Un método para medir la productividad del equipo de pruebas en la estimación del esfuerzo de pruebas de software

A method for measuring software test team productivity

Ing. Diana Maria Torres Ricaurte

Universidad Nacional de Colombia

Facultad de Minas, Departamento de Ciencias de la Computación y de la Decisión

Medellín, Colombia

Junio de 2013

Un método para medir la productividad del equipo de pruebas en la estimación del esfuerzo de pruebas de software

A method for measuring software test team productivity

Diana Maria Torres Ricaurte

Tesis de investigación presentada como requisito parcial para optar al título de:

Magíster en Ingeniería de Sistemas

Director:

Ph.D. Carlos Mario Zapata Jaramillo

Línea de Investigación:

Calidad de software

Universidad Nacional de Colombia

Facultad de Minas, Departamento de Ciencias de la Computación y de la Decisión

Medellín, Colombia

2013

A Dios por darme la vida y tantas bendiciones cada día. A mi familia por que son el apoyo en todos los proyectos que emprendo. A mi esposo Andrés Julián por ser ante todo el amigo que habla con sinceridad y logra hacerme reflexionar. A Andrés Tomás por que es el hijo amoroso y mi compañero incondicional. A mi madre Francis por que es el ejemplo de mujer emprendedora y cariñosa. A mis hermanos Luisa Fernanda y Carlos Andrés por crecer a mi lado y brindarme todo su amor. A mi sobrina Sofía por darme alegrías con su espontaneidad. A mi padre Carlos Iván porque sus enseñanzas siguen guiando mi vida.

AGRADECIMIENTOS

Agradezco ante todo al profesor Carlos Mario Zapata quien con paciencia y ejemplo logró formar de mí una investigadora.

A todo el equipo de la empresa SEQUAL S.A. por toda la colaboración que brindaron en el proceso de validación de esta Tesis.

A los panelistas quienes aportaron todo su conocimiento y experiencia en el desarrollo del estudio Deplhi.

A los docentes y compañeros de curso quienes diariamente con sus críticas hicieron aportes que ayudaron a pulir este trabajo.

Al docente, y mi esposo, Andrés Julián Saavedra por que sus comentarios y nuestras conversaciones fueron aportes muy importantes en el desarrollo y culminación de este trabajo.

Por ultimo, a todas las personas que de una u otra forma contribuyeron en el desarrollo de este trabajo.

RESUMEN

La productividad del equipo de pruebas es la relación entre un número de unidades de salida y el esfuerzo necesario para ejecutarlas. Los métodos para la estimación del esfuerzo requerido en las pruebas incluyen un factor de productividad del equipo de pruebas, pero ese factor se calcula con datos históricos que generalmente no poseen todas las empresas. Estos métodos, además, no incluyen las características que pueden influir en el cálculo de la productividad. Por ello, en esta Tesis se propone un método para medir la productividad del equipo de pruebas, usando como unidades de salida el número de casos de prueba ejecutados y como unidad de entrada el esfuerzo requerido para ejecutarlo. Este método contribuye a obtener datos de las características que influyen sobre la productividad, para que los coordinadores de pruebas puedan definir índices, basados en estos datos, con los cuales se puedan tomar decisiones acerca del esfuerzo de pruebas. El método se condensa en unos formatos que permiten la recolección de la información desde los actores que la originan. Para validar el método se recolectó información de proyectos de prueba en ejecución que pertenecen a una empresa de calidad de software dedicada a la pruebas.

Palabras Clave: Equipo de prueba, productividad, método, medición

ABSTRACT

Test team productivity is expressed as the ratio of a unit of output to the effort used to produce it. Test effort estimation methods include a test team productivity factor, but such a factor is estimated with historical data—and software companies usually lack such quality and productivity data. Additionally, test effort estimation methods are not based on test process factors. For this reason, in this Thesis I propose a method for measuring software test team productivity. In this method, test team productivity is determined by the number of executed test cases and the effort required to execute them. The proposed method provides a formal procedure to collect data about factors affecting test productivity, so test manager can use the measurement results for making decisions about test productivity. The method includes forms to collect information from stakeholders. The proposed method is validated in a software quality company by collecting information about its test execution projects.

Keywords: Test team, productivity, method, measure

TABLA DE CONTENIDO

1 INTRODUCCIÓN 1

1.1 JUSTIFICACIÓN 1

1.2 PLANTEAMIENTO DEL PROBLEMA 1

1.3 OBJETIVO GENERAL 2

1.4 OBJETIVOS ESPECÍFICOS 3

1.5 METODOLOGÍA A UTILIZAR 3
1.5.1 Exploración 3
1.5.2 Análisis 4
1.5.3 Desarrollo del método de medición 4
1.5.4 Validación 4

1.6 ALCANCE DEL TRABAJO 4

1.7 ESTRUCTURA DE LA TESIS 5

2 MARCO TEÓRICO 6

2.1 PRUEBAS DE SOFTWARE 6

2.2 CONCEPTOS DE MEDICIÓN DEL SOFTWARE 8
2.2.1 Medidas de software 8
2.2.2 Entidades de software 8
2.2.3 Atributos de calidad 8
2.2.4 Métricas de software 8
2.2.4 Unidad de medición 9
2.2.5 Escala de medición 9
2.2.6 Tipo de métrica 9
2.2.7 Estimación 10
2.2.8 Métricas de pruebas de software 10
2.2.9 Método de medición 10

2.3 PROCESO DE MEDICIÓN DEL SOFTWARE 11

2.4 ESTIMACIONES DEL ESFUERZO DE PRUEBAS 12
2.4.1 Estimaciones basadas en el esfuerzo de desarrollo del software 12
2.4.2 Estimaciones basadas en el tamaño del software 13
2.4.3 Estimaciones basadas en los casos de prueba 14
2.4.4 Estimaciones basadas en las actividades de pruebas 15

2.5 PRODUCTIVIDAD 16

2.6 ESQUEMAS PRECONCEPTUALES 16

2.7 LA TÉCNICA DELPHI 17

3. ANTECEDENTES 19

4. SOLUCIÓN PROPUESTA 29

4.1 OBJETIVOS DEL MÉTODO DE MEDICIÓN DE PRODUCTIVIDAD DEL EQUIPO DE PRUEBAS 29

4.2 DISEÑO Y SELECCIÓN DEL METAMODELO DE LA PRODUCTIVIDAD DEL EQUIPO DE PRUEBAS 30
4.2.1 Características del proyecto de pruebas 30
4.2.2 Características de la aplicación de software 30
4.2.3 Características de la administración del proyecto de pruebas 30
4.2.4 Validación de las características que afectan la productividad del equipo de pruebas 31

4.3 CARACTERIZACIÓN DE LOS CONCEPTOS DE PRODUCTIVIDAD DEL EQUIPO DE PRUEBAS 34
4.3.1 Los casos de prueba 35
4.3.2 Los probadores 36
4.3.3 El equipo de pruebas 40
4.3.4 El ambiente de pruebas 41
4.3.5 Estabilidad de los requisitos de software 42
4.3.6 Nivel de complejidad de la programación 42
4.3.7 Grado de innovación del proyecto de software 43
4.3.8 Factores limitantes 44
4.3.9 Participación de los usuarios 45
4.3.10 Criticidad 45
4.3.11 Grado de innovación del proyecto de pruebas 46
4.3.12 Complejidad de los datos 46
4.3.13 Complejidad de la organización 47
4.3.14 Concurrencia de desarrollo 48

4.4 RECOPILACIÓN DE LA INFORMACIÓN ACERCA DE PRODUCTIVIDAD DEL EQUIPO DE PRUEBAS 49

4.5 DEFINICIÓN DE LOS PASOS DEL MÉTODO DE MEDICIÓN DE LA PRODUCTIVIDAD DEL EQUIPO DE PRUEBAS. 51

5. VALIDACIÓN 69

6. CONCLUSIONES Y TRABAJO FUTURO 116

6.1 CONCLUSIONES 116

6.2 TRABAJO FUTURO 117

7. REFERENCIAS 118

LISTADO DE FIGURAS

Figura 1: Proceso de pruebas 6
Figura 2: Modelo detallado del proceso de medición 11
Figura 3: Modelo de calificación de los probadores 26
Figura 4: Esquema preconceptual del concepto de caso de prueba 36
Figura 5: Esquema preconceptual del concepto de probador 37
Figura 6: Esquema preconceptual del concepto de equipo de pruebas 40
Figura 7: Esquema preconceptual del concepto de ambiente de pruebas 42
Figura 8: Esquema preconceptual de los conceptos de estabilidad de los requisitos de software, nivel de complejidad de la programación y grado de innovación del proyecto de software 43
Figura 9: Esquema preconceptual del concepto de riesgo 44
Figura 10: Esquema preconceptual del concepto de participación de los usuarios 45
Figura 11: Esquema preconceptual del concepto de criticidad 46
Figura 12: Esquema preconceptual del concepto de grado de innovación del proyecto de pruebas 46
Figura 13: Esquema preconceptual del concepto de complejidad de los datos 47
Figura 14: Esquema preconceptual del concepto de complejidad de los datos 48
Figura 15: Esquema preconceptual del concepto de complejidad de los datos 49
Figura 16: Esquema preconceptual del método de medición de la productividad del equipo de pruebas de software 50

