de las requeridas inicialmente.
- Proyectos Fracasos (Failed): Aquellos proyectos cancelados antes de estar en la etapa de Terminado, es decir que nunca fueron implementados.

A la clasificación anterior, hay que adicionarle que el éxito o fracaso de un proyecto también depende de la utilidad luego de ser implementado en producción. El éxito en la etapa de implementación puede medirse con base en los tres elementos principales en la disciplina de Manejo de Proyectos (Costo, Calidad y Tiempo).

En la siguiente grafica y según el Standish Group podremos observar los cambios que se han venido efectuando en el desarrollo de software año tras año. Aunque el éxito se ha ido incrementado en los proyecto, no se ha incrementado en la medida esperada.

**Figura 7.** Los reportes “The Standish Group - Chaos” desde 1994 hasta el 2004



**Fuente:** SQA - Center of Excellence. "CHAOS REPORT" from STANDISH Group [<sup>41</sup>]

Además, podemos observar que en el año 2002 se tuvo un porcentaje mayor de proyectos exitosos, pero desafortunadamente para los siguientes años este porcentaje se redujo.

## Reporte 2006

Figura 8. Standish Group Chaos Report - 2006



**Fuente:** LAMENCA, José Ramón. Modelos de mejora de procesos de software. [42]

- De una inversión en proyectos de \$255 billones, se desperdician \$55 billones.
- De cada 100 proyectos, 94 se reinician.
- Al terminar un producto, tan sólo están incluidas el 52% de las funciones y propiedades requeridas y/o solicitadas.
- En promedio los costos de un proyecto se elevan en un 143% más de lo estimado.

La perspectiva que tiene el reporte del Grupo Standish, es que va aumentar los niveles de éxito de los proyectos. Por que día tras día se esta trabajando más en la implementación de nuevas metodologías, nuevas prácticas, nuevos modelos.



## **1.5. OTROS REPORTES**

Para tener varios puntos de vistas de como realizar nuestro proyecto hemos optado por revisar otros reportes que nos pueden servir como guía adicional al CHAOS Report. La idea además era encontrar reportes de este estilo a nivel mundial y en nuestro medio. Efectivamente hay algunos estudios de este tipo que hemos encontrado a nivel mundial, aunque en nuestro medio no encontramos mucho, por lo que confiamos en que nuestro proyecto puede ser el inicio de una evaluación que se realice de ahora en adelante para nuestras compañías.

### **1.5.1. Historia de Otros Reportes**

En 1968, se comienzan a distribuir los primeros computadores y adicionalmente el desarrollo de software crece rápidamente, generando “la crisis del software”. De este término se hablo por primera vez en una la conferencia en Alemania, este se dio debido a la gran cantidad de problemas que presentaban y hoy en día se siguen presentando, en el desarrollo de software [<sup>43</sup>].

Para resolver los diferentes inconvenientes que se presentan en el desarrollo de software, se hacen la siguiente pregunta: ¿Qué podía hacerse ante una situación en la que los proyectos software tenían un alto riesgo de fracasar? La conclusión a la que llegaron fue que el desarrollo de software debe aplicar los métodos de la ingeniería, como lo hacen en la construcción de hardware, aviones, barcos, puentes o edificios.

### **1.5.2. Reportes Encontrados a nivel mundial**

A continuación vamos a realizar una reseña de cada uno de los reportes que hemos encontrado y que nutren nuestro estudio:

### **1.5.2.1. Informes de GAO (Government Account Office)**

GAO es la oficina de cuentas del gobierno Federal de los Estados Unidos. Es una oficina que trabaja en conjunto con el senado de Estados Unidos y por esto se han ganado sobre nombres tales como "congressional watchdog" (Vigilante del Congreso) y tiene entre sus objetivos velar por el dinero del Estado para el beneficio de todos los estadounidenses, haciendo de [<sup>44</sup>]:

- Auditores para determinar las inversiones.
- Investigadores en problemas de pleitos ilegales y actividades impropias.
- Reportando como los programas de gobierno van y si están logrando los objetivos.
- Analizando opciones consideradas por el congreso.
- Expidiendo decisiones y opiniones legales.

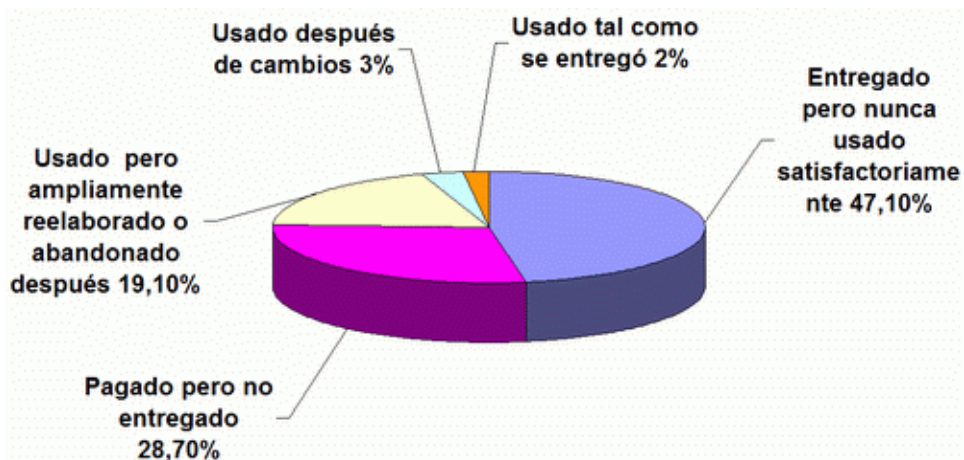
Ellos avisan al congreso y a los altos ejecutivos de las agencias caminos para un mejor comportamiento del Gobierno.

En ese orden de ideas GAO es la entidad que se encarga de estudiar y analizar el gasto público en las inversiones de tecnología, a continuación veamos algunos informes realizados por esta entidad relacionados con tecnologías de información:

- [<sup>45</sup>] En 1979, El GAO realizo un estudio en Estados Unidos, en el que selecciono nueve proyectos de desarrollo de software. La suma de los contratos de los proyectos seleccionados sumaban 6.800.000 dólares.

119.000 dólares del total de los contratos, era de un proyecto que se había utilizado tal cual como se había entregado. El resto de los 6.800.000 dólares fueron invertidos en proyectos cancelados o no satisfactorios.

**Figura 9.** Resultados del Informe de GAO [46]

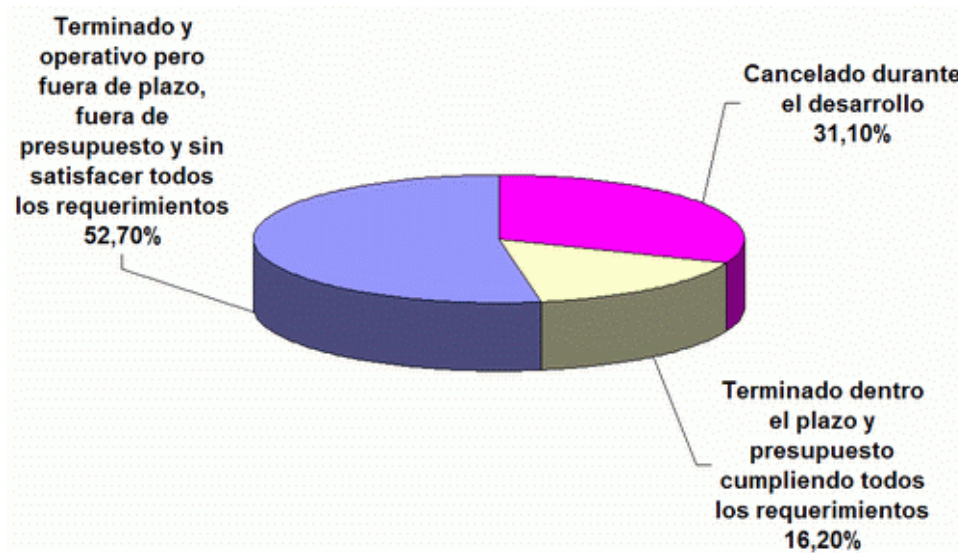


**Fuente:** Government Account Office, GAO [GAO, 1979].

En 1995, el Grupo Standish realizó un estudio, el informe CHAOS, mucho más amplio y significativo que el del GAO cuyos resultados, a pesar de haber pasado más de 25 años, no reflejaban una mejoría sustancial en el desarrollo de proyectos [TSG, 1995].

Los resultados generales, que pueden verse en la figura 9, si se compara con los de [GAO, 1979] presentan una mejora en los proyectos que se entregan cumpliendo todos sus requisitos, 2% frente al 16.2%, sólo el 9% en grandes compañías, pero empeoran ligeramente respecto a los que se abandonan durante el desarrollo, 28.7% frente a 31.1%.

**Figura 10.** Resultados del informe CHAOS [47]



**Fuente:** Grupo Standish [TSG 1995]

- Information Technology Managment: Es un informe que surgió debido a leyes promulgadas por el congreso y se busca tener una guía de desarrollo de planes estratégicos para lograr los resultados deseados por las compañías y tener un control por parte de ellas en la inversión de tecnología. Se habla para cada uno de los sectores económicos y se les da su recomendación particular. Adicionalmente tiene en su parte inicial una gráfica donde nos muestra las agencias que utilizan las prácticas de la planeación estratégica / performance. [48]
- OMB Can More Effectively use its Investment Review: Es un documento que detalla las inversiones federales en Estados Unidos para el año 2005 que fueron de más de 60 billones de pesos. Se encontró efectividad en la inversión; sin embargo en este documento se hace un énfasis para lograr una optimización en el proceso, con una lista de chequeo que busca tener un mayor control sobre la inversión en tecnología. Se

puede consultar en la siguiente dirección <http://www.gao.gov/new.items/d05571t.pdf>  
Año 2005. [49]

Los reportes de GAO son reportes en los que el contexto de la investigación es 100% en inversiones públicas y tienen un componente diferente al de CHAOS en cuanto GAO puede obtener una información más certera y precisa de los proyectos en tecnología en el sector público. Adicional, si observamos estos informes se hace un énfasis mayor en los procesos y en el como hacer las cosas, mientras que el CHAOS es un reporte de como se hicieron los proyectos y cuales fueron los factores de fracaso.

De estos informes GAO hemos decidido tomar la forma en la que analizan la información después de tener los datos y la estructura de sus documentos. Nos llamo mucho la atención porque siempre enfatizan al principio del documento un objetivo claro para que el lector no tenga duda alguna del contenido del informe\*.

### **1.5.2.2. Informes de FINANCIAL EXPRESS**

Financial Express, es un portal de noticias Indio que está asociado con la Assian CERC donde se da información actualizada de cerca de 550 compañías. Para muchos lectores Financial Express es una fuente verídica debido a su seriedad.

---

\* Para consultar mas acerca de los informes de GAO visitar el link [http://www.gao.gov/docsearch/locate?s=searched=1&o=10&order\\_by=rel&old\\_keyword=technology&ft=&search\\_type=publications&add\\_topic=&remove\\_topic=&add\\_type=&remove\\_type=&add\\_fed\\_type=&remove\\_fed\\_type=&add\\_fed\\_desc=&remove\\_fed\\_desc=&add\\_year=&remove\\_type=&keyword=technology](http://www.gao.gov/docsearch/locate?s=searched=1&o=10&order_by=rel&old_keyword=technology&ft=&search_type=publications&add_topic=&remove_topic=&add_type=&remove_type=&add_fed_type=&remove_fed_type=&add_fed_desc=&remove_fed_desc=&add_year=&remove_type=&keyword=technology)

Financial Express ha dedicado una porción de sus reportes a sistemas como lo siguientes:

- **Tectonic shift seen in IT services next decade:** Este es un reporte muy optimista en el que se muestra que la industria de TI para el 2020 va a crecer en un 80% en ingresos según el presidente de Nasscom, aunque con la actual crisis no se sabe como va a impactar el crecimiento presupuestado por el presidente de Nasscom. Este reporte se basa en la opinión del presidente de Nasscom, en nuevas tendencias de leyes y en algunos estudios con el gobierno Indio, lo que hemos visto con muy buenos ojos de este reporte es la proyección de personas expertas, esto nos puede ayudar en la parte donde los expertos hacen un análisis de los datos recolectados, principalmente porque allí lo que buscamos es una interpretación que permita descubrir tendencias a futuro. Adicionalmente, el informe nos da un pequeño análisis de lo que dice el experto contextualizándolo a hoy lo que da un conocimiento mayor del que tenemos solo con el análisis del experto. [<sup>50</sup>]
- **China to be world's top outsourcing destination:** Este es un reporte interesante del 2005 en el que se muestra la tendencia del Outsourcing de las empresas Estadounidenses en diferentes partes del mundo, así como la utilización del offshoring. Según el informe que como el anterior es basado en tendencias y en la voz de un experto, la India en ese entonces era el destino por excelencia para el outsourcing y según estimaciones en el 2015 el país número uno en Outsourcing sería la China. El reporte en el que se basaron es de Frank-Jrgen Richter, presidente de Horasis. Es interesante saber que el top 5 de países se encuentra Costa Rica como representante latinoamericano, China, India como la principal, República Checa y Hungría, pero las proyecciones para el futuro son China como la principal, India, USA, Brasil y Rusia. [<sup>51</sup>]

Lo que pudimos apreciar en los reportes y noticias de The Financial Express es que se encargan de poner a un experto a sustenten sus documentos, debido a que consultan

fuentes secundarias o porque realizan una entrevista. Esto es muy interesante para nuestro esquema porque la idea es tener la posibilidad de apoyar nuestro estudio en mínimo uno o más expertos para tener una mejora sustancial en nuestro informe.

### **1.5.2.3. Informes ESPITI (European Software Process Improvement Training Initiative)**

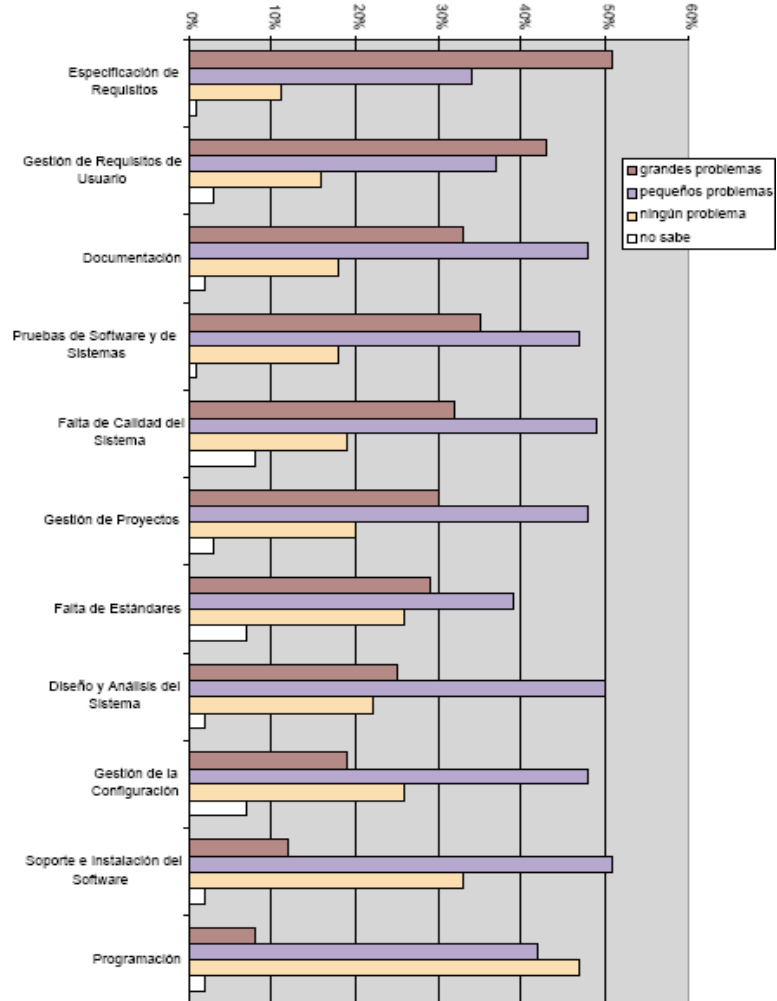
ESPITI (European Software Process Improvement Training Initiative) es un programa de la Unión Europea para mejorar la toma de conciencia acerca de los procesos de software.

En 1996, ESPITI realizó una investigación con el mismo objetivo que los estudios de GAO y CHAOS. Buscaba identificar los principales problemas en el desarrollo de software a nivel europeo. Los resultados arrojados por esta investigación fueron muy similares a los resultados de CHAOS. Mostraron que la mayoría de los problemas provenían de la especificación, la gestión y la documentación de los requisitos de software.

Estos resultados demuestran que aunque las herramientas para construir software han evolucionado considerablemente, hoy en día se sigue produciendo software que no es satisfactorio para clientes y usuarios. Esto indica que los principales problemas que han dado origen a la crisis del software persisten en las primeras etapas del desarrollo, cuando hay que decidir las características del producto software a desarrollar.

Con este estudio también se demostró que un cambio en los requisitos, cuando ya se tiene listo el producto, implica altos costos y mucho esfuerzo para adaptarlo al sistema. El costo es entre 60 y 100 veces superior a si se hubiera corregido en etapas tempranas del desarrollo [<sup>52</sup>].

**Figura 11. Resultados del informe ESPITI**



**Fuente:** Informe ESPITI



## GLOSARIO

**ARQUITECTURA DE SOFTWARE:** se refiere a un grupo de abstracciones y patrones que nos brindan un esquema de referencia útil para guiarnos en el desarrollo de software dentro de un sistema informático.

**COMUNICACIÓN:** es un proceso de transmitir ideas, o bien símbolos, que tienen el mismo significado para dos o más sujetos los cuales intervienen en una interacción.

**ELICITACIÓN:** obtener información necesaria para un proyecto.

**ESTRATEGIA:** diseña planes de acción efectivos para alcanzar las metas.

**ESTRUCTURA ORGANIZACIONAL:** estructura interna de autoridad y comunicación de una organización que define la forma como deben manejarse los programas y departamentos.

**EXPERTO:** es una persona que tiene conocimiento y experiencia en una actividad o campo de la técnica.

**GERENCIA DE PROYECTOS:** es la ciencia y el arte de administrar, dirigir y coordinar el talento humano, los recursos económicos, los recursos materiales y los recursos logísticos e informáticos para lograr objetivos y resultados previamente determinados, mediante la ejecución de un proyecto específico.

**HIPÓTESIS:** una hipótesis puede definirse como una solución provisional (tentativa) para un problema dado.

**INTERRELACIÓN:** es la asociación o conexión entre conjuntos de entidades.

**IDÓNEO:** Adj. Que tiene competencia para una cosa.

**MÉTODOS ÁGILES:** formas de trabajar más rápida que las demás.

**PLANIFICACIÓN:** predeterminación de un curso de acción para alcanzar los objetivos organizacionales.

**RIESGO:** La posibilidad de sufrir un daño o pérdida; peligro.

**REQUISITOS:** es una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio.

**ROL/ROLES:** papel que desempeña alguien en un análisis.

**SEI:** es un centro de investigación y desarrollo patrocinado por el Departamento de Defensa de los Estados Unidos de América y gestionado por la Universidad Carnegie-Mellon. "CMMI" es una marca registrada del SEI.

