



**UNIVERSIDAD CATÓLICA ANDRÉS BELLO  
VICERRECTORADO ACADÉMICO  
DIRECCIÓN GENERAL DE ESTUDIOS DE POSTGRADO  
ÁREA DE CIENCIAS ADMINISTRATIVAS Y DE GESTIÓN  
POSTGRADO EN GERENCIA DE PROYECTOS**

**TRABAJO ESPECIAL DE GRADO**

***DISEÑO DE INSTRUCCIÓN PARA EL ADIESTRAMIENTO DE  
DESARROLLADORES DE SOFTWARE EN SSCA***

Presentado a la Universidad Católica Andrés Bello,

Por:

**FELIPE ALBERTO FREITES AMER**

Como requisito parcial para optar al grado de:

**ESPECIALISTA EN GERENCIA DE PROYECTOS**

Asesor:

**ARIADNA FLORES VITELLI**

**Caracas, Junio de 2.007**

## **DEDICATORIA**

A mi mamá, Elvira, quien siempre me ha apoyado incondicionalmente enseñándome a ser lo mejor que puedo ser... ¡Gracias por estar ahí siempre!

A mi Papá, Juvenal, quien me ha dado los mejores ejemplos ... Gracias por darme alas y dejarme volar.

A mi Hermana Elvira, quien es mi mayor ejemplo cada vez que doy un nuevo paso en mi vida.

A todos Ustedes quienes son mi principal motivación les dedico mi logro, porque sé cuan orgullosos se sienten cada vez que alcanzo una nueva meta.

## **RECONOCIMIENTOS**

A mi Tutor Lic. Ariadna Flores Vitelli, por su acertada intervención para el desarrollo y culminación de este trabajo.

A la Lic. Elvira E. Amer P., por guiar cada etapa en el desarrollo de este trabajo, fue un gran apoyo.

A los expertos y a la organización que aportaron su conocimiento y experiencia para el desarrollo de esta investigación.

Y por último, quiero expresar mi especial reconocimiento a todas las personas que de una u otra forma colaboraron para llevar a cabo este estudio.

**UNIVERSIDAD CATÓLICA ANDRÉS BELLO**  
**VICERRECTORADO ACADÉMICO**  
**DIRECCIÓN GENERAL DE ESTUDIOS DE POSTGRADO**  
**ÁREA DE CIENCIAS ADMINISTRATIVAS Y DE GESTIÓN**  
**POSTGRADO EN GERENCIA DE PROYECTOS**  
**TRABAJO ESPECIAL DE GRADO**

*DISEÑO DE INSTRUCCIÓN PARA EL ADIESTRAMIENTO DE DESARROLLADORES DE  
SOFTWARE EN SSCA*

Asesor: Flores Vitelli, Ariadna  
Autor: Freitas Amer, Felipe  
Fecha: Junio, 2007

El presente proyecto está destinado a la elaboración de un Diseño de Instrucción para el adiestramiento de los desarrolladores de *software* de SSCA. A partir de un diagnóstico previo, se determinó que las constantes quejas de los clientes, relacionadas con los retrasos en la entrega de producto y sobre defecto en las aplicaciones, más los informes de los Líderes sobre las fallas y la dificultad que presentan los desarrolladores para resolverlas sin su intervención, surgían generalmente en la etapa de construcción. Para formular el diseño de instrucción fue necesario identificar con la mayor precisión posible las necesidades de conocimientos y habilidades existentes en el personal que se ocupa de esa etapa. El presente trabajo está enmarcado en el tipo investigación descriptiva y se realizó en tres fases. En la fase 1, se realizó un diagnóstico para obtener los datos requeridos para fundamentar el contenido del diseño, esto incluyó revisión documental, llevada a efecto en el repositorio de la empresa y trabajo de campo, en el cual se realizaron entrevistas a cinco (5) líderes supervisores de los desarrolladores, seleccionados intencionalmente por su formación académica, su amplia experiencia, su nivel de competencia y compromiso organizacional. Seguidamente, en la fase 2, se analizaron cuantitativamente los datos obtenidos en el repositorio de la empresa y cualitativamente los datos obtenidos en el trabajo de campo, con lo que se definieron precisamente los requerimientos de adiestramiento. Por último, en la fase 3, se formuló el diseño instruccional fundamentado en las necesidades de adiestramiento de los desarrolladores identificadas. El diseño instruccional es del tipo teórico - práctico y está dirigido a desarrolladores recién contratados. Como recomendación, se debe realizar un piloto que permita validar y ajustar el diseño realizado. Una vez validado, el producto pasará a ser un activo disponible a todas las unidades de la organización, quienes serán responsables de determinar en qué momento necesitan ejecutarlo. A largo plazo, se espera que se mejoren los tiempos de respuesta de las actividades asignadas y que se mejore la calidad del producto elaborado.

**Palabras clave:** *Inducción, ingeniería de software, diseño de sistemas, programación, aplicaciones*

# ÍNDICE GENERAL

|                                                  |           |
|--------------------------------------------------|-----------|
| DEDICATORIA .....                                | II        |
| RECONOCIMIENTOS .....                            | III       |
| RESUMEN .....                                    | IV        |
| INTRODUCCIÓN .....                               | 1         |
| <b>CAPÍTULO I: EL PROBLEMA .....</b>             | <b>3</b>  |
| 1.    PLANTEAMIENTO DEL PROBLEMA .....           | 3         |
| 2.    OBJETIVOS DEL PROYECTO .....               | 6         |
| 2.1.  OBJETIVO GENERAL .....                     | 6         |
| 2.2.  OBJETIVOS ESPECÍFICOS.....                 | 6         |
| 3.    JUSTIFICACIÓN DEL PROYECTO .....           | 6         |
| 4.    LIMITACIONES .....                         | 8         |
| <b>CAPÍTULO II: MARCO REFERENCIAL .....</b>      | <b>9</b>  |
| 1.    MARCO CONCEPTUAL .....                     | 9         |
| 1.1  GERENCIA DE PROYECTOS .....                 | 9         |
| 1.2  TEORÍAS DE ENSEÑANZA.....                   | 12        |
| 1.3  COMPETENCIAS PROFESIONALES.....             | 20        |
| 1.4  DESARROLLO DE SOFTWARE .....                | 22        |
| 2.    TÉRMINOS ASOCIADOS .....                   | 32        |
| <b>CAPÍTULO III: MARCO METODOLÓGICO .....</b>    | <b>36</b> |
| 1.    TIPO Y DISEÑO DE LA INVESTIGACIÓN .....    | 36        |
| 2.    UNIDAD DE ANÁLISIS .....                   | 38        |
| 3.    POBLACIÓN Y MUESTRA.....                   | 38        |
| 4.    RECOLECCIÓN DE LA DATOS.....               | 39        |
| 5.    PROCESAMIENTO DE LOS DATOS .....           | 41        |
| 6.    METODOLOGÍA.....                           | 42        |
| 7.    RESULTADOS ESPERADOS E IMPLICACIONES.....  | 43        |
| 8.    CONSIDERACIONES ÉTICAS.....                | 43        |
| <b>CAPÍTULO IV: MARCO ORGANIZACIONAL.....</b>    | <b>45</b> |
| 1.    ANTECEDENTES.....                          | 45        |
| 2.    OBJETIVO.....                              | 46        |
| 3.    PERFIL .....                               | 46        |
| 4.    VISIÓN.....                                | 46        |
| 5.    MISIÓN .....                               | 46        |
| 6.    POLÍTICA DE LA CALIDAD .....               | 47        |
| 7.    OBJETIVOS GENERALES DE LA CALIDAD .....    | 47        |
| 8.    COMPROMISO.....                            | 47        |
| 9.    ESTRATEGIA COMPETITIVA.....                | 48        |
| 10.  MODELO DE GESTIÓN .....                     | 48        |
| 11.  ESTRUCTURA DE TRABAJO.....                  | 49        |
| 12.  ESTRUCTURA ORGANIZATIVA .....               | 49        |
| <b>CAPÍTULO V: DESARROLLO DEL PROYECTO .....</b> | <b>51</b> |

|                                                            |                                                                     |           |
|------------------------------------------------------------|---------------------------------------------------------------------|-----------|
| 1.                                                         | FASE 1: DIAGNÓSTICO.....                                            | 51        |
| 1.1.                                                       | <i>Conocimientos mínimos del desarrollador: .....</i>               | <i>51</i> |
| 1.2.                                                       | <i>Conocimientos de los Desarrolladores - Situación Actual.....</i> | <i>55</i> |
| <b>CAPÍTULO VI: RESULTADOS DEL PROYECTO .....</b>          |                                                                     | <b>73</b> |
| 1.                                                         | FASE 2: ANÁLISIS .....                                              | 73        |
| 2.                                                         | FASE 3: DISEÑO .....                                                | 78        |
| 3.                                                         | CRONOGRAMA DE ACTIVIDADES.....                                      | 83        |
| 4.                                                         | ESTIMACIÓN DE COSTO DEL CURSO .....                                 | 84        |
| <b>CAPÍTULO VII: EVALUACIÓN DE LOS RESULTADOS.....</b>     |                                                                     | <b>86</b> |
| <b>CAPÍTULO VIII: CONCLUSIONES Y RECOMENDACIONES .....</b> |                                                                     | <b>88</b> |
| 1.                                                         | CONCLUSIONES .....                                                  | 88        |
| 2.                                                         | RECOMENDACIONES .....                                               | 90        |
| <b>BIBLIOGRAFÍA .....</b>                                  |                                                                     | <b>91</b> |
| <b>ANEXOS .....</b>                                        |                                                                     | <b>94</b> |
| 1.                                                         | ANEXO 1 - ENTREVISTA.....                                           | 95        |

## ÍNDICE DE TABLAS

|                                                                  |    |
|------------------------------------------------------------------|----|
| TABLA 1 NIVELES DE MADUREZ.....                                  | 28 |
| TABLA 2 ERRORES REGISTRADOS EN JIRA.....                         | 57 |
| TABLA 3 RETRASO DE LOS PROYECTOS .....                           | 58 |
| TABLA 4 ERRORES DE DOCUMENTACIÓN .....                           | 59 |
| TABLA 5 RESULTADOS DE LA REVISIÓN DEL FUENTE .....               | 60 |
| TABLA 6 ERRORES DETECTADOS CON FXCOP .....                       | 61 |
| TABLA 7 REQUERIMIENTOS DE ADIESTRAMIENTO.....                    | 77 |
| TABLA 8 CONTENIDO PROGRAMÁTICO DEL CURSO DE ADIESTRAMIENTO ..... | 80 |
| TABLA 9 CRONOGRAMA DE EJECUCIÓN DEL CURSO .....                  | 83 |
| TABLA 10 COSTO DE EJECUCIÓN DEL CURSO PARA 5 PARTICIPANTES ..... | 84 |
| TABLA 11 OBJETIVOS VERSUS CUMPLIMIENTO .....                     | 86 |

## ÍNDICE DE FIGURAS

|                                              |    |
|----------------------------------------------|----|
| FIGURA 1 FASES DEL DISEÑO INSTRUCCIONAL..... | 15 |
| FIGURA 2 MODELO DE DICK Y CAREY .....        | 17 |
| FIGURA 3 FLUJOGRAMA DE GESTIÓN .....         | 48 |
| FIGURA 4 ORGANIGRAMA DE SSCA.....            | 50 |



## INTRODUCCIÓN

Los conocimientos, destrezas y competencias de los ingenieros contratados por SSCA con frecuencia están por debajo del nivel requerido para incorporarse a los proyectos de desarrollo de *software* sin causar inconvenientes, ruido en la aplicación o retrasos. Dichas deficiencias se observan en mayor o menor medida tanto en los recién contratados como en los que tienen algún tiempo trabajando en la empresa. Esto trae consigo déficit en la calidad de servicio que ha certificado la empresa y que la hace competitiva en el mercado del producto a nivel nacional e internacional.

Considerado el problema y las circunstancias, se ha decidido proveer a los ingenieros de un adiestramiento intensivo que en muy pocas semanas les permita adquirir o reforzar el conjunto de conocimientos, competencias, y destrezas requeridos por el modelo de negocios de la empresa y que lo hagan de forma operativa. El proyecto está destinado a la elaboración de un Diseño de Instrucción para el adiestramiento de los desarrolladores de *software*. Para esto es preciso identificar con la mayor precisión posible las necesidades de conocimientos y habilidades existentes en ellos. La implementación del Diseño de Instrucción reducirá el riesgo de re-trabajo o fallas en la calidad del *software*, lo cual implicará ahorros en tiempo y por ende ahorros económicos. Adicionalmente mejorará el nivel técnico del personal e impulsará el cumplimiento de las metas de calidad.

El presente trabajo se estructuró en ocho capítulos: en el Capítulo I, ***El Problema***, se detalla la problemática a ser tratada, la justificación del proyecto destacando su relevancia, se identifican los objetivos de la investigación y se establece el alcance del trabajo a ser realizado. A continuación, en el Capítulo II, ***Marco Referencial***, se incorporan las bases teóricas que sustentan la investigación. Incluye conceptos de Gerencia de Proyectos, teorías sobre adiestramiento e inducción y teorías sobre desarrollo de *software*. Seguidamente, en el Capítulo III,

**Marco Metodológico**, se expone el tipo y diseño de la investigación, la unidad de análisis, la población y muestra objeto de estudio, los distintos métodos y técnicas que posibilitaron la obtención los datos para la investigación, así como las técnicas utilizadas para el análisis de los datos producto de la investigación. En el capítulo IV, **Marco Organizacional**, se describe la organización donde se realizó la investigación, se hace referencia a sus objetivos, misión, visión, estrategias de calidad, modelo de gestión y su estructura organizacional. En el capítulo V, **Desarrollo del Proyecto**, se presenta la primera fase metodológica, **Diagnóstico**, mostrando primero el perfil deseado de los desarrolladores y luego la situación actual de SSCA mediante tablas y cuadros, utilizando la información concerniente a la revisión documental y el trabajo de campo. El capítulo VI, **Resultados del Proyecto**, incluye la fase 2, **Análisis**, donde se analizan e interpretan los datos obtenidos para dar respuesta a los objetivos generales: definir los conocimientos, competencias y destrezas específicas mínimas del cargo de desarrollador de *Software* en SSCA y evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de *software* en SSCA, se definen cuáles son las necesidades de adiestramiento de los desarrolladores; y la fase 3, **Diseño**, que muestra el desarrollo del diseño instruccional para adiestramiento de desarrolladores de *software* de SSCA para atender esas necesidades. En el Capítulo VII, **Evaluación de los resultados**, se evalúa el cumplimiento de los objetivos planteados en el capítulo I. En el capítulo VIII se resumen las **Conclusiones y Recomendaciones** del trabajo realizado. Y por último se presenta la Bibliografía utilizada en la investigación, así como los Anexos.

## CAPÍTULO I: EL PROBLEMA

### 1. PLANTEAMIENTO DEL PROBLEMA

SSCA es una empresa Venezolana proveedora de servicios de tecnología de la información al mercado global. Se dedica al desarrollo de aplicaciones, a la consultoría en tecnología de la información, y al *outsourcing* de procesos de negocios.

Específicamente para el desarrollo de aplicaciones, la empresa cuenta con una metodología madurada por más de 15 años y por cada proyecto se conforma un Equipo de Desarrollo del Proyecto (*PRT - Project Realization Team*) que involucra miembros especialistas de cada una de las áreas del desarrollo de una aplicación.

En los proyectos que actualmente se están ejecutando en SSCA se observan retrasos y esto repercute sobre la calidad de servicio requerida por el cliente. Esta característica se ha observado también en proyectos realizados anteriormente.

Cada uno de los miembros del equipo asignados a cada proyecto tiene bien definidas sus responsabilidades y debe poseer la capacidad básica necesaria para llevar a cabo el proyecto; sin embargo las competencias de los desarrolladores contratados por SSCA con frecuencia no alcanzan el nivel esperado y esto se debe a que no poseen los conocimientos y destrezas necesarios para asumir el proceso de desarrollo, estando por debajo del nivel requerido para incorporarse a los proyectos de desarrollo de *software* sin causar ruido, inconvenientes o retrasos. Dichas deficiencias se observan en mayor o menor medida tanto en los recién contratados como en los que tienen algún tiempo trabajando en la empresa.

Las causas de origen de las deficiencias observadas son variadas y están en su mayor parte relacionadas con las características del entorno, formación

académica: pensum y métodos de enseñanza en las universidades, o han adquirido la experiencia laboral que les sirve de fundamento para el desempeño en otras empresas. Es difícil para SSCA atenderlas directamente, ya que el proceso de selección de personal no resuelve el problema, y modificar el entorno es una labor de magnitud considerable.

Contar con personal entrenado en tecnologías y metodologías de trabajo de ingeniería de *software* reduce el riesgo de re-trabajo o fallas en la calidad del *software*, lo cual implica ahorros en tiempo y, por ende, ahorros económicos.

Se desea que todo desarrollador pueda incorporarse a cualquier proyecto en pleno conocimiento de los procedimientos y métodos de la empresa, acoplándose al equipo de trabajo, y produciendo *software* de la calidad requerida por el modelo de negocios de la empresa.

Para SSCA ha sido un reto tratar de superar esta dificultad. Las estrategias empleadas para solventar la situación se han enfocado en el reclutamiento de nuevos recursos humanos así como enviar a su personal a recibir adiestramiento; aun los resultados se hacen esperar y el déficit en el proceso de desarrollo se mantiene.

De continuar esta situación, la empresa puede perder credibilidad con los clientes por no producir con la calidad y en el tiempo requeridos; lo cual puede inducir al cliente a reducir la contratación de sus proyectos con la empresa e inclusive llegar a prescindir de sus servicios como proveedora; lo que sin duda afectaría directamente los capitales y la estabilidad de la empresa.

Considerado el problema y las circunstancias, surgen las interrogantes ¿se podría resolver el problema si se facilita a los desarrolladores de SSCA un curso de entrenamiento intensivo, que en muy pocas semanas permita que los participantes

adquieran y/o refuercen el conjunto mínimo de conocimientos y destrezas requeridos para asumir sus competencias en el modelo de negocios de la empresa? ¿Qué contenido programático debe cubrir el diseño de instrucción para adiestrar a los desarrolladores *software* en SSCA?

Para ofrecer un curso destinado a resolver la problemática identificada se amerita realizar un diagnóstico y un análisis de los conocimientos requeridos, así como una evaluación para desarrollar un diseño de instrucción. Es por esto que se pretende realizar una investigación mediante la cual se pueda diseñar el programa de instrucción para adiestrar a los desarrolladores de *software* en SSCA

## **2. OBJETIVOS DEL PROYECTO**

### **2.1.Objetivo General**

- 2.1.1. Formular un Diseño de Instrucción para el adiestramiento del Desarrollador de *Software* en SSCA

### **2.2.Objetivos Específicos**

- 2.2.1. Definir los conocimientos, competencias y destrezas específicas mínimas del cargo de desarrollador de *Software* en SSCA
- 2.2.2. Evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de *Software* en SSCA
- 2.2.3. Desarrollar un Diseño de Instrucción que permita cubrir las diferencias detectadas entre los conocimientos definidos y los reales diagnosticados para los desarrolladores de *Software* en SSCA
- 2.2.4. Estimar el costo de inversión y operación del Diseño de Instrucción clase 5.

## **3. JUSTIFICACIÓN DEL PROYECTO**

Actualmente SSCA posee una certificación de calidad CMMI nivel 2. Dicha acreditación es re-certificada anualmente, lo cual la obliga a cumplir con una serie de estándares de calidad. Esto, eventualmente, podría resultar afectado por cuanto que entre el personal existen debilidades en la etapa de construcción, en aspectos relacionados con el trabajo de los desarrolladores, el cual es deficiente al asumir sus competencias, por presentar déficit de los conocimientos necesarios para cumplir con dichos estándares. Sólo por esto la empresa podría perder la certificación, lo que implicaría perder una de las ventajas competitivas que ofrece. Esta certificación es, además, un requisito indispensable para licitar en proyectos en Estados Unidos, siendo éste uno de sus más lucrativos mercados.

Es un hecho que en el presente año uno de sus proyectos más grandes para ese importante cliente fue cancelado y, dentro de las causas que obligaron esta

acción, alegada por el cliente, fue falta de calidad en el código y significativo retraso en las entregas. Más aun, este cliente informó que no realizaría más proyectos con la empresa. Las pérdidas ocasionadas por la cancelación de este proyecto fueron estimadas por encima 300 mil dólares.

Por otra parte, según el *Project Management Institute* (PMI) (2004), para que un proyecto sea considerado exitoso debe terminar dentro del tiempo y presupuesto estimado, con la calidad requerida y con el cliente satisfecho; por lo tanto, si los proyectos no poseen la calidad necesaria, ni terminan en el tiempo estipulado, no sólo significa que no se trata de proyectos exitosos, sino también que el cliente no estará satisfecho y la empresa automáticamente se coloca en riesgo para mantener su nivel de competitividad y prestigio en el mercado del producto.