LISTADO DE TABLAS

Tabla 1: Simbología de los Esquemas Preconceptuales 16
Tabla 2: Escala nominal del TPA 20
Tabla 3: Resumen de las técnicas de estimación de la productividad de pruebas de software 26
Tabla 4: Resultado estudio Delphi para identificar las características que afectan la productividad del equipo de pruebas 31
Tabla 5: Atributos de los conceptos que caracterizan la productividad del equipo de pruebas 33
Tabla 6: Formato del método de medición de la productividad del equipo de pruebas para el gerente de proyecto 47
Tabla 7: Formato del método de medición de la productividad del equipo de pruebas para el coordinador de pruebas 49
Tabla 8: Formato del método de medición de la productividad del equipo de pruebas para el probador 58
Tabla 9: Datos del método para medir la productividad del equipo de pruebas-Gerente de proyecto 66
Tabla 10: Datos del método para medir la productividad del equipo de pruebas-Coordinador de pruebas 68
Tabla 11: Datos del método para medir la productividad del equipo de pruebas-Probador No.1 73
Tabla 12: Datos del método para medir la productividad del equipo de pruebas-Probador No.2 77
Tabla 13: Datos del método para medir la productividad del equipo de pruebas-Gerente de proyecto No.2 84
Tabla 14: Datos del método para medir la productividad del equipo de pruebas-Coordinador de pruebas 86
Tabla 15: Datos del método para medir la productividad del equipo de pruebas-Probador No.3 93
Tabla 16: Datos del método para medir la productividad del equipo de pruebas-Probador No.4 98
Tabla 17: Datos del método para medir la productividad del equipo de pruebas-Probador No.5 101

Tabla 18: Datos del método para medir la productividad del equipo de pruebas-Probador No.6 105

1. INTRODUCCIÓN

1.1 Justificación

Las pruebas de software constituyen la mitad del costo total del proyecto de desarrollo de software (Harrold, 2000). Por esta razón, es importante cumplir con los objetivos de las pruebas dentro del tiempo y el presupuesto estimados. Con el objetivo de planear de manera acertada la asignación de los recursos y el presupuesto, los gestores de proyectos usan métricas de software en el proceso de pruebas (Farooq, Quadri, & Ahmad, 2011). Estas métricas, además, permiten obtener valores tangibles de las características del proceso de pruebas que servirán para estimar valores de las medidas en proyectos futuros o incompletos (Jones, 2008).

Una métrica de la productividad del equipo de pruebas de software permite a los gestores de proyectos controlar la asignación del personal y mejorar la estimación del tiempo y el presupuesto necesarios para reducir los costos de las pruebas. Por esta razón, es importante tener un método que proponga una manera sistemática de medir la productividad del equipo de pruebas de software basado en las características particulares del equipo de pruebas.

El método de medición de la productividad del equipo de pruebas debe permitir una caracterización numérica de los atributos que representan características del equipo de pruebas que afectan la productividad y, además, debe ser usable para los diferentes actores que participan en el proceso de pruebas.

1.2 Planteamiento del problema

Los métodos que estiman el esfuerzo de pruebas usan diferentes criterios, a saber: el esfuerzo de desarrollo (Nageswaran, 2001), el tamaño del software (Nageswaran, 2001; Jones, 2008), la enumeración de los casos de prueba (Aranha & Borba, 2007; Xiaochum, Bo, Li, Junbo, & Lu, 2008) y las actividades de

prueba (Abhishek, Kumar, Vitta, & Srivastava, 2010; Mizuno, Shigematsu, Takagi, & Kikuno, 2002). Tales criterios se basan en características del proceso de pruebas como son: la complejidad de los casos de prueba, la experiencia y el conocimiento de los probadores y el factor de productividad, entre otros. El factor de productividad representa el número de horas empleadas para ejecutar una unidad de prueba (Veenendaal & Dekkers, 1999; Aranha & Borba, 2009).

Los métodos que usan el factor de productividad lo obtienen de información en bases datos históricas o con base en el criterio del coordinador de pruebas (Veenendaal & Dekkers, 1999; Aranha & Borba, 2009; Xiaochum, Bo, Li, Junbo, & Lu, 2008). Sin embargo, las bases de datos históricas no están disponibles en todas las organizaciones. Tampoco es seguro que, si están disponibles, contengan información acerca de los atributos necesarios para realizar el cálculo de la productividad. Por otra parte, la subjetividad presente en el juicio del experto condiciona la corrección de la medición.

Por esta razón, en esta Tesis se propone un método para medir de forma estructurada la productividad de un equipo de pruebas. Este método proporciona una lista de características mínimas que representan la productividad del equipo de pruebas y las relaciones entre estas características. Mediante los valores de la productividad del equipo de pruebas se pueden ejercer acciones de mejora en el proceso de pruebas. El método, además, proporciona una lista de chequeo acerca de la información de los proyectos actuales que se debe recolectar en bases de datos históricas con el objetivo de preparar la estimación de proyectos futuros.

1.3 Objetivo General

Proponer un método para medir la productividad del equipo de pruebas en la estimación del esfuerzo de pruebas.

1.4 Objetivos Específicos

- Identificar, en las estimaciones del esfuerzo de pruebas existentes, las características que afectan la productividad del equipo de pruebas.
- Categorizar las características que afectan la productividad del equipo de pruebas de acuerdo con el nivel de afectación que tienen dentro de la estimación del esfuerzo de pruebas.
- Integrar en un método de medición las características de la productividad del equipo de pruebas.
- Validar el método de medición con datos obtenidos de la industria.

1.5 Metodología a utilizar

La metodología que se usa para desarrollar esta Tesis de Maestría sigue cuatro etapas, en las cuales se cubren los pasos propuestos por Jacquet y Abran (1997) para el diseño de un método de medición:

1.5.1 Exploración

En esta etapa se tienen dos metas principales. La primera es adquirir todo el marco conceptual que sustenta la temática que se aborda en este trabajo. Los temas que se deben comprender claramente son: el proceso de medición del software, los criterios de estimación del esfuerzo en pruebas de software y el criterio de productividad.

La segunda meta consiste en revisar la literatura para identificar aquellos métodos de estimación del esfuerzo de pruebas de software que tienen en cuenta el factor de productividad, comprender la forma en que estos métodos obtienen este factor de productividad y reconocer las características que representan la productividad del equipo de pruebas.

1.5.2 Análisis

En esta etapa, la meta es, con base en la revisión de la literatura que se obtuvo en la etapa anterior, caracterizar el concepto de productividad del equipo de pruebas de software. Se deben definir los objetivos que permitan plantear el método de medición, el rol de la persona para quien se diseña el método de medición y el conjunto mínimo de características y sus atributos que representan el concepto de productividad.

1.5.3 Desarrollo del método de medición

En esta etapa, el objetivo es establecer las reglas de asignación numéricas para los atributos que se definieron en la etapa anterior y las relaciones entre estos atributos. Posteriormente, se integran los atributos y las relaciones en un modelo que representa la productividad del equipo de pruebas de software. Finalmente, mediante una herramienta conocida como panel Delphi se someten a evaluación de expertos los atributos que se propusieron en el modelo y se obtiene consenso acerca del nivel de impacto que tienen sobre la productividad del equipo de pruebas.

1.5.4 Validación

En esta etapa se valida el método propuesto, cotejando los valores obtenidos de mediciones de la productividad del equipo de pruebas, realizadas con datos de una empresa privada de *testing*, con la productividad del equipo de pruebas *in situ***.**

1.6 Alcance del trabajo

El método de medición de la productividad del equipo de pruebas permite medir la productividad de forma estructurada, incluyendo las características mínimas que afectan la productividad. El método brinda a los coordinadores de pruebas directrices acerca del tipo de información que se debe almacenar en bases de

datos históricas, de manera que sean útiles para la estimación de la productividad en proyectos futuros.

El método de medición se valida sometiendo a verificación los valores de la productividad del equipo de pruebas medidos mediante el método propuesto.

1.7 Estructura de la Tesis

Esta Tesis se organiza de la siguiente manera: en el Capitulo 2 se presenta el marco teórico relacionado con las pruebas de software, las mediciones de software, la estimación del esfuerzo de pruebas de software y la productividad; en el Capitulo 3 se presentan los antecedentes; en el Capitulo 4 se presenta el método de medición de la productividad del equipo de pruebas propuesto; en el Capitulo 5 se muestran los resultados de la validación y, finalmente, en el Capitulo 6 se presentan las conclusiones y el trabajo futuro que se deriva de este trabajo.

2 MARCO TEÓRICO

2.1 Pruebas de software

El propósito de las pruebas de software es identificar y corregir defectos de los productos de software (Myers, Sandler, Badgett, & Thomas, 2004). Las actividades del proceso de pruebas se distribuyen en las siguientes etapas: preparación, planeación, ejecución, análisis y seguimiento (Tian, 2005). La mayor parte de las actividades se centra en la ejecución de las pruebas (véase la Figura 1). Esta etapa consiste en ejecutar el software, observar su funcionamiento y comunicar las observaciones. Con este objetivo, la principal herramienta que se usa en la etapa de ejecución la constituyen los casos de prueba (Tian, 2005).