# 1. MARCO TEÓRICO

Debido a los grandes retos que enfrentan las compañías desarrolladoras de software de nuestra ciudad y a las exigencias de sus clientes, hemos visualizado que existe en el medio una gran necesidad de medir el estado de los proyectos ejecutados en nuestra ciudad, y que las conclusiones permiten alcanzar niveles de calidad más altos en el desarrollo de software y también conocer los posibles puntos débiles y sus fortalezas.

Por tal motivo hemos decidido crear un reporte llamado "**REPORTE MADI**" en el cual queremos explorar a modo general las tendencias de ejecución de todas las compañías desarrolladoras de nuestra ciudad en los proyectos de software y como estos proyectos llegan a ser exitosos o no exitosos. Para poder lograr esto hemos adoptado una metodología que tiene cierta similitud con un reporte llamado CHAOS que hace este tipo de estudios, pero vamos a tener unas ligeras modificaciones porque hemos encontrado otros reportes y notas que nos han enriquecido. Ahora vamos a hablar de nuestro estudio y su estructura.

## 1.1. ESTRUCTURA DEL REPORTE MADI

El Reporte MADI comienza con una encuesta desarrollada que pregunta acerca de los siguientes ítems:

- Usuario(s) Líder(es) del Proyecto: Los expertos de los temas son los usuarios, queremos ver su importancia dentro de los proyectos.
- Soporte de la Alta Gerencia: El compromiso de la alta gerencia de las compañías usuarias con el proyecto es un tema importante.

- Requisitos del Proyecto: ¿Serán los requisitos la guía del proyecto o son simplemente los caprichos de los usuarios?
- Planeación del Proyecto: La planeación es la mejor carta de presentación de como vamos a trabajar.
- Competencias del Equipo de Trabajo: Es indispensable un equipo de trabajo antes que un trabajo grupal.
- Acerca del proyecto: Generalidades del proyecto y cuando lo consideramos exitoso o no.

Le plantearémos a las empresas de desarrollo y/o departamentos o gerencias de informática de compañías reconocidas en nuestra ciudad, participar en la creación del nuevo **"REPORTE MADI"**. Las empresas que pensamos invitar a participar son: Tecnofactor, Dicosoft, AvanSoft, AXEDE, Cognox, Ceiba, MVM, PSL, Insitu, Sonda, Cognox, Websys, Intergrupo, Gerencia de Desarrollo Inversura y Centro de Informática de la Universidad EAFIT.

Luego de que tengamos diligenciadas las encuestas vamos a realizar una tabulación de los resultados donde vamos a desaparecer los nombres de los proyectos y de las compañías que nos colaboraron y vamos a dejar los datos compactados como si fueran uno solo, con esto pretendemos dos cosas:

- Lograr los niveles de confidencialidad acordados con todas las empresas que nos colaboraron.
- Poder generalizar la información para que todo quede apuntando solo a un rubro y dar mayor facilidad en el manejo de los datos y la información.

Ya con los datos tabulados realizaremos el análisis de los resultados arrojados por la investigación. Plasmaremos la información más relevante generada por la tabulación de los datos y adicionalmente comenzamos a introducir como herramienta todo el análisis estadístico.

Cuando tengamos un análisis completo de la tabulación, consultaremos un experto para que nos ayude a interpretar los resultados obtenidos. Ya teniendo clara las ideas del experto e investigando sobre el tema, procederemos a redactar el “**Reporte MADI**”.

## **1.2. BENEFICIARIOS**

Con este proyecto se beneficiaran las siguientes personas y entidades:

1. Las compañías desarrolladoras de software, que tendrán la oportunidad de visualizar de manera macro las tendencias de la ciudad en cuanto al desarrollo de software, lo que les va a permitir lograr un mejoramiento continuo en sus proyectos.
2. Los usuarios de las compañías desarrolladoras, debido a que mejoran los procesos y sus solicitudes de una manera u otra pueden ser escuchadas con mayor atención por parte de los desarrolladores y analistas.
3. Los estudiantes de ingeniería de sistemas, porque tendrán una fuente de consulta del estado actual de las empresas en las que aspiran trabajar o las que serán su posible competencia en el futuro.
4. La Universidad EAFIT, gracias a que este proyecto quedará como material de consulta en la biblioteca, lo que enriquecerá su capital intelectual.
5. Otros estudiantes, ya que este proyecto puede ser el inicio de otros trabajos o proyectos de grado.

### **1.3. PLANTEAMIENTO DE HIPÓTESIS**

Tenemos dos hipótesis por las que consideramos que actualmente los proyectos fracasan o no son 100% exitosos:

- Mala gestión sobre los usuarios que participan en el proyecto.
- Metodologías utilizadas en el desarrollo del proyecto.

#### **1.3.1. Mala Gestión de Usuarios**

Pensamos que los proyectos tienden a fracasar por que durante el desarrollo del mismo no se tiene en cuenta al/los usuario(s), quienes son finalmente, los que tienen el conocimiento del negocio.

La etapa más crucial de la participación de los usuarios en el desarrollo de un proyecto es la elicitación de requisitos, pues es aquí en donde debe haber una estrecha comunicación entre usuario y desarrollador. En esta etapa el usuario expresa todas sus necesidades en su lenguaje natural, el cual posteriormente es traducido en un lenguaje técnico por un ingeniero de requisitos.

Lo anteriormente mencionado no limita la participación de los usuarios en las otras etapas del proyecto, ya que él es el responsable de tomar las decisiones de los cambios funcionales del software.

Como conclusión, la oportuna y constante participación de los Usuario es la clave principal para que el desarrollo se termine con éxito.

#### **1.3.2. Metodologías utilizadas en el desarrollo del proyecto**

El otro gran problema para nosotros en los proyectos que se desarrollan actualmente son los esquemas de trabajo adoptados. En este aspecto se nos ha olvidado la tarea de

hacer la selección óptima entre método, marco metodológico, modelos de ciclo de vida y técnicas.

Consideramos que en nuestras compañías se utiliza con mayor frecuencia la siguiente configuración:

- Modelo de Ciclo de Software: Cascada.
- Método: Enfoque orientado a objetos.
- Marco metodológico: RUP.
- Técnicas: UML.

Nuestra primera hipótesis es que el ciclo de vida utilizado actualmente es muy lento (no por esto es malo) y las necesidades actuales nos dictan que debe ser mucho más ágil, como con los ciclos actuales debemos esperar a que se acaben todas las etapas para ver como funciona el sistema. La segunda es que el método aunque es muy bueno, ya debe tomarse un método más avanzado que nos permita mayor agilidad y podamos reutilizar todo lo desarrollado anteriormente. La tercera hipótesis es el marco metodológico, esta comprobado en la experiencia de casi todos los analistas y desarrolladores de la ciudad que este es marco tiende al fracaso y por esta razón empresas como Google, AT&T, Microsoft y SUN Microsystems están usando marcos más ágiles como Crystal y XP.

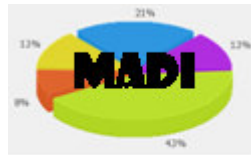
Finalmente, se encuentra como técnica UML que es una buena forma de entender muchos elementos de los desarrollos por lo que consideramos que es una buena puesta a punto; sin embargo es de resaltar que muchas personas no conocen en detalle el significado de muchos objetos contenidos allí por lo que nuestras empresas fallan en su uso.

Dado que esta configuración de nuestras compañías que no es la óptima, nosotros consideramos una mejor configuración que nos ayudaría a mejorar nuestra reputación y nuestro trabajo:

- Modelo de Ciclo de Software: Iterativo, Evolutivo o prototipado.
- Método: SOA o Enfoque Orientado a Aspectos.
- Marco metodológico: XP, Crystal o Scrum.
- Técnicas: UML en caso de ser necesario.

Esto no quiere decir que no se deben utilizar otras configuraciones porque esto depende del proyecto, sin embargo para el mercado local la configuración expuesta podría ayudarles para mejorar su esquema de trabajo.





## 2. DESARROLLO DE LA PROPUESTA

### 2.1. INTRODUCCIÓN ENCUESTA

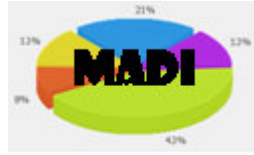
#### REPORTE MADI (Basado en el Reporte CHAOS)

“Standish Group” es una compañía Norteamericana enfocada a ayudar a las compañías de tecnología a tener éxito en sus desarrollos tecnológicos. Para lograr este objetivo existen una serie de reportes estadísticos y analíticos entre los cuales encontramos el reporte CHAOS. El reporte CHAOS consiste en el estudio de los factores de éxito, problemas o fracasos en proyectos de Software y es un estudio que se basa en una detallada encuesta con los directivos de las grandes empresas desarrolladoras de Estados Unidos. Luego de ser realizada la encuesta se organiza la información para generar un completo documento donde se analiza la situación general de la calidad con la que fueron ejecutados estos proyectos, lo interesante es que hace un análisis de los motivos que originaron los “fracasos”, permitiéndonos poner foco en ellos al dirigir nuevos proyectos. En el reporte CHAOS no se muestra información confidencial de las empresas que participan en el estudio, solo se muestra información forma general.

El reporte MADI lo que intenta es investigar y documentar en las empresas de desarrollo de nuestra ciudad, las causas generales por las que un proyecto informático puede llegar a fracasar o ser un éxito. Simulando en cierta forma lo que Standish Group ha realizado en Estados Unidos con el reporte CHAOS. Es importante que los encuestados contesten basado en lo que ocurrió en la realidad con los proyectos, sin ocultar ningún detalle, para así tener una muestra más confiable.

Cada una de las empresas que hagan parte de este proyecto va a tener el informe final.

***La información suministrada por cada una de las empresas es confidencial y no se hará pública para cuidar sus intereses.***



## 2.2. ENCUESTA

### 1 Usuario(s) Líder(es) del Proyecto.

1.1 ¿Durante todo el tiempo que duro el proyecto estuvo involucrado el usuario líder?

SI

NO

En caso de que su respuesta sea negativa indicar en que fases del proyecto estuvo presente.

1. \_\_\_\_\_

2. \_\_\_\_\_

3. \_\_\_\_\_

1.2 ¿Cómo fue la relación durante el proyecto con los usuarios?

Excelente

Buena

Regular

Mala

1.3 ¿En general considera que los usuarios fueron los correctos en el desarrollo de este proyecto?

SI

NO

¿Por qué?

Expertos en el proceso a automatizar

SI

NO

Tenían tiempo disponible para el proyecto

SI

NO

Tomaban decisiones

SI

NO

Expresaban claramente sus requisitos

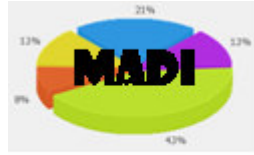
SI

NO

Estaban comprometidos con el proyecto

SI

NO



1.4 ¿Involucrar a los usuarios fue tarea fácil?

SI

NO

En caso de que su respuesta sea negativa poner las razones por las que no fue fácil involucrarlos.

1. \_\_\_\_\_

2. \_\_\_\_\_

3. \_\_\_\_\_

## 2 Soporte de la Gerencia

2.1 ¿Contó con el apoyo de un (unos) directivo(s) de la empresa usuaria durante el desarrollo del proyecto?

SI

NO

En caso de que su respuesta sea negativa mencionar porque no hubo apoyo.

1. \_\_\_\_\_

2. \_\_\_\_\_

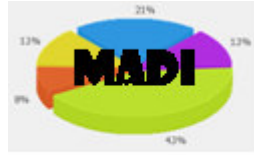
3. \_\_\_\_\_

2.2 ¿El directivo encargado mostró interés en los resultados del proyecto?

SI

NO

\_\_\_\_\_  
\_\_\_\_\_



2.3 En alguna fase del proyecto ¿Se presentaron Inconvenientes?

SI

NO

¿Como los manejo el equipo de trabajo?

- a. Si se presentaron inconvenientes y se superaron.
- b. Si se presentaron inconvenientes, se superaron y se aprendió de ellos para futuros proyectos.
- c. Si se presentaron inconvenientes, pero se ocultaron.

### 3 Requisitos del Proyecto

3.1 ¿Se tenía definida una visión (Alcance) en el proyecto?

SI

NO

3.2 ¿La definición de los usuarios de sus necesidades fue?

Excelente

Buena

Regular

Mala

3.3 ¿Se contemplo desde el inicio del proyecto una evaluación de riesgos?

SI

NO

---

---

3.4 ¿Fue el proyecto medible o bien dimensionado desde su inicio?  
(Según la definición de los requisitos)

Tiempo

SI

NO

Costos

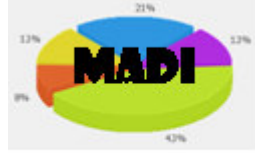
SI

NO

Recursos (Maquinas, personas)

SI

NO



3.5 ¿Cree que fue correcta la fase de elicitación de requisitos?

SI

NO

En caso de ser negativa la respuesta indique las posibles causas de los inconvenientes.

1. \_\_\_\_\_

2. \_\_\_\_\_

3. \_\_\_\_\_

3.6 El equipo de trabajo ¿Priorizó correctamente los requisitos que se definieron en el proyecto?

SI

NO

\_\_\_\_\_  
\_\_\_\_\_

3.7 ¿Se hizo un prototipo ó interfaz grafica del proyecto en esta etapa inicial?

SI

NO

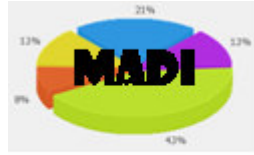
#### 4 Planeación del Proyecto

4.1 ¿Se tenía un plan definido de trabajo para el proyecto?

SI

NO

\_\_\_\_\_  
\_\_\_\_\_



4.2 ¿Involucraba constantemente este plan a los usuarios?

SI

NO

4.3 ¿Se hizo una definición clara de las fases del proyecto?

SI

NO

---

---

4.4 ¿Se manejo durante el proyecto la gestión de cambios?

SI

NO

---

---

4.5 ¿Se pudo medir el progreso del proyecto en cualquier momento de este?

SI

NO

4.6 ¿La definición de los tiempos límites para cada una de las fases fue?

Excelente

Buena

Regular

Mala

¿Se cumplieron estos tiempos?

Siempre

A veces

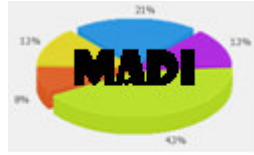
Nunca

4.7 ¿Se realizo una definición clara de los Roles en el proyecto?

SI

NO

En caso de que su respuesta sea negativa mencione los aspectos que se pueden mejorar.



1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_

4.8 ¿Cada integrante del proyecto tenía claro conocimiento de su rol?

En caso de que su respuesta sea negativa mencione las causas por las que los integrantes del equipo no estuvieron enterados

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_

4.9 ¿Se realizó un seguimiento periódico del avance del proyecto?

¿Con que periodicidad?

Diario

Semanal

Quincenal

Mensual

4.10 ¿En el desarrollo del proyecto se utilizó una ó varias metodologías y/o técnicas propias de la ingeniería de software?

SI

NO

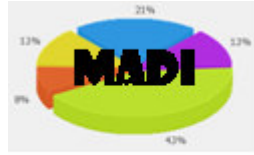
¿Si se utilizó una metodología(s) y/o técnica(s) decir cual ó cuales?

RUP

CRYSTAL

AUP

AM –Agile Modeling



CASCADA

X UML

XP

SOA

SCRUM

OSS

Otra. Cual(es) \_\_\_\_\_

\_\_\_\_\_

## 5 Competencias del Equipo de Trabajo

5.1 ¿Las habilidades de los integrantes del equipo de desarrollo fueron?

Excelente

Buena

Regular

Mala

5.2 ¿El equipo de trabajo fue conformado adecuadamente? (Analistas de desarrollo, calidad, arquitectura, diseñadores, entre otros).

SI

NO

En caso de que su respuesta sea negativa, ¿Por qué cree usted que fue mal conformado?

1. \_\_\_\_\_

2. \_\_\_\_\_

3. \_\_\_\_\_

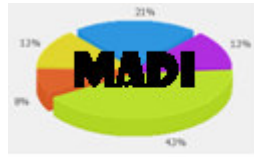
5.3 Durante la ejecución del proyecto ¿Se necesito de un programa de capacitación para incrementar las habilidades del equipo de desarrollo y poder dar mejores resultados?

Si

NO

No aplica





5.4 ¿Se otorgaron incentivos al equipo de trabajo por su buen desempeño?

SI

NO

---

---

5.5 Los miembros del equipo de desarrollo ¿estaban comprometidos con el proyecto?

SI

NO

5.6 En caso de que su respuesta sea negativa, indicar razones por las que no estuvieran interesados.

1. \_\_\_\_\_

2. \_\_\_\_\_

3. \_\_\_\_\_

5.7 ¿Se evidencio un trabajo en equipo durante el proyecto?

SI

NO

## 6 Acerca del proyecto

6.1 ¿Esta alineado el proyecto con los objetivos estratégicos de la empresa usuaria?

SI

NO

6.2 ¿Durante el proyecto se definieron indicadores que permitieran medir si los objetivos planteados se lograron o no?

SI

NO

6.3 ¿En que porcentaje fueron alcanzados los objetivos planteados en el proyecto?

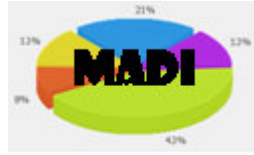
a) 100% - 80%

b) 79% - 65%

c) 64% - 50%

d) Menos de 50%

6.4 En general como considera el resultado del proyecto



- a) Totalmente Exitoso.
- b) Exitoso con algunos inconvenientes.
- c) No exitoso: se entrego pero hubo sobrecosto y desfase en el tiempo.

En caso de que su respuesta sea la b, señale los inconvenientes presentados:

- a) Problemas en la planeación
- b) Sobrecosto
- c) Desfase en el tiempo
- d) Falta de Recursos

¿Durante el año 2008 su empresa cancelo algún proyecto?

SI

NO

En caso de que su respuesta sea afirmativa, ¿Cuál(es) fue (ron) el(los) motivo(s) para cancelarlo?

---

---

## **EMPRESAS ENTREVISTADAS EN LA CIUDAD DE MEDELLIN**

En Colombia el sector de la informática está conformado por varias empresas [1]: Empresas desarrolladoras de software, Consultoras de TI, redes y telecomunicaciones, comercializadoras de equipos de cómputo, empresas de testing, integradora de TI, entre otras.

En Colombia este sector ha mostrado un gran crecimiento, por encima de las tasas de crecimiento del país. Actualmente hay más de 3.000 empresas activas que emplean a unas 32.000 personas y generan para el país aproximadamente 158 millones de dólares en impuestos. También han ayudado a reducir la brecha digital.

Con los resultados positivos de este sector, en Colombia hay confianza y muchas posibilidades para crear nuevas empresa, establecer relaciones comerciales con compañías colombianas y/o internacionales que tengan el interés de invertir en el país.

Desde el punto de vista empresarial, el desarrollo de software está concentrado en las microempresas con un 58 por ciento del total del mercado; seguido de las pequeñas empresas, las cuales producen un 34 por ciento. Estos datos denotan un común denominador en la economía colombiana, puesto que son las micro, pequeña y mediana empresa las mayores dinamizadoras de la economía y el desarrollo en todos los sectores económicos.