Es por ello que se hace imperiosamente necesario incorporar estrategias que contribuyan al fortalecimiento del equipo de producción y, dado que se han identificado debilidades puntuales en el proceso de desarrollo, se debe gerenciar la capacitación de los desarrolladores, en el menor tiempo posible; de esta manera se justifica el Diseño de un Programa de instrucción para el adiestramiento de tan importante recurso. EL producto de este estudio pasará a formar parte de las herramientas disponibles para cada una de las unidades de la organización, siendo éstas las responsables de identificar la frecuencia de ejecución de acuerdo a las necesidades de sus equipos de trabajo.

De contar con el personal entrenado, a pesar que no garantiza en un 100% el éxito del proyecto, se reduce la posibilidad de retrasos, ya que los desarrolladores no tendrán que aprender sobre la marcha desde cero los conocimientos básicos que les estén haciendo falta. Una vez capacitados se pueden dedicar en mayor medida a desarrollar, esforzándose en perfeccionar sus asignaciones, y como no deben aprender sino aplicar conocimientos, también utilizaran productivamente el

tiempo en la ejecución del trabajo, garantizando efectivamente el éxito de la empresa.

El presente trabajo de investigación se realizó en un lapso de 12 semanas en las instalaciones de SSCA, en su sede de Caracas, Venezuela, y permitirá medir resultados, que a su vez puedan servir de referencia a otras empresas así como a investigadores futuros.

Adicionalmente, permite aplicar los conocimientos adquiridos durante la ejecución del postgrado de Gerencia de Proyectos de la Universidad Católica Andrés Bello y es requisito para obtener el título de Especialista en Gerencia de Proyectos.

#### **4. LIMITACIONES**

Por asuntos estratégicos, la compañía donde se realizó la investigación se reserva el derecho de aparecer citada en el presente documento. Para efectos de mantener una redacción consistente se le denominará Servicios de Software de Calidad Acreditada, SSCA.



## **CAPÍTULO II: MARCO REFERENCIAL**

### **1. MARCO CONCEPTUAL**

#### **1.1 GERENCIA DE PROYECTOS**

##### **1.1.1 Proyecto**

Según Palacios (2000), un proyecto es un trabajo que realiza una organización con el objetivo de dirigirse hacia una situación deseada. También es conocido como un conjunto de actividades orientadas a un fin común, que tienen un comienzo y una terminación. En pocas palabras, un proyecto es un trabajo temporal y cuyo resultado es un producto o servicio único, en un ambiente de incertidumbre (planificación), recursos limitados y procedimientos preestablecidos.

Siguiendo en ese mismo orden de ideas, de acuerdo con Palacios, se describen las Fases de un Proyecto:

Visión: es la fase en donde el proyecto es identificado para el plan de negocios y se asegura su alineación con los objetivos del negocio.

Conceptualización: es la fase en donde se seleccionan las mejores opciones y las mejoras en la precisión de estimados de costos y tiempo.

Definición: es la fase para completar el Alcance de las opciones seleccionadas y desarrollar un plan de ejecución detallado que le permita a la corporación comprometer los fondos u obtener el financiamiento requerido.

Implantación: contratar y materializar el plan de ejecución del proyecto hasta la completación.

Operación: Operacionalizar el proyecto y analizar el cumplimiento de las expectativas del negocio

### **1.1.2 Gerencia de Proyectos**

Según el PMI (2004), “la dirección de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer los requisitos del proyecto. La dirección de proyectos se logra mediante la aplicación e integración de los procesos de dirección de proyectos de inicio, planificación, ejecución, seguimiento y control y cierre”.

Las áreas de conocimiento en la Gerencia de Proyectos son:

Gestión de la Integración del Proyecto

Gestión del Alcance del Proyecto

Gestión del Tiempo del Proyecto

Gestión de los Costes del Proyecto

Gestión de la Calidad del Proyecto

Gestión de los Recursos Humanos del Proyecto

Gestión de las Comunicaciones del Proyecto

Gestión de los Riesgos del Proyecto

Gestión de Adquisiciones del Proyecto

A los efectos del presente estudio, a continuación se describirán sólo aquellas áreas que son relevantes.

### **1.1.3 Gestión de la Calidad del Proyecto**

#### **Planificación de la Calidad**

Programa de trabajo mediante el cual se analizan los principales problemas de calidad y oportunidades de mejoramiento continuo de la calidad de la empresa o institución y las relaciones con el entorno.

En su forma básica consiste en:

- Analizar las tendencias posibles de evolución hacia la calidad de la empresa y de los proyectos

- Concretar las metas de la calidad a conseguir, estableciendo los objetivos de la calidad
- Sistematizar el conjunto de evidencias y parámetros de medición de la calidad que permitirán verificar los resultados de la evolución
- Definir las reglas de adaptación que permitirán cambiar de estrategia a lo largo de todo el proceso.

### **Realizar Aseguramiento de Calidad**

Aseguramiento de Calidad son las actividades realizadas para verificar si los procesos ejecutados aseguran que el proyecto cumplirá con los estándares de calidad.

Sus propósitos son:

- Evaluar si los procesos y procedimientos son adecuados y seguidos
- Identificar problemas emergentes y recomendar acciones correctivas
- Verificar que el proyecto cumple o excede los objetivos

### **Realizar Control de Calidad**

Realizar control de calidad implica supervisar los resultados específicos del proyecto, para determinar si cumplen con las normas de calidad relevantes e identificar los modos de eliminar las causas de resultados insatisfactorios. Esto debería ser realizado durante todo el proyecto.

## **1.1.4 Gestión de los Recursos Humanos del Proyecto**

### **Planificación de los Recursos Humanos**

Identificando, documentando, y asignando los roles del proyecto, responsabilidades, y relaciones que le toca a todos y cada uno de los participantes en el proyecto

### **Contratar el Equipo del Proyecto**

Conseguir los recursos humanos idóneos y necesarios para trabajar en el proyecto.

### **Desarrollar el Equipo del Proyecto**

Crecimiento organizacional mediante el aprovechamiento de las habilidades y competencias que aporta cada individuo y que se convierten en fortalezas para realizar el proyecto con alto rendimiento grupal e individual. Así como también mejorar las competencias y la interacción de los miembros del equipo para lograr el mejor rendimiento del proyecto.

### **Gestionar el Equipo del Proyecto**

Hacer un seguimiento del rendimiento de los miembros del equipo, proporcionar retroalimentación, resolver polémicas y coordinar cambios a fin de mejorar el rendimiento del proyecto.

## **1.2 TEORÍAS DE ENSEÑANZA**

### **1.2.1 Inducción:**

#### ***¿Por qué tener inducción?***

Michael Meighan (1992) asevera que la inducción sistemática en la organización con frecuencia se pasa por alto y esto da como resultado la pérdida de una serie de beneficios:

- **Financieros:** una rotación de personal innecesariamente grande aumenta los costos de reclutamiento.
- **Motivacionales:** el personal que se somete a programas de entrenamiento para inducción de calidad tiene más probabilidades de comprometerse a más largo plazo con la organización.

- Capacitación y desarrollo: como la inducción es posiblemente la primera experiencia que el recién entrado tiene respecto al entrenamiento de la organización, es importante que cause buena impresión.
- Vigilancia del ambiente externo: los recién llegados pueden ser una medida efectiva de la forma como se observa la organización desde fuera.
- Influencia del personal actual: la inducción puede tener un efecto benéfico en el personal actual, a través de la participación en el proceso.

### ***¿Quién debe recibir inducción?***

La inducción es una necesidad para todos aquellos que ingresan a organizaciones nuevas, desempeñando nuevos cargos o iniciando nuevas carreras. La siguiente lista indica quienes deben recibir inducción:

- Mujeres que regresan al trabajo
- Desempleados de mucho tiempo
- Grupos étnicos
- Antiguo personal de servicio
- Reinsertados Sociales: Discapacitados, Exdelincuentes
- Bachilleres
- Profesionales
- Recién graduados

La inducción a veces tiene que ver con gente que empieza a trabajar por primera vez, y esto implica una más amplia significación si se toma en consideración todas las cosas nuevas que se hacen.

Todos los programas de inducción se deben basar en necesidades individuales, pero han de tener en cuenta las necesidades de grupos particulares.

La inducción también es importante para la gente que asume nuevos cargos dentro de una organización o cuando se introducen nuevos desarrollos.

### **1.2.2 ¿Qué es un módulo instruccional?**

De acuerdo a Gloria Yukavetsky (2003), un módulo instruccional es un material didáctico que contiene todos los elementos que son necesarios para el aprendizaje de conceptos y destrezas al ritmo de/la estudiante y sin el elemento presencial continuo del instructor.

Es deseable tener un fundamento teórico y práctico al crear o diseñar módulos instruccionales. Por esto se utiliza la metodología que se maneja para la elaboración de los mismos, la metodología del Diseño Instruccional. Por otro lado, el diseñador instruccional es la persona que, por lo general, tiene el conocimiento para llevar a cabo la creación de módulos instruccionales. Sin embargo, siempre y cuando se cumpla con los requisitos metodológicos, cualquier persona puede crear un módulo instruccional.

### **El Diseño Instruccional**

El Diseño Instruccional (DI) es un proceso fundamentado en teorías de disciplinas académicas, especialmente en las disciplinas relativas al aprendizaje humano, que tiene el efecto de maximizar la comprensión, uso y aplicación de la información, a través de estructuras sistemáticas, metodológicas y pedagógicas. Una vez diseñada la instrucción, deberá probarse, evaluarse y revisarse, atendiéndose de forma efectiva las necesidades particulares del individuo.

En su definición más sencilla, el DI es una metodología de planificación pedagógica, que sirve de referencia para producir una variedad de materiales educativos, atemperados a las necesidades estudiantiles, asegurándose así la calidad del aprendizaje.

El DI se nutre de las:

### **Ciencias Sociales**

- La psicología- a través del estudio de las diferencias individuales.
- Teorías de la conducta humana- a través de las teorías conductistas, cognoscitivistas, y constructivistas.

### **Ciencias de la Ingeniería**

- Teoría de sistemas

### **Ciencias de la Información (Informática)**

- Tecnologías del campo de la informática: computadoras, programados, multimedios, telecomunicaciones, micro-onda, satélites, etc.

### **Ciencias**

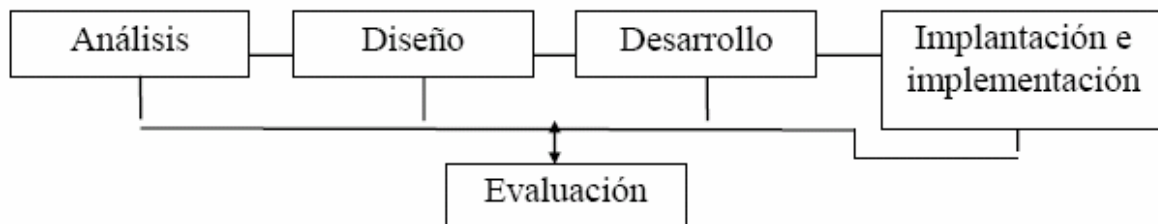
- Método científico

### **Las fases del DI**

La secuencia o fases del DI constituyen el armazón procesal sobre el cual se produce la instrucción de forma sistemática.

Las fases del DI se resumen en el siguiente esquema:

**Figura 1 Fases del Diseño Instruccional**



Fuente: Gloria J. Yukavetsky (2003)

La fase de **Análisis** constituye la base para las demás fases del Diseño Instruccional. En esta fase se define el problema, se identifica la fuente del problema y se determinan las posibles soluciones. En esta fase se utilizan diferentes métodos de investigación, tal como el análisis de necesidades. El producto de esta fase se compone de las metas instruccionales y una lista de las tareas a enseñarse. Estos productos serán los insumos de la fase de diseño.

En la fase de **Diseño** se utiliza el producto de la fase de Análisis para planificar una estrategia y así producir la instrucción. En esta fase se hace un bosquejo de cómo alcanzar las metas instruccionales. Algunos elementos de esta fase incluyen hacer una descripción de la población a impactarse, llevar a cabo un análisis instruccional, redactar objetivos, redactar ítemes para pruebas, determinar cómo se divulgará la instrucción, y diseñar la secuencia de la instrucción. El producto de la fase de **Diseño** es el insumo de la fase de Desarrollo.

En la fase de **Desarrollo** se elaboran los planes de la lección y los materiales que se van a utilizar. En esta fase se elabora la instrucción, los medios que se utilizarán en la instrucción y cualquier otro material necesario, tal como guías de una lección.

En la fase de **Implantación** e **Implementación** se divulga eficiente y efectivamente la instrucción. La misma puede ser implantada en diferentes ambientes: en el salón de clases, en laboratorios o en escenarios donde se utilicen las tecnologías relacionadas a la computadora. En esta fase se propicia la comprensión del material, el dominio de destrezas y objetivos, y la transferencia de conocimiento del ambiente instruccional al ambiente de trabajo.

En la fase de **Evaluación** se evalúa la efectividad y eficiencia de la instrucción. La fase de **Evaluación** deberá darse en todas las fases del proceso instruccional. Existen dos tipos de evaluación: la **Evaluación Formativa** y la **Evaluación**



**Sumativa.** La **Evaluación Formativa** es continua, es decir, se lleva a cabo mientras se están desarrollando las demás fases. El objetivo de este tipo de evaluación es mejorar la instrucción antes de que llegue a la etapa final. La **Evaluación Sumativa** se da cuando se ha implantado la versión final de la instrucción. En este tipo de evaluación se verifica la efectividad total de la instrucción y los hallazgos se utilizan para tomar una decisión final, tal como continuar con un proyecto educativo o comprar materiales instruccionales.

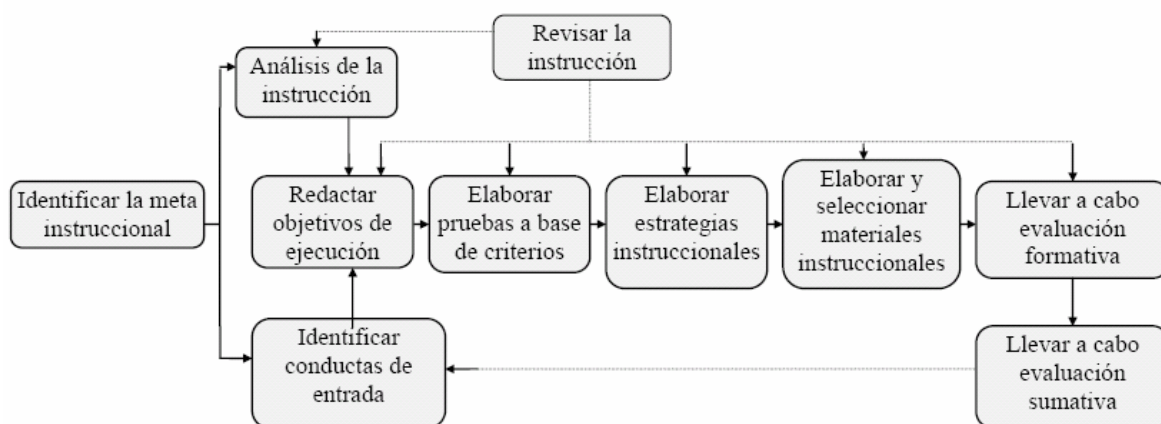
Los modelos de DI se pueden utilizar para producir los siguientes materiales: módulos para lecciones, los cursos de un currículo universitario, y cursos de adiestramientos variados para el mundo del trabajo.

### Ejemplos de modelos de Diseño Instruccional

Existen varios modelos de DI. A continuación se presentan dos: el modelo de Dick y Carey y el modelo ADDIE.

#### Modelo de Dick y Carey

Figura 2 Modelo de Dick y Carey



Fuente: Gloria J. Yukavetsky (2003)

Este modelo utiliza el enfoque de sistemas para el diseño de la instrucción. Es uno de los más conocidos por su naturaleza estructurada. El modelo describe todas las fases de un proceso interactivo, que comienza identificando las metas instruccionales y termina con la Evaluación Sumativa. El modelo se puede aplicar a múltiples escenarios, desde el ambiente educativo hasta el laboral.

### **Modelo ADDIE**

El modelo ADDIE (Análisis, Diseño, Desarrollo, Implantación y Evaluación) es un modelo comúnmente utilizado en el diseño de la instrucción tradicional, aunque más en el medio electrónico (un ejemplo de un medio electrónico es la Internet). Es un modelo genérico y se compone de las siguientes fases:

#### **Análisis**

En la fase de Análisis se determina lo siguiente:

- Las características de la audiencia
- Lo que necesita aprender la audiencia
- El presupuesto disponible
- Los medios de difusión
- Si existen limitaciones
- Fecha límite para entregar o implantar la instrucción
- Las actividades que necesitan hacer los estudiantes para el logro de las competencias

#### **Diseño**

En la fase de Diseño se lleva a cabo lo siguiente:

- Selección del mejor ambiente (ya sea electrónico o tradicional) examinando los tipos de destrezas cognitivas que se requieren para el logro de la meta.
- Señalamiento de los objetivos instruccionales
- Selección de estrategias pedagógicas.
- Bosquejo de unidades, lecciones y módulos.

- Diseño del contenido del curso teniendo en cuenta los medios interactivos electrónicos.

## **Desarrollo**

En la fase de desarrollo se hace lo siguiente:

- Se selecciona, obtiene o se crea el medio requerido.
- Se utiliza la Internet para presentar la información en formatos variados multimediales (la palabra multimediales proviene de la palabra multimedia que significa dos o más medios integrados a una aplicación, programa o experiencia de aprendizaje), para atender las preferencias del estudiantado.
- Se determinan las interacciones apropiadas. Las mismas deben dirigir al estudiante hacia una experiencia creativa, innovadora y de exploración.
- Planificación de actividades que le permitan al estudiantado construir un ambiente social de apoyo.

## **Implantación**

En la fase de implantación se:

- Duplican y distribuyen los materiales.
- Implanta e implementa el curso.
- Resuelven problemas técnicos y se discuten planes alternos.

## **Evaluación**

En la fase de evaluación se lleva a cabo lo siguiente:

- Desarrollo de pruebas para medir los estándares instruccionales.
- Implantación de pruebas y evaluaciones
- Evaluación continúa.
- Planificación de evaluaciones estudiantiles del curso para mantener al instructor consciente de las necesidades de éstos/as.
- Desarrollo de evaluaciones formativas para evaluar el curso.

- Desarrollo de evaluaciones sumativas para emitir un juicio de la efectividad de la instrucción.

### **1.3 COMPETENCIAS PROFESIONALES**

Incorporar la teoría sobre competencias profesionales de Nelson López (2002) se consideró relevante puesto que de éstas se puede definir precisamente esa personalidad profesional indispensable al asumir cargos, actividades, responsabilidades, esa personalidad profesional que le permite a la persona auto identificar sus potencialidades, su capacidad de respuesta, que le permite comprometerse con lo que es capaz de hacer y prepararse para asumir nuevos retos.

#### ***Primer Elemento de Competencia:***

Este le permitirá basar su práctica profesional en el conocimiento de un modelo conceptual: “Modelo conceptual: es un conjunto de conceptos o una imagen mental que permite la representación de la realidad”.

*Tener Claro:*

1. El Objetivo de su profesión
2. La finalidad de su actividad
3. Su Papel
4. Las dificultades existentes que justifican su acción resolutive.
5. La naturaleza de sus intervenciones.
6. Las consecuencias de estas intervenciones.

#### ***Segundo Elemento de Competencia:***

*Conocimientos Científicos :*

1. Bases Teóricas que fundamenten la práctica según su especificidad.
2. Técnicas y Procedimientos para el desempeño de asignaciones específicas.

*Conocimientos Éticos:*

1. Deontología de la carrera que ejerce.

*Conocimientos Legales :*

1. Fundamentos Legales para el Ejercicio de la carrera
2. Marco Legal: Institucionales.

***Tercer Elemento de Competencia:***

Conocimiento del proceso científico aplicado a los procesos de la empresa:

- Investigación
- Análisis
- Interpretación
- Planificación
- Implementación
- Evaluación

“Le confiere rigor científico, orienta la calidad del producto, mediante la especificidad y el mejor uso de los recursos”

***Competencia Fundamental:***

La relación entre miembros del equipo:

- Comunicación Funcional: Facilita, clarifica y proporciona eficiencia a los constantes intercambios.
- Comunicación Pedagógica o de Enseñanza: Proporciona informaciones necesarias sobre aspectos del saber y del hacer cotidianos.
- Comunicación de Relación de Ayuda: Establece lazos significativos entre los miembros de equipos, involucrados en una misma causa: aceptación, respeto cálido y la comprensión empática: le conduce a avanzar en su evolución, sentirse seguro, más aceptado, a ver la vida, el trabajo de forma positiva.