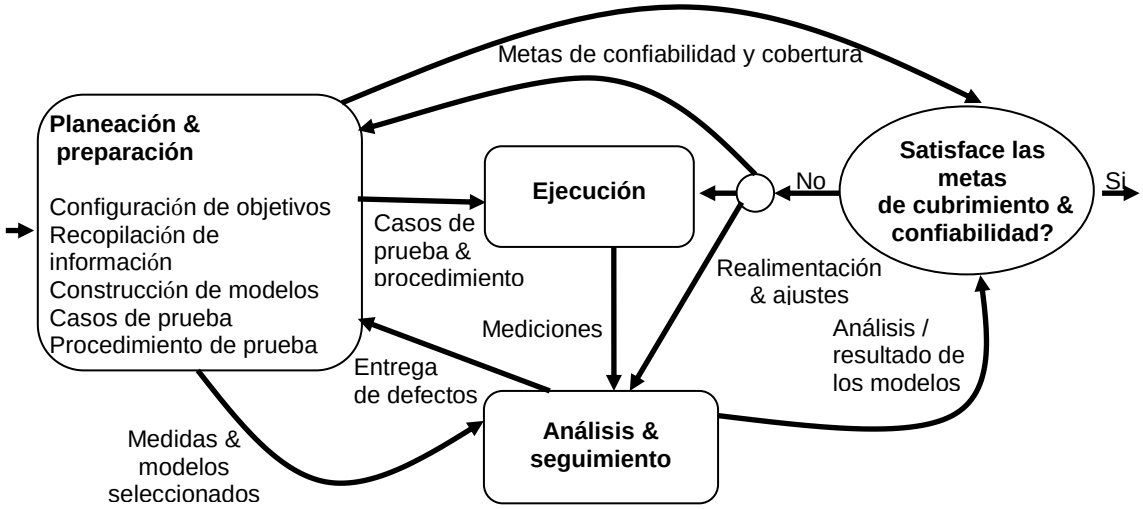


Figura 1: Proceso de pruebas (Tian, 2005)

Los casos de prueba son artefactos que permiten evaluar de forma estructurada el grado de conformidad de la aplicación de software con sus especificaciones, sus requisitos o los riesgos del negocio (Zapata, Ochoa, & Cardona, 2011; Everett & McLeod, 2007). En su forma básica, un caso de prueba consiste en un conjunto de entradas y los resultados que se espera generar con ellos, donde las entradas son los parámetros con los que se configura la aplicación que se va a probar y los resultados que se esperan son descripciones de las salidas que se prevé obtener. La ejecución de un caso de prueba consiste en seguir los lineamientos del caso de

prueba y diligenciar un documento donde se reporta el proceso y los resultados de la ejecución del caso de prueba (Tian, 2005). Un conjunto de casos de prueba que se organizan bajo algún criterio y se ejecutan en una secuencia, se conoce como *suite* de pruebas. Generalmente, el criterio que se usa para organizar los casos de prueba en una *suite* es el enfoque de la prueba.

Dependiendo del enfoque, las pruebas se pueden clasificar en pruebas estáticas, funcionales, estructurales y de rendimiento. Las pruebas estáticas tienen como objetivo reducir los defectos del software mediante la revisión de la documentación. Las pruebas funcionales tienen como objetivo validar que el comportamiento de la aplicación de software corresponde a la funcionalidad del negocio reportada en los documentos de requisitos y de especificaciones de la aplicación. Las pruebas estructurales tienen como objetivo validar la plataforma sobre la cual se instala la aplicación de software. Las pruebas de rendimiento tienen como objetivo validar el tiempo de respuesta bajo cargas de trabajo y configuraciones controladas (Everett & McLeod, 2007).

Personas con conocimientos acerca de las técnicas y procesos de pruebas deben ejecutar y administrar las actividades de prueba. Los actores involucrados en el proceso de pruebas se organizan en equipos de prueba con alguna estructura, siguiendo un modelo horizontal o un modelo vertical. En el modelo horizontal, el equipo de prueba ejecuta un mismo tipo de prueba para diferentes productos de software de la organización. En el modelo vertical, el equipo de prueba ejecuta una o más tareas de prueba en torno de un producto de software (Tian, 2005). Los roles que se reconocen en estos equipos se relacionan con las responsabilidades de los actores, así: el coordinador de pruebas, el diseñador de pruebas y el probador. El coordinador es quien lidera el equipo, gestiona y administra los recursos. El diseñador de pruebas es quien se encarga de diseñar los casos de prueba. Por último, el probador es quien se encarga de ejecutar las pruebas; en este rol también pueden participar los usuarios finales, los programadores y los diseñadores de software (Zapata et al., 2011).

2.2 Conceptos de medición del software

2.2.1 Medidas de software

Las medidas de software garantizan que, durante el ciclo de vida del desarrollo, las entidades de software cumplen las características de calidad. Medir la calidad del software implica verificar que posee el conjunto esperado de atributos de calidad (IEEE std. 1061, 1998). La medición de un atributo de una entidad es el proceso de caracterizar el atributo en términos de números o símbolos (Habra, Abran, Lopez & Sellami, 2008).

2.2.2 Entidades de software

Las entidades de software, según la norma *ISO/IEC/IEEE 24765:2010,* representan los objetos del mundo real asociados con los productos, procesos y recursos del desarrollo del software que se pueden medir (Habra *et al.*, 2008), y sobre los cuales se almacena información porque se consideran relevantes.

2.2.3 Atributos de calidad

Los atributos de calidad representan propiedades asociadas con la calidad de las entidades de software. Dichas propiedades se pueden asignar con un valor cuantitativo ISO/IEC 24765:2010 (Habra *et al.*, 2008).

2.2.4 Métricas de software

Las métricas de software, según la norma *ISO/IEC/IEEE 24765:2010,* son medidas cuantitativas que representan el grado en que una entidad de software posee un atributo de calidad. Las métricas se expresan en términos de una unidad de medición, poseen una escala y un tipo.

2.2.4 Unidad de medición

La unidad de medición, según la norma *ISO/IEC/IEEE 24765:2010,* es un atributo escalar de una entidad definido y adoptado por convención, mediante la cual otras cantidades de la misma clase se pueden comparar para expresar su magnitud.

2.2.5 Escala de medición

La escala de medición es un conjuto de valores con ciertas propiedades asociados con un atributo (Habra *et al.*, 2007). Existen cinco tipos de escalas: nominal, ordinal, intervalar, racional y absoluta. Una escala nominal es un conjunto desordenado de categorías, tal que la única comparación posible es “pertenece a”. Una escala ordinal es un conjunto lineal ordenado de categorías, tal que las comparaciones “mayor que” o “menor que” son válidas. Una escala intervalar es una escala ordinal con intervalos consistentes entre puntos en la escala, tal que la adición y la substracción tienen sentido pero no la multiplicación y la división. Una escala racional es una escala intervalar donde los intervalos entre los puntos son constantes, tal que todas las operaciones aritméticas son validas. Finalmente, una escala absoluta es una escala racional que es un recuento del número de ocurrencias (Everett & McLeod, 2007).

2.2.6 Tipo de métrica

El tipo de métrica sirve para identificar la forma como se obtiene su valor. El tipo de métrica puede ser: métrica base, métrica derivada o indicador. Una métrica base se obtiene de la observación de un atributo y se mide empleando un método de medición. Una métrica derivada, como indica su nombre, se deriva de una métrica base o de otra métrica derivada y se obtiene mediante una función de cálculo. A su vez, un indicador se obtiene de una métrica base o derivada mediante un modelo de análisis (Ferreira *et al.*, 2006, García *et al.*,2006).

2.2.7 Estimación

La estimación es el proceso de predecir, mediante los atributos y métricas recogidos durante el desarrollo de proyectos de software que se completaron, los valores de proyectos incompletos o por empezar (Jones, 2008). Algunas de las técnicas de estimación son: el juicio de expertos, los modelos de analogías y los modelos de simulación, entre otros. La técnica de juicio de expertos consiste en consultar uno o más expertos para determinar según su criterio los valores estimados para ciertos atributos. La técnica de modelos de analogías consiste en comparar valores estimados que se obtienen de proyectos pasados y su herramienta primordial son las bases de datos históricas. La técnica de modelos de simulación consiste en simular situaciones reales y observar el comportamiento del sistema, lo cual se usa para predecir el desempeño en diferentes escenarios.

2.2.8 Métricas de pruebas de software

Las métricas de pruebas son las métricas que se producen durante la etapa de pruebas del ciclo de vida del desarrollo de software. Existen dos tipos de métricas de pruebas de software: las métricas de procesos y las métricas de productos. Con las métricas de procesos se obtiene información acerca de las actividades que comprenden el proceso de pruebas y su objetivo principal es monitorear el progreso de las pruebas. Algunas de las métricas de procesos son: el número de defectos detectados en las pruebas, la efectividad en la remoción de defectos y el esfuerzo de pruebas, entre otras. Con las métricas de productos se obtiene información relacionada con las pruebas y el proceso de pruebas de un producto de software posterior a las actividades de ejecución de las pruebas y su principal objetivo es conocer la calidad del producto. Algunas de las métricas de producto son: la portabilidad, la confiabilidad y la eficiencia, entre otras (Farooq *et al.*, 2011).

2.2.9 Método de medición

Un método de medición es la descripción de una secuencia lógica de operaciones acerca de cómo realizar una actividad de medición. Específicamente, consiste en

definir la manera de representar el atributo de una entidad que se quiere medir con un valor. El método de medición se debe implementar con algunas operaciones concretas que se logran mediante instrumentos de medición o con operaciones prácticas como: selección, conteo, cálculo y comparación, entre otras (Habra *et al.*, 2008).