A pesar del continuo crecimiento del sector del software, Colombia requiere de una transformación que permita exportar aplicaciones de primer nivel. De acuerdo con Fedesoft, menos del 10 por ciento de las empresas de software del país tienen o han tenido experiencias en el mercado internacional, lo cual no ha permitido ser reconocidos

en el comercio exterior como exportadores de software sino como buenos “artesanos de software”.

Como conclusión de estos hechos tenemos un gran reto en nuestro país, exportar e innovar con desarrollos propios dentro del mercado internacional.

De acuerdo con Fedesoft [2], en Colombia solo se ha trabajado el tema de la calidad alrededor de la certificación ISO. No obstante, en los negocios del software hay un estándar de calidad más exigente, el CMMI (Capability Maturity Model Integration). Este posee cinco niveles y solo una empresa de nuestra geografía está certificada en nivel cinco: PSL software, Productora de Software S.A., ubicada en Medellín.

Para convertirnos en un país competitivo en el desarrollo de software y dejar de lado el calificativo de artesanos, necesitamos inicialmente dos cosas importantes:

1. Un estudio que nos permita saber el estado en el que se encuentran nuestras compañías desarrolladoras y que podemos mejorar.
2. Adoptar medidas que nos lleven a lograr los estándares de calidad como CMMI, para que la industria de tecnología en Colombia, se internacionalice.

En nuestro proyecto de grado vamos a trabajar el primer punto y para esto hemos contado con la participación de empresas de desarrollo y departamentos o gerencias de informática de compañías reconocidas y que tienen certificaciones CMMI nivel 3 y 4 u otras certificaciones en Medellín.

A continuación enunciaremos las diferentes empresas que nos brindaron apoyo.

### **2.2.1. Gerencia de Desarrollo Inversura**

La gerencia de Inversura, es un Departamento encargado de cubrir las necesidades tecnológicas e informáticas de las compañías Suramericana de Seguros S.A., Compañía Suramericana de servicios de Salud (Susalud) y Compañía Suramericana de servicios de Riegos Profesionales (Suratep). Estas compañías desde el año 2006 decidieron conformar un solo departamento de Informática que las atendiera, con el fin de reducir costos y tener una mayor productividad que le permita a estas compañías ser competitivas a nivel nacional e internacional.

Esta gerencia está dedicada única y exclusivamente a estas tres compañías y con la unión han logrado consolidar muchos servicios informáticos y generar soluciones cada vez más óptimas.

### **2.2.2. Centro de Informática de la Universidad EAFIT**

Centro de Informática de la Universidad EAFIT es la encargada de prestar todos los servicios informáticos y tecnológicos a todas las áreas de la Universidad, con el fin de optimizar por medio de la tecnología todos los procesos para el mejor funcionamiento de la Universidad.

Este Centro ha logrado grandes avances tecnológicos que ha puesto al servicio de la Universidad. Actualmente algunos aplicativos de la Universidad son vendidos a otras entidades educativas para que mejoren sus servicios.

### **2.2.3. Intergrupo**

Tiene como promesa hacia sus clientes el ser una empresa que provee soluciones que les ayudan a transformar sus operaciones y les permiten construir nuevos modelos de negocios usando tecnologías de punta, permitiéndoles ahorrar costos y mejorar la eficiencia de su negocio.

Su misión es ser líderes en la prestación de servicios de consultoría, desarrollo de software e integración de productos y servicios en tecnología de punta, con un

excelente servicio al cliente, promoviendo el crecimiento profesional y humano de sus colaboradores.

#### **2.2.4. Ceiba**

Esta comprometido con sus clientes en proveerles sistemas de información adecuados a sus necesidades, apoyados en profesionales íntegros y de alta competencia técnica y en procesos estructurados y en constante evolución.

Proveen soluciones a la medida que soportan los procesos de la organización, aplicación de buenas prácticas de ingeniería de software y equipo humano y técnico de primer nivel [3].

#### **2.2.5. MVM Ingeniería de Software S.A**

Proveedor líder de soluciones IT de negocio que permiten a sus clientes aprovechar al máximo las tecnologías de información necesarias para el logro de las metas corporativas. El éxito se basa en la construcción de relaciones sólidas y de largo plazo con sus clientes, manteniendo personal altamente calificado dentro de un excelente clima organizacional. Para atender sus diferentes líneas de servicio, MVM se soporta en las mejores prácticas del mercado y en modelos ya probados en empresas similares y de alta trayectoria en el mercado internacional. Es así como, para el desarrollo a la medida utiliza una metodología basada en el marco metodológico UP (Unified Process), con notación UML (Unified Modeling Language).

Adicionalmente, se respalda en la evolución permanente del sistema de gestión de la compañía, desarrollado sobre CMMI. Para el servicio de soporte, mantenimiento y evolución de aplicaciones, MVM ofrece un modelo de servicio basado en las mejores prácticas de ITIL.

### **2.2.6. Tecnofactor S.A.**

Es una empresa orientada a la prestación de servicios de desarrollo e integración de soluciones en tecnologías de información y comunicación. Específicamente la Empresa ofrece soluciones como el desarrollo de software a la medida, capacitación especializada y consultoría. Aseguran la calidad de sus servicios. Trabajan con metodologías ágiles, flexibles y de alto rendimiento; que facilitan un acompañamiento profesional, con personal capacitado, comprometido y en constante desarrollo. En TECNOFACTOR S.A. están siempre dispuestos a conocer del negocio de nuestros clientes para ser un nuevo integrante en su estrategia de negocio.

### **2.2.7. Sonda**

Es uno de los principales integradores y proveedores de servicios de TI en América Latina, con ventas anuales de más de US\$539,5 millones y una organización de más de 4.500 personas. Su misión es agregar valor a sus clientes, mediante el mejor uso de las tecnologías de información, a través de la provisión de servicios y soluciones de calidad que apoyen su gestión productiva y empresarial. SONDA tiene un extenso alcance en la región, con operaciones en Argentina, Brasil, Chile, Colombia, Costa Rica, Ecuador, México, Perú y Uruguay [4].

### **2.2.8. Cognox Ltda.**

Es una empresa dedicada al desarrollo y mejoramiento de los Sistemas de Información de sus clientes, como una respuesta a la transformación de las necesidades empresariales en nuevos modelos de negocios, apoyados en tecnologías informáticas de punta, basados en la amplia experiencia, trayectoria y conocimiento de su talento humano [5].

Como empresa, les ofrece a sus clientes soluciones acordes a sus necesidades, analizadas, diseñadas y desarrolladas con una metodología de trabajo, que garantiza

un producto final con altos estándares de calidad y funcional para la organización, diseñado con las mejores prácticas de programación y completamente administrable por el usuario final.

### **2.2.9. Insitu**

Insitu Mobile Software S.A. es una empresa dedicada a la prestación y comercialización de servicios informáticos, basados en tecnologías móviles, generando soluciones innovadoras; permitiendo el acceso a información oportuna y en el sitio de trabajo. Un ejemplo común de sus soluciones es el acceso a sistemas de inventarios para la revisión de éste y realización de pedidos al proveedor, todo en un ambiente de movilidad y amplia cobertura.

### **2.2.10. Dicosoft**

Compañía nacida entre 2006 y 2007, que se ha dedicado al desarrollo de software para Universidades, aunque su portafolio de soluciones se aplica a cualquier empresa. Esta es una de las nuevas compañías que busca realizar software para apoyar a la dirección y administración de las organizaciones.

Como producto estrella tiene el sistema Gestor de prácticas profesionales implantado en la Universidad EAFIT, Universidad de Medellín, Institución Universitaria de los Libertadores y la Fundación Universitaria INPAHU [6].



## 2.3. RESULTADO GENERAL

### 2.3.1. Resultados de Encuestas

#### Convenciones

Para comprender la preguntas de nuestra encuesta, vamos a presentarles las convenciones que se deben tener en cuenta:

- Preguntas de Tipo S - N - N/A

**S:** Si

**N:** No

**N/A:** No Aplica

- Preguntas cualitativas

**E:** Excelente

**R:** Regular

**B:** Buena

**M:** Mala

- Preguntas de tipo S – A - N

**S:** Siempre

**A:** A veces

**N:** Nunca

- Preguntas de tipo A – B – C - D

Pregunta 1.2.3:

**A:** Si se presentaron inconvenientes y se superaron

**B:** Si se presentaron inconvenientes, se superaron y se aprendió de ellos para futuros proyectos

**C:** Si se presentaron inconvenientes, pero se ocultaron

Pregunta 1.6.3:

**A:** 100% - 80%

- B:** 79% - 65%
- C:** 64% - 50%
- D:** Menos de 50%

**Preguntas Generales:**

En general ¿Cómo considera el resultado del Proyecto?

- A:** Totalmente exitoso
- B:** Exitoso con algunos inconvenientes
- C:** No exitoso: Se entrego, pero hubo sobrecostos y desfase en el tiempo

En caso de que su respuesta sea la B, señale los inconvenientes presentados:

- A:** Problemas de planeación
- B:** Sobrecosto
- C:** Desfase en el tiempo
- D:** Falta de recursos

**Resultado General**

Luego de realizar las encuestas, las hemos tabulado y agrupado para una lectura fácil de los resultados, a continuación les presentamos el resultado generado por las encuestas.

#	Pregunta	Respuestas				Total
<b>1.1. Usuario(s) Líder(es) del Proyecto.</b>						
1.1.1	Durante todo el tiempo que duro el proyecto estuvo involucrado el usuario líder	<b>S</b>	<b>N</b>	<b>N/A</b>		
		77%	15%	8%		100%
1.1.2	Como fue la relación durante el proyecto con los usuarios	<b>E</b>	<b>B</b>	<b>R</b>	<b>M</b>	
		24%	76%	0%	0%	100%

1.1.3	En general considera que los usuarios fueron los correctos en el desarrollo de este proyecto	<b>S</b>	<b>N</b>	<b>N/A</b>		
		81%	12%	8%		100%
	<b>Porque</b>	<b>S</b>	<b>N</b>	<b>N/A</b>		
	¿Experto en el proceso a automatizar?	56%	40%	4%		100%
	Tenían tiempo disponible para el proyecto	52%	44%	4%		100%
	Tomaban decisiones	56%	40%	4%		100%
	Expresaban claramente sus requisitos	56%	40%	4%		100%
1.1.4	Estaban comprometidos con el proyecto	88%	8%	4%		100%
	Involucrar a los usuarios fue tarea fácil	<b>S</b>	<b>N</b>	<b>N/A</b>		
		69%	23%	8%		100%

<b>1.2. Soporte de la Gerencia</b>						
1.2.1	¿Contó con el apoyo de un(os) directivo(s) de la empresa usuaria durante el desarrollo del proyecto	<b>S</b>	<b>N</b>	<b>N/A</b>		
		81%	12%	8%		100%
1.2.2	¿El directivo encargado mostró interés en los resultados del proyecto?	<b>S</b>	<b>N</b>	<b>N/A</b>		
		77%	12%	12%		100%
1.2.3	En alguna fase del proyecto ¿Se presentaron Inconvenientes?	<b>S</b>	<b>N</b>	<b>N/A</b>		
		92%	4%	4%		100%
	Como manejo los inconvenientes el equipo de trabajo	<b>A</b>	<b>B</b>	<b>C</b>		
		13%	83%	4%		100%
<b>1.3. Requisitos del Proyecto</b>						
1.3.1	¿Se tenía definida una visión (alcance) en el proyecto?	<b>S</b>	<b>N</b>			
		100%	0%			100%
1.3.2	La definición de los usuarios de sus necesidades fue:	<b>E</b>	<b>B</b>	<b>R</b>	<b>M</b>	
		21%	63%	17%	0%	100%
1.3.3	¿Se contemplo desde el principio una evaluación de riesgos?	<b>S</b>	<b>N</b>			
		40%	60%			100%

1.3.4	¿Fue el proyecto medible o bien dimensionado desde el principio (Según la definición de los requisitos)	<b>S</b>	<b>N</b>			
	Tiempo	58%	42%			100%
	Costos	63%	38%			100%
	Recursos	50%	50%			100%

1.3.5	¿Cree que fue correcta la fase de elicitación de requisitos?	<b>S</b>	<b>N</b>			
		64%	36%			100%
1.3.6	El equipo de trabajo ¿Priorizó correctamente los requisitos que se definieron en el proyecto	<b>S</b>	<b>N</b>			
		84%	16%			100%
1.3.7	¿Se hizo un prototipo o interfaz gráfica del proyecto en esta etapa inicial?	<b>S</b>	<b>N</b>	<b>N/A</b>		
		69%	23%	8%		100%

#### 1.4. Planeación del Proyecto

1.4.1	¿Se tenía un plan definido de trabajo para el proyecto?	<b>S</b>	<b>N</b>			
		100%	0%			100%
1.4.2	¿Involucraba constantemente este plan a los usuarios?	<b>S</b>	<b>N</b>	<b>N/A</b>		
		69%	23%	8%		100%
1.4.3	¿Se hizo una definición clara de las fases del proyecto?	<b>S</b>	<b>N</b>			
		81%	19%			100%
1.4.4	¿Se manejo durante el proyecto la gestión de cambios?	<b>S</b>	<b>N</b>			
		58%	42%			100%
1.4.5	¿Se pudo medir el progreso del proyecto en cualquier momento de este?	<b>S</b>	<b>N</b>			
		96%	4%			100%
1.4.6	¿La definición de los tiempos límites para cada una de las fases fue?	<b>E</b>	<b>B</b>	<b>R</b>	<b>M</b>	
		4%	69%	27%	0%	100%
	¿Se cumplieron estos tiempos?	<b>S</b>	<b>A</b>	<b>N</b>		
		31%	58%	12%		100%
1.4.7	¿Se realizó una definición clara de los roles en el proyecto?	<b>S</b>	<b>N</b>			
		88%	12%			100%
1.4.8	¿Cada integrante del proyecto tenía claro conocimiento de su rol?	<b>S</b>	<b>N</b>			
		88%	12%			100%

1.4.9	¿Se realizó un seguimiento periódico del avance del proyecto?	<b>S</b>	<b>N</b>			
		100%	0%			100%
	Con que periodicidad	<b>Diaria</b>	<b>Sem.</b>	<b>Quincenal</b>	<b>Men.</b>	
		15%	56%	22%	7%	100%
1.4.10	¿En el desarrollo del proyecto se utilizó una o varias metodología(s) y/o técnica propias de la ingeniería del software?	<b>S</b>	<b>N</b>			
		100%	0%			100%
		<b># Por C/U</b>				
	RUP	30%				
	AUP	0%				
	CASCADA	15%				
	XP	4%				
	SCRUM	4%				
	CRYSTAL	0%				
	AM - AGILE MOLDELLING	0%				
	UML	30%				
	SOA	9%				
	OSS	0%				
	COMPONENTES	7%				
	100%					
<b>1.5. Competencias del Equipo de Trabajo</b>						
1.5.1	¿Las habilidades de los integrantes del equipo de desarrollo fueron?	<b>E</b>	<b>B</b>	<b>R</b>	<b>M</b>	
		50%	46%	4%	0%	100%
1.5.2	¿Su equipo de trabajo fue conformado correctamente (Analistas de desarrollo, Calidad, Arquitectura, entre otros)?	<b>S</b>	<b>N</b>			
		88%	12%			100%
1.5.3	Durante la ejecución del proyecto ¿Se necesitó de un programa de capacitación para incrementar las habilidades del equipo de desarrollo y poder dar mejores resultados?	<b>S</b>	<b>N</b>	<b>N/A</b>		
		46%	42%	12%		100%
1.5.4	¿Se otorgaron incentivo al equipo de trabajo por su buen desempeño?	<b>S</b>	<b>N</b>			
		17%	83%			100%
1.5.5	Los miembros del equipo de desarrollo ¿estaban comprometidos con el proyecto?	<b>S</b>	<b>N</b>			
		96%	4%			100%

1.5.6	¿Se evidenció un trabajo en equipo durante el proyecto?	<b>S</b>	<b>N</b>			
		92%	8%			100%

<b>1.6. Acerca del proyecto</b>						
1.6.1	¿Está alineado el proyecto con los objetivos estratégicos de la empresa usuaria?	<b>S</b>	<b>N</b>			
		96%	4%			100%
1.6.2	Durante el proyecto ¿se definieron indicadores que permitieran medir si los objetivos planteados se lograron o no?	<b>S</b>	<b>N</b>			
		46%	54%			100%
1.6.3	¿En que porcentaje fueron alcanzados los objetivos planteados en el proyecto?	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	
		77%	15%	4%	4%	100%
1.6.4	En general ¿Cómo considera el resultado del Proyecto?	<b>A</b>	<b>B</b>	<b>C</b>		
		31%	58%	12%		100%
	En caso de que su respuesta sea la B, señale los inconvenientes presentados	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	
		38%	21%	25%	17%	100, %
	Durante el año 2008 su empresa ¿cancelo algún proyecto?	<b>S</b>	<b>N</b>			
		40%	60%			100%

Ahora anexamos algunos complementos de algunas preguntas:

**Pregunta 1.3.2:** En caso de tener problema en la definición ¿cual cree que fue la causa?

No sabían como debía funcionar	Falta de algunas definiciones al inicio	No sabían como definir lo que querían	No sabían lo que querían	Cambios de Usuarios	Falta de Usuarios
14%	29%	36%	14%	7%	14%

**Pregunta 1.3.5:** En caso de ser negativa la respuesta indique las posibles causas de los inconvenientes.

Inconvenientes de roles y/o usuarios	Cambios de ideas	Decisión Alta Gerencia de hacer rápido esta parte	Falta de diccionario de datos	Falta detalle en necesidades
17%	33%	17%	17%	17%

**Pregunta proyectos Cancelados:** En caso de que su respuesta sea afirmativa, ¿Cuál(es) fue (ron) el(los) motivo(s) para cancelarlo?

Estudio de Factibilidad al inicio del Proyecto	Alta desviación del costo del proyecto
75%	25%

## 2.4. ANÁLISIS DE LAS ENCUESTAS

Luego de revisar los resultados de las encuestas vamos a realizar el análisis en tres grandes rubros:

- Proyectos

- Usuarios y Requisitos
- Metodología, Planeación y Equipos de trabajo

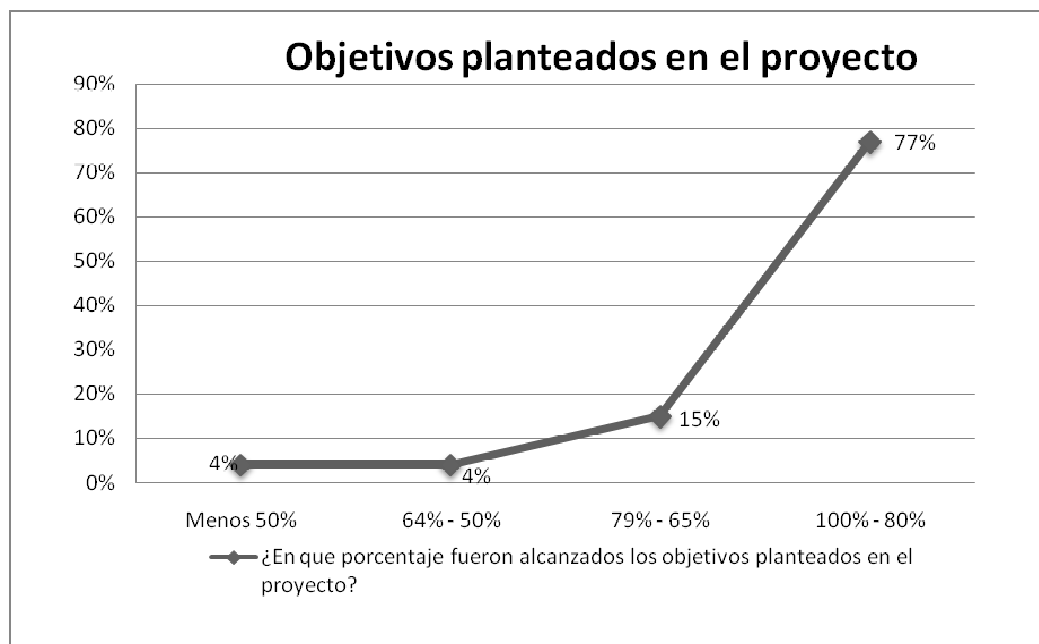
Hemos decidido tomar estos puntos porque hemos encontrado aspectos interesantes que merecen ser ilustrados, algunos confirman las hipótesis planteadas y otros puntos que aparecen allí no los habíamos tenido en cuenta.