## 1.4 DESARROLLO DE SOFTWARE

A continuación se presentan una serie de conceptos que deben ser manejados por los desarrolladores y que serán mencionados ampliamente a lo largo del presente documento.

### 1.4.1 INGENIERÍA DE SOFTWARE

Según Microsoft (2005), “la Ingeniería del *Software* se podría definir como el establecimiento y aplicación de principios de la Ingeniería para obtener *software*. Teniendo en cuenta factores tan importantes como el coste económico, la fiabilidad del sistema y un funcionamiento eficiente que satisfaga las necesidades del usuario.”

Como concepto de *Software* se entienden las aplicaciones, toda aquella parte de la informática que se desarrolla, y que posee un difícil mantenimiento debido al corto período de tiempo en el que queda obsoleto.

La **Ingeniería del software** se preocupa de la fiabilidad y el rendimiento, tratando de reducir costes y complejidad. Los modelos que utiliza están basados en teorías matemáticas sólidas.

Una parte básica de la ingeniería del *software* es la **Especificación**, que consiste en la descripción del comportamiento externo del sistema, desde el punto de vista del usuario o del entorno. Define aquello que debería hacer el sistema.

En el desarrollo de un proyecto intervienen diferentes etapas que se interrelacionan y complementan para con la finalidad de alcanzar los objetivos iniciales. Estas etapas pueden variar dependiendo del ciclo de vida que se haya decidido para el sistema, pero las principales fases de un proyecto son el análisis de requerimientos, la especificación, el diseño, el desarrollo o implementación, y por último el mantenimiento.

**Análisis de Requerimientos:** Antes de empezar un proyecto, es necesario saber cuáles son los objetivos y funcionalidades que debe cumplir. La fase de análisis consiste en detectar estas funcionalidades y expresarlas en forma de requerimientos.

**Especificación:** Para realizar la especificación, en la mayoría de proyectos se utiliza UML (*Unified Modeling Language*). No obstante, UML no es el único lenguaje utilizado, existen otros como por ejemplo ORM (Object-Relational mapping).

**Diseño:** En la fase de diseño se determina la arquitectura general del sistema y su comportamiento dinámico, adaptando la especificación realizada en la etapa anterior. En esta fase, se decide qué tecnología se utilizará en el sistema aprovechando y adaptando sus ventajas.

**Desarrollo y trabajo en equipo:** La implementación es una fase generalmente muy extensa y requiere de un equipo de desarrolladores para llevarse a cabo. La coordinación entre los miembros de este equipo es muy importante si se desea finalizar el proyecto en el tiempo previsto.

**Mantenimiento:** La fase de mantenimiento de un proyecto contiene las fases de implantación, pruebas, depuración y control de rendimiento, etapas que permiten ajustar el sistema de forma que sus funcionalidades correspondan con los objetivos iniciales.

A continuación, y de acuerdo a la etapa, se describen áreas de conocimiento que debe manejar el desarrollador de *software*:

#### 1.4.1.1 Análisis y diseño

##### UML (Unified Modeling Language)

Según Wikipedia (2007), UML es el lenguaje más extendido para la especificación y diseño de proyectos, hasta el punto de convertirse en un estándar en 1997. Define dos modelos básicos: el modelo conceptual y el modelo de comportamiento.

- *Modelo Conceptual*: Es el utilizado en la especificación del sistema, representa los conceptos más significativos en el dominio del problema utilizando Clases de Objetos, Asociación entre clases de Objetos y Atributos de las clases. Nos describe la parte estática del problema, es una fotografía del mundo real. La parte dinámica del sistema no queda definida en este modelo.
- *Modelo de Comportamiento*: Utilizado en la parte de diseño del sistema, define la parte dinámica, es decir, cual debe ser el comportamiento en cada situación y la forma de proceder. Los diagramas de secuencia y de estados son parte de este modelo.

##### Programación Orientada a Objetos

Según Wikipedia (2007), la **Programación Orientada a Objetos** es un paradigma de programación que define los programas en términos de "clases de objetos", objetos que son entidades que combinan *estado* (es decir, datos), *comportamiento* (esto es, procedimientos o *métodos*) e identidad (propiedad del objeto que lo diferencia del resto). La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar. De esta forma, un objeto contiene toda la información, (los denominados atributos) que permite definirlo e identificarlo frente a otros objetos pertenecientes a otras clases (e incluso entre objetos de una misma clase, al poder tener valores bien diferenciados en sus atributos). A su vez, dispone de mecanismos de interacción (los llamados métodos) que favorecen la comunicación entre objetos



(de una misma clase o de distintas), y en consecuencia, el cambio de estado en los propios objetos. Esta característica lleva a tratarlos como unidades indivisibles, en las que no se separan (ni deben separarse) información (datos) y procesamiento (métodos).

### **Patrones de diseño**

Según Wikipedia (2007), los **Patrones de Diseño** (*Design Patterns*) son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de *software* y otros ámbitos referentes al diseño de interacción o interfaces.

Un **Patrón de Diseño** es una solución a un problema de diseño no trivial que es efectiva (ya se resolvió el problema satisfactoriamente en ocasiones anteriores) y reusable (se puede aplicar a diferentes problemas de diseño en distintas circunstancias).

#### **1.4.1.2 Desarrollo o construcción**

##### **Manejo de Excepciones**

Según Wikipedia (2007), el **Manejo de excepciones** es una estructura de control de los lenguajes de programación diseñada para manejar condiciones anormales que pueden ser tratadas por el mismo programa que se desarrolla.

##### **Convenios de Nombre**

Según Wikipedia (2007), un convenio de nombres es una colección de reglas que especifican una serie de nombres. El propósito es que los usuarios de estos nombres podrán deducir su información de uso, basados en la secuencia de los caracteres y conocimiento de las reglas que se siguieron.

Convenios bien escogidos ayudan a los usuarios a navegar en largas estructuras de código.

## Reutilización de Código

Según Wikipedia (2007), la **reutilización de código** se refiere al comportamiento y a las técnicas que garantizan que una parte o la totalidad de un programa informático existente se pueda emplear en la construcción de otro programa. De esta forma se aprovecha el trabajo anterior, se economiza tiempo, y se reduce la redundancia.

La manera más fácil de reusar código es copiarlo total o parcialmente desde el programa antiguo al programa en desarrollo. Pero es trabajoso mantener múltiples copias del mismo código, por lo que en general se elimina la redundancia dejando el código reusable en un único lugar, y llamándolo desde los diferentes programas. Este proceso se conoce como abstracción. La abstracción puede verse claramente en las bibliotecas de *software*, en las que se agrupan varias operaciones comunes a cierto dominio para facilitar el desarrollo de programas nuevos. Hay bibliotecas para convertir información entre diferentes formatos conocidos, acceder a dispositivos de almacenamiento externos, proporcionar una interfaz con otros programas, manipular información de manera conocida (como números, fechas, o cadenas de texto).

Para que el código existente se pueda reusar, debe definir alguna forma de comunicación o interfase. Esto se puede dar por llamadas a una subrutina, a un objeto, o a una clase.

### 1.4.2 CALIDAD EN EL DESARROLLO DE SOFTWARE:

#### RUP

Según Wikipedia (2007), el **Proceso Racional Unificado (RUP)**, el original inglés *Rational Unified Process*) 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. RUP es en realidad un refinamiento realizado por *Rational Software* del más genérico Proceso Unificado.

El RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante:

- Concepción: se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos
- Elaboración: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos
- Construcción: se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario
- Implementación: se instala el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

## **CMM**

Según Carolina Urrutia (sin fecha), el *Capability Maturity Model* (CMM) para *software* es un conjunto de guías ampliamente aceptadas para desarrollar organizaciones de *software* de alto desempeño. Watts Humphrey y sus colegas de IBM desarrollaron el concepto original para el CMM en los años 80.

Humphrey determinó que la calidad de una aplicación estaba directamente relacionada a la calidad del proceso utilizado para desarrollarla. Para mejorar los procesos de desarrollo de aplicaciones, Humphrey quería implementar el ciclo de mejora continua de W. Edwards Deming (Ej.: *plan-do-check-act.*). Sin embargo, los desarrolladores de aplicaciones han estado instalando métodos y tecnologías de desarrollo mejorados sin obtener mucha mejora.

Humphrey observó que las prácticas de desarrollo de *software* mejoradas no sobrevivían a menos que haya un cambio en el comportamiento de una organización de desarrollo de aplicaciones (DA) para soportarlas. Su visión única era que las organizaciones DA tenían que remover los impedimentos para la mejora continua en un orden específico si querían tener éxito. Consecuentemente, Humphrey diseñó el esquema de maduración de procesos, un camino de evolución para ayudar a las organizaciones DA a incrementar la capacidad de sus procesos DA en cinco etapas.

Durante la evolución a través de los cinco niveles de maduración, las prácticas de desarrollo son transformadas desde un estado indisciplinado, ad hoc, hasta procesos disciplinados capaces de obtener resultados predecibles.

**Tabla 1 Niveles de Madurez**

| Áreas de Procesos o Grupos de Prácticas del CMM |                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nivel de Madurez                                | Mejoras implantadas                                                                                                                                                                                                                                                                                                                                                                                                         |
| 5. Optimizado                                   | <ul style="list-style-type: none"> <li>• Desarrollo de infraestructura de cambios</li> <li>• Evaluación e implantación de mejoras</li> <li>• Eliminación de causas de defectos</li> </ul>                                                                                                                                                                                                                                   |
| 4. Administrado cuantitativamente               | <ul style="list-style-type: none"> <li>• Administración de procesos cuantitativamente</li> <li>• Establecimiento de “baseline” de capacidad</li> </ul>                                                                                                                                                                                                                                                                      |
| 3. Definido                                     | <ul style="list-style-type: none"> <li>• Establecimiento de infraestructura de mejoras</li> <li>• Identificación procesos de <i>software</i> requeridos</li> <li>• Definición de procesos de <i>software</i> comunes</li> <li>• Procesos de implantación y administración</li> <li>• Recolección de datos</li> <li>• Entrenamiento en toda la organización</li> <li>• Coordinación con grupos no-<i>software</i></li> </ul> |
| 2. Repetible                                    | <ul style="list-style-type: none"> <li>• Administración de requerimientos</li> <li>• Planeamiento y control de proyectos</li> <li>• Administración de contratistas</li> <li>• Administración de configuración del producto</li> <li>• Medición de proyectos</li> <li>• Políticas de cumplimiento</li> </ul>                                                                                                                 |
| 1. Inicial                                      | <ul style="list-style-type: none"> <li>• Procesos no requeridos</li> </ul>                                                                                                                                                                                                                                                                                                                                                  |

Fuente: Urrutia, Carolina (sin fecha)

## PRUEBAS UNITARIAS

Según Wikipedia (2007), en programación, una **prueba unitaria** es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. Luego, con las Pruebas de Integración, se podrá asegurar el correcto funcionamiento del sistema o subsistema en cuestión.

La idea es escribir casos de prueba para cada función no trivial o método en el módulo, de forma que cada caso sea independiente del resto.

### Características:

Para que una prueba unitaria sea *buena* se deben cumplir los siguientes requisitos:

- **Automatizable:** no debería requerirse una intervención manual. Esto es especialmente útil para integración continua.
- **Completas:** deben cubrir la mayor cantidad de código.
- **Repetibles o Reutilizables:** no se deben crear pruebas que sólo puedan ser ejecutadas una sola vez. También es útil para *integración continua*.
- **Independientes:** la ejecución de una prueba no debe afectar a la ejecución de otra.
- **Profesionales:** las pruebas deben ser consideradas igual que el código, con la misma profesionalidad, documentación, etc.

Aunque estos requisitos no tienen que ser cumplidos a rajatabla, se recomienda seguirlos o de lo contrario las pruebas pierden parte de su función.

### Ventajas:

El objetivo de las pruebas unitarias es aislar cada parte del programa y mostrar que las partes individuales son correctas. Proporcionan un contrato escrito que el trozo de código debe satisfacer. Estas pruebas aisladas proporcionan cinco ventajas básicas:

1. **Fomentan el cambio:** Las pruebas unitarias facilitan que el programador cambie el código para mejorar su estructura (lo que se ha dado en llamar refactorización), puesto que permiten hacer pruebas sobre los cambios y así asegurarse de que los nuevos cambios no han introducido errores.
2. **Simplifica la integración:** Puesto que permiten llegar a la fase de integración con un grado alto de seguridad de que el código está funcionando correctamente. De esta manera se facilitan las pruebas de integración.

3. **Documenta el código:** Las propias pruebas son documentación del código puesto que ahí se puede ver cómo utilizarlo.
4. **Separación de la interfaz y la implementación.**
5. **Los errores están más acotados y son más fáciles de localizar:** dado que tenemos pruebas unitarias que pueden desenmascararlos.

## **2. TÉRMINOS ASOCIADOS**

Los términos descritos a continuación permiten aclarar conceptos que serán manejados a lo largo del presente documento, dichos términos fueron extraídos del Manual de Gestión de Operaciones de la Organización, versión 13/03/2006.

### **Manual de la Calidad**

Es el documento que especifica el Sistema de Gestión de la Calidad de SSCA de acuerdo con la política y los objetivos de la calidad establecidos y la norma aplicable, el mismo es de uso externo.

### **Manual de Gestión**

Es el documento que contiene los objetivos específicos de cada unidad funcional y el conjunto de procedimientos documentados del sistema de gestión de la calidad destinados a planificar y gerenciar las actividades que afectan a la calidad dentro de cada unidad de la organización, el mismo es de uso interno.

### **Cliente**

Organización que contrata los servicios y que recibe el resultado de los procesos de ejecución de proyectos informáticos. Generalmente, la organización designa a un responsable para representarla.

### **Usuario**

Es la persona o personas que van a tener contacto directo con el sistema o solución a implantar.

### **Proyecto de Software**

Proceso que consiste en un conjunto de actividades coordinadas y controladas con fechas de inicio y de finalización, llevadas a cabo para lograr una solución computacional conforme con los requisitos específicos del cliente, incluyendo las limitaciones de tiempo, costo y recursos.



## **Producto de Software**

Conjunto completo de programas de computadora, procedimientos, documentación, y datos asociados para ser despachados a un usuario.

## **Elemento de Software**

Cualquier parte identificable de un producto de *software* en un paso intermedio o en el paso final del desarrollo

## **Requisito**

Son todas aquellas necesidades y/o expectativas que tiene el cliente / usuario del sistema con relación a la solución que se le está ofreciendo. Representan las capacidades y los objetivos con los cuales la solución debe conformarse y es el hilo de rosca común para todas las actividades del desarrollo del proyecto.

## **Gestión de requisitos**

Es el conjunto de actividades que incluyen la identificación, control, comunicación de la información a través del Equipo del Proyecto, y el seguimiento a los requisitos y a los cambios que se presenten, en cualquier momento del desarrollo del proyecto.

## **Especificación de requisitos**

Es la conceptualización documentada de los requisitos. Puede ser un documento escrito, un modelo gráfico, un modelo matemático formal, una colección de escenarios de uso ó casos de uso, un prototipo o una combinación de lo anterior.

## **Elementos de la Configuración (EC)**

Son cada uno de los ítems o artefactos que componen el producto del desarrollo del proyecto. Son considerados Elementos de la Configuración (EC's) todas las especificaciones, los diseños, las comunicaciones, código fuente, código

compilado, bases de datos, herramientas, y en general todo documento o artefacto producido como parte del proceso de desarrollo del producto.

### **Repositorio**

Es el lugar donde se almacenan y residen todos los ECS de un proyecto.

### **Línea de Base ó Base Line**

Es el conjunto de ECS's que impactan críticamente al proyecto y a los que se le hace un seguimiento y control formal de su estado. Todos los elementos y productos que se entregan al cliente forman parte de la línea base. Los ECS's que conforman la línea base necesitan ser aprobados por las funciones competentes, y sólo pueden ser modificados a través de procedimientos formales de control de cambios. Es una fotografía del producto de *software* en un momento dado, y un subconjunto del repositorio del proyecto.

### **Control de Versiones**

Es un mecanismo ó método para hacer seguimiento de los cambios producidos en los ECS's de un proyecto de *software*. El control de versiones permite llevar cuenta de los cambios hechos a los ECS a medida que se desarrolla el producto, y volver a versiones anteriores de los ECS cuando se hayan hecho cambios incorrectos o se requiera dar mantenimiento a entregas previas del producto.

### **Auditoría**

Proceso sistemático, independiente y documentado para obtener evidencias de verificaciones, y evaluarlas de manera objetiva, con el fin de determinar el grado en que se cumplen los criterios acordados.

### **Verificación**

Comprobación mediante la aportación de evidencia objetiva de que se han cumplido los requisitos especificados. El término “verificado” se utiliza para

designar el estado correspondiente. Esta confirmación puede comprender acciones tales como:

La elaboración de cálculos alternativos,

La comparación de una especificación de un diseño nuevo con una especificación de un diseño similar probado,

La realización de pruebas o demostraciones, y

La revisión de los documentos antes de su liberación.

### **Desviación**

Es la acción o resultado de apartarse de los requisitos originalmente especificados para un proceso o producto.

### **Conformidad / No conformidad**

Cumplimiento / Incumplimiento de un requisito.

### **Acción correctiva**

Es la acción tomada para eliminar la causa de una no conformidad detectada u otra situación indeseable.

### **Acción preventiva**

Es la acción tomada para eliminar la causa de una no conformidad potencial u otra situación indeseable antes de que ésta ocurra.

### **Iteración**

Iteración es la repetición de una serie de instrucciones en un programa de computadora. Puede usarse tanto como un término genérico (como sinónimo de repetición) así como para describir una forma específica de repetición con un estado mutable.

## CAPÍTULO III: MARCO METODOLÓGICO

### 1. TIPO Y DISEÑO DE LA INVESTIGACIÓN

El método que se va a seguir en la investigación determina la necesidad de definir el tipo de estudio, que hace referencia a la profundidad con la que se espera abordar el problema de conocimiento.

La presente investigación, se enmarca dentro de la modalidad de proyecto factible, ya que consiste en la elaboración de una propuesta de un modelo viable u operativo, o una posible solución a un problema de tipo práctico; y está apoyado en una investigación de campo no experimental de nivel descriptivo.

En lo concerniente a la formulación de proyectos factibles, éstos enfatizan en la elaboración de una propuesta que se sustenta en un modelo operativo y funcional para resolver problemas o situaciones planteadas, o satisfacer una o varias necesidades de una institución, empresa o grupo social en particular. Ello se sustenta en el concepto de **Proyecto Factible**, según el Manual de Trabajos de Grado de Especialización, Maestrías y Tesis Doctorales de la UPEL (1998):

Consiste en la investigación, elaboración y desarrollo de una propuesta de un modelo operativo viable para solucionar problemas, requerimientos o necesidades de organizaciones o grupos sociales, puede referirse a la formulación de políticas, programas, tecnologías, métodos o procesos. El proyecto debe tener apoyo en una investigación de tipo documental, de campo o ambas modalidades (p. 7)

Este estudio es de naturaleza descriptiva, ya que el mismo busca especificar las propiedades importantes de personas y equipos que han sido sometidos a un análisis profundo.

Al respecto, Hernández Sampieri (1999), cuando desarrolla el propósito de los estudios descriptivos, señala que “Muy frecuentemente el propósito del investigador es describir situaciones y eventos. Esto es, decir como es y se manifiesta determinado fenómeno. *Los estudios descriptivos buscan especificar las propiedades importantes, de personas, grupos, comunidades o cualquier otro fenómeno que sea sometido a análisis* (Dankhe, 1986). Miden o evalúan diversos aspectos, dimensiones o componentes del fenómeno o fenómenos a investigar. Desde el punto de vista científico describir es medir...”.

También Hernández Sampieri (2.000), plantea que “...los estudios e investigaciones de carácter descriptivo pueden ofrecer la posibilidad de predicciones aunque sean rudimentarias”.

En cuanto a la estrategia del estudio en cuestión, es una investigación de campo, pues la misma abarca, tal como lo define el manual de la U.P.E.L. (1990):

“... el análisis sistemático de problemas con el propósito de describirlos, explicar sus causas y efectos, entender su naturaleza y factores constituyentes o predecir su concurrencia. Los datos de interés son recogidos en forma directa de la realidad por el investigador, en este sentido se trata de investigaciones a partir de datos primarios u originales. Según los objetivos propuestos, la investigación puede ser de carácter exploratoria, descriptiva, explicativa o evaluativa.”

Por otra parte, el diseño o estrategia para responder la interrogante objeto de investigación, es de tipo no experimental debido a que se tiene como finalidad analizar como se manifiesta una o varias variables tal y como se presentan en su contexto, ya que en ésta no se construye ni se provoca ninguna situación nueva, simplemente se observa un fenómeno ya existente sin intervención en el comportamiento de la o las variables.