2.3 Proceso de medición del software

Un proceso de medición consiste, básicamente, en la aplicación de cuatro pasos, así: diseñar el método; aplicar las reglas de medición al software o a la parte del software que se va a medir; analizar los resultados obtenidos y explotar los resultados de medición en un modelo cualitativo o cuantitativo (Jacquet & Abran, 1997). En la Figura 2 se muestra el modelo detallado del proceso de medición que los autores proponen.

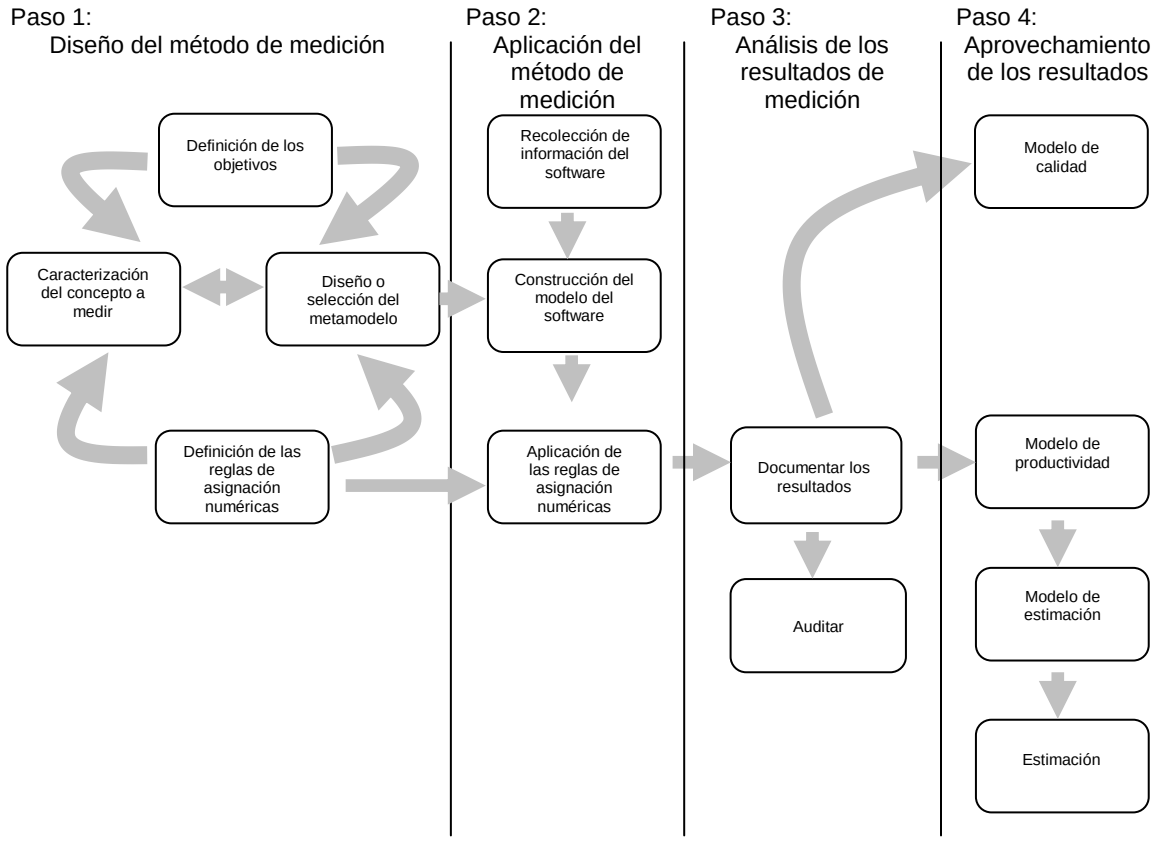


Figura 2: Modelo detallado del proceso de medición (Jacquet & Abran, 1997)

Como se aprecia en la Figura 2, el paso uno consta de cuatro subpasos para diseñar el método de medición. En el subpaso uno el principal objetivo es establecer: qué, para quién y para qué se quiere medir. En los subpasos dos y tres se define el conjunto de las características y las relaciones entre éstas, las cuales representan la entidad que se quiere medir. En el subpaso cuatro se definen las reglas de asignación numéricas; este proceso consiste en mapear las entidades del mundo real con su valor correspondiente, lo cual puede incluir operaciones de conteo, cálculos o medición con instrumentos dependiendo de la manera seleccionada para obtener la medida. Reiteradamente, se deben realizar refinamientos de estos primeros pasos hasta que se logren identificar los conceptos que representan el concepto que se quiere medir. En el paso dos se aplica el método de medición y consta de 3 subpasos. El subpaso dos se puede obviar si con los pasos anteriores se logró identificar el modelo. En el paso tres se documentan y auditan los resultados obtenidos después de medir. Finalmente, en el paso cuatro se usan los resultados obtenidos.

2.4 Estimaciones del esfuerzo de pruebas

La estimación del esfuerzo de pruebas de software proporciona criterios de decisión a los coordinadores de pruebas para la asignación de los recursos de personal y del tiempo dedicado al proceso de pruebas. El esfuerzo de pruebas es una estimación de la cantidad de hombres por unidad de tiempo necesarios para completar un conjunto de pruebas (Everett & McLeod, 2007). Los métodos que estiman el esfuerzo de pruebas se pueden clasificar según los criterios de estimación que usan.

2.4.1 Estimaciones basadas en el esfuerzo de desarrollo del software

Lo métodos que usan este criterio se basan en la premisa de que el esfuerzo en las pruebas depende del esfuerzo del desarrollo (Nageswaran, 2001). El proceso de estimar el esfuerzo de pruebas consiste, primero, en estimar el esfuerzo de desarrollo mediante cualquiera de las técnicas que existen, por ejemplo el modelo constructivo de costos (COCOMO por sus siglas en inglés), el modelo de

administración del ciclo de vida del software (SLIM por sus siglas en inglés) o el modelo de análisis de puntos de función (FPA por sus siglas en inglés). Posteriormente, mediante alguna heurística que relaciona los valores de esfuerzo del desarrollo se obtiene un valor que indica el esfuerzo de pruebas. Como los métodos en esta categoría se basan en estimaciones del esfuerzo de desarrollo, la mayor cantidad de actividades necesarias para la estimación se ejecutan una sola vez.

2.4.2 Estimaciones basadas en el tamaño del software

El criterio de estimación para los métodos en esta categoría es el tamaño de la aplicación de software, porque parten de la premisa de que el esfuerzo de pruebas es directamente proporcional al tamaño del software. Particularmente, Jones (2008) afirma que el número de casos de prueba se estima con el número de puntos de función de acuerdo con la Ecuación 1.

$$\textit{Numero de casos de prueba} = (\textit{Puntos de función})^{1.25} \quad (1)$$

El resultado de esta operación es el número estimado del volumen de errores que se pueden encontrar en la aplicación de software en desarrollo. Finalmente, el valor del esfuerzo de pruebas se obtiene aplicando un factor de conversión que se calcula con información de proyectos pasados.

Nageswaran (2001) desarrolló un método de estimación del esfuerzo que se basa en el método de puntos de casos de uso, el cual sirve para estimar el tamaño del software, conocido como puntos de casos de prueba. En este método se usa información de los casos de uso y sus actores y los factores ambientales del proceso de pruebas. Por último, aplicando un factor de conversión se obtiene un valor del esfuerzo de pruebas.

El método de puntos de casos de prueba se deriva de los casos de uso, que son artefactos que están disponibles en etapas tempranas del ciclo de vida de desarrollo, por lo cual las estimaciones del esfuerzo se pueden realizar desde el inicio del desarrollo.

Veenendaal y Dekkers (1999) desarrollan el método conocido como análisis de puntos de prueba, en el cual el número de puntos de prueba se deriva de los puntos de función. Posteriormente, operando el número de puntos de pruebas junto con el factor ambiental y el factor de productividad se obtiene el esfuerzo de pruebas.

El método de análisis de puntos de prueba se enfoca en la importancia relativa de cada función en el proceso de pruebas, optimizando el tiempo dedicado a las pruebas.

2.4.3 Estimaciones basadas en los casos de prueba

Las aproximaciones agrupadas en esta categoría se basan en el número de casos de prueba y los diferentes escenarios que conforman las pruebas para calcular el esfuerzo de pruebas. Específicamente, Arahna y Borba (2007, 2009) definieron un método que estima el esfuerzo de pruebas usando la *suite* de pruebas como entrada principal. La *suite* de pruebas se define en un lenguaje controlado para luego asignar a cada prueba en la *suite* un número de puntos de ejecución según los criterios de complejidad y de tamaño; por último, se ajusta la suma de los puntos de ejecución con un factor de conversión para obtener el esfuerzo. Este método estima el esfuerzo para casos de prueba de manera individual y los que se generan de manera automática.

Xiaochum et al. (2008) proponen un método para estimar el esfuerzo de pruebas con la premisa: “similares pruebas tienen costos similares”. Los autores proponen una herramienta conocida como vector de ejecución con la que se caracteriza una *suite* de pruebas. El vector de ejecución consta de tres atributos que son: el número de casos de prueba, la complejidad de ejecución de las pruebas y la clasificación de los probadores. Una vez se crea el vector de ejecución para el caso específico, se compara este vector con otros vectores en bases de datos históricas. Con el valor del esfuerzo que se conoce se trata de estimar el esfuerzo del vector de ejecución actual mediante algoritmos de aprendizaje.