## 2.4.1. Proyectos

### 2.4.1.1. Proyectos Exitosos y Proyectos Fracasados

La gestión de proyectos es una buena guía para que los participantes logren alinearse con los objetivos planteados, por este motivo en la encuesta preguntamos el rango de porcentajes en los que el proyecto ha alcanzado los objetivos. Un gráfico que muestra los resultados lo encontramos a continuación:

**Figura 1.** Resultados del Reporte MADi



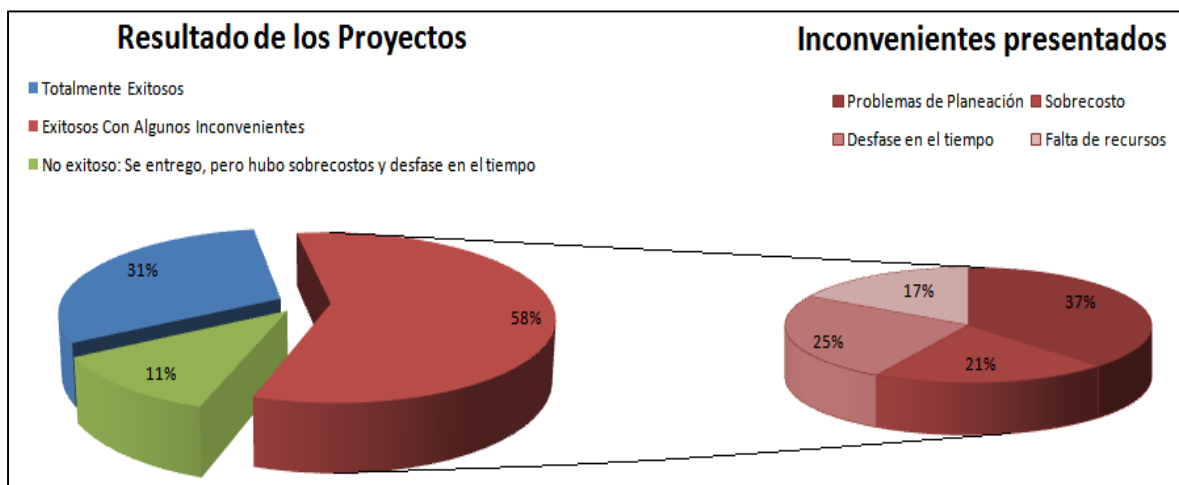
**Fuente:** Reporte MADi



Como lo vemos en la gráfica la tendencia es que el 77% de los proyectos en nuestra ciudad alcanzan entre el 80% y el 100% de los objetivos planteados inicialmente. El otro 23% de los proyectos no logran a cumplir las metas especificadas en un alto porcentaje y son aplicaciones que tienden a no ser exitosas.

En este orden de ideas encontramos información acerca de cómo las compañías de desarrollo percibían el resultado del proyecto y nos encontramos con lo siguiente:

**Figura 2.** Resultados del Reporte MADI



**Fuente:** Reporte MADI

El 11% de los proyectos pueden considerarse un fracaso, el 31% de los proyectos que enfrentamos son exitosos y el 58% presentan algún inconveniente.

Si analizamos más a fondo el 58% de los proyectos que presentan inconvenientes tenemos 4 rubros en los que podemos clasificar los problemas que se presentan:

- Falta de recursos: 17%.
- Desfase en el tiempo: 25%.

- Sobrecosto: 21%.
- Problemas de Planeación: 37%.

De estas cifras podemos inferir que el punto más alarmante en los proyectos de software que no son totalmente exitosos en nuestra ciudad es que se realizan planeaciones defectuosas que hacen que el desarrollo fracase. Si ponderamos la cifra de la mala planeación de la gráfica anterior  $X = 0.37 * 0.58 = 0.2146$  obtenemos que el 21.46% del total de proyectos posee problemas de planeación aunque son considerados como fracasos.

Además de la planeación podemos observar que el desfase en los tiempos, presenta un alto porcentaje en los proyectos que no son totalmente exitosos con un 25% al igual que el sobrecosto con un 21%.

Al ver las cifras presentadas anteriormente, podemos inferir que en el desarrollo de aplicaciones los pecados más frecuentes en los que estamos incurriendo son:

- Mala planeación desde el inicio del proyecto.
- Problemas de tiempo.
- Mal dimensionamiento económico de los proyectos.

#### **2.4.1.2. Proyectos Cancelados**

La pregunta acerca de los proyectos cancelados se enfocó en conocer el porcentaje de las compañías desarrolladoras que durante el año 2008 cancelaron algún(os) proyecto(s), el resultado se observa en la figura 14 y pensamos es coherente con lo que pensamos en la hipótesis.

**Figura 3.** Resultados del Reporte MADI



**Fuente:** Reporte MADI

El 60% de las compañías que participaron en nuestra encuesta no cancelaron ningún proyecto y el 40% si cancelo algún proyecto. Este es un índice alentador si miramos que del 40% de estos proyectos, el 75% se cancelo porque se hizo un estudio de factibilidad y se determino que no era conveniente realizar el proyecto.

Estudio de Factibilidad al inicio del PY	Alta desviación del costo del proyecto
75%	25%

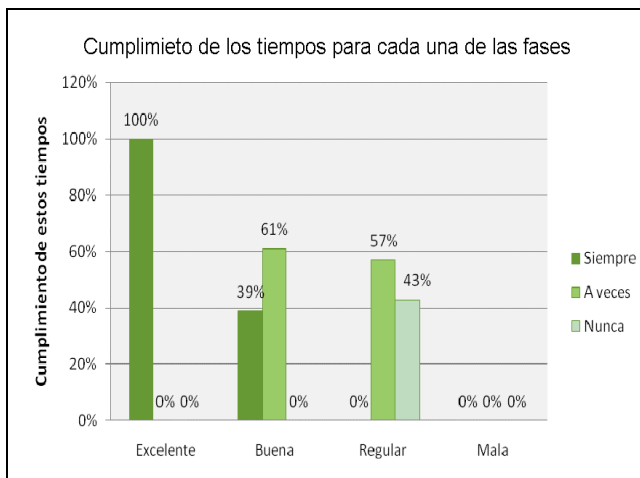
Por esta razón no se perdió mucho dinero en proyectos que se encontraban en ejecución y fueron cancelados.

Respecto a este índice con el reporte CHAOS, debemos estar muy complacidos con el resultado porque el índice de proyectos cancelados presentados por este reporte es muy alto en comparación con lo que ocurre en nuestra ciudad. Adicionalmente, en los proyectos estadounidenses se perdió más cantidad de dinero.

### 2.4.1.3. Estimación de Tiempos

Además de la planeación, el tiempo es un factor determinante en el fracaso o éxito del proyecto. Veamos la figura 15 que construimos para evaluar las estimaciones de tiempo contra la efectividad en cada una de las fases.

**Figura 4.** Resultados del Reporte MAD I



**Fuente:** Reporte MAD I

Como vemos hay 3 ítems a calificar, para el cumplimiento de los tiempos reales, esto quiere decir, por ejemplo, que si la planeación de los tiempos fue buena, las empresas opinaron que a veces cumplieron los tiempos en un 61% y siempre cumplieron los tiempos en un 39%.

Si hubiéramos explorado con mayor detenimiento cuando la respuesta es que la planeación no es buena y en como se

cumplen los tiempos, creemos que esto implicaría supuestos como horarios extras de parte de cada uno de los participantes, baja calidad en el producto software y pruebas funcionales y no funcionales defectuosas.

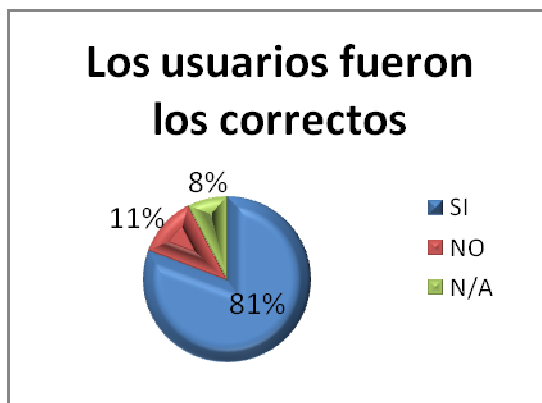
## 2.4.2. Requisitos y usuarios

### 2.4.2.1. Usuarios

En el tema de los usuarios, buscamos comprender las relaciones interpersonales entre el equipo de desarrollo y los usuarios del proyecto, adicionalmente el concepto del desarrollador respecto al usuario (que tan experto es en el tema).

En la figura 16 vemos el resultado a la pregunta *¿Los usuarios fueron los correctos para el proyecto?*

**Figura 5.** Resultados del Reporte MADI

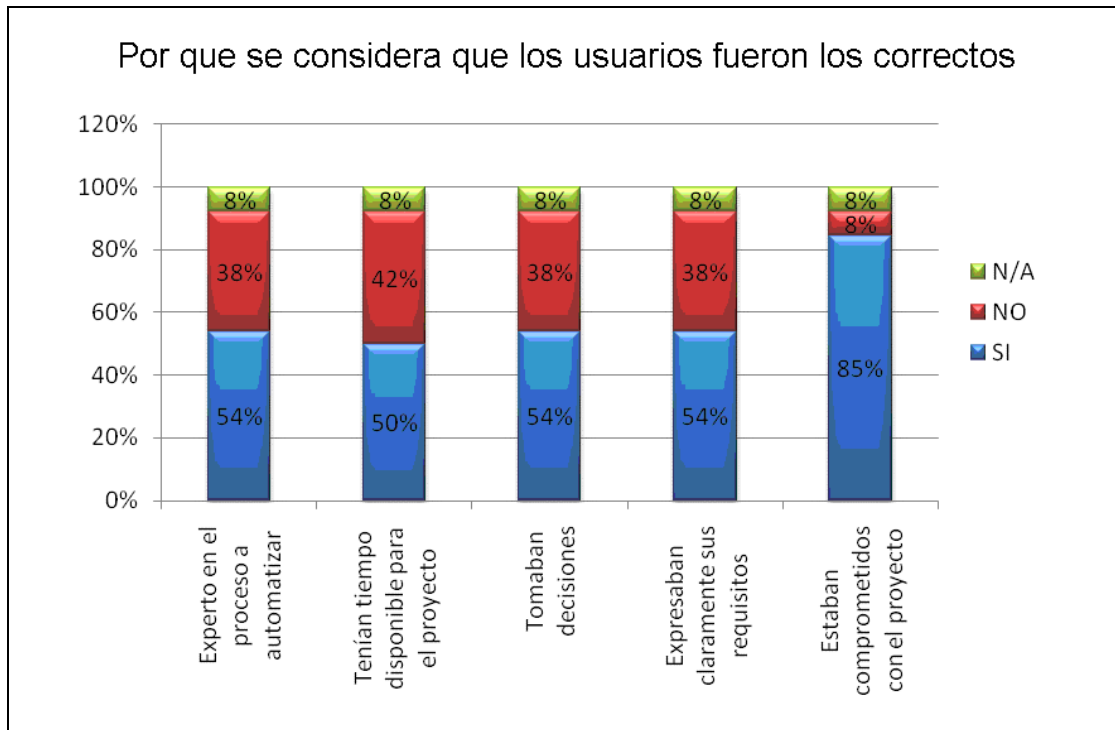


**Fuente:** Reporte MADI

Como vemos en la figura 16, el 81% de los proyectos tenían los usuarios correctos y el 11% se considero que no eran los correctos, el 8% de los proyectos de la muestra no tenían usuario porque eran proyectos realizados para cada una de las casas desarrolladoras, por lo que no implicaron interacción con un usuario externo a la compañía.

En la figura 17, vemos el concepto que tienen los desarrolladores de sus usuarios, en los puntos mas relevantes a la hora de realizar un desarrollo de software.

**Figura 6.** Resultados del Reporte MADI



**Fuente:** Reporte MADI

Para mostrar esta pregunta utilizamos la gráfica de barras apiladas donde apreciamos, los resultados de las preguntas las presentamos a continuación:

- ¿Eran los usuarios expertos en el proceso a automatizar? Según la percepción de los desarrolladores en el 54% de los proyectos el usuario si era experto en el proceso a automatizar, el 38% no era experto en el tema y en el 8% no aplicaba esta pregunta por ser desarrollos internos en la compañía desarrolladora. Lo que nos indica la primera barra es que es posible que en el 38% de los proyectos, los desarrolladores trabajaron realizando un esfuerzo mayor al que comunmente deberían, porque les toco aprender del proceso a automatizar por medios diferentes a la orientación del usuario líder.

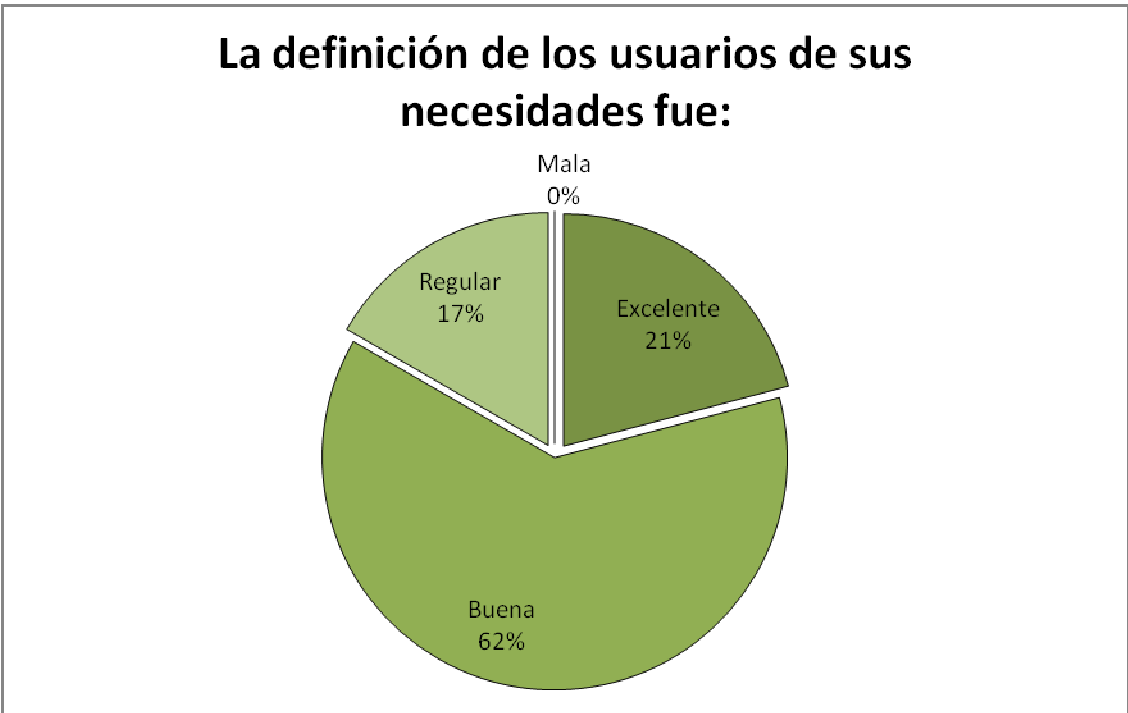
- ¿Tenían tiempo disponible para el proyecto? Este es un punto importante, porque el tiempo que los usuarios dediquen al proyecto es de vital importancia a la hora de tomar decisiones en el desarrollo, para nuestros encuestados el 50% de los usuarios tenían tiempo disponible y el 42% no lo tenían. Este índice es un llamado a nuestros usuarios a que estén más atentos a los desarrollos de software, porque en gran medida el éxito del proyecto depende de ellos y del tiempo que le dediquen.
- ¿Tenían los usuarios capacidad para tomar decisiones? Para el 54% de los proyectos teníamos usuarios con capacidad de tomar decisiones, el 38% no tenía esta capacidad. Cuando ocurre el segundo caso el proyecto tiende a retrasarse porque el tiempo que transcurre para tomar una decisión, no se planea y no es factible determinarlo. El mensaje que podríamos darle a nuestras compañías de desarrollo es que incluyan este tiempo en su planeación cuando tengan un usuario sin capacidad de decisión y controlen en la medida de lo posible este tipo de situaciones, porque si el proyecto se demora nunca se va a decir que el retraso fue por problemas en la toma de decisiones .
- Los usuarios ¿Eran claros expresando sus requisitos? El 54% de los proyectos contaron con un usuario que expresaba correctamente sus requisitos y en el 38% era todo lo contrario. Para los proyectos con el último caso es importante que piensen en buscar metodologías de elicitación de requisitos como workshops y modelos ciclo de vida como el prototipado o el evolutivo que poseen un manejo óptimo para este tipo de situaciones. Aunque hay una mayoría que expresan bien sus requisitos, un 38% que no lo hacen es una alerta que deben mirar bien las empresas desarrolladoras.
- ¿Los usuarios estaban comprometidos con el proyecto? Esta pregunta da un porcentaje del 85% de usuarios comprometidos con el proyecto y 8% de usuarios no comprometidos con el proyecto. Este ítem se comporta según lo esperado y es una fortaleza que deben aprovechar las empresas de software.

Otro dato importante es la relación entre usuarios y el equipo de desarrollo, en términos generales la relación fue buena (76%) y excelente (24%). De este resultado inferimos que en el tema de las relaciones sociales, los usuarios y desarrolladores crean una relación de trabajo buena, aunque es posible mejorarla más en busca de un mayor beneficio en el desarrollo del proyecto.

Como fue la relación durante el proyecto con los usuarios	Excelente	Buena	Regular	Mala	Total
	24%	76%	0%	0%	0%

Finalmente, queremos mostrar la figura 18 en la que se puede apreciar claramente si la definición de las necesidades por parte de los usuarios fue excelente, buena, regular o mala.