En cuanto a los diseños no experimentales, el Manual para la elaboración del Trabajo de Investigación de la ESGA (2002) establece:

Se aplica en las investigaciones de campo en las que no hay manipulación de variables, porque la acción de las variables ya se dio en la realidad, el investigador no intervino en ello. Se observan las variables y sus relaciones en un contexto natural, el investigador toma los datos directamente de la realidad. (p. s/n)

Así mismo, esta investigación también es de tipo documental, puesto que para cumplir con los objetivos planteados en el desarrollo de la investigación se basará en la revisión de fuentes documentales de todas las referencias teóricas necesarias y que estén al alcance, según lo afirma Rísquez, Fuenmayor y Pereira (1999), esto se hace con la finalidad de realizar la investigación obteniendo los datos necesarios para responder la interrogante planteada, conducente a formular el Diseño de Instrucción para el adiestramiento del desarrollador de *software* en SSCA, acertadamente para el abordaje de sus necesidades reales de conocimientos y destrezas.

## **2. UNIDAD DE ANÁLISIS**

En este caso específico se tomaron en cuenta tres unidades de análisis. En primer lugar se tomaron documentos contentivos de normas, procedimientos y estándares concernientes a la Operación dentro de la Organización. En segundo lugar, los documentos de los proyectos almacenados en repositorio de información de la empresa y aplicaciones asociadas que generan estadísticas. Y en tercer lugar personas calificadas para validar la gestión de los desarrolladores durante la ejecución de dichos proyectos.

## **3. POBLACIÓN Y MUESTRA**

La población del primer caso, esta constituida por documentos de normas, procedimientos e instrucciones de trabajo ubicados en la Intranet de la empresa; dicha muestra estuvo conformada por los Manuales de Gestión de cada una de las

áreas de conocimiento definidas en la Metodología de SSCA (Desarrollo, Gerencia de Proyecto, Calidad, Arquitectura, Configuración, Validación y Verificación, Manejo de Requerimientos, Diseño Gráfico, entre otros).

En el segundo caso, en los dos últimos años se ejecutaron diez (10) proyectos de los cuales se seleccionaron dos. Dichos proyectos fueron evaluados revisando los documentos almacenados en el repositorio de información tales como Planes de Proyectos, Minutas, Reportes de Estatus, Reportes de Tiempo, Cronogramas, Informes de *Peer Reviews*, Documentos de Diseño y Diseño Detallado, entre otros, así como también las aplicaciones generadoras de estadísticas CVS, Manejador de tareas (Jira), Sistema de control de tiempo, etc.

Con respecto a la tercera unidad de análisis, dentro de SSCA existen aproximadamente veinte (20) Líderes, de los cuales se tomó una muestra de cinco (5) que participaron como supervisores de los desarrolladores de *software* que ejecutaron los proyectos seleccionados en la muestra anterior.

#### **4. RECOLECCIÓN DE LA DATOS**

La recolección de la información se llevo a cabo en dos procesos. El primero, un arqueo bibliográfico para las dos primeras unidades de análisis; y el segundo, entrevistas a cinco (5) líderes encargados de supervisar a los desarrolladores.

Ambos métodos son explicados a continuación, según Bandín (2004), quien plantea que:

El arqueo bibliográfico o revisión de la literatura consiste en detectar, obtener y consultar la biblioteca y otros materiales que pueden ser útiles para los propósitos del estudio, de donde se debe extraer y recopilar la información revelante y necesaria que atañe a nuestro problema de investigación. Esta revisión debe ser selectiva. (p.23).

Las fuentes de información disponibles para llevar a cabo este proceso se clasificaron en fuentes primarias o directas y fuentes secundarias, a tal respecto el mismo autor refiere que:

Las primarias proporcionan datos de primera mano y un ejemplo de éstas son los libros, antologías, artículos de publicaciones periódicas, monografías, tesis y disertaciones, documentos oficiales, reportes de asociaciones, trabajos presentados en conferencia o seminarios, artículos periodísticos, testimonios de expertos, películas, documentales y videocintas. (p. 23).

Siguiendo el mismo orden de ideas, Bandín, al referirse a las secundarias, las define como “compilaciones, resúmenes y listados de referencias publicados en un área en particular (son listados de fuentes primarias). Es decir, reprocesan información de primera mano”. (p. 23).

El segundo proceso de recolección de la información se hizo a través de un estudio de campo, que según Carlos Sabino (2000) “se basa en informaciones o datos primarios, obtenidos directamente de la realidad”, por lo que se hizo directamente entrevistas en persona a los líderes encargados de supervisar a los desarrolladores.

El instrumento de recolección de información de la investigación se diseñó según el objeto de estudio. La modalidad entrevista semi-estructurada fue el instrumento de recolección de información de elección, puesto que se formulan determinadas preguntas abiertas y cerradas, relativas a las variables en estudio.

La entrevista, según Sabino (2000):

“desde el punto de vista del método, es una forma específica de interacción social que tiene por objeto recolectar datos para una investigación. El investigador formula preguntas a las personas capaces de aportarle datos de interés, estableciendo un diálogo peculiar, asimétrico, donde una de las partes busca recoger informaciones y la otra es la fuente de esas informaciones”.



Y de acuerdo con los objetivos del estudio resulta útil porque:

“La ventaja esencial de la entrevista reside en que son los mismos actores sociales quienes proporcionan los datos relativos a sus conductas, opiniones, deseos, actitudes, y expectativas. Nadie mejor que la misma persona involucrada en el proceso para hablarnos acerca de aquello que piensa y necesita, de lo que ha experimentado o piensa hacer”. (p. 155)

Con este método se pretende estimular al entrevistado a hablar sin limitaciones sobre lo consultado, de modo que se pueda obtener la información requerida acerca del proceso, aportes y barreras.

## **5. PROCESAMIENTO DE LOS DATOS**

Los datos obtenidos, fueron organizados y tabulados para su análisis e interpretación, utilizando para ello las técnicas del análisis cualitativo (análisis de contenido) y análisis cuantitativo, que permitió obtener los datos necesarios para formular adecuadamente un Diseño de Instrucción para el adiestramiento del Desarrollador en SSCA

Para la primera y segunda unidad de análisis se realizó un análisis de contenido, que según Arias, F. (1999) “es la técnica dirigida a la cuantificación y clasificación de las ideas de un texto, mediante categorías preestablecidas” (p. 77). Por ello, el análisis cualitativo se refiere al tratamiento que se le va a dar a la información recabada de las fuentes, tanto primarias como secundarias, y al análisis de las respuestas emitidas por las personas durante la aplicación de la entrevista lo cual permitirá revisarla, relacionarla e interpretarla adecuadamente para llegar a conclusiones pertinentes respecto al problema planteado.

A su vez, en el análisis de contenido, el investigador utilizó el método de análisis y síntesis, el cual según explica Méndez, C. (1995), el análisis “es el proceso de conocimiento que se inicia por la identificación de cada una de las partes que

caracterizan una realidad” (p. 131) y la síntesis como “el proceso que procede de lo simple a lo complejo, de la causa a los efectos, de la parte al todo, de los principios a las consecuencias” (p. 132). En este sentido, estas dos técnicas, por ser procesos que se complementan, permitieron al investigador percibir la realidad en estudio a través de la clasificación y organización de la información relacionada con los conocimientos, competencias y destrezas específicas mínimas requeridas para el cargo de desarrollador de *software* en SSCA y los que posee actualmente.

El análisis cuantitativo se realizó mediante las técnicas de la estadística descriptiva, utilizando para ello la tabulación de datos así como su representación en gráficos, acorde con lo planteado por Sabino, C. (1996), al referirse al análisis cuantitativo:

Este tipo de operación se efectúa naturalmente con toda la información numérica resultante de la investigación. Ésta, luego del procedimiento sufrido, se nos presentará como un conjunto de cuadros, tablas y medidas, a las cuales habrá que pasar en limpio, calculando sus porcentajes y otorgándoles formas definitivas (p. 172)

## **6. METODOLOGÍA**

El proyecto está dividido en 3 fases:

**Fase 1: Diagnóstico**

**Fase 2: Análisis**

**Fase 3: Diseño**

En la fase de diagnóstico se identifica cual es el estado actual del departamento, cuales son las áreas que necesitan ser reforzadas, y cual es el perfil ideal que debe poseer el desarrollador. Seguidamente, una fase de análisis donde se determina la brecha existente entre el deber ser y la realidad existente en la

organización. Y por último se realiza la propuesta del diseño de instrucción con el fin de cerrar las brechas presentes en los desarrolladores.

## **7. RESULTADOS ESPERADOS E IMPLICACIONES**

Al concluir el proyecto se contará con un diseño de Instrucción para el adiestramiento de Desarrolladores de *Software* completo, que pueda ser inmediatamente puesto al servicio del personal al que está dirigido para su puesta en práctica, permitiendo así el adiestramiento *insourcing* y que, por otra parte, sea de fácil manejo para que constantemente se ejecute en el proceso de inducción al personal que se va incorporando a la empresa.

A mediano plazo se espera que el personal mejore sus prácticas de desarrollo y, adicionalmente, reduzca los tiempos de entrega. Esto será traducido en una mejora en la calidad del producto entregado a los clientes que implicará un mayor prestigio para la empresa.

## **8. CONSIDERACIONES ÉTICAS**

Como pautas éticas se aplicaron las dictadas por el *Project Management Institute (PMI)*. De acuerdo al código de ética de los miembros del PMI (*Project Management Institute, 2004*), los profesionales dedicados a la Gerencia de Proyectos deben comprometerse a:

- Mantener altos estándares de una conducta íntegra y profesional.
- Aceptar las responsabilidades de sus acciones.
- Buscar continuamente mejorar sus capacidades profesionales.
- Practicar la justicia y honestidad.
- Alentar a otros profesionales a actuar de una manera ética y profesional.

Algunas obligaciones profesionales indicadas por el PMI que son particularmente aplicables a este trabajo son:

### **Comportamiento Profesional**

- Revelar completa y oportunamente cualquier conflicto profesional.
- Respetar y proteger apropiadamente los derechos intelectuales de otros; revelar y reconocer apropiadamente las contribuciones profesionales, intelectuales y de investigación de otros.

Procurar mejorar sus capacidades, habilidades y conocimientos profesionales, y dar a conocer sus calificaciones profesionales de forma sincera y certera.

## **CAPÍTULO IV: MARCO ORGANIZACIONAL**

En este Capítulo se hace referencia al Marco Organizacional de la empresa objeto de estudio y la fuente utilizada es Documental Institucional, por lo que es copia fiel de contenido del Manual de Gestión de Operaciones de la Organización, versión 13/03/2006.

### **1. ANTECEDENTES**

SSCA es un grupo comercial conformado por tres empresas localizadas en Estados Unidos, Perú y Venezuela. Para efectos del Sistema de Gestión de Calidad éstas empresas adoptan el logo de la organización consolidadora, es decir, SSCA, y todos los procesos que dichas empresas desarrollan son iguales, con la excepción de que para efectos contables, algunos de sus registros se desarrollan en papelería de cada una de las razones sociales enumeradas.

SSCA es una empresa venezolana de consultoría para el desarrollo de *software*. Tiene más de 16 años de existencia, su sede principal está ubicada en Caracas, Venezuela y el centro global de desarrollo de *software* está localizado en la ciudad de Mérida. Cuenta, además, con oficinas en Chicago, Estados Unidos de América y Lima, Perú. A través de estas oficinas y con un gran equipo de profesionales calificados, SSCA ofrece sus servicios al mundo, habiendo ejecutado exitosamente proyectos en Venezuela, Colombia, Perú, Trinidad, Brasil, Estados Unidos de América, Francia, Finlandia, Holanda y Australia.

## 2. OBJETIVO

Servimos a clientes en el mercado nacional y *Near Shore* desde nuestro Centro Global de Desarrollo, con énfasis en los atributos de soporte metodológico en ingeniería de *software*, dominio de tecnologías de punta, calidad certificada, procesos de producción costo-efectivos y capacidad de comunicación eficiente, apoyándonos en procesos claves de mercadeo, servicios, operaciones, gestión de la calidad y gestión de capital humano, para ofrecer servicios de desarrollo de aplicaciones y soluciones tecnológicas innovadoras, a la medida de las necesidades de los clientes, con la mejor relación precio-valor del mercado.

## 3. PERFIL

- Proveedora de Servicios de Tecnología de la Información al mercado global.
- Dedicada al desarrollo de aplicaciones, a la consultoría en tecnologías de la información, y al *outsourcing* de procesos de negocios.
- Basada en el *Global Delivery Model* para la entrega de servicios bajo el concepto “*Near Shore Software Factory*”, desde Centro Global de Desarrollo ubicado en el Parque Tecnológico de Mérida.
- Provee servicios a las principales corporaciones de Venezuela y el mundo.

## 4. VISIÓN

En armonía con el ecosistema, con crecimiento sostenido, gente entusiasta y a la vanguardia tecnológica, seremos una organización venezolana reconocida por su capacidad de innovación y por la calidad en la prestación de servicios de tecnologías de información al mercado global.

## 5. MISIÓN

Nuestro equipo de profesionales de alto desempeño establece relaciones a largo plazo con los clientes en el mercado global, para proveerles servicios de TI que

agregan valor a sus negocios, con tecnologías de vanguardia y estándares de calidad internacional

## **6. POLÍTICA DE LA CALIDAD**

Proveer servicios de TI que agreguen valor y satisfagan a nuestros clientes en el mercado global, mediante el uso de tecnologías de vanguardia, estándares de calidad internacional y con el compromiso en el mejoramiento continuo de los procesos.

## **7. OBJETIVOS GENERALES DE LA CALIDAD**

- Mantener una cartera amplia de clientes satisfechos
- Incrementar las ventas y utilidades de la organización
- Mantener un capital humano competente, entusiasta, experto y comprometido
- Mejorar la eficiencia de los procesos

## **8. COMPROMISO**

La ejecución de proyectos es realizada siguiendo la más moderna tecnología de desarrollo de *software*, complementada con plataformas tecnológicas y herramientas de desarrollo de punta. En permanente vigilia tecnológica, los profesionales de SSCA garantizan a sus clientes la calidad y la vigencia en el tiempo de las soluciones implementadas, dado el alto nivel de escalabilidad de nuestros sistemas bajo la concepción de sistemas abiertos.

Es un compromiso de SSCA prestar productos y servicios de primera calidad que cumplan con los requerimientos de sus clientes. Para ello se cuenta con un Sistema de Gestión de la Calidad documentado, el cual se encuentra descrito en el Manual de la Calidad.

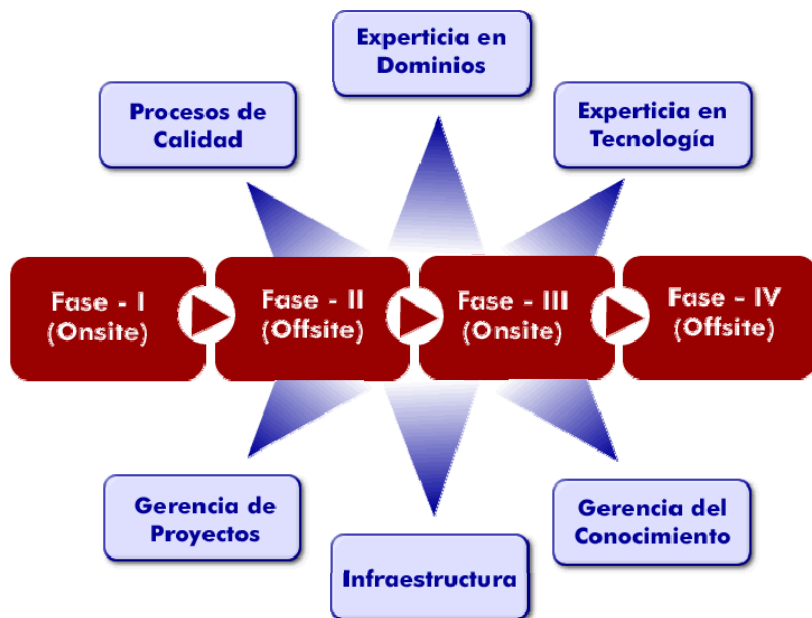
## 9. ESTRATEGIA COMPETITIVA

Servimos a clientes en el mercado nacional y *near shore* desde nuestro Centro Global de Desarrollo, con énfasis en los atributos de soporte metodológico en ingeniería de *software*, dominio de tecnologías de punta, calidad certificada, procesos de producción costo–efectivos y capacidad de comunicación eficiente; apoyándonos en procesos claves de mercadeo, servicios, operaciones, aseguramiento de la calidad y gestión de capital humano, para ofrecer servicios de desarrollo de aplicaciones y soluciones tecnológicas innovadoras, a la medida de las necesidades de los clientes, con la mejor relación precio-valor del mercado.

## 10. MODELO DE GESTIÓN

El modelo de gestión de SSCA puede visualizarse en la siguiente figura:

Figura 3 Flujograma de Gestión



Fase-1 (En las oficinas del cliente):

- Análisis y planificación
- Diseño de alto nivel.
- Diseño de las interfaces de usuarios.

Fase-2 (En el Centro Global de Desarrollo):

- Gerencia del proyecto.
- Diseño de detalles.
- Codificación.
- Pruebas.
- Documentación.

Fase-3 (En las oficinas del cliente):

- Pruebas en sitio.
- Implementación.
- Entrenamiento.
- Soporte post-implementación.

Fase-4 (En el Centro Global de Desarrollo):

- Soporte.
- Mantenimiento.

Fuente: Manual de Gestión de SSCA (03/2006)



Nuestro modelo de entrega global se basa en un equipo con 20 a 30% de los miembros ubicados en el sitio del cliente y 70 a 80% en el Centro Global de Desarrollo.

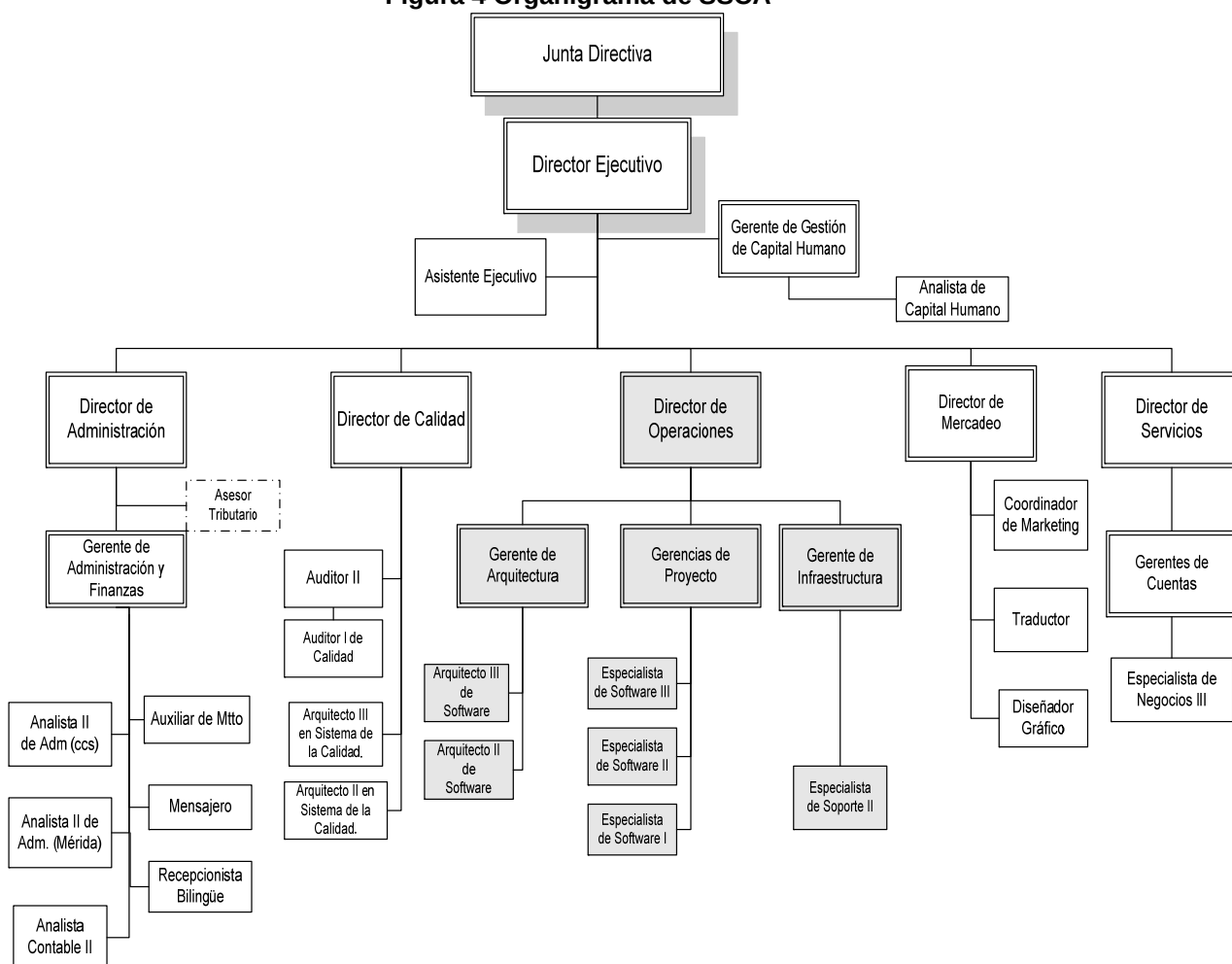
#### **11. ESTRUCTURA DE TRABAJO**

- Más de 80 profesionales especializados en tecnologías de vanguardia, expertos en el diseño y desarrollo de soluciones integrales de TI.
- Estaciones de trabajo con capacidad de 8.000 horas al mes.
- Infraestructura física de aproximadamente 500 metros cuadrados.
- Mecanismos automatizados de gestión para el trabajo colaborativo: email, chat, video-conferencia, conferencias telefónicas y redes privadas virtuales, con estándares internacionales de seguridad.
- Capacidad multilingüe para comunicarse con clientes de habla hispana e inglesa, principalmente.
- Conjunto de tecnologías que demanda el mercado, habilitadas permanentemente para investigación y desarrollo.