2.4.4 Estimaciones basadas en las actividades de pruebas

Las aproximaciones en esta categoría dividen el proceso de pruebas en etapas que agrupan un conjunto de actividades de prueba. El esfuerzo total se obtiene operando el esfuerzo en cada etapa representado por medio de unas características que afectan el esfuerzo.

Abhishek et al. (2010) proponen un método de estimación donde el proceso de pruebas se divide en dos etapas: precodificación y postcodificación. En la etapa de precodificación, el esfuerzo se representa con tres características de los artefactos de desarrollo: los casos de uso y sus actores y los factores ambientales y técnicos. En la etapa de postcodificación, el esfuerzo se representa con tres características del código que son: las variables, la complejidad de los componentes y la criticidad de los componentes. Con esta información, más información histórica y una tasa de error aceptable, se entrenan redes neuronales. La salida de las redes neuronales es el esfuerzo de pruebas. Este método puede usar los casos de uso o puntos de función para calcular el esfuerzo en la etapa de precodificación, lo cual lo hace compatible con más metodologías de desarrollo.

Mizuno et al. (2002) proponen un método de estimación del esfuerzo de pruebas para cumplir el nivel de calidad, pues consideran que el nivel de calidad se logra cuando se mantiene el número de defectos residuales en los valores permitidos, que se definen *a priori*. El método divide el proceso de pruebas en dos etapas, la primera de las cuales se realiza paralela a las actividades del diseño y codificación del ciclo de vida de desarrollo. En esta etapa se realizan exclusivamente pruebas estáticas de los artefactos en la medida en que se generan. La segunda etapa se realiza paralela a la fase de pruebas y depuración, la cual consiste en realizar varios ciclos seguidos de estas actividades. El proceso de estimación consta de tres fases: primero, se define una relación entre el esfuerzo para ejecutar todas las actividades y el número de defectos aceptables; luego, con el valor de prueba obtenido se aplica un modelo de regresión múltiple; por último, con la transformación del modelo de regresión se obtiene una ecuación que representa el esfuerzo. El método propuesto estima el esfuerzo antes que las actividades de

pruebas se inicien. El esfuerzo de las etapas de diseño y codificación se calcula una sola vez para el proyecto de software.

2.5 Productividad

La productividad de una empresa se define como la razón entre una salida que se produce y la entrada que se usa para producir la salida (Coelli, Rao, O'Donnell, & Battese, 2005). En la Ecuación 2.2 se muestra esta relación.

Productividad = Salidas / Entradas (2.2)

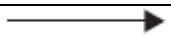
También, se suele entender la productividad como un factor de productividad total, el cual representa la productividad cuando se mide teniendo en cuenta todos los factores de producción. Por otra parte, existen mediciones parciales de productividad que no incluyen la totalidad de factores, pero que pueden arrojar valores de productividad promedio en condiciones aisladas.

La medición de la productividad en el software es similar a la medición de la productividad que se da en economía. Generalmente, se representa como la razón entre el tamaño del producto y el esfuerzo del proyecto (Kitchenham & Mendes, 2004). Las unidades de salida que constituyen el tamaño del producto de manera tangible son las líneas de código y la documentación, mientras que las unidades de entrada se derivan de los recursos y el personal necesarios para completar el proyecto (IEEE std. 1045, 1993).

2.6 Esquemas Preconceptuales

Los esquemas preconceptuales son herramientas para la representación de un domino especifico, los cuales usan una simbología que quienes conocen acerca del dominio pueden validar (Zapata, Giraldo, & Londoño, 2011). La simbología que usan y la explicación de cada elemento se presentan en la Tabla 1.

Tabla 1: Simbología de los Esquemas Preconceptuales

Símbolo	Definición
CONCEPTO	Representan los sustantivos del dominio
RELACIÓN ESTRUCTURAL	Son relaciones permanentes entre dos conceptos, usando los verbos “ser” y “tener”
RELACIÓN DINAMICA	Son relaciones transitorias de actividad u operación entre dos conceptos. Por ejemplo: ejecuta y asigna
 CONEXIÓN	Sirven para unir conceptos a relaciones estructurales o dinámicas y viceversa
IMPLICACIÓN	Son relaciones causa-efecto entre dos relaciones dinámicas
NOTA O INSTANCIA	Sirven para indicar los posibles valores que puede tomar un concepto
REFERENCIA	Sirven para unir elementos que están físicamente distantes en el esquema
CONECTOR DE NOTAS	Sirve para ligar un concepto con el conjunto de notas o valores asociados a él

2.7 La Técnica Delphi

Consiste en aplicar una serie de cuestionarios secuenciales, o “rondas”, con retroalimentación controlada, buscando obtener el consenso de la opinión de un grupo de expertos (Listone & Turoff, 1975). El método se puede aplicar cuando el juicio individual de un experto es relevante y se combina con otros juicios para lograr acuerdo acerca de un conocimiento incompleto de un problema o fenómeno (Skulmoski, Hartman, & Krahn, 2007).

Para aplicar el panel Delphi se siguen de manera general varios pasos (Skulmoski et al., 2007): desarrollo de la pregunta de investigación; diseño de la investigación; selección de la muestra de la investigación; desarrollo del cuestionario de la

primera ronda; aplicación de la prueba piloto; aplicación y análisis del primer cuestionario; desarrollo del cuestionario de la siguiente ronda, aplicación y análisis de la siguiente ronda; verificación, generalización y documentación de los resultados. Los pasos se pueden adaptar a las necesidades de la investigación.

Adicionalmente, se deben tener en cuenta algunas consideraciones para garantizar el éxito de la técnica Delphi. Skulmoski et al. (2007) exponen las siguientes consideraciones a tener en cuenta: las opciones metodológicas, la pregunta inicial, el criterio de experticia, el número de participantes, el número de rondas, el modo de interacción, el rigor metodológico, los resultados, las verificaciones adicionales y la publicación del instrumento Delphi.

3. Antecedentes

Obtener un valor de la productividad en pruebas de software es una actividad que hace parte de algunos de los métodos de estimación del esfuerzo de pruebas. Estos métodos estiman el valor de la productividad mediante alguna técnica que, generalmente, se basa en la información de proyectos anteriores, por lo que se presume la existencia de bases de datos históricas. Pese a este requisito, los métodos no proponen una manera de recopilar la información necesaria para lograr una estimación cercana a las condiciones del proyecto de pruebas, o en otras palabras, una manera de medir la productividad en pruebas. En el pasado se propusieron métodos para medir la productividad del software sin especificar las actividades y las características propias del proceso de pruebas.

En el estándar IEEE Std 1045-1992 se define un marco para la medición de la productividad del software. El objetivo es mejorar la precisión en la recolección y el reporte de información relacionada con la productividad del proceso de desarrollo de software. Los productos de salida definidos para medir la productividad son el código y la documentación, mientras que el producto de entrada es el esfuerzo. Los productos de salida y de entrada se caracterizan mediante atributos, los cuales son medidas directas del proceso de desarrollo.

De manera general, los atributos que representan el código y la documentación son respectivamente: las declaraciones en el código fuente y las páginas de la documentación. Por otra parte, los atributos que representa el esfuerzo, medido en número de hombres por hora, son: las horas del personal dedicado directamente al proceso de desarrollo y las horas del personal de soporte (IEEE std. 1045, 1992).

Si bien el estándar recopila en detalle una serie de atributos para medir los productos de salida y de entrada, dichos atributos representan el proceso de desarrollo como una unidad, sin separar los procesos asociados con las etapas del ciclo de vida del desarrollo del software. Esto representa una desventaja si se quiere medir la productividad, de manera independiente, de los grupos de trabajo en los que se dividen las organizaciones de desarrollo de software. Por esta razón,

el estándar representa una base para crear un método de medición de la productividad del equipo de pruebas pero no es, en sí mismo, una herramienta para medirla.

Por otro lado, se encuentran los métodos que estiman el esfuerzo en pruebas de software, los cuales usan diferentes técnicas para estimar la productividad del equipo de pruebas. Algunas de esas técnicas son: el juicio de expertos (Veenendaal & Dekkers, 1999; Nageswaran, 2001), el análisis de información almacenada en bases de datos históricas (Veenendaal & Dekkers, 1999; Nageswaran, 2001; Jones, 2008; Aranha & Borba, 2009; Xiaochum et al., 2008), y los modelos de simulación (Aranha, Borba, & Lima, 2006), entre otras.

En el modelo de estimación del esfuerzo de pruebas de Jones (2008), primero se calcula el número de potenciales defectos y luego, mediante un factor de conversión que se deriva de información en bases de datos históricas, se obtiene el esfuerzo en términos de hombres por hora. El factor de conversión es un número que representa el número de personas entre analistas, programadores, probadores, etc., necesarios para completar el proceso de desarrollo (Jones, 2008).

El factor de conversión, entendido como un factor de productividad del proceso de pruebas, presenta algunas desventajas: la primera es que no es una estimación de la productividad en el proceso de pruebas sino del desarrollo en general. La segunda, es que se liga con la estimación de los puntos de función, cuyo método de análisis aún presenta problemas de construcción, problemas adaptativos y, adicionalmente, tiene un alto costo de ejecución (Xiaochun, Bo, Fan, Yi, & Lu, 2008). Por último, usa información en bases de datos históricas, las cuales no están disponibles en todas las empresas y, si lo están, no poseen toda la información necesaria. Jones (1998) presenta las típicas lagunas de datos históricos acerca de costos y recursos de software.