**Figura 7.** Resultados del Reporte MADI



**Fuente:** Reporte MADI



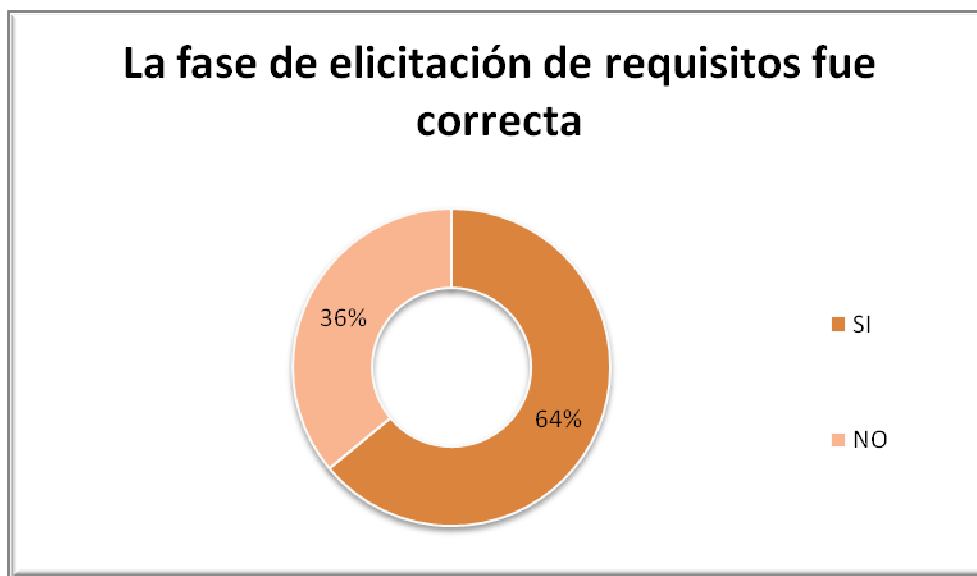
En términos generales las definiciones fueron entre buenas y excelente, con un 83% en la suma de estos dos resultados y solo en el 17% de los proyectos la definición de los usuarios fue regular. En ningún proyecto encontramos que la definición de los proyectos fue mala.

Analizando los resultados de estas gráficas, vemos que debemos buscar una estrategia óptima para el manejo del tiempo con nuestros usuarios. Como lo vimos hay tiempos ocultos durante el proyecto que no son inherentes a las compañías desarrolladoras, pero si debemos tenerlos presentes para poder realizar los ajustes necesarios a los cronogramas, y minimizar retrasos que en ocasiones se han dado en el desarrollo de proyectos de software.

#### 2.4.2.2. Requisitos

Luego de revisar el tema de los usuarios, consideramos importante revisar como fue la elicitación de requisitos. En la figura 19 mostramos los resultados.

**Figura 8.** Resultados del Reporte MAD I



**Fuente:** Reporte MAD I

El 64% de los proyectos contó con una correcta elicitación de requisitos, mientras que el 36% no tuvo este mismo comportamiento. Cabe resaltar que aún teniendo una buena o excelente relación con el usuario, la fase de elicitación puede no haber sido correcta, no solo por la definición del usuario, sino por la gestión que realizó el ingeniero de requisitos con las necesidades del usuario.

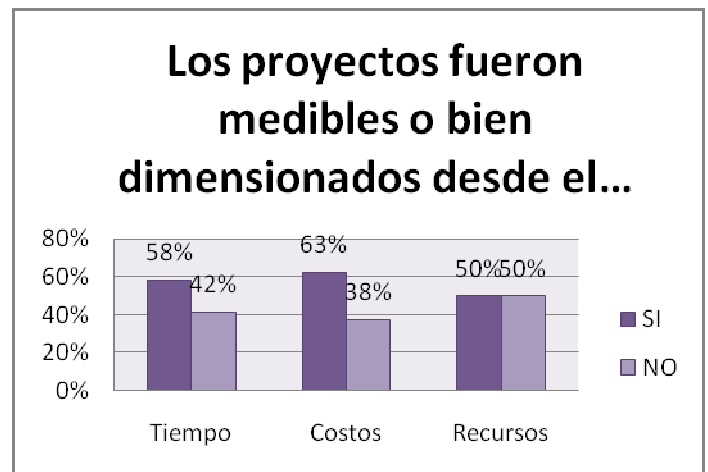
Los requisitos nos permiten dimensionar el proyecto y con esto se realiza la planeación de los recursos y el esfuerzo requerido, para precisar más este tema presentamos la figura 20.

Nuestra hipótesis es que los proyectos con una buena fase de elicitación de requisitos, deberían cumplir una buena planeación de recursos. Sin embargo, si analizamos la diferencia entre el resultado de la figura 19, donde 36% de los proyectos no lograron hacer una buena elicitación contra los problemas de planeación (figura 20) en tiempo, costos y recursos, encontramos una diferencia significativa que presentamos a continuación:

- Tiempo: 6%
- Costos: 2%
- Recursos: 14%

Las diferencias de las que hablamos en el párrafo anterior son inherentes a la planeación del proyecto de software y es un problema que debemos resolver en nuestras empresas de desarrollo de software.

**Figura 9.** Resultados del Reporte MAD I



**Fuente:** Reporte MAD I

### 2.4.3. Metodología, Planeación y Equipos de trabajo

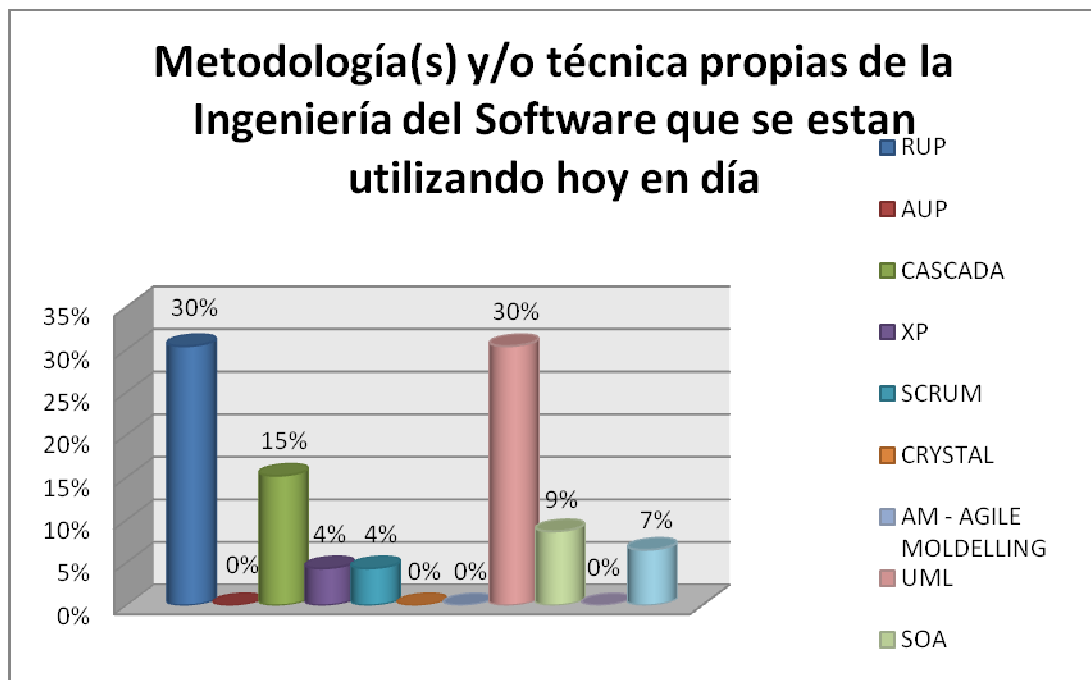
En nuestra hipótesis este es uno de los puntos más importantes sobre la planeación ya tenemos algunas inferencias y sobre la metodología no tenemos todavía ninguna inferencia.

Es importante resaltar que vamos a tocar el tema de planeación desde el punto de vista de la gestión del control de cambios y desde los indicadores de medición del avance del proyecto.

#### 2.4.3.1. Metodología

Se refiere a la falta de metodología y queremos, a través de la siguiente figura, confirmar si lo que pensamos se refleja en las encuestas. Ver figura 21.

**Figura 10.** Resultados del Reporte MAD I



**Fuente:** Reporte MAD I

El modelo de ciclo de vida más utilizado en los proyectos es el de cascada con un 15% de participación porcentual y el marco metodológico preferido es RUP con un 30% de participación porcentual en los proyectos. Nuestra hipótesis es que esta es la combinación más utilizada en nuestro medio y hemos confirmado, lo único que no logramos identificar es si todavía se utiliza mucho la programación orientada a objetos, sin embargo nos alienta que el informe muestra que hay empresas que comienzan a utilizar SOA como una alternativa en su modelo de programación (SOA tuvo una elección en el 9% de los proyectos).

Adicionalmente nos encontramos que se está utilizando otros marcos diferentes a RUP como lo son XP con un 4%, SCRUM con otro 4% y AM con un 7%.

En este aspecto seguimos con la idea que es mejor tomar una configuración en un nivel más alto al que presentamos hoy como el mas utilizado como lo son modelos de ciclo de vida por prototipos o evolutivos, marcos de referencia de la familia Crystal o los que ya se han comenzado a utilizar como XP, SCRUM y AM y la metodología SOA sin dudas nos esta marcando una nueva forma de programar.

### 2.4.3.2. Planeación

El objetivo de la planeación es definir unas metas, teniendo en cuenta los riesgos potenciales y los posibles problemas que se van a presentar y que puedan afectar el logro de dichas metas. Hoy en día las empresas de desarrollo de nuestra ciudad, no están muy concientes de la

¿Se contemplo desde el principio una evaluación de riesgos?	Si	No
	40%	60%

importancia que tiene una metodología de evaluación, identificación, control y solución de los riesgos que se encuentran en todo el ciclo de vida. Esto se ve reflejado en los resultados de la encuesta puesto que en el 60% de los proyectos no se contempla una evaluación de riesgos. El 40% de los proyectos de las empresas de software reconocen la importancia de la evaluación de riesgos desde el inicio del proyecto.

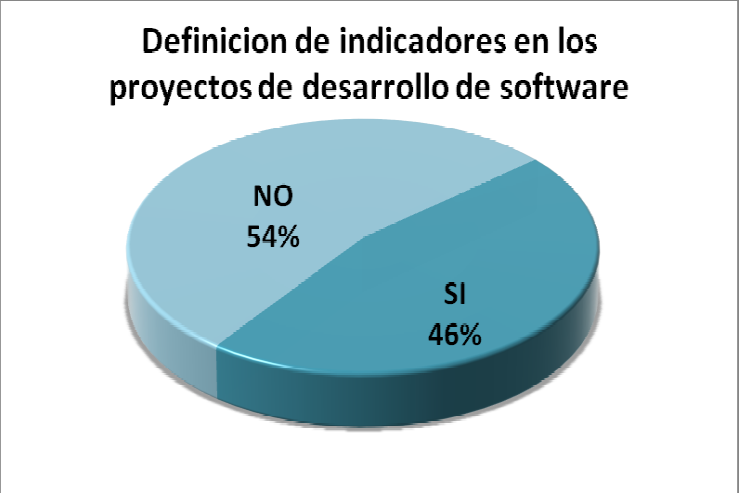
Otra actividad que se debe tener en cuenta durante todo el ciclo del proyecto es la gestión de cambios. Los cambios en el desarrollo de software pueden presentarse en cualquier momento, por lo que se debe estar preparado. La gestión de Cambios sirve para identificar, organizar y controlar las modificaciones que sufre un software durante su desarrollo.

¿Se maneja durante el proyecto la gestión de cambios?	Si	No
	58%	42%

Según los resultados obtenidos con relación a este tema, el 58% de los proyectos manejan la gestión de cambios, mientras que un 42% de los proyectos no tienen en cuenta la administración de cambios, lo cual es preocupante dada la naturaleza dinámica de los sistemas.

Ahora revisemos el manejo de indicadores en los proyectos de desarrollo. En la Figura 22 se puede observar que en el 54% de los proyectos no se definieron indicadores que les permitiera medir el alcance de los objetivos planteados. El 46% de los proyectos si tienen en cuenta dentro de la plantación la definición de indicadores.

**Figura 11.** Resultados del Reporte MADI

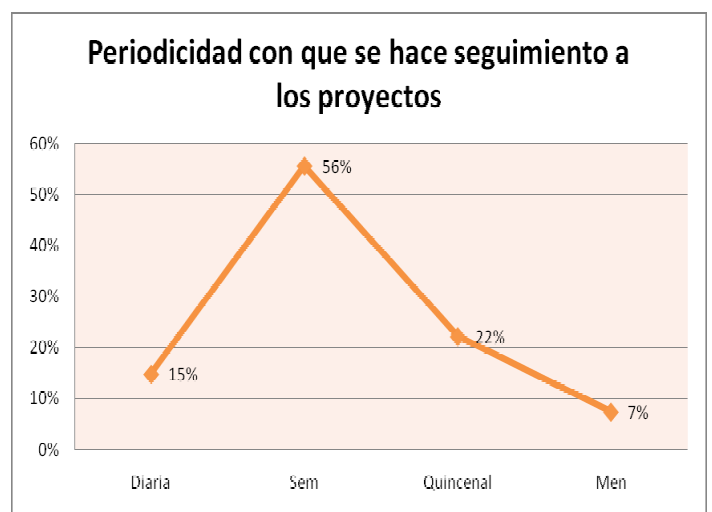


**Fuente:** Reporte MADI

Una vez tabulados los resultados de las preguntas sobre la etapa de planeación, hemos visto un comportamiento muy dividido entre los analistas para definir las mejores prácticas en la ejecución de proyectos de desarrollo de software.

Para profundizar el tema de planeación de proyectos, queremos revisar la periodicidad con que se realiza seguimiento a los proyectos. Como nos muestra la figura 23 el 56% de los equipos de trabajo se reunieron semanalmente para analizar los avances del proyecto. El 22% de los proyectos hicieron reuniones quincenalmente. Los integrantes de los proyectos que se reunieron diariamente fueron el 15% y el 7% de los equipo tuvieron

**Figura 12.** Resultados del Reporte MAD I



**Fuente:** Reporte MAD I

seguimiento mensualmente. Lo que marca una tendencia a que los seguimientos se realicen semanalmente, y consideramos esta periodicidad como prudente.

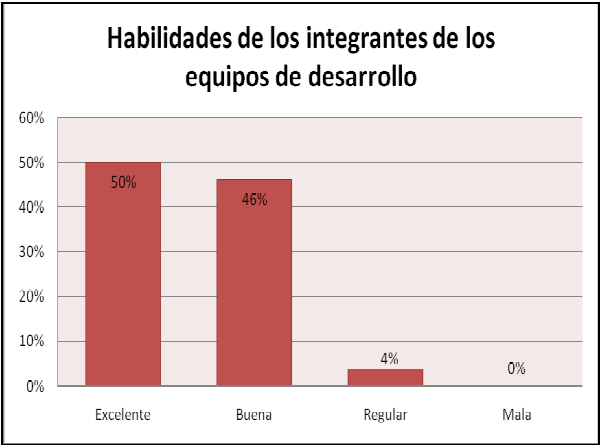
#### **2.4.3.3. Equipo de Trabajo**

Para que un proyecto evolucione satisfactoriamente es importante que los integrantes del equipo de trabajo estén comprometidos, sean habilidosos y quieran participar en el proyecto. Adicionalmente, también es importante que el equipo de trabajo sea reconocido por su buen desempeño. Podemos deducir a partir de la figura 24 que más

de la mitad de los integrantes de los proyectos encuestados tienen excelentes habilidades. El 46% de los integrantes son buenos y el 4% les falta destreza.

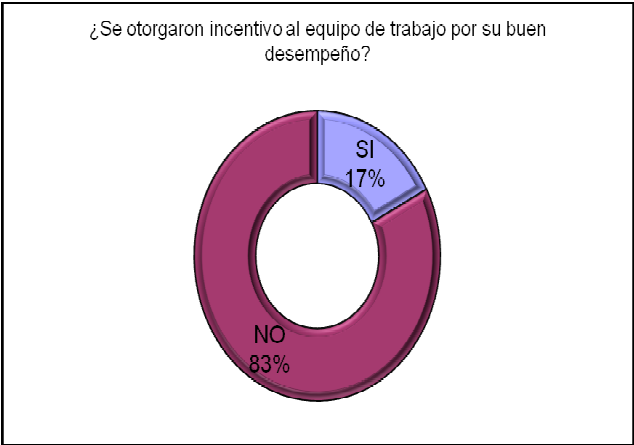
Para estimular el equipo de trabajo podemos ver que el 17% de los integrantes tuvieron incentivos de conocimiento y reconocimiento. El 83% no recibieron incentivos de ningún tipo. Esto puede ser un factor por el que la gente no esta comprometida con los proyectos.

**Figura 13.** Resultados del Reporte MADI



**Fuente:** Reporte MADI

**Figura 14.** Resultados del Reporte MADI



**Fuente:** Reporte MADI





## CONCLUSIONES

- Confirmamos nuestra hipótesis acerca de la configuración que utilizan nuestros desarrolladores en la ejecución de proyectos de ingeniería de software (el modelo de ciclo de vida cascada con un 15%, método orientado a objetos y marco metodológico RUP con un 30%), aunque también descubrimos una tendencia hacia el método de desarrollo SOA que se viene marcando principalmente en las compañías grandes con un 9% de utilización. Adicional logramos ver reflejado en los resultados de las encuestas que las empresas pequeñas impulsan marcos metodológicos ágiles como lo son SCRUM con un 4% de participación, al igual que XP con un 4%. Ninguno de los encuestados señaló el método orientado a objetos como su preferido, así como tampoco nombraron otro método de desarrollo diferente a los propuestos.
- El manejo de usuarios y la elicitación de requisitos es un tema que nos sorprendió, debido a que en nuestra hipótesis planteamos que este sería uno de los aspectos negativos, sin embargo encontramos que este aspecto no marco en forma negativa, debido a que encontramos cosas como que los usuarios estaban comprometidos con los proyectos en un 85%, las relaciones con los usuarios fueron buenas o excelentes para el 100% de los encuestados y para el 81% de los proyectos los usuarios fueron los correctos y solo el 17% califico como regular la elicitación de requisitos. Por lo tanto podemos inferir que el manejo de usuarios y la elicitación de requisitos debe ser levemente mejorada para que logremos unos indicadores más altos en la producción de proyectos de ingeniería de software.

- La planeación es un tema que las compañías de desarrollo de software deben replantear con urgencia, debido a que este es uno de los puntos más débiles a la hora de desarrollar software en nuestra ciudad. Los resultados de los proyectos nos mostraron que el 31% de los proyectos son exitosos y el restante 71% presento inconvenientes o no fue exitoso. Esta cifra se debe reducir, si analizamos mas a fondo que encontramos indicios del por que se presenta esta situación y vemos que los problemas de planeación en puntos importantes como la gestión de control de cambios con un 42% de proyectos que no lo manejan, un 54% de proyectos sin indicadores y un índice de proyectos sin control de riesgos del 60% puede ser unas de las causas. Adicional a esto nos encontramos con otros problemas en la planeación como proyectos que no fueron medibles en tiempo o mal dimensionamiento desde el principio (figura 20) con un 42%, un 50% en recursos y el 38% en los costos. Como podemos ver hay varias razones por las que la planeación tiene un índice negativo y es importante que comencemos a buscar soluciones para reducir la brecha que hay entre los proyectos exitosos y los no exitosos o exitosos con inconvenientes.
- A diferencia de lo que nos dicen algunos gerentes de empresas de desarrollo de la ciudad en la revista Dinero [5], en el 96% de los proyectos con los que trabajamos, los integrantes del equipo tenían buenas o excelentes habilidades para desarrollar el proyecto y el 4% tenían habilidades regulares, sin embargo nos encontramos con algo muy desalentador y es que las compañías desarrolladoras de software de Medellín no tenían establecido en un 83% de los proyectos, un esquema de incentivos que permita mejorar la productividad de los desarrolladores y los motive a sacar adelante los proyectos en que participan. Esta puede ser una de las causas por las que en Medellín sean menos las

personas con motivación a estudiar Ingeniería de Sistemas y busquen alternativas diferentes.