#### **12. ESTRUCTURA ORGANIZATIVA**

A continuación se presenta el organigrama de SSCA:

**Figura 4 Organigrama de SSCA**



Fuente: Manual de Gestión (03/2006)

La presente investigación está enmarcada dentro de Operaciones, en Gerencia de Proyectos y específicamente para los **Especialistas de Software I** que son quienes se desempeñan como desarrolladores de *software* durante la construcción de una aplicación. Es a este rol, al que se le realizará el estudio para determinar las necesidades de adiestramiento.

## **CAPÍTULO V: DESARROLLO DEL PROYECTO**

### **1. FASE 1: DIAGNÓSTICO**

#### **1.1. Conocimientos mínimos del desarrollador:**

Se realizó una revisión bibliográfica de los procedimientos y manuales de la organización localizados en la intranet, a través de la cual se tomó en cuenta los siguientes:

- PGC-OP-01 Planificación del Proyecto (05/06/2006)
- PGC-OP-02 Seguimiento y Control del Proyecto (11/06/2006)
- PGC-OP-03 Gestión de la Configuración (21/06/2006)
- PGC-OP-04 Gestión de Requisitos (14/04/2006)
- PGC-OP-05 Control de Calidad de Proyectos (19/03/2006)
- PGC-OP-06 Diseño y Desarrollo Técnico (25/06/2006)
- PGC-OP-08 Pruebas Funcionales (12/04/2006)
- PGC-OP-09 Implantación (11/04/2006)
- Manual de Gestión Dirección de Operaciones (13/03/2006)
- Manual de Gestión Dirección de Calidad (07/06/2006)
- Manual de Gestión Dirección de Arquitectura (03/06/2006)

En virtud de su pertinencia en cuanto a la descripción que plantea sobre el objeto de estudio, refiere que en cuanto al perfil que debe poseer el Desarrollador de *Software* se puede decir que :

Debe ser un profesional universitario de las carreras licenciado en informática, ingeniero de sistemas o ingeniero de computación. Con Experiencia en .Net, lenguaje orientado a objetos, control de versiones, capacidad de innovación, alto desempeño, vocación de servicio al cliente, generar valor y mejoramiento continuo

a los procesos. Experiencia Superior a un (1) año. Manejar metodologías de desarrollo RUP o MSF.

Así mismo, se considera el perfil del desarrollador en la empresa en estudio, donde se le asignan las responsabilidades a referir seguidamente:

- Desarrollo de los requerimientos.
- Estimación tamaño de los elementos y esfuerzo.
- Diseño detallado de la solución.
- Construcción de programas.
- Corrección de fallas y defectos
- Planificación de pruebas al sistema.
- Implantación de la solución.
- Reporte e informe de avance de actividades.

En ese mismo orden de ideas, durante la revisión documental Institucional se examinaron manuales de Calidad, Operaciones, Recursos Humanos, y Mercadeo, pudiendo determinarse los requerimientos de conocimientos mínimos que cada parte considera que un desarrollador debe manejar. Cada requerimiento general fue traducido a uno o más requerimientos suficientemente específicos como para poder determinar las actividades que pudieran atenderlos. Los mismos se presentan a continuación:

### **1.1.1 REQUERIMIENTOS DEL DESARROLLADOR**

#### **Procesos Organizacionales:**

REQ-1-CMM: Comprender la importancia de la certificación de calidad en el modelo de negocios de SSCA y entender el compromiso que eso implica para cada uno de los miembros de la empresa.

REQ-2-CMM: Conocer en forma general el conjunto de roles, procedimientos, prácticas, y documentos que constituyen que conforman el modelo de calidad de SSCA.

REQ-3-CMM: Conocer la conformación de los PRT por proyecto, la motivación del mismo, y las responsabilidades pertinentes a cada uno de los roles.

#### **Gestión de requisitos:**

REQ-4-CMM: Conocimiento de los documentos relacionados con la gestión de requisitos y comprensión de los mismos.

REQ-5-CMM: Comprensión de requisitos funcionales y no funcionales y de cómo los mismos deben transformarse en diseño detallado y código ejecutable.

REQ-6-CMM: Comprensión de casos de uso y de cómo los mismos deben transformarse en diseño detallado y código ejecutable.

REQ-7-CMM: Comprensión de lo que son los criterios de aceptación para requisitos y casos de uso y de cómo los mismos se traducen en pruebas funcionales.

REQ-8-CMM: Comprensión de los documentos de arquitectura de proyectos, en particular de los diagramas UML de implantación, componentes, clases, e interacciones que los componen.

#### **Planificación:**

REQ-9-CMM: Comprensión del documento que describe el plan de proyecto, y de los compromisos compartidos que los hitos allí especificados implican para todos los participantes.

REQ-10-CMM: Rol en las tareas de planificación semanal y diaria, estimación de tareas, distinción entre trabajo y retrabajo.

#### **Seguimiento y control:**

REQ-11-CMM: Uso de las herramientas de control de tareas (Jira).

REQ-12-CMM: Registro del tiempo trabajado usando las herramientas correspondientes (SCP o Jira).

REQ-13-CMM: Conceptos de gestión de riesgos, e identificación y procesamiento de los riesgos detectables durante las actividades de desarrollo de *software*.

#### **Gestión de configuración:**

REQ-14-CMM: Concepto de gestión de configuraciones (gestión del cambio) y comprensión del rol de los distintos integrantes del equipo de proyecto en dicha gestión.

REQ-15-CMM: Control de versiones usando las herramientas adoptadas por SSCA (CVS y Eclipse), incluyendo el desarrollo de versiones (*releases*) paralelas (*branching*).

REQ-16-CMM: Procedimientos de integración continua, integración diaria, y *releases*.

### **Gestión de calidad:**

REQ-17-CMM: Conceptos de conformidad con los procesos y procedimientos y características y rol de las auditorías de calidad.

REQ-18-CMM: Conocimiento sobre el uso de los canales de comunicación establecidos para los proyectos: listas por proyecto, listas por rol PRT (arquitectura, configuraciones, infraestructura, etc.).

REQ-19-CMM: Conocimiento sobre el uso de la documentación necesaria para el seguimiento del proyecto como las minutas y los reportes semanales.

REQ-20-CMM: Papel de pos *peer-review*, procedimientos, estándares de codificación y documentación, y listas de verificación de SSCA.

### **Desarrollo:**

REQ-21-OPR: Diseño detallado usando UML y las herramientas de diseño seleccionadas por SSCA.

REQ-22-OPR: Conocimiento del lenguaje de programación .Net y de sus estilos y librerías fundamentales.

REQ-23-OPR: Conocimiento de las librerías de Manejo de Excepciones y *Logging*.

REQ-24-OPR: Construcción de pruebas unitarias de regresión usando *NUnit*.

REQ-25-OPR: Automatización de la construcción del *software*, su comprobación unitaria, empaquetamiento, e implantación.

### **Arquitectura y diseño:**

REQ-26-ARQ: Metodologías de diseño y construcción de *software* con énfasis en la metodología de SSCA.

REQ-27-ARQ: Conceptos y prácticas de diseño y programación estructurada y orientada a objetos.

REQ-28-ARQ: Identificación, clasificación, y reparación de defectos. Técnicas de *debugging*.

REQ-29-ARQ: Componentización, separación por capas, y *model-view-controller* (MVC). Conceptos, importancia, y maneras de obtenerla en el código.

REQ-30-ARQ: Programación basada en contratos, invariantes, precondiciones, y poscondiciones, manejo de errores y uso de excepciones.

REQ-31-ARQ: Patrones de diseño (*Software Design Patterns*) fundamentales.

REQ-32-ARQ: Refactorización: como transformar programas para adaptarlos a nuevos requerimientos sin alterar su funcionalidad y afectar su calidad.

**Destrezas personales:**

REQ-33-RH: Comunicación.

REQ-34-RH: Manejo del cambio.

REQ-35-RH: Trabajo en equipo.

REQ-36-RH: Liderazgo.

REQ-37-RH: Negociación.

REQ-38-RH: Presentaciones orales.

REQ-39-RH: Preparación de informes técnicos.

REQ-40-RH: Técnicas de estudio, lectura veloz, y manejo del tiempo.

## **1.2. Conocimientos de los Desarrolladores - Situación Actual**

Para determinar el estado actual del equipo de los desarrolladores primeramente se realizó una revisión bibliográfica de los proyectos, para dicha revisión se utilizaron una serie de herramientas corporativas.

Las herramientas corporativas son todas aquellas herramientas particulares de la organización que el personal debe conocer y aprovechar para mejorar su labor dentro de la empresa. Algunas de estas herramientas son comunes entre una organización y otra debido a su funcionalidad, sin embargo los métodos y procedimientos de uso son generalmente diferentes entre cada organización. Por lo cual, es importante que el personal que requiera su utilización sepa a ciencia cierta cuáles son procedimientos de uso así como los objetivos y ventajas de utilizarlas.

En cuanto a este importante aspecto, dentro de SSCA y a propósito del presente estudio se revisaron todos los Documentos disponibles, de los cuales se obtuvo información valiosa por sus aportes, se trata de:

#### **1.2.1 Repositorio del Proyecto (BÚHO):**

Es el repositorio de información de los proyectos de la organización. Está basado en tecnología Web por lo cual es accesible desde Internet. Es un administrador de contenidos cuyas principales funciones son:

- Centralizar la información
- Compartir la información dentro de todos los departamentos
- Mantener versiones de los documentos
- Estructurar los proyectos

Dentro de los documentos que se pueden encontrar se pueden mencionar:

- Planes de proyectos
- Documentos de especificación
- Minutas de reuniones
- Documentos de arquitectura
- Definiciones y procedimientos de trabajo
- Formatos y estándares de la organización.
- Manuales y documentos técnicos
- Cronogramas de trabajo
- Casos de uso
- Documentos de diseño

#### **1.2.2 Administrador de Tareas (JIRA):**

Es el Administrador de Tareas de la organización. Es un sistema de *workflow* que permite registrar un incidente, evento, inconformidad o información referente al



proyecto, el cual puede ser asignado a un grupo o área que será responsable de atenderlo o revisarlo. Durante el proceso de atención los relacionados pueden intercambiar información escribiendo comentarios o anexando archivos. Al tratarse de un *workflow*, los casos van pasando varios estatus hasta llegar al estatus resuelto. Es un sistema web por lo que está disponible desde Internet.

Adicionalmente, permite hacer seguimiento estadístico de la resolución y atención de los casos y sirve además como base del conocimiento para búsqueda de soluciones.

En Jira son reportados todos los casos o inconsistencias encontradas por el equipo de pruebas, el cliente, el líder del proyecto o el arquitecto del proyecto, básicamente, aunque cualquier miembro del equipo puede crear casos en Jira. Por lo tanto, los casos reportados en esta herramienta son exclusivos de programación y errores funcionales.

De la revisión realizada se pudieron obtener los siguientes resultados:

**Tabla 2 Errores registrados en Jira**

|                       | Proyecto 1 |    |     |     |     |       |        | Proyecto 2 |    |    |    |     |     |       |        |
|-----------------------|------------|----|-----|-----|-----|-------|--------|------------|----|----|----|-----|-----|-------|--------|
| Iteración             | 1          | 2  | 3   | 4   | 5   | Total | %      | 1          | 2  | 3  | 4  | 5   | 6   | Total | %      |
| Errores identificados | 120        | 80 | 200 | 120 | 350 | 870   | 72,32% | 50         | 80 | 45 | 80 | 120 | 440 | 815   | 65,62% |
| Diseño                | 1          | 2  | 10  | 5   | 50  | 68    | 5,65%  | 5          | 2  | 6  | 40 | 20  | 150 | 223   | 17,95% |
| Malas Practicas       | 30         | 20 | 35  | 21  | 127 | 233   | 19,37% | 10         | 15 | 4  | 6  | 37  | 80  | 152   | 12,24% |
| Información           | 4          | 5  | 10  | 0   | 3   | 22    | 1,83%  | 0          | 0  | 0  | 5  | 2   | 15  | 22    | 1,77%  |
| Rendimiento           | 0          | 0  | 0   | 0   | 10  | 10    | 0,83%  | 0          | 0  | 0  | 0  | 0   | 30  | 30    | 2,42%  |
| Totales               |            |    |     |     |     | 1203  |        |            |    |    |    |     |     | 1242  |        |

Fuente: Repositorio de Proyectos Ejecutados (2006)

Los cuales ponen en evidencia que el principal problema encontrado son los errores de programación. Con respecto a problemas de rendimiento, malas prácticas y problemas de diseño, tiene una tendencia creciente que es pronunciada en la última iteración, esto es resultado, de que se hace una revisión

más exhaustiva en estos tres aspectos en las fases finales, cuando empiezan las pruebas de aceptación.

### 1.2.3 SCT (Sistema de Control de Tiempo):

Es un sistema que permite llevar el control del tiempo del personal dentro de la organización. Permite registrar el tiempo dedicado por día a cada proyecto y específicamente en cada tarea que se haya especificado en el cronograma. Dichas tareas están asociadas a una fase del proyecto, lo cual permite llevar estadísticas para mejorar los procesos de estimación. Adicionalmente es utilizado para hacer la facturación a los clientes de las horas ejecutadas de los proyectos.

Adicionalmente, se revisaron los cronogramas del proyecto, obteniendo las fechas de entregas y se cruzó con la información que aparece en el SCT y con los informes de estatus. Los resultados fueron:

**Tabla 3 Retraso de los proyectos**

| <b>Proyecto 1: Fase Construcción</b> | <b>Retraso en la entrega (Semanas)</b> |
|--------------------------------------|----------------------------------------|
| Iteración 1                          | 2                                      |
| Iteración 2                          | 3                                      |
| Iteración 3                          | 2                                      |
| Iteración 4                          | 2                                      |
| Iteración 5                          | 12                                     |
| <b>Total</b>                         | <b>21</b>                              |
| <b>Proyecto 2: Fase Construcción</b> | <b>Retraso en la entrega (Semanas)</b> |
| Iteración 1                          | 0                                      |
| Iteración 2                          | 2                                      |
| Iteración 3                          | 1                                      |
| Iteración 4                          | 3                                      |
| Iteración 5                          | 2                                      |
| Iteración 6                          | 9                                      |
| <b>Total</b>                         | <b>17</b>                              |

Fuente: Repositorio de Proyectos Ejecutados (2006)

Mediante el análisis de los datos de la tabla 3 se permite señalar que sólo en una de las iteraciones hubo una entrega a tiempo. Por otra parte, en la última iteración de cada proyecto se puede observar un retraso marcado, lo que indica que el producto nunca estuvo listo en el tiempo que se negoció.

#### 1.2.4 Documentación del código

La documentación del código, a pesar de ser un proceso que se va realizando durante cada una de las iteraciones, es en la fase final cuando se realiza una última revisión, esto motivado a que los esfuerzos de las iteraciones previas son orientados a producir módulos funcionales.

**Tabla 4 Errores de Documentación**

|                                         | Proyecto 1 | Proyecto 2 |
|-----------------------------------------|------------|------------|
| Errores de documentación identificados. | 450        | 1483       |
| Días estimados para corregir.           | 37,5       | 123,583333 |

Fuente: Repositorio de Proyectos Ejecutados (2006)

Cada proyecto posee estrategias distintas para llevar el control de la documentación del código, sin embargo, como se puede visualizar en la tabla 4, al final siempre resulta en un tiempo considerable. El procedimiento ideal es incluir dentro de las actividades de desarrollo, la documentación del código, es decir, que el desarrollo del caso de uso no esté completo hasta que se haya incorporado la documentación, así se evita esta fase final de corrección de la documentación.

#### 1.2.5 Peer Reviews – Revisión del código

Los *Peer Reviews* son realizados al menos cada mes por el líder técnico y el arquitecto, básicamente chequean el código fuente utilizando listas de verificación, luego se realiza una reunión para discutir las fallas, cuales son las causas y cual es la forma correcta de realizar el desarrollo del fuente, por último se realiza una asignación de los errores encontrados para hacer la corrección. Se hizo una

revisión de los informes almacenados en el repositorio y el resultado fue el siguiente:

**Tabla 5 Resultados de la revisión del fuente**

|                                    | Proyecto 1 |     |     |    |     |       |        | Proyecto 2 |     |     |    |     |     |       |        |
|------------------------------------|------------|-----|-----|----|-----|-------|--------|------------|-----|-----|----|-----|-----|-------|--------|
| Iteración                          | 1          | 2   | 3   | 4  | 5   | Total | %      | 1          | 2   | 3   | 4  | 5   | 6   | Total | %      |
| Documentación de código            | 0          | 107 | 83  | 0  | 212 | 402   | 39,37% | 0          | 120 | 300 | 41 | 442 | 323 | 1226  | 44,70% |
| Manejo de excepciones              | 0          | 75  | 76  | 34 | 42  | 227   | 22,23% | 0          | 113 | 6   | 22 | 66  | 114 | 321   | 11,70% |
| Métodos recursivos                 | 0          | 1   | 0   | 2  | 2   | 5     | 0,49%  | 0          | 2   | 3   | 0  | 0   | 4   | 9     | 0,33%  |
| Convenios de nombre                | 0          | 7   | 105 | 0  | 83  | 195   | 19,10% | 0          | 42  | 49  | 73 | 93  | 323 | 580   | 21,14% |
| Patrones de diseño                 | 0          | 12  | 5   | 0  | 8   | 25    | 2,45%  | 0          | 14  | 5   | 10 | 13  | 22  | 64    | 2,33%  |
| Pruebas unitarias                  | 0          | 56  | 25  | 0  | 45  | 126   | 12,34% | 0          | 39  | 37  | 49 | 99  | 38  | 262   | 9,55%  |
| Reutilización de código            | 0          | 12  | 12  | 0  | 4   | 28    | 2,74%  | 0          | 2   | 4   | 2  | 9   | 89  | 106   | 3,86%  |
| Utilización eficiente de las capas | 0          | 5   | 6   | 0  | 2   | 13    | 1,27%  | 0          | 5   | 15  | 42 | 14  | 99  | 175   | 6,38%  |
| Totales                            |            |     |     |    |     | 1021  |        |            |     |     |    |     |     | 2743  |        |

Fuente: Repositorio de Proyectos Ejecutados (2006)

Como puede observarse, la mayor cantidad de errores está concentrada en errores de documentación de código y convenio de nombres. A pesar de que estos errores son fácilmente solventables, pues no amerita más de cinco minutos su resolución, por su alta cantidad se traducen en un esfuerzo considerable. Por ejemplo, para 402 errores se necesitan aproximadamente 33,5 horas para resolverlos. El manejo de excepciones y las pruebas unitarias son los siguientes en la lista, los cuales tienen una cantidad alta de incidencias. Estos dos son altamente impactantes ya que depende de ellos la realización de las pruebas, lo cual repercute en la calidad del producto. Por último, utilización eficiente de las capas, reutilización de código, patrones de diseño y métodos recursivos en ese orden, los cuales son pocos pero son básicos para la orientación a objetos lo que indica que los desarrolladores no están claros en estos conceptos y fallan en su utilización.

### 1.2.6 Fxcop

De los análisis efectuados por *Fxcop* se obtiene un reporte detallado de los problemas relacionados con el código fuente.