El método de puntos de casos de uso de Nageswaran (2001) utiliza un factor de conversión que se define como el número de hombres por hora necesarios para probar los diferentes tipos de puntos de casos de uso del proyecto de desarrollo. El coordinador de pruebas o de desarrollo es quien estima el factor de conversión

de acuerdo con su criterio y experiencia (De Almeida, De Abreau, & Moraes, 2009).

La principal deficiencia del factor de conversión es que su unidad de estimación son los puntos de casos de uso, que no representan unidades tangibles asociadas con el proceso de pruebas (Yi, Bo, & Xiaochun, 2008). Por otra parte, la precisión de la estimación depende de la experiencia y el análisis de la persona quien realiza la estimación, sin depender en gran medida de las condiciones y características del proceso de pruebas.

Para el método de análisis de puntos de prueba (TPA) de Veenendaal & Dekkers (1999), la productividad se define como el tiempo necesario para ejecutar un punto de prueba y tiene dos componentes: una figura de productividad y un factor ambiental. La figura de productividad se basa en el conocimiento y habilidades del equipo de pruebas. Los autores destacan las siguientes áreas del conocimiento y habilidades: componentes específicos de pruebas, gestión de pruebas, técnicas de prueba, aseguramiento de calidad y habilidades sociales (Veenendaal & Pol, 1997). Para estimar la denominada figura de productividad, el método se basa en el análisis de información de proyectos terminados, por lo cual requiere bases de datos históricas. Los autores proponen un rango que varía entre 0,7 y 2,0. El factor ambiental indica el grado en que el ambiente de pruebas influye sobre la productividad y se define mediante las seis variables ambientales siguientes: herramientas de pruebas, pruebas del desarrollo, bases de pruebas, ambiente de desarrollo, ambiente de pruebas y el repositorio de pruebas. Para estimar el factor ambiental se define para cada variable ambiental una escala nominal; el valor seleccionado para la variable debe ser uno de los propuestos en la escala, ya que los valores intermedios no se permiten. Un resumen de las escalas nominales propuestas para las seis variables ambientales se presenta en la Tabla 2.

Tabla 2: Escala nominal del TPA (Veenendaal & Dekkers, 1999)

Herramientas de Pruebas	
1	La prueba usa lenguajes de consulta como SQL, un registro y alguna herramienta de reproducción
2	La prueba usa lenguajes de consulta como SQL pero no usa registros y herramientas de reproducción
Herramientas de Pruebas	
4	Las herramientas de prueba no están disponibles

Tabla 2: Escala nominal del TPA (Veenendaal & Dekkers, 1999) (Continuación)

Pruebas del Desarrollo	
2	Está disponible un plan de pruebas de desarrollo y el equipo de pruebas tiene familiaridad con los casos de prueba y los resultados
4	Está disponible un plan de pruebas de desarrollo
8	No está disponible un plan de pruebas
Bases de Pruebas	
3	Durante el desarrollo se usa documentación estándar y plantillas y las inspecciones se programan
6	Durante el desarrollo se usa documentación estándar y plantillas
12	La documentación del sistema no se desarrolló usando documentación estándar
Ambiente de Desarrollo	
2	El sistema se desarrolló usando lenguajes 4GL, con sistema de administración de bases de datos con numerosas restricciones
4	El sistema se desarrolló usando lenguajes 4GL en combinación con lenguajes 3GL
8	El sistema se desarrolló exclusivamente con lenguajes de programación 3GL
Ambiente de pruebas	
1	El ambiente de pruebas se usó en el pasado
2	Las pruebas se realizan en un ambiente de pruebas recién equipado pero similar a otros que ya se probaron
4	El ambiente se considera experimental
Repositorio de pruebas	
1	Se dispone de un conjunto de datos de prueba y casos de uso
2	Se dispone de un conjunto de datos de prueba
4	No hay repositorio de pruebas

El método presenta la figura de productividad como una manera de incluir las condiciones del personal del equipo de pruebas, pero su estimación es meramente subjetiva, ya que se emplea el juicio del coordinador de pruebas o la persona que haga las veces de administrador. El proceso de estimación de la figura de productividad no presenta un método estructurado que garantice su corrección. Una vez más, se requiere información almacenada en bases de datos históricas, con las desventajas que ya se expusieron. Por otra parte, aunque el método hace una recopilación de factores ambientales que impactan el proceso de pruebas, no hay una relación entre estos y la productividad del equipo de pruebas. Por lo tanto, no se puede concluir qué incidencia tiene la presencia o no de estos factores en la productividad que alcanza el equipo de pruebas.

Aranha, Borba y Lima (2006) desarrollan un modelo de capacidad del equipo de ejecución de pruebas. Esta medida de capacidad hace parte del método de estimación del esfuerzo de pruebas que proponen Aranha y Borba (2009). El modelo de capacidad consiste en estimar la capacidad de un equipo de ejecución de pruebas según el número de ciclos de prueba que ejecuta por día, donde un

ciclo de pruebas es un conjunto de casos de prueba agrupados bajo algún criterio. La capacidad del equipo de ejecución de pruebas se estima dividiendo el número de hombres disponibles por día entre el esfuerzo necesario para ejecutar el ciclo de prueba, como se muestra en la Ecuación 2.

$$Capacidad = \frac{NumerodeHombresporDía}{EsfuerzoporCiclo} \quad (2)$$

Donde el número de hombres por día se calcula multiplicando el número de probadores disponibles por el número de horas trabajadas por día, como se ve en la Ecuación 3.

$$NúmerodeHombresporDía = Númerode\,\mathrm{Pr}obadores * NúmerdeHorasTrabajadasporDía \quad (3)$$

El esfuerzo por ciclo se calcula sumando el esfuerzo (en horas) consumido en configurar el ambiente de pruebas y el esfuerzo consumido en las actividades de prueba, como se muestra en la Ecuación 4.

$$EsfuerzoporCiclo = EsfuerzoenlaConfiguraciónde\,l\,Ambientede\,\mathrm{Pr}uebas + EsfuerzodeEjecuciónde\,\mathrm{Pr}uebas \quad (4)$$

Donde el esfuerzo consumido en las actividades de prueba se obtiene multiplicando la productividad del equipo de pruebas (promedio de horas que tarda en ejecutar un caso de prueba) por el número de casos de prueba que contiene el ciclo de prueba, como se ve en la Ecuación 5.

(5)

La forma de calcular el esfuerzo consumido en configurar el ambiente de pruebas no se especifica en el modelo.

La productividad del equipo de pruebas y la configuración del ambiente de pruebas son variables en cada ciclo de pruebas, por lo cual, los autores proponen usar información en bases de datos históricas para estimar sus valores, usando una medida estadística conocida como la media, cuando los datos son simétricos. En caso de que no se disponga de bases de datos históricas, proponen predecir valores usando técnicas de simulación.

El modelo de capacidad del equipo de ejecución de pruebas no permite especificar un equipo de prueba y un proceso de pruebas particular, porque no tiene en

cuenta características de los profesionales que conforman los equipos de prueba, los casos de prueba para el producto de software y los factores externos que pueden llegar a impactar esta capacidad. Por lo tanto, su estimación es demasiado general y poco ajustada a las condiciones propias del proceso de pruebas en ejecución. Adicionalmente, requiere información muy específica en bases de datos históricas (por ejemplo el esfuerzo por ciclo de pruebas) la cual seguramente no se encuentra disponible. Esta condición compromete los resultados de la estimación. Como una muestra, se asume que la información del número de casos de prueba que se ejecutan por día se puede calcular al realizar una consulta del reporte de los casos de prueba que los probadores ejecutan por día. El esfuerzo por ciclos de prueba sería una estimación con base en esta información y las estimaciones traen consigo errores de estimación relativos, que se propagarán en la siguiente estimación.

Silva et al. (2008), en su método para estimar el esfuerzo de ejecución de una *suite* de casos de prueba, se enfocan en la eficiencia del equipo. Con este objetivo se define una métrica denominada eficiencia acumulada, la cual parte de la premisa que el tiempo requerido para ejecutar casos de prueba manuales decrece lentamente mientras el probador se familiariza con el sistema. La eficiencia acumulada hasta el día i es una relación entre el número de pasos de los casos de prueba ejecutados hasta el día i y el tiempo que se tarda en ejecutarlos. La métrica de eficiencia acumulada se muestra en la Ecuación 6.

$$E(f_i) = \frac{\sum_{d_0}^{d_i} CPPasos}{\sum_{d_0}^{d_i} Tiempo} \quad (6)$$

Donde $E(f_i)$ es la eficiencia acumulada hasta el día i (d_i), $CPPasos$ es el número de pasos de los casos de prueba y $Tiempo$ es el tiempo de ejecución de los casos de prueba.

El modelo de eficiencia acumulada no se presenta como una medida de productividad del equipo de pruebas. Sin embargo, debido a que relaciona las salidas (por ejemplo, el número de pasos de los casos de prueba ejecutados) y las entradas (por ejemplo el tiempo que se emplea para ejecutarlas), se puede decir

que representa en una forma muy limitada la productividad del equipo de pruebas. Además, se limita exclusivamente a la productividad de pruebas de una aplicación de software específica o de aplicaciones de software con características similares.