## BIBLIOGRAFÍA

HARTMANN, Deborah, "Interview: Jim Johnson of the Standish Group".  
Internet: <http://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS>.  
THE STANDISH GROUP INTERNATIONAL. 1994; 1995. "The CHAOS  
REPORT". Disponible en Internet:  
[http://www3.uta.edu/faculty/reyes/teaching/general\\_presentations/chaos1994.pdf](http://www3.uta.edu/faculty/reyes/teaching/general_presentations/chaos1994.pdf)

INSTITUTO COLOMBIANO DE NORMAS TÉCNICAS Y CERTIFICACION.

Trabajos escritos: presentación y referencias bibliográficas. Sexta  
actualización. Bogotá: ICONTEC, 2008.

FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES. "Normas de  
presentación para trabajo de grado Fundación universitaria Los Libertadores:  
Normas de presentación de trabajos". Bogotá: FUNDACIÓN UNIVERSITARIA  
LOS LIBERTADORES, 2008.

THE STANDISH GROUP INTERNATIONAL. 2000; 2001. "Extreme CHAOS".  
Disponible en Internet:

<http://www.smallfootprint.com/Portals/0/StandishGroupExtremeChaos2001.pdf>

o

[http://www3.uta.edu/faculty/reyes/teaching/general\\_presentations/extreme\\_chaos.pdf](http://www3.uta.edu/faculty/reyes/teaching/general_presentations/extreme_chaos.pdf).

THE STANDISH GROUP INTERNATIONAL. 1995; 1995. "The Standish Group  
Report". Disponible en Internet:

<http://net.educause.edu/ir/library/pdf/NCP08083B.pdf>

MCCONNELL, Steve. Desarrollo Y Gestión De Proyectos Informáticos.  
Washington, Estados Unidos: Escala, 1997. 710 p. (ISBN: 8448112296).

SOMMERVILLE, Ian. ingeniería del software. Escocia: Escala, 2005. 687 p.  
(ISBN: 8478290745).

[<sup>1</sup>] WORDPRESS. 2008; 2008; Definición de proyecto. Versión 1. Disponible  
desde Internet en: <http://definicion.de/proyecto/>.

[<sup>1</sup>] VARAS C., Marcela. Gestión de proyectos de desarrollo de software.  
Trabajo de Grado ingeniería de sistemas. Ciudad de Concepción: Universidad  
de Concepción. Facultad de ingeniería. Departamento de ingeniería informática  
y Ciencias de la Computación, 2000. p 6.

[<sup>1</sup>] VARAS C., Marcela. Gestión de proyectos de desarrollo de software.  
Trabajo de Grado ingeniería de sistemas. Ciudad de Concepción: Universidad  
de Concepción. Facultad de ingeniería. Departamento de ingeniería informática  
y Ciencias de la Computación, 2000. p 7.

[<sup>1</sup>] VARAS C., Marcela. Gestión de proyectos de desarrollo de software. Trabajo de Grado ingeniería de sistemas. Ciudad de Concepción: Universidad de Concepción. Facultad de ingeniería. Departamento de ingeniería informática y Ciencias de la Computación, 2000. p 7.

[<sup>1</sup>] DINERO.COM. 16, Septiembre, 2009; 2009. No hay Ingenieros de Sistemas. Disponible en Internet en: <http://www.dinero.com/noticias-productividad/no-ingenieros-sistemas/52476.aspx>

[<sup>1</sup>] VARAS C., Marcela. Gestión de proyectos de desarrollo de software. Trabajo de Grado ingeniería de sistemas. Ciudad de Concepción: Universidad de Concepción. Facultad de ingeniería. Departamento de ingeniería informática y Ciencias de la Computación, 2000. p 8-9.

[<sup>1</sup>] VARAS C., Marcela. Gestión de proyectos de desarrollo de software. Trabajo de Grado ingeniería de sistemas. Ciudad de Concepción: Universidad de Concepción. Facultad de ingeniería. Departamento de ingeniería informática y Ciencias de la Computación, 2000. p 9.

[<sup>1</sup>] VARAS C., Marcela. Gestión de proyectos de desarrollo de software. Trabajo de Grado ingeniería de sistemas. Ciudad de Concepción: Universidad de Concepción. Facultad de ingeniería. Departamento de ingeniería informática y Ciencias de la Computación, 2000. p 22 - 30.

[<sup>1</sup>] USR.CODE. Implementación y debugging, "Ciclo de Vida del Software". Disponible en Internet: <http://img.redusers.com/imagenes/libros/lpcu097/capitulogratis.pdf>

[<sup>1</sup>] USR.CODE. Implementación y debugging, "Ciclo de Vida del Software". Disponible en Internet: <http://img.redusers.com/imagenes/libros/lpcu097/capitulogratis.pdf>

[<sup>1</sup>] VARAS C., Marcela. Gestión de proyectos de desarrollo de software. Trabajo de Grado ingeniería de sistemas. Ciudad de Concepción: Universidad de Concepción. Facultad de ingeniería. Departamento de ingeniería informática y Ciencias de la Computación, 2000. p 23

[<sup>1</sup>] USR.CODE. Implementación y debugging, “Ciclo de Vida del Software”.

Disponible en Internet:

<http://img.redusers.com/imagenes/libros/lpcu097/capitulosgratis.pdf>

[<sup>1</sup>] VARAS C., Marcela. Gestión de proyectos de desarrollo de software. Trabajo de Grado ingeniería de sistemas. Ciudad de Concepción: Universidad de Concepción. Facultad de ingeniería. Departamento de ingeniería informática y Ciencias de la Computación, 2000. p 23

[<sup>1</sup>] VARAS C., Marcela. Gestión de proyectos de desarrollo de software. Trabajo de Grado ingeniería de sistemas. Ciudad de Concepción: Universidad de Concepción. Facultad de ingeniería. Departamento de ingeniería informática y Ciencias de la Computación, 2000. p 24

[<sup>1</sup>] VARAS C., Marcela. Gestión de proyectos de desarrollo de software. Trabajo de Grado ingeniería de sistemas. Ciudad de Concepción: Universidad de Concepción. Facultad de ingeniería. Departamento de ingeniería informática y Ciencias de la Computación, 2000. p 24

[<sup>1</sup>] VARAS C., Marcela. Gestión de proyectos de desarrollo de software. Trabajo de Grado ingeniería de sistemas. Ciudad de Concepción: Universidad de Concepción. Facultad de ingeniería. Departamento de ingeniería informática y Ciencias de la Computación, 2000. p 25

[<sup>1</sup>] VARAS C., Marcela. Gestión de proyectos de desarrollo de software. Trabajo de Grado ingeniería de sistemas. Ciudad de Concepción: Universidad de Concepción. Facultad de ingeniería. Departamento de ingeniería informática y Ciencias de la Computación, 2000. p 26

[<sup>1</sup>] ALLSOFT SA; 2009; 2009; Modelos de Desarrollo. Monterrey, México. Disponible en Internet: <http://www.slideshare.net/inventa2/modelos-de-desarrollo>

[<sup>1</sup>] VARAS C., Marcela. Gestión de proyectos de desarrollo de software. Trabajo de Grado ingeniería de sistemas. Ciudad de Concepción: Universidad de Concepción. Facultad de ingeniería. Departamento de ingeniería informática y Ciencias de la Computación, 2000. p 27

[<sup>1</sup>] VARAS C., Marcela. Gestión de proyectos de desarrollo de software. Trabajo de Grado ingeniería de sistemas. Ciudad de Concepción: Universidad de Concepción. Facultad de ingeniería. Departamento de ingeniería informática y Ciencias de la Computación, 2000. p 29

[<sup>1</sup>] VARAS C., Marcela. Gestión de proyectos de desarrollo de software. Trabajo de Grado ingeniería de sistemas. Ciudad de Concepción: Universidad de Concepción. Facultad de ingeniería. Departamento de ingeniería informática y Ciencias de la Computación, 2000. p 29

[<sup>1</sup>] ALLSOFT SA; 2009; 2009; Modelos de Desarrollo. Monterrey, México. Disponible en Internet: <http://www.slideshare.net/inventa2/modelos-de-desarrollo>

[<sup>1</sup>] ALLSOFT SA; 2009; 2009; Modelos de Desarrollo. Monterrey, México. Disponible en Internet: <http://www.slideshare.net/inventa2/modelos-de-desarrollo>.

[<sup>1</sup>] ANAYA, Raquel. Visión General del desarrollo de software. Clases de materia Análisis y Diseño Orientado a Objetos (17, Abril: Medellín Colombia). Memorias. Medellín: Universidad EAFIT, 2007, p. 13.

[<sup>1</sup>] ANAYA, Raquel. Visión General del desarrollo de software. Clases de materia Análisis y Diseño Orientado a Objetos (17, Abril: Medellín Colombia). Memorias. Medellín: Universidad EAFIT, 2007, p. 13.

[<sup>1</sup>] Wikimedia Foundation, Inc.2009; 2009; Definición de Arquitectura Orientada a Servicios (SOA). Disponible en Internet:

[http://es.wikipedia.org/wiki/Arquitectura\\_orientada\\_a\\_servicios](http://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios)

[<sup>1</sup>] ANAYA, Raquel. Visión General del desarrollo de software. Clases de materia Análisis y Diseño Orientado a Objetos (17, Abril: Medellín Colombia). Memorias. Medellín: Universidad EAFIT, 2007, p. 13.

[<sup>1</sup>] Wikimedia Foundation, Inc.2009; 2009; Definición de Proceso Unificado de Racional (RUP). Disponible en Internet: <http://es.wikipedia.org/wiki/RUP>

[<sup>1</sup>] Wikimedia Foundation, Inc.2008; 2008; Definición de El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP). Disponible en Internet: [http://es.wikipedia.org/wiki/Agile\\_Unified\\_Process](http://es.wikipedia.org/wiki/Agile_Unified_Process)

[<sup>1</sup>] Wikimedia Foundation, Inc.2009; 2009; Definición de Programación extrema (XP). Disponible en Internet: [http://es.wikipedia.org/wiki/Programación\\_Extrema](http://es.wikipedia.org/wiki/Programación_Extrema)

[<sup>1</sup>] SANCHEZ DIAZ, Sonia. 2007. Métodos Ágiles. Revista Software Guro. México. Disponible en Internet: <http://www.sg.com.mx/content/view/487>

[<sup>1</sup>] SCOTT W. Ambler. 2002; Modelos Ágiles. Definición de Agile Modeling (AM). Disponible en Internet:<http://www.agilemodeling.com/shared/AMPamphletSpanish.pdf>

[<sup>1</sup>] Wikimedia Foundation, Inc.2009; 2009; Definición de SCRUM. Disponible en Internet: <http://es.wikipedia.org/wiki/Scrum>

[<sup>1</sup>] The world business organization, 2005; Software de código abierto. Definición de open source software (OSS). Disponible en Internet: [http://www.iccwbo.org/uploadedFiles/ICC/policy/e-business/Statements/373-466\\_OSS\\_spanish.pdf](http://www.iccwbo.org/uploadedFiles/ICC/policy/e-business/Statements/373-466_OSS_spanish.pdf)

[<sup>1</sup>] Wikimedia Foundation, Inc.2009; 2009; Definición de Lenguaje Unificado de Modelado (UML). Disponible en Internet: <http://es.wikipedia.org/wiki/UML>

[<sup>1</sup>] The Standish Group. 1995. CHAOS Report. Contextualización sobre el reporte Chaos. Disponible en Internet: <http://net.educause.edu/ir/library/pdf/NCP08083B.pdf>

[<sup>1</sup>] Wikimedia Foundation, Inc.2009; 2009; Definición de Capability Maturity Model Integration (CMMI). Disponible en Internet: <http://es.wikipedia.org/wiki/CMMI>

[<sup>1</sup>] Wikimedia Foundation, Inc.2009; 2009; Definición de ITIL. Disponible en Internet: <http://es.wikipedia.org/wiki/ITIL>

[<sup>1</sup>] Wikimedia Foundation, Inc.2009; 2009; Definición de ISO/IEC 15504. Disponible en Internet: [http://es.wikipedia.org/wiki/ISO/IEC\\_15504](http://es.wikipedia.org/wiki/ISO/IEC_15504)

[<sup>1</sup>] The Standich Group. 1995. CHAOS Report. Contextualización sobre el reporte Chaos. Disponible en Internet:



<http://net.educause.edu/ir/library/pdf/NCP08083B.pdf>

[<sup>1</sup>] SQA - Center of Excellence. "CHAOS REPORT" from STANDISH Group.  
Disponible en Internet: <http://sqa-coe.com/default.aspx>

[<sup>1</sup>] LAMENCA, José Ramón. Modelos de mejora de procesos de software.  
Barcelona.

2009. Disponible en Internet:

[http://217.116.17.204/content/pdf/030309\\_rlamenca\\_ESI.pdf](http://217.116.17.204/content/pdf/030309_rlamenca_ESI.pdf)

[<sup>1</sup>] HUANCA PINTO, Nelson. La Paz Bolivia. Sistema experto basado en reglas para la documentación de requisitos de Software. Descripción del objeto bajo estudio. Disponible en Internet:

<http://www.monografias.com/trabajos26/sistema-documentacion/sistema-documentacion.shtml>

[<sup>1</sup>] The U.S. Government Accountability Office (GAO). Disponible en Internet:

<http://www.gao.gov/index.html>

[<sup>1</sup>] HUANCA PINTO, Nelson. La Paz Bolivia. Sistema experto basado en reglas para la documentación de requisitos de Software. . Descripción del objeto bajo estudio Disponible en Internet:

<http://www.monografias.com/trabajos26/sistema-documentacion/sistema-documentacion.shtml>

[<sup>1</sup>] HUANCA PINTO, Nelson. La Paz Bolivia. Sistema experto basado en reglas para la documentación de requisitos de Software. Descripción del objeto bajo estudio. Disponible en Internet:

<http://www.monografias.com/trabajos26/sistema-documentacion/sistema-documentacion.shtml>

[<sup>1</sup>] HUANCA PINTO, Nelson. La Paz Bolivia. Sistema experto basado en reglas para la documentación de requisitos de Software. Descripción del objeto bajo estudio. Disponible en Internet:

<http://www.monografias.com/trabajos26/sistema-documentacion/sistema-documentacion.shtml>

[<sup>1</sup>] United States General Accounting Office. 2004. Information technology management. Governmentwide Strategic Planning, Performance Measurement, and Investment Management Can Be Further Improved Disponible en Internet:

<http://www.gao.gov/new.items/d0449.pdf>. Año 2004

[<sup>1</sup>] POWNER, David A. United States General Accounting Office. 2005.

Information technology. OMB Can More Effectively Use Its Investment Reviews. Disponible en Internet: <http://www.gao.gov/new.items/d05571t.pdf>

[<sup>1</sup>] Indian Express Newspapers Limited. 2008. Financial Express. Disponible desde Internet en:

<http://www.financialexpress.com/search.php?q=technology+reports+2005&ao=or&col=all&dte=ed>

[<sup>1</sup>] The Indian Express Newspapers Limited. 2005; 2005. China to be world's top outsourcing destination. Disponible en Internet:

<http://www.financialexpress.com/news/China-to-be-world%92s-top-outsourcing-destination/130598/>

[<sup>1</sup>] HUANCA PINTO, Nelson. La Paz Bolivia. Sistema experto basado en reglas para la documentación de requisitos de Software. Descripción del objeto bajo estudio. Disponible en Internet:

<http://www.monografias.com/trabajos26/sistema-documentacion/sistema-documentacion.shtml>

[<sup>1</sup>] Cultura E. Sector informático de Colombia - Medellín. Disponible en Internet: <http://www.culturaemedellin.gov.co/sites/CulturaE/CulturaE/default.aspx>

[<sup>1</sup>] Fedesoft. 2007. Disponible en Internet: <http://www.fedesoft.org/>

[<sup>1</sup>] Ceiba Software House S.A. Disponible en Internet: <http://www.ceiba.com.co/>

[<sup>1</sup>] SONDA. Disponible en Internet: <http://www.sonda.com/>

[<sup>1</sup>] Cognox. Disponible en Internet: <http://www.cognox.com/>

[<sup>1</sup>] DICOSOFT. Disponible en Internet: <http://www.dicosoftware.com/>

# ANEXO

---

## THE STANDISH GROUP REPORT

© The Standish Group 1995. Reprinted here for sole academic purposes with written permission from The Standish Group.

### CHAOS

*"The Roman bridges of antiquity were very inefficient structures. By modern standards, they used too much stone, and as a result, far too much labor to build. Over the years we have learned to build bridges more efficiently, using fewer materials and less labor to perform the same task."*

Tom Clancy (The Sum of All Fears)

---

### INTRODUCTION

In 1986, Alfred Spector, president of Transarc Corporation, co-authored a paper comparing bridge building to software development. The premise: Bridges are normally built on-time, on-budget, and do not fall down. On the other hand, software never comes in on-time or on-budget. In addition, it always breaks down. (Nevertheless, bridge building did not always have such a stellar record. Many bridge building projects overshot their estimates, time frames, and some even fell down.)

One of the biggest reasons bridges come in on-time, on-budget and do not fall down is because of the extreme detail of design. The design is frozen and the contractor has little flexibility in changing the specifications. However, in today's fast moving business environment, a frozen design does not accommodate changes in the business practices. Therefore a more flexible model must be used. This could be and has been used as a rationale for development failure.

But there is another difference between software failures and bridge failures, beside 3,000 years of experience. When a bridge falls down, it is investigated and a report is written on the cause of the failure. This is not so in the computer industry where failures are covered up, ignored, and/or rationalized. As a result, we keep making the same mistakes over and over again.

Consequently the focus of this latest research project at The Standish Group has been to identify: -

- The scope of software project failures
- The major factors that cause software projects to fail
- The key ingredients that can reduce project failures

### FAILURE RECORD

In the United States, we spend more than \$250 billion each year on IT application development of approximately 175,000 projects. The average cost of a development project for a large company is \$2,322,000; for a medium company, it is \$1,331,000; and for a small company, it is \$434,000. A great many of these projects will fail. Software development projects are in chaos, and we can no longer imitate the three monkeys -- hear no failures, see no failures, speak no failures.

The Standish Group research shows a staggering 31.1% of projects will be cancelled before they ever get completed. Further results indicate 52.7% of projects will cost 189% of their original estimates. The cost of these failures and overruns are just the tip of the proverbial iceberg. The lost opportunity costs are not measurable, but could easily be in the trillions of dollars. One just has to look to the City of Denver to realize the extent of this problem. The failure to produce reliable software to handle baggage at the new Denver airport is costing the city \$1.1 million per day.