**Tabla 6 Errores detectados con Fxcop**

|                                          | Proyecto 1   | %      | Proyecto 2       | %      |
|------------------------------------------|--------------|--------|------------------|--------|
| Convenio de Nombres                      | 945          | 34,81% | 689              | 41,41% |
| Estándares de lenguaje                   | 570          | 20,99% | 440              | 26,44% |
| Manejo de Excepciones                    | 1200         | 44,20% | 535              | 32,15% |
| <b>Total</b>                             | <b>2715</b>  |        | <b>1664</b>      |        |
| <b>Horas estimadas para correcciones</b> | <b>452,5</b> |        | <b>277,33333</b> |        |

Fuente: Repositorio de Proyectos Ejecutados (2006)

Como se puede observar, el principal problema está en el manejo de excepciones. Sin embargo, el convenio de nombres también tiene una cantidad significativa de errores. Los estándares del lenguaje son los que tienen menor puntuación a pesar de que no deben ser desestimados.

El análisis general de los resultados presentados en las Tablas permite definir parcialmente los conocimientos, competencias y destrezas específicas del cargo de desarrollador de *Software* en SSCA; sin embargo, en la intención de formular un diseño de instrucción que se ajuste a las necesidades reales de la empresa y de los Desarrolladores se consideró prudente llevar a efecto el estudio de campo.

## 2. Estudio de Campo-Aplicación de las entrevistas

Seguidamente se presentan diecisiete (17) cuadros que sirven para representar las 17 interrogantes con las respectivas respuestas dadas por los cinco (5) Líderes seleccionados intencionalmente en la búsqueda de datos para definir parcialmente los conocimientos, competencias y destrezas específicas del cargo de desarrollador de *Software* en SSCA.

## Cuadro N° 1

**Evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de *Software* en SSCA para el desarrollo de un diseño de Instrucción para el adiestramiento de Desarrolladores de *Software*.**

| 1. ¿Cuántos desarrolladores tiene a su cargo? |    |
|-----------------------------------------------|----|
| Entrevistado 1                                | 18 |
| Entrevistado 2                                | 10 |
| Entrevistado 3                                | 10 |
| Entrevistado 4                                | 14 |
| Entrevistado 5                                | 10 |

Fuente: Entrevistas aplicadas (2007)

De acuerdo a la tabla anterior se puede deducir que los líderes poseen experiencia en la supervisión de desarrolladores, la menor cantidad de supervisados es de 10, lo cual es un indicador significativo, considerando todo el trabajo que implica facilitar la gestión a 10 desarrolladores.

## Cuadro N° 2

**Evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de *Software* en SSCA para el desarrollo de un diseño de Instrucción para el adiestramiento de Desarrolladores de *Software*.**

| 2. ¿Qué nivel de capacitación poseen? |                                                                                                                                                                                                                                    |
|---------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Entrevistado 1                        | Todos mis desarrolladores están en la línea de ingenieros de sistemas o informática. Algunos no están graduados: pero sólo les falta un semestre o la tesis                                                                        |
| Entrevistado 2                        | Mis desarrolladores son ingenieros de sistemas o información, la mayoría están recién graduados, sólo tengo 1 con más de 2 años de experiencia. Adicionalmente están recientes dentro de la organización, no tienen más de un año. |
| Entrevistado 3                        | Son ingenieros. Unos con más experiencia que otros; pero las fallas siempre están presentes, a casi todos les falta mayor conocimiento específico sobre la fase de construcción, en la que trabajan.                               |
| Entrevistado 4                        | Son ingenieros de sistemas o licenciados en computación, han tenido poca experiencia en el desarrollo de <i>software</i> de alto nivel.                                                                                            |
| Entrevistado 5                        | Ingenieros de sistemas, algunos están cursando postgrado.                                                                                                                                                                          |

Fuente: Entrevistas aplicadas. (2007)

Existe predominio en la afirmación de que los desarrolladores poseen poca experiencia. Se refieren a dificultad para evitar errores en esa figura; lo que es

indicativo que existe déficit de conocimientos y de destrezas, en consecuencia se infiere bajo nivel de competencia. Lo que, sin duda, es un indicador de la necesidad de adiestramiento específico.

### **Cuadro N° 3**

**Evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de *Software* en SSCA para el desarrollo de un diseño de Instrucción para el adiestramiento de Desarrolladores de *Software*.**

| <b>3. ¿Los desarrolladores entienden los procesos de la empresa y su rol dentro del equipo del proyecto?</b> |                                                                                                                                                                                                                                               |
|--------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Entrevistado 1</b>                                                                                        | Sí, ellos adicionalmente a sus actividades de desarrollo son parte del grupo de arquitectura e innovación por lo tanto están en un continuo mejoramiento de lo procesos organizacionales.                                                     |
| <b>Entrevistado 2</b>                                                                                        | Sí tienen la visión general, sin embargo no están inmersos en el proceso. Hay muchas de sus responsabilidades que no pueden realizar sin una guía. Les hace falta cierta instrucción.                                                         |
| <b>Entrevistado 3</b>                                                                                        | Sinceramente pienso que no lo saben, o al menos desconocen como llevar a cabalidad su rol. La mayoría espera a que su líder le de todas las indicaciones, muchas veces no producen por sí mismos.                                             |
| <b>Entrevistado 4</b>                                                                                        | Hay un grupo que está claro, pero es minoría, el grupo que falla programa por ensayo y error y no estructura su trabajo.                                                                                                                      |
| <b>Entrevistado 5</b>                                                                                        | La mayoría no, los procedimientos y normas que se establecen para el desarrollo con frecuencia no son cumplidos, considero que esta situación es producida por desconocimiento por parte de los desarrolladores de algunos aspectos técnicos. |

Fuente: Entrevistas aplicadas. (2007)

En este cuadro sólo el entrevistado 1 indica que el grupo conoce los procesos de la empresa y su rol dentro del equipo de trabajo, el resto afirma que desconocen los procesos. Esta es uno de las áreas que deben ser cubiertas por el curso. Está asociado al REQ-2-CMM y REQ-3-CMM.

#### Cuadro N° 4

**Evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de *Software* en SSCA para el desarrollo de un diseño de Instrucción para el adiestramiento de Desarrolladores de *Software*.**

| <b>4. ¿Los desarrolladores entienden todos los requisitos del sistema: criterios de aceptación, rendimiento, acuerdos de codificación?</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Entrevistado 1</b>                                                                                                                      | Sí, los requerimientos están documentados en UML, lo cual permite un fácil entendimiento por quien los revise. Los criterios de aceptación incluyen los de rendimiento, para sintonizar estos requerimientos se hace una fase de una o dos semanas de inmersión y análisis de los requerimientos, por supuesto siempre hay fallas. Con respecto a los acuerdos de codificación el ambiente de trabajo es configurado para que cumpla con los acuerdos establecidos por el arquitecto o el cliente.                                                                                                  |
| <b>Entrevistado 2</b>                                                                                                                      | No, a pesar que los documentos están en UML y divididos por casos de uso existen oportunidades en que no entienden qué les están solicitando, ya sea porque no entienden los diagramas o porque no están suficientemente claros, en dichos casos se explica el requerimiento al desarrollador. La mayoría de ellos no es muy dado a la consulta del manual y por lo tanto generalmente sólo leen los casos de uso que tienen asignados por lo que al menos que se les participe en una reunión los criterios de aceptación, de rendimiento y acuerdos de codificación, no se enteran de cuáles son. |
| <b>Entrevistado 3</b>                                                                                                                      | No, creo que tienen problemas para entender UML, adicionalmente cuando codifican no están enfocados en el rendimiento para nada, sólo están tratando de que les funcione el caso de uso que estén trabajando.                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Entrevistado 4</b>                                                                                                                      | No, generalmente entienden el caso de uso asignado, pero no todos los requerimientos del sistema, esto no les da la visibilidad completa del sistema y por lo tanto trabajan de manera aislada. Los acuerdos de codificación son generalmente incumplidos y se tiene que recurrir a Fxcop para validar los problemas, esto siempre ocupa un buen lapso de tiempo para corregir estos errores, y genera retardo en la productividad.                                                                                                                                                                 |
| <b>Entrevistado 5</b>                                                                                                                      | No, los únicos que entienden claramente los requerimientos son los que tienen mayor experiencia, pero los más novatos siempre necesitan que se les explique que se necesita y como hacerlo. A pesar de que se configura el ambiente de trabajo para que cumplan con los acuerdos de codificación siempre hay problemas que indican que hacen caso omiso de las reglas.                                                                                                                                                                                                                              |

Fuente: Entrevistas aplicadas. (2007)

Nuevamente sólo el Entrevistado 1 afirma que su equipo conoce y maneja los requisitos en su totalidad. El resto de los entrevistados afirma que no conocen lo suficiente de UML y de los procedimientos para seguir un convenio de nombres. Los hallazgos están relacionados con los requerimientos: REQ-41-CMM, REQ-42-CMM, REQ-43-CMM, REQ-44-CMM, REQ-45-CMM y REQ-46-OPR.



### Cuadro N° 5

**Evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de *Software* en SSCA para el desarrollo de un diseño de Instrucción para el adiestramiento de Desarrolladores de *Software*.**

| <b>5. ¿Los desarrolladores manejan la Metodología de Desarrollo de la Organización y/o alguna otra metodología, como CMM, RUP, MSF?</b> |                                                                                                                                                                              |
|-----------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Entrevistado 1</b>                                                                                                                   | Sí, ellos deben al menos manejar alguna metodología para poder ser contratados.                                                                                              |
| <b>Entrevistado 2</b>                                                                                                                   | No, se les impartió una charla sobre CMM pero creo que se debería fortalecerse con otras charlas o algo más formal, sobre todo de la metodología de la organización.         |
| <b>Entrevistado 3</b>                                                                                                                   | Sí, los desarrolladores de Java generalmente manejan RUP y el perfil de mis desarrolladores incluye que deben manejar Java.                                                  |
| <b>Entrevistado 4</b>                                                                                                                   | No lo creo, además pienso que la metodología la deben llevar los líderes.                                                                                                    |
| <b>Entrevistado 5</b>                                                                                                                   | No, los desarrolladores se empeñan en programar desordenadamente. Se les impartió un curso de CMM; pero les pareció fastidioso, no hubo mucho provecho de ese entrenamiento. |

Fuente: Entrevistas aplicadas. (2007)

Tres de los entrevistados indican que no manejan metodologías de desarrollo. Es necesario que se entienda como son organizados los procesos para llevar a cabo una ejecución del proyecto exitoso. Los requisitos identificados de este cuadro son: REQ-47-ARQ.

### Cuadro N° 6

**Evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de *Software* en SSCA para el desarrollo de un diseño de Instrucción para el adiestramiento de Desarrolladores de *Software*.**

| <b>6. ¿Los desarrolladores conocen los patrones de diseño?</b> |                                                                                                                                                                         |
|----------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Entrevistado 1</b>                                          | Sí, ellos deben saber como implementar lo establecido por el arquitecto.                                                                                                |
| <b>Entrevistado 2</b>                                          | No, no creo que tengan completamente claro la orientación a objetos, sólo hacen lo que el arquitecto les indique.                                                       |
| <b>Entrevistado 3</b>                                          | Sólo 1 de los desarrolladores sabe que son patrones de diseño.                                                                                                          |
| <b>Entrevistado 4</b>                                          | Al menos saben que es SOA por uno de los proyectos que se ejecutó; pero no tienen la iniciativa de investigar y conocer más de arquitectura a menos que se les imponga. |
| <b>Entrevistado 5</b>                                          | Creo que sólo manejan el de programación multicapas.                                                                                                                    |

Fuente: Entrevistas aplicadas. (2007)

Cuatro de los entrevistados indican que los desarrolladores no conocen en profundidad lo que son los patrones de diseño. Los requisitos asociados a este cuadro son: REQ-48-ARQ.

### **Cuadro N° 7**

**Evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de *Software* en SSCA para el desarrollo de un diseño de Instrucción para el adiestramiento de Desarrolladores de *Software*.**

| <b>7. ¿Los desarrolladores han presentado problemas con los documentos de diseño detallados?</b> |                                                                                                                                                                                               |
|--------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Entrevistado 1</b>                                                                            | No, existe un formato bien explícito de cómo hacer un documento de diseño detallado, además todos ellos están por certificarse en UML.                                                        |
| <b>Entrevistado 2</b>                                                                            | Sí. De hecho en el último proyecto el arquitecto dio una charla de cómo realizarlos, pero lo que hicieron no puede llamarse diseño detallado, realmente ni siquiera le vieron la importancia. |
| <b>Entrevistado 3</b>                                                                            | Sí, muchos, imagínate que ni siquiera saben identificar que amerita y que no un diseño detallado.                                                                                             |
| <b>Entrevistado 4</b>                                                                            | Sí, creo que mi equipo tiene problemas serios con UML.                                                                                                                                        |
| <b>Entrevistado 5</b>                                                                            | No, los documentos que han hecho están dentro del nivel aceptable.                                                                                                                            |

Fuente: Entrevistas aplicadas. (2007)

Tres de los entrevistados demandan adiestramiento en diseño detallado, adicionalmente indican que los desarrolladores no manejan en profundidad UML. Requisitos asociados al cuadro: REQ-49-ARQ, REQ-50-OPR, REQ-51-CMM y REQ-52-CMM.

### Cuadro N° 8

**Evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de *Software* en SSCA para el desarrollo de un diseño de Instrucción para el adiestramiento de Desarrolladores de *Software*.**

| <b>8. ¿Qué nivel de conocimiento del Lenguaje de programación poseen?</b> |                                                                                                                                                                          |
|---------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Entrevistado 1</b>                                                     | Tienen un nivel básico, manejan las estructuras.                                                                                                                         |
| <b>Entrevistado 2</b>                                                     | Un nivel bajo, no conocen las bases teóricas del framework, muchos de los errores de Fxcop serían evitados si tuvieran un mejor conocimiento del lenguaje.               |
| <b>Entrevistado 3</b>                                                     | Tienen los conocimientos necesarios para hacer un programa funcional pero no con los mejores métodos.                                                                    |
| <b>Entrevistado 4</b>                                                     | No se si será problema de lenguaje de programación pero muchas veces ni siquiera le dan a F1 para pedir ayuda y re-implementan cosas que el lenguaje de por sí las hace. |
| <b>Entrevistado 5</b>                                                     | Tengo sólo dos personas expertas en el lenguaje, que se encargan de darle apoyo a los demás integrantes del equipo.                                                      |

Fuente: Entrevistas aplicadas. (2007)

Todos los entrevistados coinciden que la mayoría de sus desarrolladores necesitan un refuerzo en lenguaje de programación. REQ-53-OPR y REQ-54-OPR.

### Cuadro N° 9

**Evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de *Software* en SSCA para el desarrollo de un diseño de Instrucción para el adiestramiento de Desarrolladores de *Software*.**

| <b>9. ¿Tienen un correcto manejo de excepciones?</b> |                                                                                                                                                           |
|------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Entrevistado 1</b>                                | Sí, es lo primero que deben atacar antes de iniciar la programación.                                                                                      |
| <b>Entrevistado 2</b>                                | En .Net es muy sencillo olvidarse de las excepciones por lo que existen problemas si no se detecta a tiempo que no se están empleando de manera correcta. |
| <b>Entrevistado 3</b>                                | No, la mayoría no sabe cómo se usan correctamente.                                                                                                        |
| <b>Entrevistado 4</b>                                | No, es una de los principales problemas de calidad de las aplicaciones de este grupo.                                                                     |
| <b>Entrevistado 5</b>                                | Tienen problemas con el despliegue de mensajes acordes a la excepción que viene subiendo.                                                                 |

Fuente: Entrevistas aplicadas. (2007)

Cuatro de los entrevistados indican que tienen problemas con el manejo de excepciones. Requerimiento asociado: REQ-55-OPR.

### **Cuadro N° 10**

**Evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de *Software* en para el desarrollo de un diseño de Instrucción para el adiestramiento de Desarrolladores de *Software*.**

| <b>10. ¿Tienen algún problema con las pruebas, unitarias y funcionales?</b> |                                                                                                                                                                                                                                                                                      |
|-----------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Entrevistado 1</b>                                                       | El único problema es que no las están haciendo antes de hacer el caso de uso.                                                                                                                                                                                                        |
| <b>Entrevistado 2</b>                                                       | Sí, es otro problema que afrontamos, la falta de pruebas unitarias. No saben cómo hacerlas y mucho menos utilizar NUnit, o si las hacen prueban algo que no era necesario probar.                                                                                                    |
| <b>Entrevistado 3</b>                                                       | Sí, no logro hacer que dentro de su procedimiento regular ejecutan las pruebas unitarias para validar que no rompan la estabilidad de la aplicación.                                                                                                                                 |
| <b>Entrevistado 4</b>                                                       | Sí, por falta de experiencia hacen pruebas que no prueban sino los “camino felices”. Adicionalmente si hay retrasos lo primero que obvia el desarrollador son las pruebas unitarias y funcionales lo que hace que cuando llega al departamento de pruebas la aplicación no funciona. |
| <b>Entrevistado 5</b>                                                       | Sí, con la falta de experiencia hace que no prueben bien y entregan casos de uso que con 2 intentos fallan.                                                                                                                                                                          |

Fuente: Entrevistas aplicadas. (2007)

En este cuadro todos los entrevistados coinciden en que los desarrolladores tienen problemas con las pruebas unitarias. Requisito asociado: REQ-56-OPR, REQ-57-ARQ .

### **Cuadro N° 11**

**Evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de *Software* en SSCA para el desarrollo de un diseño de Instrucción para el adiestramiento de Desarrolladores de *Software*.**

| <b>11. ¿Cómo es realizado el proceso de corrección de errores por los desarrolladores?</b> |                                                                                                                                                                                               |
|--------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Entrevistado 1</b>                                                                      | Se hace la corrección del error y luego se ejecutan las pruebas unitarias para validar que no afectó otro módulo de la aplicación.                                                            |
| <b>Entrevistado 2</b>                                                                      | Los resultados de pruebas son sometidos a un proceso de corrección y luego de corregidos vuelven a validar la aplicación.                                                                     |
| <b>Entrevistado 3</b>                                                                      | Generalmente los errores son corregidos por el que construyó el modulo sin embargo existen ocasiones en que deben corregir módulos que desconocen, el líder se encarga de explicar el módulo. |
| <b>Entrevistado 4</b>                                                                      | La corrección de errores ha sido nuestro principal problema, los errores que se corrigen generan otros errores en otras zonas del sistema.                                                    |
| <b>Entrevistado 5</b>                                                                      | Generalmente no los corrigen, interviene el líder y resuelve.                                                                                                                                 |

Fuente: Entrevistas aplicadas. (2007)

Dos de los entrevistados indican que tienen problemas en el proceso de corrección de errores. Requerimientos asociados: REQ-58-OPR y REQ-59-ARQ.

### **Cuadro N° 12**

**Evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de *Software* en SSCA para el desarrollo de un diseño de Instrucción para el adiestramiento de Desarrolladores de *Software*.**

| <b>12. ¿Tienen problemas con el control de versiones y la integración continua?</b> |                                                                                                                                                              |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Entrevistado 1</b>                                                               | No, el manejo del CVS está estandarizado.                                                                                                                    |
| <b>Entrevistado 2</b>                                                               | De vez en cuando se rompe el código, además tienen problemas de sincronización cuando hay muchos cambios.                                                    |
| <b>Entrevistado 3</b>                                                               | Sí, cuando existen varias ramas de código generadas por los ciclos de pruebas hace que la sincronización sea un desastre nunca saben que han hecho y que no. |
| <b>Entrevistado 4</b>                                                               | Sí, continuamente rompen el CVS, no hacen el proceso de sincronización como está explícito en el documento.                                                  |
| <b>Entrevistado 5</b>                                                               | No, sólo de vez en cuando se rompe el CVS por descuido.                                                                                                      |

Fuente: Entrevistas aplicadas. (2007)

Cuatro de los entrevistados indican que el CVS se rompe, esto no sucedería si todos llevan a cabo el procedimiento. Los requerimientos de este cuadro son: REQ-60-CMM y REQ-61-CMM.

### **Cuadro N° 13**

**Evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de *Software* en SSCA para el desarrollo de un diseño de Instrucción para el adiestramiento de Desarrolladores de *Software*.**

| <b>13. ¿Realizan de manera acorde el uso de las herramientas de control de tareas?</b> |                                                                                                                                                     |
|----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Entrevistado 1</b>                                                                  | Hay ciertos problemas con los casos en curso.                                                                                                       |
| <b>Entrevistado 2</b>                                                                  | Sí, generalmente no documentan toda la información de los casos, no cambian el estatus a trabajando para saber que están atacando en la actualidad. |
| <b>Entrevistado 3</b>                                                                  | No, no hay problemas con Jira.                                                                                                                      |
| <b>Entrevistado 4</b>                                                                  | Sí, siento que no se explota todo el potencial de la herramienta por falta de adiestramiento.                                                       |
| <b>Entrevistado 5</b>                                                                  | No, es una aplicación bien sencilla de utilizar.                                                                                                    |

Fuente: Entrevistas aplicadas. (2007)

Tres de los entrevistados indican que no se utiliza correctamente la herramienta.