En el método de estimación del esfuerzo basado en el vector de ejecución (Xiaochum et al., 2008) la productividad del equipo de pruebas se representa mediante dos elementos: la complejidad de ejecución y la calificación de los probadores. La complejidad de ejecución es una característica del esfuerzo de ejecución de las pruebas y es inherente a la *suite* de pruebas. La calificación de los probadores es una categorización de los probadores de acuerdo con dos aspectos: el conocimiento y la experiencia.

Para obtener un valor de la complejidad de ejecución, se realiza una lista de todos los factores de complejidad que pueden impactar la ejecución de las pruebas, tales como: las entradas de las pruebas con los pasos y los datos de prueba; la verificación y validación de las salidas; y el ambiente de pruebas. La lista de factores puede ser el resultado del estudio de expertos. Una vez se tenga la lista de factores de complejidad se debe asignar a cada factor un peso y un nivel en una escala ordinal. Nuevamente, se puede someter al estudio de expertos o puede ser el resultado del análisis de información en bases de datos históricas. Finalmente, se obtiene un valor estimado de la complejidad de ejecución de acuerdo con la regla de cálculo que se muestra en la Ecuación 7.

$$Complejidad\,de\,Ejecución\,de\,las\,Pruebas = \sum_{i}^{n}[Escala(f_i) * Peso(f_i)] \qquad (7)$$

Donde f_i es el factor de complejidad i de la lista de factores de complejidad, $Escala(f_i)$ es el valor asignado al factor i en la escala ordinal y $Peso(f_i)$ es el valor del peso asignado al factor i.

Para obtener una calificación de los probadores, se mide el conocimiento, de acuerdo con el conocimiento que tienen del negocio en un componente o caso de uso asociado con la *suite* de pruebas. Por otra parte, se mide la experiencia de acuerdo con los años de experiencia en pruebas de software. En la Figura 3 se muestra el modelo de calificación de los probadores que proponen los autores.

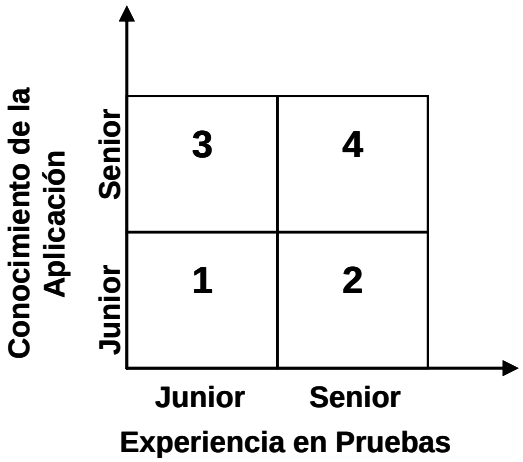


Figura 3: Modelo de calificación de los probadores (Xiaochum et al., 2008)

En este método, los factores de complejidad parametrizan las características de las pruebas y del ambiente de pruebas. Por su parte, la calificación de los probadores da una idea más real de las condiciones del equipo de pruebas. Sin embargo, más que los otros, este método depende de la información en bases de datos históricas, de la precisión y plenitud de sus datos. Adicionalmente, las bases de datos deben contener información de vectores de ejecución similares para que las estimaciones de los valores del esfuerzo de pruebas puedan converger en un algoritmo evolutivo. Los autores reportan que tuvieron ausencia de información al momento de realizar las validaciones. No obstante, no proponen una forma estructurada para recopilar la información necesaria y tampoco el mínimo número de vectores de ejecución que se considera significativo para la estimación.

Una síntesis de la revisión de los métodos que estiman la productividad en pruebas de software se muestra en la Tabla 3. En su mayoría, los métodos que estiman la productividad en pruebas se pueden agrupar según la técnica de estimación que usan, así: el juicio de expertos, los modelos de analogía y los modelos de simulación. Como se observa en la tabla, los métodos presentan una técnica de estimación alternativa al análisis de información en bases de datos históricas, lo que se debe a que, generalmente, las organizaciones de software no cuentan con estas bases de datos o si las tienen omiten hasta el 75% de la información del trabajo realizado en la ejecución de los proyectos de software incluyendo la información de la fase de pruebas.

Tabla 3: Resumen de las técnicas de estimación de la productividad de pruebas de software

Método de estimación de la productividad en pruebas	Técnicas de estimación			Unidades	Características del proceso de pruebas identificadas					Método de recolección de la información
	Juicio experto	Análisis de inf. en BD históricas	Simul.		Equipo	Prob.	Pru.	Amb.	Herr.	
Aranha *et al.*, 2006		X	X	Ciclos de prueba						No determinado/ Aleatoriamente
Jones, 2008		X		Puntos de función						No determinado
Nageswaren, 2001	X	X		Puntos de casos de uso				X		No determinado
Silva *et al., 2008*		X		*Suite* de casos de prueba			X			No determinado
Veenendaal & Dekkers, 1999	X	X		Puntos de función	X			X	X	No determinado
Xiaochum *et al.*, 2008	X	X		*Suite* de pruebas		X	X	X		No determinado

Simul. = Modelos de simulación, Prob.= Probadores, Pru.=Pruebas, Amb.= Ambiente de pruebas, Herr.= Herramientas de pruebas

Por otra parte, las unidades producidas en la mitad de los casos son artefactos que no se emplean directamente en el proceso de pruebas. Las características propias de las pruebas se incluyen de manera parcial en algunas de las aproximaciones y en otras no se contemplan. Por último, aunque todos usan como técnica de estimación el análisis de información en bases de datos históricas no se propone una manera de recolectar la información necesaria.

En la revisión de los métodos de estimación de la productividad en pruebas se evidencia la necesidad de información acerca de los factores que afectan la productividad, como son: los factores ambientales de las pruebas, los factores que se refieren específicamente al equipo de pruebas, los factores de complejidad de las unidades de producción de las pruebas e, inclusive, los factores del desarrollo de software. Sin embargo, ninguno de los métodos propone una forma de adquirir la información desde las fuentes que la generan, con el objetivo que se almacene en bases de datos históricas, que luego se puedan consultar y analizar para lograr estimaciones más precisas.

4. Solución propuesta

Con los resultados de la revisión de la literatura que se realizó en el Capítulo de Antecedentes, se identificaron las características que los métodos de estimación del esfuerzo valoran en sus estimaciones de la productividad del equipo de pruebas, la cuales sirven como base para el método propuesto. En este Capítulo se presenta el método para la medición de la productividad del equipo de pruebas.

4.1 Objetivos del método de medición de productividad del equipo de pruebas

El método de medición propuesto se enfoca en el proceso de ejecución de pruebas de software. La ejecución de un caso de prueba comienza con la asignación del caso de prueba a un probador y termina cuando el probador reporta el caso de prueba con estado “cerrado”. Por lo anterior la unidad mínima producida, como se define en IEEE std-1045 (1992), es el caso de prueba ejecutado. Los tipos de pruebas que se incluyen en el método son las pruebas estructurales, funcionales y de desempeño.

El método de medición propuesto se crea con los siguientes objetivos:

1. Proporcionar a los coordinadores de pruebas criterios para estimar con mayores bases la asignación de recursos, tiempo y personal en el proceso de ejecución de pruebas de software.

2. Definir una manera estructurada de recolectar la información que caracteriza la productividad del equipo de pruebas.

3. Identificar los actores que administran información acerca de la productividad del equipo de pruebas de software.

4. Definir el impacto sobre la productividad del equipo de pruebas de cada una de las características de la productividad.

4.2 Diseño y selección del metamodelo de la productividad del equipo de pruebas

Las características que representan el concepto de productividad del equipo de pruebas de software se agrupan en las siguientes categorías: las características del proyecto de pruebas, las características de la aplicación de software y las características de la administración del proyecto de pruebas.

4.2.1 Características del proyecto de pruebas

Agrupan todos los conceptos que representan la productividad y dependen específicamente del proyecto de pruebas que se encuentra en ejecución. Son: los casos de prueba, los probadores, el equipo de pruebas y el ambiente de pruebas. En su mayoría, los probadores son quienes conocen la información de los conceptos de esta categoría.

4.2.2 Características de la aplicación de software

Agrupan los conceptos que se refieren al producto de software y que afectan la productividad del equipo de pruebas. Los conceptos que conforman esta categoría son: la estabilidad de los requisitos, el código fuente, los escenarios de desarrollo paralelo y el grado de innovación. El gerente de software o el representante del cliente es quien administra la información de estos conceptos.

4.2.3 Características de la administración del proyecto de pruebas

Agrupa los conceptos que se refieren a las decisiones administrativas en torno al proyecto de pruebas. Los conceptos que se incluyen son: los factores limitantes, la criticidad, la participación de los usuarios, los datos de prueba, la organización y el desarrollo paralelo del proyecto de pruebas. El coordinador de pruebas es quien administra la información referente a estos conceptos.

4.2.4 Validación de las características que afectan la productividad del equipo de pruebas

Para validar las características que se identifican como candidatas para afectar la productividad, y sus atributos, se usa una herramienta conocida como panel Delphi (Listone & Turoff, 2002) con la que se busca obtener consenso acerca de ciertos temas a partir de la consulta a un grupo de expertos. El consenso se logra después de varias rondas.