Based on this research, The Standish Group estimates that in 1995 American companies and government agencies will spend \$81 billion for cancelled software projects. These same organizations will pay an additional \$59 billion for software projects that will be completed, but will exceed their original time estimates. Risk is always a factor when pushing the technology envelope, but many of these projects were as mundane as a drivers license database, a new accounting package, or an order entry system.

On the success side, the average is only 16.2% for software projects that are completed on-time and on-budget. In the larger companies, the news is even worse: only 9% of their projects come in on-time and on-budget. And, even when these projects are completed, many are no more than a mere shadow of their original specification requirements. Projects completed by the largest American companies have only approximately 42% of the originally-proposed features and functions. Smaller companies do much better. A total of 78.4% of their software projects will get deployed with at least 74.2% of their original features and functions.

This data may seem disheartening, and in fact, 48% of the IT executives in our research sample feel that there are more failures currently than just five years ago. The good news is that over 50% feel there are fewer or the same number of failures today than there were five and ten years ago.

## METHODOLOGY

The survey made by The Standish Group was as thorough as possible, short of the unreachable goal of surveying every company with MIS in the country. The results are based on what we at The Standish Group define as "key findings" from our research surveys and several personal interviews. The respondents were IT executive managers. The sample included large, medium, and small companies across major industry segments, e.g., banking, securities, manufacturing, retail, wholesale, health care, insurance, services, and local, state, and federal organizations. The total sample size was 365 respondents and represented 8,380 applications. In addition, The Standish Group conducted four focus groups and numerous personal interviews to provide qualitative context for the survey results.

For purposes of the study, projects were classified into three resolution types: -

- Resolution Type 1, or project success: The project is completed on-time and on-budget, with all features and functions as initially specified.
- Resolution Type 2, or project challenged: The project is completed and operational but over-budget, over the time estimate, and offers fewer features and functions than originally specified.
- Resolution Type 3, or project impaired: The project is cancelled at some point during the development cycle.

Overall, the success rate was only 16.2%, while challenged projects accounted for 52.7%, and impaired (cancelled) for 31.1%.

## FAILURE STATISTICS

The Standish Group further segmented these results by large, medium and small companies. A large company is any company with greater than \$500 million dollars in revenue per year, a medium company is defined as having \$200 million to \$500 million in yearly revenue, and a small company is from \$100 million to \$200 million.

The figures for failure were equally disheartening in companies of all sizes. Only 9% of projects in large companies were successful. At 16.2% and 28% respectively, medium and small companies were somewhat more successful. A whopping 61.5% of all large company projects were challenged (Resolution Type 2) compared to 46.7% for medium companies and 50.4% for small companies. The most projects, 37.1%, were impaired and subsequently cancelled (Resolution Type 3) in medium companies, compared to 29.5% in large companies and 21.6% in small companies.

### Restarts

One of the major causes of both cost and time overruns is restarts. For every 100 projects that start, there are 94 restarts. This does not mean that 94 of 100 will have one restart, some projects can have several restarts. For example, the California Department of Motor Vehicles project, a failure scenario summarized later in this article, had many restarts.

### Cost Overruns

Equally telling were the results for cost overruns, time overruns, and failure of the applications to provide expected features. For combined Type 2 and Type 3 projects, almost a third experienced cost overruns of 150 to 200%. The average across all companies is 189% of the original cost estimate. The average cost overrun is 178% for large companies, 182% for medium companies, and 214% for small companies.

Cost Overruns	% of Responses
Under 20%	15.5%
21 - 50%	31.5%
51 - 100%	29.6%
101 - 200%	10.2%
201 - 400%	8.8%
Over 400%	4.4%

### Time Overruns

For the same combined challenged and impaired projects, over one-third also experienced time overruns of 200 to 300%. The average overrun is 222% of the original time estimate. For large companies, the average is 230%; for medium companies, the average is 202%; and for small companies, the average is 239%.

Time Overruns	% of Responses
Under 20%	13.9%
21 - 50%	18.3%
51 - 100%	20.0%
101 - 200%	35.5%
201 - 400%	11.2%
Over 400%	1.1%

## Content Deficiencies

For challenged projects, more than a quarter were completed with only 25% to 49% of originally-specified features and functions. On average, only 61% of originally specified features and functions were available on these projects. Large companies have the worst record with only 42% of the features and functions in the end product. For medium companies, the percentage is 65%. And for small companies, the percentage is 74%.

% of Features/Functions	% of Responses
Less Than 25%	4.6%
25 - 49%	27.2%
50 - 74%	21.8%
75 - 99%	39.1%
100%	7.3%

Currently, the 365 companies have a combined 3,682 applications under development. Only 431 or 12% of these projects are on-time and on-budget.

## SUCCESS/FAILURE PROFILES

The most important aspect of the research is discovering why projects fail. To do this, The Standish Group surveyed IT executive managers for their opinions about why projects succeed. The three major reasons that a project will succeed are user involvement, executive management support, and a clear statement of requirements. There are other success criteria, but with these three elements in place, the chances of success are much greater. Without them, chance of failure increases dramatically.

Project Success Factors	% of Responses
1. User Involvement	15.9%
2. Executive Management Support	13.9%
3. Clear Statement of Requirements	13.0%
4. Proper Planning	9.6%
5. Realistic Expectations	8.2%
6. Smaller Project Milestones	7.7%
7. Competent Staff	7.2%
8. Ownership	5.3%
9. Clear Vision & Objectives	2.9%
10. Hard-Working, Focused Staff	2.4%
Other	13.9%

The survey participants were also asked about the factors that cause projects to be challenged.

Project Challenged Factors	% of Responses
1. Lack of User Input	12.8%
2. Incomplete Requirements & Specifications	12.3%
3. Changing Requirements & Specifications	11.8%
4. Lack of Executive Support	7.5%
5. Technology Incompetence	7.0%
6. Lack of Resources	6.4%
7. Unrealistic Expectations	5.9%
8. Unclear Objectives	5.3%
9. Unrealistic Time Frames	4.3%
10. New Technology	3.7%
Other	23.0%

Opinions about why projects are impaired and ultimately cancelled ranked incomplete requirements and lack of user involvement at the top of the list.

Project Impaired Factors	% of Responses
1. Incomplete Requirements	13.1%
2. Lack of User Involvement	12.4%
3. Lack of Resources	10.6%
4. Unrealistic Expectations	9.9%
5. Lack of Executive Support	9.3%
6. Changing Requirements & Specifications	8.7%
7. Lack of Planning	8.1%
8. Didn't Need It Any Longer	7.5%
9. Lack of IT Management	6.2%
10. Technology Illiteracy	4.3%
Other	9.9%

Another key finding of the survey is that a high percentage of executive managers believe that there are more project failures now than five years ago and ten years ago. This despite the fact that technology has had time to mature.

	Than 5 Years Ago	Than 10 Years Ago
Significantly More Failures	27%	17%
Somewhat More Failures	21%	29%
No Change	11%	23%
Somewhat Fewer Failures	19%	23%
Significantly Fewer Failures	22%	8%

## FOCUS GROUPS

To augment the survey results, The Standish Group conducted four focus groups with IT executives of major companies. The attendees were from a cross section of industries, including insurance, state and federal government, retail, banking, securities, manufacturing and service. Two of the focus groups were in Boston. The other two, in San Francisco. Each focus group had an average of ten participants with an overall total of forty-one IT executives. The purpose of these particular focus groups was to solicit opinions on why projects fail. In addition, The Standish Group conducted interviews with various IT managers. Some of their comments are enlightening about the variety of problems besetting project development.

Many of the comments echoed the findings of The Standish Group survey. "We have 500 projects. None are on-time and on-budget. This year, 40% will get cancelled," said Edward, Vice President of MIS at a pharmaceutical company.

Other comments went directly to the reasons for failure. Jim, the Director of IT at a major medical equipment manufacturer, said: "Being that it's a mindset, it's very difficult to get all of the management -- it's even on the local level, not even on a worldwide level -- to get all of the management to agree on a set of rules.... That's a challenge in itself because you have to, in some cases, convince them that this is best for the company, not necessarily best for them, but best for the company. And you have to have that buy-in. If you don't have that buy-in, you're going to fail. I don't care how big or how small the project is."

John, Director of MIS at a government agency added: "Probably 90% of application project failure is due to politics!" And Kathy, a programmer at a telecommunication company, offered an even more scathing comment on politics: "Sometimes you have to make a decision you don't like. Even against your own nature. You say well, it's wrong, but you make that decision anyway. It's like taking a hammer to your toe. It hurts."

Bob, the Director of MIS at a hospital, commented on external factors contributing to project failure. "Our biggest problem is competing priorities," he said. "We just had a reorganization today. So now that's going to sap all the resources. And explaining to senior management that, 'Well, it's really taking us the time we said it was going to take. But because you've reorganized the company, I'm going to take another six months on this other project, because I'm doing something else for you.' That's the biggest issue I have." Bill, the Director of MIS at a securities firm, added: "Changes, changes, changes; they're the real killers."

Some of the comments were darkly humorous. "Brain-dead users, just plain brain-dead users," said Peter, an application analyst at a bank. "When the projected started to fail," said Paul, a programmer at a personal products manufacturer, "the management got behind it -- way behind."

The comment most indicative of the chaos in project development came from Sid, a project manager at an insurance company. "The project was two years late and three years in development," he said. "We had thirty people on the project. We delivered an application the user didn't need. They had stopped selling the product over a year before."

## CASE STUDIES

For further insight into failure and success, The Standish Group looked carefully at two famous Resolution Type 3 (cancelled) projects and two Resolution Type 1 (successful) projects. For purposes of comparison, the project success criteria from the survey of IT executive managers was used to create a "success potential" chart. The success criteria were then weighted, based on the input from the surveyed IT managers. The most important criterion, "user involvement," was given 19 "success points". The least important -- "hard-working, focused staff" -- was given three points. Two very important success criteria -- "realistic expectations" and "smaller project milestones" -- were weighted at ten and nine points respectively. Finally, as presented later in this report, each of the case studies was graded.

### California DMV

In 1987, the California Department of Motor Vehicles (DMV) embarked on a major project to revitalize their drivers license and registration application process. By 1993, after \$45 million dollars had already been spent, the project was cancelled.

According to a special report issued by DMV, the primary reason for redeveloping this application was the adoption new technology. They publicly stated: "The specific objective of the 1987 project was to use modern technology to support the DMV mission and sustain its growth by strategically positioning the DMV data processing environment to rapidly respond to change." Also, according to the DMV special report "The phasing was changed several times, but the DMV technical community was never truly confident in its viability...."

The project had no monetary payback, was not supported by executive management, had no user involvement, had poor planning, poor design specifications and unclear objectives. It also did not have the support of the state's information management staff.

The DMV project was not rocket science. There are much harder applications than driver licenses and registrations. But because of internal state politics, unclear objectives, and poor planning, the project was doomed from the start.



## American Airlines

Early in 1994, American Airlines settled their lawsuit with Budget Rent-A-Car, Marriott Corp. and Hilton Hotels after the \$165 million CONFIRM car rental and hotel reservation system project collapsed into chaos.

This project failed because there were too many cooks and the soup spoiled. Executive management not only supported the project, they were active project managers. Of course, for a project this size to fail, it must have had many flaws. Other major causes included an incomplete statement of requirements, lack of user involvement, and constant changing of requirements and specifications.

## Hyatt Hotels

While Marriott and Hilton Hotels were checking out of their failed reservation system, Hyatt was checking in. Today, you can dial from a cellular airplane telephone at 35,000 feet, check into your Hyatt hotel room, schedule the courtesy bus to pick you up, and have your keys waiting for you at the express desk. This new reservation system was ahead of schedule, under budget, with extra features -- for a mere \$15 million of cold cash. They used modern, open systems software with an Informix database and the TUXEDO transaction monitor, on Unix-based hardware.

Hyatt had all the right ingredients for success: user involvement, executive management support, a clear statement of requirements, proper planning, and small project milestones.

## Banco Itamarati

A year after a strategic redirection, Banco Itamarati, a privately-held Brazilian bank, produced an annual net profit growth of 51% and moved from 47th to 15th place in the Brazilian banking industry. Three fundamental reasons account for Banco Itamarati's success. First, they had a clear vision with documented specific objectives. Second, their top-down level of involvement allowed Banco Itamarati to stay on course. And finally, the bank produced incremental, measurable results throughout the planning/implementation period.

Banco Itamarati's clear business goal is to be one of Brazil's top five privately-held banks by the year 2000. Their objectives include maintaining a close relationship with their customers to improve and maintain an understanding of their needs, offering competitive financial solutions, guaranteeing customer satisfaction, and finally producing balanced results for the Itamarati Group. Banco Itamarati's objectives were incorporated into a strategic plan that clearly identified measurable results and individual ownership.

Their strategic plan made technology a key component of the business strategy. Itamarati used Itautec's GRIP OLTP monitor as a basic tool for integrating software components. According to Henrique Costabile, Director of Organization Development, "We are one of the first banks to implement a client-server architecture that maximizes the potential of this architecture." Executive leadership, a well-communicated plan, and a skilled diverse team provided the foundation for Banco Itamarati to achieve their long-term goal, potentially ahead of schedule.

## CASE STUDY CONCLUSIONS

The study of each project included adding up success points on the "success potential" chart.

Success Criteria	Points	DMV	CONFIRM	HYATT	ITAMARATI
1. User Involvement	19	NO ( 0)	NO ( 0)	YES (19)	YES (19)
2. Executive Management Support	16	NO ( 0)	YES (16)	YES (16)	YES (16)
3. Clear Statement of Requirements	15	NO ( 0)	NO ( 0)	YES (15)	NO ( 0)
4. Proper Planning	11	NO ( 0)	NO ( 0)	YES (11)	YES (11)
5. Realistic Expectations	10	YES (10)	YES (10)	YES (10)	YES (10)
6. Smaller Project Milestones	9	NO ( 0)	NO ( 0)	YES (9)	YES (9)
7. Competent Staff	8	NO ( 0)	NO ( 0)	YES (8)	YES (8)
8. Ownership	6	NO ( 0)	NO ( 0)	YES (6)	YES (6)
9. Clear Vision & Objectives	3	NO ( 0)	NO ( 0)	YES (3)	YES (3)
10. Hard-Working, Focused Staff	3	NO ( 0)	YES ( 3)	YES ( 3)	YES ( 3)
TOTAL	100	10	29	100	85

With only 10 success points, the DMV project had virtually no chance of success. With 100 success points, Hyatt's reservation project had all the right ingredients for success. With only 29 success points, the CONFIRM project had little chance of success. With 85, Itamarati, while not as assured as Hyatt, started with a high success probability.

## THE BRIDGE TO SUCCESS

Notwithstanding, this study is hardly in-depth enough to provide a real solution to such a daunting problem as the current project failure rates. Application software projects are truly in troubled waters. In order to make order out of the chaos, we need to examine why projects fail. Just like bridges, each major software failure must be investigated, studied, reported and shared.

Because it is the product of the ideas of IT managers, the "Success Potential" chart can be a useful tool for either forecasting the potential success of a project or evaluating project failure.

Research at The Standish Group also indicates that smaller time frames, with delivery of software components early and often, will increase the success rate. Shorter time frames result in an iterative process of design, prototype, develop, test, and deploy small elements. This process is known as "growing" software, as opposed to the old concept of "developing" software. Growing software engages the user earlier, each component has an owner or a small set of owners, and expectations are realistically set. In addition, each software component has a clear and precise statement and set of objectives. Software components and small projects tend to be less complex. Making the projects simpler is a worthwhile endeavor because complexity causes only confusion and increased cost.

There is one final aspect to be considered in any degree of project failure. All success is rooted in either luck or failure. If you begin with luck, you learn nothing but arrogance. However, if you begin with failure and learn to evaluate it, you also learn to succeed. Failure begets knowledge. Out of knowledge you gain wisdom, and it is with wisdom that you can become truly successful.

# ANEXO B

---

United States Government Accountability Office

---

**GAO**

Testimony  
Before the Committee on Government Reform,  
House of Representatives

---

For Release on Delivery  
Expected at  
10 a.m. EDT on Thursday,  
April 21, 2005

## INFORMATION TECHNOLOGY

### OMB Can More Effectively Use Its Investment Reviews

Statement of David A. Powner,  
Director, Information Technology  
Management Issues



---

GAO-05-571T



# Highlights

Highlights of GAO-05-571T, a testimony before the Committee on Government Reform, House of Representatives.

## Why GAO Did This Study

Federal spending on information technology (IT) is over \$60 billion this year and is expected to continue to rise. Accordingly, it is essential that federal IT investments are managed efficiently. Of the 1,200 major IT projects in the President's Budget for Fiscal Year 2005, OMB stated that it had placed about half—621 projects, representing about \$22 billion—on a Management Watch List to focus attention on mission-critical IT investments that need management improvements.

GAO was asked to testify on the findings and recommendations made in a report that it recently completed (GAO-05-276), which describes and assesses OMB's processes for (1) placing projects on its Management Watch List and (2) following up on corrective actions established for projects on the list.

[www.gao.gov/cgi-bin/getrpt?GAO-05-571T](http://www.gao.gov/cgi-bin/getrpt?GAO-05-571T)

To view the full product, including the scope and methodology, click on the link above. For more information, contact David Powner at (202) 512-9266 or [pownerd@gao.gov](mailto:pownerd@gao.gov).

## INFORMATION TECHNOLOGY

### OMB Can More Effectively Use Its Management Reviews

#### What GAO Found

For the fiscal year 2005 budget, OMB developed processes and criteria for including investments on its Management Watch List. In doing so, it identified opportunities to strengthen investments and promote improvements in IT management. However, it did not develop a single, aggregate list identifying the projects and their weaknesses. Instead, OMB officials told GAO that to identify projects with weaknesses, individual analysts used scoring criteria that the office established for evaluating the justifications for funding that federal agencies submit for major projects. These analysts, each of whom is typically responsible for several federal agencies, were then responsible for maintaining information on these projects. To derive the total number of projects on the list for fiscal year 2005, the office polled its individual analysts and compiled the result. However, OMB officials told GAO that because they did not see such an activity as necessary, they did not compile a single list. Accordingly, OMB has not fully exploited the opportunity to use its watch list as a tool for analyzing IT investments on a governmentwide basis.

OMB asked agencies to take corrective actions to address weaknesses associated with projects on the Management Watch List, but it did not develop a structured, consistent process for deciding how to monitor agency corrective actions. According to OMB officials, decisions on monitoring of progress were typically made by the staff with responsibility for reviewing individual agency budget submissions, depending on the staff's insights into agency operations and objectives. Because it did not consistently require or monitor agency follow-up activities, OMB did not know whether the project risks that it identified through its Management Watch List were being managed effectively, potentially leaving resources at risk of being committed to poorly planned and managed projects. In addition, because it did not consistently monitor the follow-up performed on projects on the Management Watch List, OMB could not readily tell GAO which of the 621 projects received follow-up attention.

To help enable OMB to take advantage of the potential benefits of using the Management Watch List as a tool for analyzing and following up on investments, GAO's report included recommendations that OMB develop a single, aggregate Management Watch List and that it develop and use criteria for prioritizing and monitoring the projects on the list. GAO also recommended that the office use the prioritized list for reporting to the Congress as part of its statutory reporting responsibilities. In commenting on a draft of this report, OMB did not agree that the aggregated governmentwide list recommended by GAO is necessary for adequate oversight and management. However, GAO continues to believe that an aggregated Management Watch List would contribute to OMB's ability to analyze IT investments governmentwide and track progress in addressing deficiencies.