Requerimientos asociados: REQ-62-CMM y REQ-63-CMM.

#### **Cuadro N° 14**

**Evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de *Software* en SSCA para el desarrollo de un diseño de Instrucción para el adiestramiento de Desarrolladores de *Software*.**

| <b>14. ¿Al llevar a cabo proyectos se ha producido algún retraso en la Fase de Construcción?</b> |                                                                                                                                                                          |
|--------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Entrevistado 1</b>                                                                            | Sí, los proyectos siempre están ajustados de tiempo, y siempre ocupa más.                                                                                                |
| <b>Entrevistado 2</b>                                                                            | Sí, los problemas que se van dando en el desarrollo van restando tiempo y eso origina retrasos.                                                                          |
| <b>Entrevistado 3</b>                                                                            | Sí, el retraso más marcado se encuentra en esta fase, es probable que sea debido a que es la más larga y en la que se dan más errores.                                   |
| <b>Entrevistado 4</b>                                                                            | Sí, es la fase donde se desempeñan los desarrolladores que son los que tienen menos experiencia en la organización.                                                      |
| <b>Entrevistado 5</b>                                                                            | Hay casos en que no ha habido retrasos pero son muy pocos, el desarrollo de <i>software</i> es un proceso complejo y no siempre las estimaciones de tiempo son precisas. |

Fuente: Entrevistas aplicadas. (2007)

Según este cuadro, existe un alto porcentaje de proyectos que no son terminados en el tiempo planificado, esto indica que las estimaciones no están siendo acordes a las necesidades reales de tiempo del personal con que se cuenta. Requerimiento para este cuadro: REQ-64-CMM; lo propio es poder contar con personal capaz de asumir la planificación efectivamente, con los recursos asignados y en el tiempo oportuno.

### **Cuadro N° 15**

**Evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de *Software* en SSCA para el desarrollo de un diseño de Instrucción para el adiestramiento de Desarrolladores de *Software*.**

| <b>15. ¿Considera que el tiempo planificado les resulta suficiente para realizar las tareas de desarrollo?</b> |                                                                                                                                                                  |
|----------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Entrevistado 1</b>                                                                                          | Sí; pero ocurren fallas técnicas que rompen el plan.                                                                                                             |
| <b>Entrevistado 2</b>                                                                                          | Sí, pero es difícil para desarrolladores poco expertos.                                                                                                          |
| <b>Entrevistado 3</b>                                                                                          | Sí, se planifica de acuerdo a la complejidad del caso de uso y la destreza del desarrollador establece la diferencia.                                            |
| <b>Entrevistado 4</b>                                                                                          | No, generalmente se dispone de menos tiempo del necesario por lo que se tiene que terminar trabajando fines de semana y en las noches para entregar sin demoras. |
| <b>Entrevistado 5</b>                                                                                          | Sí, siempre se puede negociar con el cliente el tiempo de entrega.                                                                                               |

Fuente: Entrevistas aplicadas. (2007)

Este cuadro reafirma lo expuesto en el cuadro 14, cuatro (4) de los Líderes encuestados reconocen que existen problemas con la estimación, es probable que no se esté analizando en profundidad la complejidad de los casos de uso. Requerimiento para este cuadro: REQ-65-CMM.

### **Cuadro N° 16**

**Evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de *Software* en SSCA para el desarrollo de un diseño de Instrucción para el adiestramiento de Desarrolladores de *Software*.**

| <b>16. ¿Está el desarrollador de acuerdo y comprometido con los tiempos de entrega?</b> |                                                                                                                                             |
|-----------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Entrevistado 1</b>                                                                   | Sí sin lugar a duda.                                                                                                                        |
| <b>Entrevistado 2</b>                                                                   | Sí puede estar de acuerdo; pero no siempre está comprometido con la entrega, muchas veces no aceptan trabajar sobre tiempo.                 |
| <b>Entrevistado 3</b>                                                                   | No, muchas veces replica por el poco tiempo que le fue asignado.                                                                            |
| <b>Entrevistado 4</b>                                                                   | Sí; pero generalmente no cumple o entrega incompleto.                                                                                       |
| <b>Entrevistado 5</b>                                                                   | Sí están; sin embargo por la falta de experiencia hace que no se cumplan los tiempos por muy comprometidos que estén para hacer la entrega. |

Fuente: Entrevistas aplicadas. (2007)

Significativos resultados, cuatro (4) de los entrevistados consideran que los desarrolladores están de acuerdo y comprometidos con los tiempos de entrega;

sin embargo se ha identificado que mayormente los retrasos se producen en la etapa del proyecto en que ellos intervienen, se deduce que los desarrolladores desconocen todas las actividades asociadas al desarrollo cuando aceptan el tiempo asignado a un caso de uso. Requerimiento para este cuadro: REQ-66-CMM y REQ-67-CMM.

### **Cuadro N° 17**

**Evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de *Software* en SSCA para el desarrollo de un diseño de Instrucción para el adiestramiento de Desarrolladores de *Software*.**

| <b>17. ¿Considera que su producto refleja la calidad requerida por el cliente?</b> |                                                                                                                                                                                                                                                                                                                                      |
|------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Entrevistado 1</b>                                                              | Sí, el cliente obtiene lo que paga.                                                                                                                                                                                                                                                                                                  |
| <b>Entrevistado 2</b>                                                              | No, el cliente hace sus revisiones exhaustivas y siempre está inconforme con algo.                                                                                                                                                                                                                                                   |
| <b>Entrevistado 3</b>                                                              | No, por la falta de experiencia se han dejado de utilizar tecnologías más de vanguardia por ser más complejas de aplicar, adicionalmente en la fase final del desarrollo se debe establecer un período para corregir todos los defectos para hacer la entrega final, la cual no es corta y no siempre se corrigen todos los errores. |
| <b>Entrevistado 4</b>                                                              | No, el utilizar personal en que la mayoría aun está en proceso de aprendizaje hace que aprendan desarrollando el proyecto y no que estén en un proceso de perfeccionamiento de las habilidades, lo cual influye en un producto que no es de alta calidad, quizás cumple los requisitos mínimos de aceptación.                        |
| <b>Entrevistado 5</b>                                                              | Sí, si no el cliente no aceptaría las aplicaciones.                                                                                                                                                                                                                                                                                  |

Fuente: Entrevistas aplicadas. (2007)

Tres de los entrevistados afirman que no, acotan que el retraso en las entregas es un factor determinante, se destaca que la debilidad está en la fase de desarrollo y aceptan que el déficit de experiencia del personal afecta el producto final y por lo tanto la satisfacción al cliente. Esto es un importante indicador de la necesidad de capacitar al personal para mejorar la calidad. Requisitos asociados: REQ-68-CMM.



## **CAPÍTULO VI: RESULTADOS DEL PROYECTO**

### **1. FASE 2: ANÁLISIS**

De acuerdo a los requerimientos listados en el capítulo anterior, un buen desarrollador debe poseer competencias en nueve áreas fundamentales, Procesos Organizacionales, Gestión de requisitos, Planificación, Seguimiento y control, Gestión de configuración, Gestión de calidad, Desarrollo, Arquitectura y diseño, y Destrezas personales. Sin embargo, según los entrevistados, la realidad de los desarrolladores de SSCA es que no manejan plenamente cada una de estas áreas. La autoridad para emitir este juicio queda evidenciada en el Cuadro 1 del capítulo anterior, donde se especifica que ninguno de los entrevistados tiene menos de 10 personas a su cargo.

Por otra parte, el que no manejen a plenitud las áreas antes señaladas no debería ser una primicia, según el Cuadro 2, la mayoría de los desarrolladores de los equipos son recién graduados o con poca experiencia laboral. Esto evidencia la necesidad que tienen de experimentar las actividades que deben realizar sin el temor a fallar, necesitan realizar un trabajo que les permita equivocarse y aprender, necesitan adiestramiento.

Con respecto al proceso de interacción entre todos los individuos que conforman el equipo del proyecto, lo que es el proyecto, el equipo, sus roles y responsabilidades, también tienen fallas, esto queda evidenciado en el Cuadro 3, en el cual la mayoría de las respuestas fueron que los desarrolladores desconocen los procesos organizacionales.

De acuerdo al Cuadro 4, los desarrolladores no manejan los requisitos del sistema en su totalidad. Este es una de las fallas más graves, el desarrollador necesita saber a ciencia cierta qué es lo que tiene que construir y qué consideraciones debe contemplar durante su construcción, por lo tanto debe estar en la capacidad

de analizar y asimilar cada uno de los documentos asociados a los requerimientos de *software*.

Las metodologías de *software* son los lineamientos que han sido puestos a prueba por gran cantidad de personas y les han resultado satisfactorios para desarrollar el *software*. El que se desconozca una metodología no quiere decir que estará destinado al fracaso, pero sí que no sabrá cada uno de los pasos que se están siguiendo dentro de la organización para desarrollar los productos. En el cuadro 5 se evidencia que la mayoría de los desarrolladores desconocen una metodología de trabajo y tampoco manejan la que utiliza la organización.

Los patrones de diseño, son utilizados con frecuencia por los arquitectos de *software* cuando realizan el diseño arquitectónico de la aplicación. Este documento del diseño arquitectónico forma parte de los requisitos del sistema, ya que el desarrollador debe cumplir con ese documento a cabalidad. Si no es entendido, suceden los mismos problemas de la gestión de requerimientos. En el cuadro 6, se puede evidenciar que la mayoría de los desarrolladores no manejan los patrones de diseño con la profundidad necesaria.

En el Cuadro 7, se puede notar que hay problemas con la elaboración de documentos de diseño detallado, estos documentos son de suma importancia, ya que permiten que quede documentado las soluciones elegidas para problemas complejos. Una vez programado, el código fuente complejo es muy difícil entenderlo, inclusive para el mismo desarrollador que lo produjo, luego de algunos meses. Por lo tanto, la norma es elaborar documentos de diseño detallado para su futura referencia.

De acuerdo al Cuadro 8, los desarrolladores tienen un nivel de conocimiento básico del lenguaje de programación, esto indica que el desarrollador no utilizará

las mejores prácticas del lenguaje a menos que tenga suficiente tiempo para investigarlas.

El manejo de excepciones es parte fundamental de cualquier aplicación de *software*, ya que dependen de ellas el cómo se comporta la aplicación ante los errores y cómo son capturados y presentados al usuario. Esto es utilizado por el equipo de pruebas para saber qué le está pasando a la aplicación cuando falla. Si no se manejan las excepciones no se podría registrar qué es lo que le está pasando y existen muchos casos en los que es muy difícil reproducir los errores. En el cuadro 9, se puede notar que hace falta que se refuerce el manejo de excepciones para algunos desarrolladores.

Muy cercano al manejo de excepciones, están las pruebas unitarias, ellas permiten saber si la aplicación está estable o no lo está. Por ejemplo, si se tiene una aplicación que posee 100 casos de uso, los cuáles están altamente interrelacionados, ¿cómo se puede saber, si se modifica alguno, cómo se afectan los demás?, allí está la importancia de desarrollar buenas pruebas unitarias, ya que basta con ejecutar las pruebas unitarias de toda la aplicación y si alguna falla, entonces ahí se vio afectado un caso de uso. Según el cuadro 10, la mayoría de los desarrolladores tienen problemas con las pruebas unitarias.

El proceso de corrección de errores es uno de los procesos que más se utilizan durante la construcción de un *software*, ya que el error es de humanos, es normal que el fuente posea errores, por lo tanto las técnicas y herramientas deben ser conocidas por los desarrolladores. Según el cuadro 11, esta es una de las fallas que actualmente poseen la mayoría de ellos.

Del Cuadro 12, se puede determinar que la mayoría de los desarrolladores no manejan a cabalidad el control de versiones y la integración continua.

La administración de las tareas es una actividad sumamente importante, el líder debe estar en la capacidad de saber qué está haciendo cada uno de sus desarrolladores en todo momento, esto es logrado a través de Jira, pero para ello debe ser utilizada de manera correcta, es decir, llevando las asignaciones por todos los pasos que están definidos dentro del administrador. De acuerdo al cuadro 13, hay ciertas deficiencias en el uso de esta herramienta.

En el cuadro 14, se evidencia que los proyectos se están retrasando en su mayoría. Este problema puede ser causado por problemas de estimaciones de tiempo . En el cuadro 15 y 16, se confirma que uno de los problemas es el de estimación de los tiempos.

En el cuadro 17, se evidencia que existe un problema de calidad con el producto entregado.

Una vez analizados los resultados que arrojó, tanto el análisis del repositorio como la entrevista a los Líderes, se infiere que existe concordancia entre los resultados. Se pudo determinar que las fallas identificadas en los proyectos están relacionadas con el déficit de conocimientos y habilidades que portan los desarrolladores, así mismo se pudo definir la situación actual en cuanto a los requerimientos de adiestramiento a satisfacer mediante el diseño instruccional para el adiestramiento de los desarrolladores de *software* de SSCA.

Resumen de los requerimientos de adiestramiento detectados, representados con los mismos códigos definidos en el capítulo V, en el apartado “requerimientos mínimos del desarrollador”.

**Tabla 7 Requerimientos de Adiestramiento**

|            |                           |
|------------|---------------------------|
| REQ-69-CMM | Procesos Organizacionales |
| REQ-2-CMM  | Procesos Organizacionales |
| REQ-3-CMM  | Procesos Organizacionales |
| REQ-70-CMM | Gestión de requisitos     |
| REQ-71-CMM | Gestión de requisitos     |
| REQ-72-CMM | Gestión de requisitos     |
| REQ-73-CMM | Gestión de requisitos     |
| REQ-74-CMM | Gestión de requisitos     |
| REQ-75-CMM | Planificación             |
| REQ-76-CMM | Planificación             |
| REQ-77-CMM | Seguimiento y control     |
| REQ-78-CMM | Seguimiento y control     |
| REQ-79-CMM | Gestión de configuración  |
| REQ-80-CMM | Gestión de configuración  |
| REQ-81-OPR | Desarrollo                |
| REQ-82-OPR | Desarrollo                |
| REQ-83-OPR | Desarrollo                |
| REQ-84-OPR | Desarrollo                |
| REQ-85-ARQ | Arquitectura y diseño     |
| REQ-86-ARQ | Arquitectura y diseño     |
| REQ-87-ARQ | Arquitectura y diseño     |
| REQ-88-ARQ | Arquitectura y diseño     |

Fuente: Elaboración Propia (2007).

En total son 22 requerimientos que representan un poco más del 50 por ciento de los requerimientos mínimos definidos en el capítulo V. Este es un porcentaje importante que evidencia que los desarrolladores no tienen los conocimientos necesarios para ejecutar los proyectos.

Por lo tanto, se recomienda la realización de un curso que cubra las áreas listadas en la Tabla 7, el cual permita abarcar todas las actividades asociadas al desarrollo de una aplicación, pudiendo brindar la oportunidad de adquirir experiencia previa a la realización de un proyecto real para los desarrolladores recién contratados y un fortalecimiento de los conocimientos para los que ya han formado parte de algún proyecto pero que aun requieren adiestramiento.

Para el curso se elaboró un diseño instruccional siguiendo el Modelo ADDIE, y se llegó hasta la etapa de diseño de dicho modelo. El diseño instruccional es detallado en el siguiente apartado.

## **2. FASE 3: DISEÑO**

El curso de adiestramiento tendrá la forma de “taller de diseño”, en el cual los participantes llevarán a cabo un proyecto de *software* al mismo tiempo que reciben la instrucción y adiestramiento pertinente, es decir, será teórico – práctico. La idea es que los desarrolladores dispongan de la oportunidad de enfrentarse a situaciones reales “planificadas” antes de enfrentarse a un proyecto real, de esta manera estarán mejor preparados para asumir los retos de proyectos más complejos. Adicionalmente, tendrán la oportunidad de ser guiados durante todo el proceso por expertos en las prácticas y procedimientos de SSCA, que en la realidad podría darse también, pero no de una forma tan amplia y didáctica como la que se proveerá en el curso.

Dichos proyectos serán llevados a cabo en estricto seguimiento de los estándares de SSCA en cuanto a procesos, procedimientos, prácticas, documentación, roles, y responsabilidades.

Las diferencias entre el proyecto ejecutado en el curso de entrenamiento y los proyectos usuales de SSCA son las siguientes:

- Los participantes pasarán entre el 25% - 50% del tiempo en las actividades de entrenamiento, adiestramiento e inducción.
- El proyecto a ser ejecutado durante el curso será un proyecto de “laboratorio”, especialmente diseñado para que sirva de práctica a los conocimientos, destrezas, y capacidades a ser impartidas. El proyecto será entregado en el estado de madurez en el cual los requisitos están identificados y documentados, hay un plan de proyecto, y el diseño arquitectónico está concluido, de manera que las tareas restantes sean sólo las de desarrollo, comprobación, e implantación.
- Los instructores prestarán apoyo a los equipos durante el desarrollo de sus proyectos.
- Habrá entregas (*Releases*), las cuales consistirán de presentaciones públicas y sesiones de discusión y crítica constructiva en las cuales participarán los instructores, los participantes, y otros miembros de la empresa como gerentes, auditores de calidad, arquitectos, y otros.
- Habrá *peer-reviews*, comprobaciones funcionales formales, y auditorías de calidad en un número mayor al usual en los proyectos ejecutados por SSCA, y en particular para proyectos tan pequeños.
- Durante el curso del proyecto se introducirán conflictos en la forma de cambios de requerimientos con el objetivo de causar stress sobre los diseños que no tengan una buena componentización, resolución de conflictos, y propiciar oportunidades para que se apliquen conocimientos de ingeniería de *software* importantes como patrones de diseño y pruebas unitarias.
- Se hará una evaluación continua sobre el progreso, los éxitos, y los fracasos del taller, la cual será reportada y analizada al culminar el proyecto.

A continuación se presenta el Diseño Instruccional dirigido a los desarrolladores de *software* de SSCA.

## DISEÑO INSTRUCCIONAL

### Objetivos:

Luego de cumplir con el curso los desarrolladores de *software* podrán ingresar a los equipos de proyectos con los conocimientos necesarios para construir una aplicación con los requerimientos de calidad de la empresa.

### Audiencia:

El presente curso está dirigido a desarrolladores de *software* con poca experiencia, que ingresen a la empresa o que hayan sido de reciente contratación.

### Estrategia de enseñanza:

La estrategia seleccionada es "*Drill and Practice*". En la cual se le asigna la práctica de una tarea hasta que se pruebe si logró el dominio del concepto y/o destreza.

### Evaluaciones:

Habrà una evaluación luego de cada inducción (Test de aprendizaje).

El desarrollo de la aplicación será constantemente validado, cuya calidad debe recibir una puntuación.

### Duración:

El curso tendrá una duración de 96 horas repartidas en 12 sesiones que serán impartidas una por semana.