---

Mr. Chairman and Members of the Committee:

Thank you for the opportunity to participate in the Committee's hearing on processes that the Office of Management and Budget (OMB) has developed as part of its efforts to identify and follow up on information technology (IT) projects that need management improvements.

As you know, the President's Budget for Fiscal Year 2005 requested over \$60 billion to fund IT, and that figure is expected to rise throughout the rest of the decade. OMB stated that of the nearly 1,200 major IT projects in the fiscal year 2005 budget, it had placed about half—621 projects, representing about \$22 billion—on its Management Watch List. For fiscal year 2006, 342 of 1,087 IT projects (representing about \$15 billion) were placed on the watch list.

At your request, we performed a review of OMB's processes for (1) placing projects on its Management Watch List and (2) following up on corrective actions established for projects on the list. Today I am summarizing the findings and recommendations of that report, which is being released today.<sup>1</sup>

---

## Results in Brief

For the fiscal year 2005 budget, OMB developed processes and criteria for including IT projects (investments) on its Management Watch List. In doing so, it identified opportunities to strengthen investments and promote improvements in IT management. However, OMB did not develop a single, aggregate list identifying the projects and their weaknesses. Instead, according to OMB officials, individual OMB analysts assigned scores to the justifications for funding (known as exhibit 300s) that are submitted by federal agencies. (These scores were based on criteria

---

<sup>1</sup> GAO, *Information Technology: OMB Can Make More Effective Use of Its Investment Reviews*, GAO-05-276 (Washington, D.C.: Apr. 15, 2005).

---

established in the office's Circular A-11.) OMB delegated individual analysts on its staff, each of whom is typically assigned responsibility for several federal agencies, with maintaining, for their respective agencies, information for the IT projects included on the list. To derive the total number of projects on the list that OMB reported (621 for fiscal year 2005), OMB polled its individual analysts and compiled the numbers. According to OMB officials, they did not construct a single list of projects meeting their watch list criteria because they did not see such an activity as necessary for performing OMB's predominant mission: to assist in overseeing the preparation of the federal budget and to supervise agency budget administration. Thus, OMB did not exploit the opportunity to use the list as a tool for analyzing IT investments on a governmentwide basis, limiting its ability to identify and report on the full set of IT investments requiring corrective actions.

OMB asked agencies to take corrective actions to address weaknesses associated with projects on the Management Watch List, but it did not develop a structured, consistent process for deciding how to follow up on these actions. According to OMB officials, decisions on follow-up and monitoring of specific projects were typically made by the OMB staff with responsibility for reviewing individual agency budget submissions, depending on the staff's insights into agency operations and objectives. Because it did not consistently monitor the follow-up performed, OMB could not tell us which of the 621 projects identified on the fiscal year 2005 list received follow-up attention, and it did not know whether the specific project risks that it identified through its Management Watch List were being managed effectively. This approach could leave resources at risk of being committed to poorly planned and managed projects. Thus, OMB was not using its Management Watch List as a tool for improving IT investments on a governmentwide basis and focusing attention where it was most needed.

To enable OMB to take advantage of the potential benefits of using the Management Watch List as a tool for analyzing and following up on IT investments, we recommended in our report that OMB develop a single, aggregate Management Watch List, and that it develop and use criteria for prioritizing and monitoring the projects on the list. We also recommended that the office use the prioritized

---

list for reporting to the Congress as part of its statutory reporting responsibilities. In commenting on a draft of this report, OMB did not agree that the aggregated governmentwide list recommended by GAO is necessary for adequate oversight and management. However, GAO continues to believe that an aggregated Management Watch List would contribute to OMB's ability to analyze IT investments governmentwide and track progress in addressing deficiencies.

---

## Background

The President's Budget for Fiscal Year 2005 identified approximately \$60 billion for IT projects. In that budget, OMB stated that, of approximately 1,200 major IT projects, about half—621 projects, representing about \$22 billion—were on a Management Watch List. In testimony in March 2004,<sup>2</sup> OMB officials explained that the fiscal year 2005 budget process required agencies to successfully correct project weaknesses and business case deficiencies of projects on the Management Watch List; otherwise, OMB would limit agencies' spending on new starts and other developmental activities.

In the most recent budget, that for fiscal year 2006, OMB continued its use of a Management Watch List. This budget includes 1,087 IT projects, totaling about \$65 billion. Of this total, 342 projects, representing about \$15 billion, are on the Management Watch List. The budget also stated that projects on the Management Watch List had to address performance, security, or other related issues before funding would be obligated in fiscal year 2006.

---

<sup>2</sup>On March 3, 2004, OMB's Deputy Director for Management and its Administrator for Electronic Government and Information Technology testified at a hearing conducted by the Subcommittee on Technology, Information Policy, Intergovernmental Relations and the Census, Committee on Government Reform, House of Representatives. The hearing topic was "Federal Information Technology Investment Management, Strategic Planning, and Performance Measurement: \$60 Billion Reasons Why."

---

According to OMB officials, the office identifies projects for the Management Watch List through their evaluation of justifications for funding that agencies submit for major IT projects as part of the budget development process. This evaluation is carried out as part of OMB's predominant mission: to assist the President in overseeing the preparation of the federal budget and to supervise budget administration in executive branch agencies. OMB is also responsible for evaluating the effectiveness of agency programs, policies, and procedures; assessing competing funding demands among agencies; and setting funding priorities. Finally, OMB is responsible for overseeing and coordinating the administration's policies regarding procurement, financial management, information, and regulations. In each of these three areas of responsibility, OMB's role is to help improve administrative management, to develop better performance measures and coordinating mechanisms, and to reduce unnecessary burden on the public.

To drive improvement in the implementation and management of IT projects, the Congress enacted the Clinger-Cohen Act in 1996, which expanded the responsibilities of the agencies and OMB under the Paperwork Reduction Act.<sup>4</sup> Under the act, agencies are required to engage in capital planning and performance- and results-based management. OMB is required to establish processes to analyze, track, and evaluate the risks and results of major capital investments in information systems made by executive agencies. OMB is also required to report to the Congress on the net program performance benefits achieved as a result of major capital investments in information systems that are made by executive agencies.<sup>4</sup>

In response to the Clinger-Cohen Act and other statutes, OMB developed section 300 of Circular A-11. This section provides policy for planning, budgeting, acquisition, and management of federal capital assets and instructs agencies on budget justification and

---

<sup>3</sup>41 U.S.C. § 3501(a)(1)(B)(vi) (OMB); 41 U.S.C. § 3506(h)(5) (agencies).

<sup>4</sup>These requirements are specifically described in the Clinger-Cohen Act, 40 U.S.C. § 11302(c).



---

reporting requirements for major IT investments.<sup>5</sup> Section 300 defines the budget exhibit 300 as a document that agencies submit to OMB to justify resource requests for major IT investments. This reporting mechanism (part of the budget formulation and review process) is intended to enable an agency to demonstrate to its own management, as well as to OMB, that it has employed the disciplines of good project management; developed a strong business case for the investment; and met other Administration priorities in defining the cost, schedule, and performance goals proposed for the investment. The exhibit 300 includes information that is intended, among other things, to help OMB and the agencies identify and correct poorly planned or performing investments (i.e., investments that are behind schedule, over budget, or not delivering expected results) and real or potential systemic weaknesses in federal information resource management (e.g., project manager qualifications).

According to OMB's description of its processes, agencies' exhibit 300 business cases are reviewed by OMB analysts from its four statutory offices—the Offices of E-Government and Information Technology (e-Gov), Information and Regulatory Affairs (OIRA), Federal Financial Management, and Federal Procurement Policy—and its Resource Management Offices (RMO). Within OIRA, each of about 12 analysts is responsible for overseeing IT projects for a specific agency or (more commonly) several agencies. According to OMB officials, the OIRA and e-Gov analysts, along with RMO program examiners, evaluate and score agency exhibit 300 business cases as part of the development of the President's Budget. The results of this review are provided to agencies through what is called the "passback" process. That is, OMB passes the requests back to agencies with its evaluation, which identifies any areas requiring remediation.

---

<sup>5</sup>OMB Circular A 11 defines a major IT investment as an investment that requires special management attention because of its importance to an agency's mission or because it is an integral part of the agency's enterprise architecture, has significant program or policy implications, has high executive visibility, or is defined as major by the agency's capital planning and investment control process.

---

The integrity of this review process presupposes that the exhibit 300s are accurate. In response to a request from this committee, we are currently reviewing the quality of the information that underlies exhibit 300s at several agencies. We will be reporting on this work in the fall of this year.

---

## OMB Established Processes and Criteria for Identifying Weak Projects, but It Did Not Use an Aggregate List to Perform Its Analysis or Oversight

According to OMB officials, including the Deputy Administrator of OIRA and the Chief of the Information Technology and Policy Branch, OMB staff identified projects for the Management Watch List through their evaluation of the exhibit 300s that agencies submit for major IT projects as part of the budget development process. The OMB officials added that the scoring of agency exhibit 300s is based on guidance in OMB Circular A-11<sup>6</sup> that is intended to ensure that agency planning and management of capital assets are consistent with OMB policy and guidance.

As described in Circular A-11, the scoring of a business case consists of individual scoring for 10 categories, as well as a total composite score of all the categories. (Examples of these 10 categories are performance goals, security and privacy, performance-based management system—including the earned value management system<sup>7</sup>—and support of the President's Management Agenda.) According to Circular A-11, scores range from 1 to 5, with 5 indicating investments whose business cases provided the best justification and 1 the least.

---

<sup>6</sup>These scoring criteria are presented in Office of Management and Budget Circular A-11, Part 7, *Planning, Budgeting, Acquisition, and Management of Capital Assets* (July 2004).

<sup>7</sup>Earned value management is a project management tool that integrates the investment scope of work with schedule and cost elements for investment planning and control. This method compares the value of work accomplished during a given period with that of the work expected in the period. Differences in expectations are measured in both cost and schedule variances.

---

OMB officials said that, for fiscal year 2005, an IT project was placed on the Management Watch List if its exhibit 300 business case received a total composite score of 3 or less, or if it received a score of 3 or less in the areas of performance goals, performance-based management systems, or security and privacy, even if its overall score was a 4 or 5. OMB reported that agencies with weaknesses in these three areas were to submit remediation plans addressing the weaknesses.

According to OMB management, individual analysts were responsible for evaluating projects and determining which projects met the criteria to be on the Management Watch List for their assigned agencies. To derive the total number of projects on the list that were reported for fiscal year 2005, OMB polled the individual analysts and compiled the numbers.

OMB officials said that they did not aggregate these projects into a single list describing projects and their weaknesses, because they did not see such an activity as necessary in performing OMB's predominant mission. Further, OMB officials stated that the limited number of analysts involved enabled them to explore governmentwide issues using ad hoc queries and to develop approaches to address systemic problems without the use of an aggregate list. They pointed at successes in improving IT management, such as better compliance with security requirements, as examples of the effectiveness of their current approach.

Nevertheless, OMB has not fully exploited the opportunity to use its Management Watch List as a tool for analyzing IT investments on a governmentwide basis. According to the Clinger-Cohen Act, OMB is required to establish processes to analyze, track, and evaluate the risks and results of major IT capital investments made by executive agencies, which aggregation of the Management Watch List would facilitate. Without aggregation, OMB's ability to conduct governmentwide analysis is limited, since no governmentwide dataset exists—only a set of subordinate datasets in the hands of individual analysts. In addition, each time an up-to-date report is required, OMB must query all its analysts to assemble an aggregate response; thus, the office cannot efficiently identify, analyze, and report on the full set of IT investments requiring corrective actions.

---

---

## OMB's Monitoring of Projects Was Inconsistent, and Agency Follow-up Activities Were Not Tracked Centrally

OMB asked agencies to take corrective actions to address weaknesses associated with projects on the Management Watch List, but it did not develop a structured, consistent process or criteria for deciding how to follow up on these actions. Instead, OMB officials, including the Deputy Administrator of OIRA and the Chief of the Information Technology and Policy Branch, said that the decision on whether and how to follow up on a specific project was typically made jointly between the OIRA analyst and the RMO program examiner who had responsibility for the individual agency, and that follow-up on specific projects was driven by a number of factors, only one of which was inclusion on the Management Watch List. According to these officials, those Management Watch List projects that did receive specific follow-up attention received feedback through the passback process, through targeted evaluation of remediation plans designed to address weaknesses, and through the apportioning of funds so that the use of budgeted dollars was conditional on appropriate remediation plans being in place.<sup>9</sup>

These officials also said that follow-up of some Management Watch List projects was done through quarterly e-Gov Scorecards; these are reports that use a red/yellow/green scoring system to illustrate the results of OMB's evaluation of the agencies' implementation of e-government criteria in the President's Management Agenda. OMB determines the scores in quarterly reviews, in which it evaluates agency progress toward agreed-upon goals along several dimensions. The e-gov scores are part of the input to the quarterly reporting on the President's Management Agenda.

OMB officials also stated that those Management Watch List projects that did receive follow-up attention were not tracked centrally, but only by the individual OMB analysts. Accordingly, OMB could not readily tell us which of the 621 watch list projects

---

<sup>9</sup>The authority for apportioning funds is specifically described in the Clinger Cohen Act, 40 U.S.C. § 11303(b)(5)(B)(ii).

---

for fiscal year 2005 were followed up on, nor could it use the list to describe the relationship between its follow-up activities and the changes in the numbers of projects on the watch list between fiscal year 2005 (621 projects) and fiscal year 2006 (342 projects).

OMB does not have specific criteria for prioritizing follow-up on Management Watch List projects. Without specific criteria, OMB staff may be agreeing to commit resources to follow up on projects that did not represent OMB's top priorities from a governmentwide perspective. For example, inconsistent attention to OMB priorities, such as earned value management, could undermine the objectives that OMB set in these areas. In addition, major projects with significant management deficiencies may have continued to absorb critical agency resources.

In order for OMB management to have assurance that IT program deficiencies are addressed, it is critical that corrective actions associated with Management Watch List projects be monitored. Such monitoring is instrumental in ensuring that agencies address and resolve weaknesses found in exhibit 300s, which may indicate underlying weaknesses in project planning or management. Tracking agency follow-up activities is essential to enabling OMB to determine progress on both specific projects and governmentwide trends. Without tracking specific follow-up activities, OMB could not readily ascertain whether the risks that it identified through its Management Watch List were being managed effectively; if they were not, funds were potentially being spent on poorly planned and managed projects.

---

In summary, OMB's scoring of agency IT budget submissions and identification of weaknesses has resulted in opportunities to strengthen investments. However, the office has not taken the next step—to develop a single, aggregate list identifying the projects and their weaknesses—and it has not developed a structured, consistent process for deciding how to follow up on corrective actions. OMB's approach does not fully exploit the insights developed through the scoring process, and it may leave unattended weak projects consuming significant budget dollars. Developing an aggregated list

---

would help OMB to realize more fully the potential benefits of using the Management Watch List as a tool for monitoring and analyzing IT investments governmentwide. Accordingly, in our report we recommended that the Director of OMB take the following four actions:

- Develop a central list of projects and their deficiencies.
- Use the list as the basis for selecting projects for follow-up and for tracking follow-up activities;
  - to guide follow-up, develop specific criteria for prioritizing the IT projects included on the list, taking into consideration such factors as the relative potential financial and program benefits of these IT projects, as well as potential risks.
- Analyze the prioritized list to develop governmentwide and agency assessments of the progress and risks of IT investments, identifying opportunities for continued improvement.
- Report to the Congress on progress made in addressing risks of major IT investments and management areas needing attention.

In commenting on a draft of this report, OMB's Administrator of the Office of E-Government and Information Technology expressed appreciation for our review of OMB's use of its Management Watch List. However, the Administrator disagreed with our assessment that an aggregated governmentwide list is necessary to perform adequate oversight and management, and that OMB does not know whether risks are being addressed. Nonetheless, based on OMB's inability to easily report which of the 621 investments on the Management Watch List remained deficient or how much of the \$22 billion cited in the President's Budget remained at risk, we continue to believe that an aggregate list would facilitate OMB's ability to track progress.

Mr. Chairman, that concludes my testimony. I would be pleased to answer any questions that you and the other Members of the Committee may have.

---

---

## Contact and Acknowledgements

For further information, please contact David A. Powner at (202) 512-9286 or Lester Diamond at (202) 512-7957. We can also be reached by e-mail at [pownerd@gao.gov](mailto:pownerd@gao.gov) or [diamondl@gao.gov](mailto:diamondl@gao.gov). Key contributors to this testimony were William G. Barrick, Barbara Collier, Lester Diamond, and Sandra Kerr.

---

---

This is a work of the U.S. government and is not subject to copyright protection in the United States. It may be reproduced and distributed in its entirety without further permission from GAO. However, because this work may contain copyrighted images or other material, permission from the copyright holder may be necessary if you wish to reproduce this material separately.



---

---

## GAO's Mission

The Government Accountability Office, the audit, evaluation and investigative arm of Congress, exists to support Congress in meeting its constitutional responsibilities and to help improve the performance and accountability of the federal government for the American people. GAO examines the use of public funds; evaluates federal programs and policies; and provides analyses, recommendations, and other assistance to help Congress make informed oversight, policy, and funding decisions. GAO's commitment to good government is reflected in its core values of accountability, integrity, and reliability.

---

## Obtaining Copies of GAO Reports and Testimony

The fastest and easiest way to obtain copies of GAO documents at no cost is through GAO's Web site ([www.gao.gov](http://www.gao.gov)). Each weekday, GAO posts newly released reports, testimony, and correspondence on its Web site. To have GAO e-mail you a list of newly posted products every afternoon, go to [www.gao.gov](http://www.gao.gov) and select "Subscribe to Updates."

---

## Order by Mail or Phone

The first copy of each printed report is free. Additional copies are \$2 each. A check or money order should be made out to the Superintendent of Documents. GAO also accepts VISA and Mastercard. Orders for 100 or more copies mailed to a single address are discounted 25 percent. Orders should be sent to:

U.S. Government Accountability Office  
441 G Street NW, Room LM  
Washington, D.C. 20548

To order by Phone: Voice: (202) 512-6000  
TDD: (202) 512-2537  
Fax: (202) 512-6061

---

## To Report Fraud, Waste, and Abuse in Federal Programs

Contact:

Web site: [www.gao.gov/fraudnet/fraudnet.htm](http://www.gao.gov/fraudnet/fraudnet.htm)  
E-mail: [fraudnet@gao.gov](mailto:fraudnet@gao.gov)  
Automated answering system: (800) 424-5454 or (202) 512-7470

---

## Congressional Relations

Gloria Jarnon, Managing Director, [JarnonG@gao.gov](mailto:JarnonG@gao.gov) (202) 512-4400  
U.S. Government Accountability Office, 441 G Street NW, Room 7125  
Washington, D.C. 20548

---

## Public Affairs

Paul Anderson, Managing Director, [AndersonP1@gao.gov](mailto:AndersonP1@gao.gov) (202) 512-4800  
U.S. Government Accountability Office, 441 G Street NW, Room 7149  
Washington, D.C. 20548