### Contenido Programático:

**Tabla 8 Contenido Programático del Curso de Adiestramiento**

|                 |                                                                                                                        |
|-----------------|------------------------------------------------------------------------------------------------------------------------|
| Calidad en SSCA | <ul style="list-style-type: none"><li>• Objetivos de calidad de la organización</li><li>• Modelo de negocios</li></ul> |
|-----------------|------------------------------------------------------------------------------------------------------------------------|



|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                          | <ul style="list-style-type: none"> <li>• Introducción a CMMI</li> </ul>                                                                                                                                                                                                                                                                                                                                                                          |
| Unidades de SSCA         | <ul style="list-style-type: none"> <li>• Equipos de trabajo por proyectos (PRT)</li> <li>• Roles y responsabilidades de cada miembro (Énfasis en desarrolladores)</li> <li>• Manuales y procedimientos de la intranet</li> </ul>                                                                                                                                                                                                                 |
| Planificación            | <ul style="list-style-type: none"> <li>• Importancia de cumplir con los tiempos estimados e hitos</li> <li>• Que es un cronograma de actividades</li> <li>• Que tareas tiene asociada el desarrollo de un caso de uso. Estimación del tiempo de actividades</li> </ul>                                                                                                                                                                           |
| Arquitectura             | <ul style="list-style-type: none"> <li>• Metodología de SSCA</li> <li>• Programación Orientada a Objetos</li> <li>• Identificación, clasificación, y reparación de defectos. Técnicas de <i>Debugging</i></li> </ul>                                                                                                                                                                                                                             |
| Gestión de Requisitos    | <ul style="list-style-type: none"> <li>• Documentos asociados al proyecto referentes a Gestión de Requisitos</li> <li>• Requisitos funcionales</li> <li>• Requisitos no funcionales</li> <li>• UML, ¿Qué es UML?, Ventajas, tipos de diagramas, Comportamientos.</li> </ul>                                                                                                                                                                      |
| Seguimiento y control    | <ul style="list-style-type: none"> <li>• Uso de las herramientas de administración de tareas. Flujo del trabajo con la herramienta. Estatus posibles de un caso. Asignación de casos. Creación de casos</li> <li>• Uso de la herramienta de registro del tiempo. Actividades que debe registrar y donde. Importancia de registrar las actividades en el ítem correcto. Importancia de registrar el tiempo utilizado.</li> </ul>                  |
| Gestión de configuración | <ul style="list-style-type: none"> <li>• Procedimiento de sincronización con el repositorio de fuentes</li> <li>• ¿Qué es el CVS?</li> <li>• Sincronización de archivos con gran cantidad de cambios y modificados por más de un usuario</li> <li>• Comparación de archivos locales</li> <li>• Sincronización con más de un árbol</li> <li>• Generación de <i>Releases</i></li> <li>• Importancia de la integración continua y diaria</li> </ul> |
| Desarrollo               | <ul style="list-style-type: none"> <li>• Documento de diseño detallado. ¿Qué necesita diseño</li> </ul>                                                                                                                                                                                                                                                                                                                                          |

|  |                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p>detallado? Estructura del documento.</p> <ul style="list-style-type: none"> <li>• .Net. ¿Qué es el <i>Framework</i>?. Librerías. Clases y estructuras. Métodos. Alcance y niveles de acceso. <i>Garbage Collection</i>. ADO.Net. Interfaz de Usuario. Manejo de excepciones en .Net y Librería de <i>Logging</i>.</li> <li>• Pruebas unitarias. De límites, de tipos de datos, etc. Utilización de <i>Nunit</i></li> </ul> |
|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### 3. CRONOGRAMA DE ACTIVIDADES

El curso de adiestramiento será impartido un (1) día a la semana por doce (12) semanas. Cada renglón de la tabla 9 representa dos (2) horas. Se realizó esta distribución para permitir que los desarrolladores puedan también realizar actividades productivas en los cuatro (4) días restantes de la semana y además para que tengan tiempo de asimilar el material que se les imparte semanalmente.

**Tabla 9 Cronograma de ejecución del curso**

| Semana 1         | Semana 2              | Semana 3                 | Semana 4   | Semana 5 | Semana 6 |
|------------------|-----------------------|--------------------------|------------|----------|----------|
| Calidad en SSCA  | Arquitectura          | Seguimiento y control    | Desarrollo | Práctica | Práctica |
| Unidades de SSCA | Arquitectura          | Seguimiento y control    | Desarrollo | Práctica | Práctica |
| Planificación    | Gestión de Requisitos | Gestión de configuración | Desarrollo | Práctica | Práctica |
| Planificación    | Gestión de Requisitos | Gestión de configuración | Desarrollo | Práctica | Práctica |

mini proyecto

| Semana 7 | Semana 8 | Semana 9 | Semana 10 | Semana 11 | Semana 12 |
|----------|----------|----------|-----------|-----------|-----------|
| Práctica | Práctica | Práctica | Práctica  | Práctica  | Práctica  |
| Práctica | Práctica | Práctica | Práctica  | Práctica  | Práctica  |
| Práctica | Práctica | Práctica | Práctica  | Práctica  | Práctica  |
| Práctica | Práctica | Práctica | Práctica  | Práctica  | Práctica  |

diseño detallado

construcción y pruebas

Fuente: Elaboración propia (2007).

#### 4. ESTIMACIÓN DE COSTO DEL CURSO

El estimado de costo que se presenta en la tabla 10 es de clase V.

Se definió el uso de cuatro (4) instructores para los cursos teóricos. A cada uno le corresponderá impartir 2 cursos. Luego, al entrar en la fase práctica, sólo quedarán tres (3) instructores que se encargarán de apoyar a los desarrolladores. Esta estimación es para cinco (5) participantes.

Se estima que la preparación del material tomará 2 semanas por curso, las cuales están incluidas en las horas de los instructores.

**Tabla 10 Costo de Ejecución del Curso para 5 participantes**

| <b>Actividades</b>                           | <b>Horas</b> | <b>Costo por Hora (Bs.)</b> | <b>Total (Bs.)</b>   |
|----------------------------------------------|--------------|-----------------------------|----------------------|
| Preparación del Material                     | 640          | 45.000,00                   | 28.800.000,00        |
| <b>Inversión preparación</b>                 |              |                             | <b>28.800.000,00</b> |
| Curso por 12 días 5 participantes            | 480          | 25.000,00                   | 12.000.000,00        |
| Instructores por 12 días                     | 176          | 45.000,00                   | 7.920.000,00         |
| Infraestructura por 12 días                  | 96           | 100.000,00                  | 9.600.000,00         |
| <b>Inversión por curso (5 participantes)</b> |              |                             | <b>29.520.000,00</b> |
| <b>Inversión Total (1er curso)</b>           |              |                             | <b>58.320.000,00</b> |

|                                                                |  |  |                      |
|----------------------------------------------------------------|--|--|----------------------|
| <b>Inversión Total (cursos subsiguientes, 5 participantes)</b> |  |  | <b>29.520.000,00</b> |
|----------------------------------------------------------------|--|--|----------------------|

Fuente: Elaboración propia (2007).

Según se puede apreciar en la tabla 10, el curso tendrá un costo de preparación de material aproximado de Bs. 28.800.00, monto que será invertido sólo la primera vez que se dicte. Cada vez que se imparta el curso, el mismo tendrá un costo de Bs. 29.520.000 para 5 participantes.

Esta inversión, a pesar de ser considerable, se puede justificar tomando en cuenta las mejoras en tiempo y calidad. Es preferible realizar una inversión de tiempo y dinero en el personal en un curso de doce días, en lugar de tenerlo trabajando por uno, dos, tres o más meses, sin poder cobrar al cliente por estar resolviendo los problemas de calidad que se hayan detectado en el producto. Adicionalmente, el curso se planificó para ser realizado solo un día a la semana, permitiendo a los desarrolladores continuar con sus actividades operativas y no detener la producción.

## CAPÍTULO VII: EVALUACIÓN DE LOS RESULTADOS

Tabla 11 Objetivos versus cumplimiento

| Objetivo                                                                                                                                                                                             | Cumplimiento                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Definir los conocimientos, competencias y destrezas específicas mínimas del cargo de desarrollador de <i>Software</i> en SSCA                                                                        | En el capítulo IV se logra este objetivo, al realizar la Fase 1 de Diagnóstico. Al culminar esta fase se listan cuáles son los requisitos mínimos que debe poseer un desarrollador de <i>software</i> en SSCA, los cuales están divididos en áreas de conocimiento.                                                                                                                                                                    |
| Evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de <i>Software</i> en SSCA                                                                    | Este objetivo fue cumplido al realizar la revisión bibliográfica y las entrevistas a los líderes. La revisión bibliográfica permitió obtener las direcciones generales de las principales necesidades de adiestramiento de los desarrolladores y las entrevistas fueron más específicas en qué exactamente conocen y qué desconocen los desarrolladores de <i>software</i> . Esto se evidencia en el Capítulo IV, Fase de Diagnóstico. |
| Desarrollar un Diseño de Instrucción que permita cubrir las diferencias detectadas entre los conocimientos definidos y los reales diagnosticados para los desarrolladores de <i>Software</i> en SSCA | Al comparar los conocimientos actuales del desarrollador con los requisitos mínimos de conocimiento, se pudo detectar cuáles son las necesidades de adiestramiento. A partir de esas necesidades y tomando en cuenta el perfil profesional de la audiencia se                                                                                                                                                                          |

|                                                                                                       |                                                                                                                                                                                                                    |
|-------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                       | realizó el desarrollo del diseño de instrucción para cubrir las necesidades de adiestramiento de los desarrolladores. El diseño instruccional puede ser revisado en el Capítulo VI, Fase 2                         |
| Estimar el costo de inversión y operación del Diseño de Instrucción clase 5.                          | Una vez hecho el diseño se puede estimar qué costo tendrá el ejecutarlo. Se hizo una estimación clase V tomando como base 4 instructores y 5 estudiantes. El costo estimado puede verse en el Capítulo VI, Fase 3. |
| Formular un Diseño de Instrucción para el adiestramiento del Desarrollador de <i>Software</i> en SSCA | Habiendo cumplido con todos los objetivos específicos que se plantearon en el Capítulo I, queda completado el objetivo general del presente documento.                                                             |

Fuente: Elaboración propia (2007).

## CAPÍTULO VIII: CONCLUSIONES Y RECOMENDACIONES

### 1. CONCLUSIONES

En SSCA, se estima que para que los procesos organizacionales sean ejecutados con calidad, necesitan personas adiestradas para poder asumir competentemente cada tarea. A nivel de los desarrolladores, se ha diagnosticado déficit de conocimientos y habilidades que no puede ser menospreciado.

El curso de adiestramiento diseñado aportará los requerimientos específicos para cubrir ese déficit, ya que fueron alcanzados los objetivos planteados durante la realización de esta investigación, la cual permitió científicamente *definir los conocimientos, competencias y destrezas específicas mínimas del cargo de desarrollador de Software en SSCA*, así como también *evaluar los conocimientos, competencias y destrezas específicas que posee actualmente el desarrollador de Software en SSCA*, lo cual llevó a *desarrollar un Diseño de Instrucción que permita cubrir las diferencias detectadas entre los conocimientos definidos y los reales diagnosticados para los desarrolladores de Software en SSCA*.

El proceso de diagnóstico fue llevado de manera satisfactoria, el principal insumo para el análisis fue el provisto por los encuestados, en este caso los líderes que supervisan a los desarrolladores, cuya experiencia fue evidenciada al momento de realizar las entrevistas, en su medio laboral.

Mediante el análisis de los resultados se pudo inferir que las necesidades de adiestramiento del personal que actualmente labora en SSCA son altas, presentan déficit elevado en cuanto a los conocimientos, habilidades y competencias que debe poseer el desarrollador. Esto indica que los problemas de calidad y de



tiempo son producto del nivel de capacitación y adiestramiento de quienes están ejecutando los proyectos, en este caso los desarrolladores.

Por lo tanto, la interrogante formulada que dio origen a este trabajo de investigación: *“¿se podría resolver el problema si se facilita a los desarrolladores de SSCA un curso de entrenamiento intensivo, que en muy pocas semanas permita que los participantes adquieran y/o refuercen el conjunto de conocimientos y destrezas requeridos para asumir sus competencias en el modelo de negocios de la empresa?”*, puede ser respondida afirmativamente, debido a que al adiestrar al personal se puede garantizar que en la etapa de construcción se minimizarán los problemas, que el mismo desarrollador estará en capacidad de identificar oportunamente el riesgo de error, que los desarrolladores podrán desempeñar su rol dentro del equipo del proyecto, con óptimo nivel de productividad.

Con respecto al diseño de instrucción, el trabajo es completo, ya que abarca todos los requerimientos identificados.

La estrategia de enseñanza empleada en el diseño es teórico-práctica, con mayor peso en la práctica porque se estimó primordial que los participantes sean preparados para afrontar problemas reales, en tiempo real.

En cuanto al objetivo general “estimar el costo de inversión y operación del Diseño de Instrucción clase 5” se alcanzó satisfactoriamente. Su relevancia está dada porque permite asegurar la factibilidad del Diseño, en cuanto a inversión.

Por otra parte, con respecto a la pregunta de investigación: *“¿Qué contenido programático debe cubrir el diseño de instrucción para adiestrar a los desarrolladores software en SSCA?”*, la Tabla 8 del capítulo V de resultados del proyecto, detalla el contenido programático del curso de adiestramiento del desarrollador de *software* en SSCA, adiestramiento sobre Procesos

Organizacionales, Gestión de requisitos, Planificación, Seguimiento y control, Gestión de configuración, Gestión de calidad, Desarrollo y Arquitectura y diseño, básicos desde el punto de vista del conocimiento teórico y práctico de todo desarrollador, especialmente en el modelo de negocios de SSCA.

## **2. RECOMENDACIONES**

Aplicar un piloto con un equipo de participantes para confirmar que se hayan incluido todos los aspectos requeridos, validar la eficiencia del curso y ajustar los posibles detalles que se detecten.

Una vez validado, se debe informar a los líderes para la planificación de los cursos.

Considerar la participación de todo el personal de desarrolladores operativos en SSCA. Durante la ejecución del curso, evaluarlos objetivamente y en base a los resultados tomar decisiones estructurales.

Con respecto a las áreas cubiertas por el curso, hacer énfasis en los aspectos del contenido dependiendo de los participantes.

Revisar y actualizar periódicamente el contenido programático del diseño de instrucción, ajustándolo a las necesidades de SSCA.

Mantener el diseño “a la mano” de los desarrolladores para eventuales consultas.

## BIBLIOGRAFÍA

- Aguilar Sanz, Joel. *El Diseño de Instrucción*. Recuperado de la World Wide Web:  
<http://ares.unimet.edu.ve/programacion/psfase3/modII/biblio/Dlevolucion.pdf> el 28 de Marzo de 2007.
- López C., Nelson (2002). *Una opción metodológica para la calidad*. Caracas: Solmaria Suniaga.
- Meighan, Michael. (1992). *Programas de Inducción Entrenamiento, Diseño y Ejecución*. Bogota: Fondo Editorial Legis.
- Microsoft. Ingeniería de Software. Recuperado de la World Wide Web:  
<http://www.microsoft.com/spanish/MSDN/estudiantes/ingsoft/default.asp> el 28 de Marzo de 2007.
- Palacios A, L. (2000). *Principios Esenciales para Realizar Proyectos. Un Enfoque Latino*. (2da Edición), Caracas: PDVSA/CIED/UCAB/OTEPi.
- Project Management Institute. (2004). *Guía de los fundamentos de la dirección de proyectos*. (3ª edición). Newton Square, USA: Autor
- Project Management Institute. (2004). *Código de ética de los miembros del PMI*. Newton Square, USA: Autor
- Rísquez, G.; Fuenmayor, E. y Pereira, B. (1999). *Metodología de la Educación. Manual Teórico Práctico*. Maracaibo: S/edt.
- Sabino, C. (2000). *El proceso de Investigación*. Caracas: Editorial Panapo.
- Sampieri, R., Collado, C. y Lucio, P. (2004). *Metodología de la Investigación*. (Tercera edición). México: McGraw-Hill/Interamericana Editores, S.A. de C.V.
- SSCA. (2005). *Ciclo de Vida del Software (Versión 2.0)*. Caracas: Autor.
- SSCA. (2006). *Manual de Gestión de Operaciones (Versión 3.0)*. Caracas: Autor.
- SSCA. (2006). *Manual de Gestión Dirección de Calidad (Versión 4.0)*. Caracas: Autor.

- SSCA. (2006). *Manual de Gestión Dirección de Arquitectura (Versión 2.3)*. Caracas: Autor.
- SSCA. (2006). *PGC-OP-01 Planificación del Proyecto (Versión 3.0)*. Caracas: Autor.
- SSCA. (2006). *PGC-OP-02 Seguimiento y Control del Proyecto (Versión 3.0)*. Caracas: Autor.
- SSCA. (2006). *PGC-OP-03 Gestión de la Configuración (Versión 2.0)*. Caracas: Autor.
- SSCA. (2006). *PGC-OP-04 Gestión de Requisitos (Versión 3.1)*. Caracas: Autor.
- SSCA. (2006). *PGC-OP-05 Control de Calidad de Proyectos (Versión 4.1)*. Caracas: Autor.
- SSCA. (2006). *PGC-OP-06 Diseño y Desarrollo Técnico (Versión 3.2)*. Caracas: Autor.
- SSCA. (2006). *PGC-OP-08 Pruebas Funcionales (Versión 3.0)*. Caracas: Autor.
- SSCA. (2006). *PGC-OP-09 Implantación (Versión 3.3)*. Caracas: Autor.
- Universidad Pedagógica Experimental Libertador, Vicerrectorado de Investigación y Postgrado. (2001). *Manual de Trabajos de Grado de Especialización y Maestría y Tesis Doctorales*. Caracas: FEDUPEL.
- Urrutia Carolina (sin fecha). **¿Qué es el Capability Maturity Model (CMM®)?**. PM Value. Recuperado de la World Wide Web: <http://www.pmvalue.com.ar/NewsLetters/Newsletter%202004-01%20CMM.pdf> el 15 de Junio de 2007.
- Wikipedia. CVS. Recuperado de la World Wide Web: <http://es.wikipedia.org/wiki/CVS> el 28 de Marzo de 2007.
- Wikipedia. UML. Recuperado de la World Wide Web: [http://es.wikipedia.org/wiki/Lenguaje\\_Unificado\\_de\\_Modelado](http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado) el 28 de Marzo de 2007.

- Wikipedia. *Programación Orientada a Objetos*. Recuperado de la World Wide Web: [http://es.wikipedia.org/wiki/Orientado\\_a\\_objetos](http://es.wikipedia.org/wiki/Orientado_a_objetos) el 28 de Marzo de 2007.
- Wikipedia. *Patrones de diseño*. Recuperado de la World Wide Web: [http://es.wikipedia.org/wiki/Patrones\\_de\\_dise%C3%B1o](http://es.wikipedia.org/wiki/Patrones_de_dise%C3%B1o) el 28 de Marzo de 2007.
- Wikipedia. *Manejo de Excepciones*. Recuperado de la World Wide Web: [http://es.wikipedia.org/wiki/Manejo\\_de\\_excepciones](http://es.wikipedia.org/wiki/Manejo_de_excepciones) el 28 de Marzo de 2007.
- Wikipedia. *Convenios de Nombre*. Recuperado de la World Wide Web: [http://en.wikipedia.org/wiki/Wikipedia:Naming\\_conventions](http://en.wikipedia.org/wiki/Wikipedia:Naming_conventions) el 28 de Marzo de 2007.
- Wikipedia. *Reutilización de Código*. Recuperado de la World Wide Web: [http://es.wikipedia.org/wiki/Reutilizaci%C3%B3n\\_de\\_c%C3%B3digo](http://es.wikipedia.org/wiki/Reutilizaci%C3%B3n_de_c%C3%B3digo) el 28 de Marzo de 2007.
- Wikipedia. *RUP*. Recuperado de la World Wide Web: [http://es.wikipedia.org/wiki/Proceso\\_Unificado\\_de\\_Rational](http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational) el 28 de Marzo de 2007.
- Wikipedia. *PRUEBAS UNITARIAS*. Recuperado de la World Wide Web: [http://es.wikipedia.org/wiki/Prueba\\_unitaria](http://es.wikipedia.org/wiki/Prueba_unitaria) el 28 de Marzo de 2007.
- Yukavetsky, Gloria J. (2004). *La Elaboración de un Módulo Instruccional*. Preparado para el Centro de Competencias de la Comunicación Universidad de Puerto Rico en Humacao. Recuperado de la World Wide Web: [http://www.ccc.uprh.edu/download/modulos/CCC\\_LEDUMI.pdf](http://www.ccc.uprh.edu/download/modulos/CCC_LEDUMI.pdf) El 25 de Junio de 2007.

## **ANEXOS**

## 1. ANEXO 1 - ENTREVISTA

### Instrumento Guía

#### ENTREVISTA

#### SOBRE DESTREZAS DE LOS DESARROLLADORES

El propósito de esta entrevista es detectar cuál es el nivel de los desarrolladores de la empresa, por favor sea objetivo y lo más abierto posible para sacar el máximo provecho.

1. ¿Cuántos desarrolladores tienes a tu cargo?
2. ¿Qué nivel de capacitación poseen?
3. ¿Los desarrolladores entienden los procesos de la empresa y su rol dentro del equipo del proyecto?
4. ¿Los desarrolladores entienden todos los requisitos del sistema (criterios de aceptación, rendimiento, acuerdos de codificación)?
5. ¿Los desarrolladores manejan la Metodología de Desarrollo de la Organización y/o alguna otra metodología, CMM, RUP, MSF, etc.?
6. ¿Los desarrolladores conocen de patrones de diseño?
7. ¿Los desarrolladores han presentado problemas con los documentos de diseño detallados?
8. ¿Qué nivel de conocimiento del Lenguaje de programación poseen?
9. ¿Tienen un correcto manejo de excepciones?
10. ¿Tienen algún problema con las pruebas, unitarias y funcionales?
11. ¿Cómo es realizado el proceso de corrección de errores?
12. ¿Tienen problemas con el control de versiones y la integración continua?
13. ¿Realizan de manera acorde el uso de las herramientas de control de tareas?
14. ¿Al llevar a cabo proyectos se ha producido algún retraso en la Fase de Construcción?
15. ¿Considera que el tiempo planificado les resulta suficiente para realizar las tareas de desarrollo?
16. ¿Está el desarrollador de acuerdo y comprometido con los tiempos de entrega?
17. ¿Considera que su producto refleja la calidad requerida por el cliente?