El objetivo del panel es que los expertos proporcionen su opinión acerca de la manera en que las características identificadas pueden llegar a influenciar la productividad, para lo cual se usan los cuatro tipos de preguntas siguientes:

1. Clasificar según el orden de importancia (influencia sobre la productividad) una lista de opciones acerca de una característica.
2. Asignar un número de casos de prueba que pueden ejecutar los probadores, quienes poseen determinadas características asociadas con la productividad.
3. Seleccionar la opción que mejor demuestre la relación entre una característica y la productividad.
4. Asignar una proporción del impacto que una característica tiene sobre la productividad. Este impacto indica una de tres opciones acerca de la característica: no impacta en ninguna manera la productividad, aumenta la productividad o disminuye la productividad.

Debido al tipo de preguntas seleccionadas, que son de carácter cerrado, se estiman convenientes dos rondas para obtener el consenso. El consenso se alcanza en más del 50% de las preguntas tras la primera ronda. El criterio que se usa para obtener el consenso, usando intervalos de confianza de la media y, eventualmente, el porcentaje de las categorías de respuesta (para las preguntas de tipo escalar), es una puntuación del 50% o más alto, de acuerdo con el criterio que proponen Villavicencio y Abran (2012). Los resultados que se obtuvieron en el estudio Delphi, tras las dos rondas, se muestran en la Tabla 3.

El panel para esta Tesis lo conforman 12 personas especialistas en el tema de calidad de software. La mitad de ellos con experiencia en la industria de pruebas de software, específicamente coordinando o administrando pruebas de software. La otra mitad del panel tiene experiencia docente universitaria en áreas de calidad y mediciones de software. El 80% de los panelistas de la industria tienen más de 4 años de experiencia coordinando proyectos de pruebas de software y liderando proyectos de calidad de software. El 33% de los panelistas tienen título de Maestría, el 25% título de Doctorado y el 48% restante tienen título de especialista o una certificación en áreas de conocimiento de calidad de software.

Los 12 panelistas recibieron el cuestionario adjunto a un correo electrónico donde se presenta el trabajo de investigación y la mecánica con la cual se lleva a cabo el estudio. El principal inconveniente que se tuvo, es que los participantes contaban con poca disponibilidad, por lo cual se presentó un amplio intervalo de tiempo entre las dos rondas. Para la segunda ronda se invitó en su totalidad el panel de la primera ronda y a la invitación responden 6 de los panelistas.

Tabla 4: Resultados estudio Delphi para identificar las características que afectan la productividad del equipo de pruebas

Atributos del caso de prueba	**Orden de impacto en la complejidad del caso de prueba (4 – Más impacto, 1 – Menos impacto)**
Número de componentes de software que involucra	4
Número de resultados esperados	3
Número de pasos	2
Número de recursos no funcionales (uso de redes, conexión con otros sistemas, bases de datos, etc.) que involucra	1
Nivel de educación	**Orden de aporte del nivel de educación sobre la productividad del probador (5- Más aporte 1- Menos aporte)**
Estudiante de pregrado	5
Técnico o tecnólogo	4
Profesional	3
Especialista	2
Magíster	1

Tabla 4: Resultado estudio Delphi para identificar las características que afectan la productividad del equipo de pruebas

Categoría de experiencia		
Nombre Categoría	**Nivel de experiencia**	**Número de Meses**
Experiencia en pruebas de software	Alto	25
	Medio	18
	Bajo	7
Experiencia en técnicas de prueba específicas	Alto	25
	Medio	18
	Bajo	7
Experiencia en la organización	Alto	38
	Medio	12
	Bajo	8
Experiencia en el dominio de la aplicación a probar	Alto	22
	Medio	7
	Bajo	4
Áreas del conocimiento	**Orden de impacto del entrenamiento en el área sobre la productividad**	
Fundamentos de pruebas de software	5	
Proceso de pruebas	4	
Niveles de prueba	3	
Técnicas de pruebas	2	
Medidas relacionadas con las pruebas	1	
	Impacto sobre la productividad	
Característica	**Tipo**	**Rango de Proporción**
Cada persona implicada directamente en el proceso de pruebas	Disminuye	10%-27%
Cada persona de soporte implicada en el proceso de pruebas	Aumenta	16%-35%
Rotación del personal del equipo de pruebas	Disminuye	39%-55%
Técnicas y herramientas de gestión de la configuración	Aumenta	50%-60%
Formato y contenido estándar para la documentación	Aumenta	20%-40%
Herramientas automatizadas de pruebas	Aumenta	45%-89%
Nivel bajo de participación de los usuarios en las pruebas	No impacta	
Nivel medio de participación de los usuarios en las pruebas	Aumenta	10%-39%
Nivel alto de participación de los usuarios en las pruebas	Aumenta	10%-37%
Cambios mayores en los requisitos del software	Disminuye	25%-33%
Riesgos asociados con la organización	Disminuye	15%-33%
Riesgos tecnológicos	Disminuye	30%-45%
Riesgos ambientales	Disminuye	10%-35%
Sincronización crítica del proyecto de pruebas	Disminuye	32%-61%
Memoria crítica del proyecto de pruebas	Disminuye	68%-100%

Tabla 4: Resultados estudio Delphi para identificar las características que afectan la productividad del equipo de pruebas (Continuación)

Características	Impacto sobre la productividad	
	Tipo	Rango de proporción
Calidad o confiabilidad crítica	Disminuye	77%-100%
Pruebas en dominios de aplicaciones nunca antes hechas por el equipo	Disminuye	53%-95%
Pruebas en dominios de aplicaciones previamente hechas por el equipo	Aumenta	10%-40%
Nivel de complejidad de la programación bajo	No impacta	
Nivel de complejidad de la programación medio	No impacta	
Nivel de complejidad de la programación alto	Disminuye	1%-65%
Nivel bajo complejidad de los datos de prueba	No impacta	
Nivel medio complejidad de los datos de prueba	Disminuye	1%-44%
Nivel alto complejidad de los datos de prueba	Disminuye	16%-77%
Solo una persona ejecutando pruebas	No impacta	
Solo un departamento ejecutando pruebas	Disminuye	1%-20%
Múltiples departamentos ejecutando pruebas	Disminuye	16%-34%
Múltiples sitios ejecutando pruebas	Disminuye	16%-55%
Subcontratación de pruebas a terceros	Disminuye	23%-44%
Escenarios de desarrollo concurrente de software y firmware	Disminuye	59%-73%
Escenarios de desarrollo concurrente de software y software	Disminuye	55%-72%
Escenarios de desarrollo concurrente de hardware, software y firmware	Disminuye	55%-72%

4.3 Caracterización de los conceptos de productividad del equipo de pruebas

Para medir la productividad del equipo de pruebas los conceptos se caracterizan mediante atributos medibles del proceso de pruebas. Los atributos de los conceptos se resumen en la Tabla 4.

Tabla 5: Atributos de los conceptos que caracterizan la productividad del equipo de pruebas

Características del proyecto de pruebas	
Los casos de prueba	
1	Complejidad de los casos de prueba
Los probadores	
1	Nivel de educación
2	Nivel de experiencia

Tabla 5: Atributos de los conceptos que caracterizan la productividad del equipo de pruebas (Continuación**)**

Los probadores	
3	Entrenamiento en la áreas del conocimiento
Características del proyecto de pruebas	
Equipo de pruebas	
1	Tamaño del equipo de pruebas
2	Rotación del personal
Ambiente de pruebas	
Características de la aplicación de software	
Estabilidad de los requisitos de software	
Nivel de complejidad de la programación	
Grado de innovación del proyecto	
Características de la administración del proyecto de pruebas	
Factores limitantes	
Participación de los usuarios	
Criticidad	
Grado de innovación	
Complejidad de los datos	
Complejidad de la organización	
Concurrencia de desarrollo	

4.3.1 Los casos de prueba

Un caso de prueba es un documento que contiene una serie de campos. Como mínimo contiene un campo para cada uno de los siguientes elementos: el título, la lista de pasos, la lista de resultados esperados, la lista de precondiciones y la lista de poscondiciones. Una representación grafica del concepto de caso de prueba usando esquemas preconceptuales se muestra en la Figura 4. Los casos de prueba que conforman un proyecto de pruebas no implican el mismo esfuerzo de ejecución. Por esta razón, el atributo que representa los casos de prueba es la complejidad del caso de prueba. Por consenso de expertos, la complejidad del caso de prueba se define con cuatro elementos que se organizan según su nivel de impacto en la complejidad del caso de prueba de mayor a menor, así:

1. El número de recursos funcionales que involucra. Es el número de componentes de software que la prueba ejercita, de alguna manera, cuando se ejecuta. Por componente de software, se entiende la definición propuesta en el estándar *ISO/IEC/IEEE 24765:2010* "Un conjunto de servicios funcionales, que, cuando se implementan representa un conjunto bien definido de funcionalidades y se distingue con un nombre único". Por ejemplo: un módulo, una función, etc.

2. El número de resultados esperados. Es la suma de los ítems de la lista de resultados esperados que componen el caso de prueba.

3. El número de pasos. Es la suma de los ítems de la lista de pasos que componen el caso de prueba.

4. El número de recursos no funcionales que involucra. Es el número de componentes de tipo no funcional que se necesitan para ejecutar la prueba. Por componente de tipo no funcional se entiende la definición propuesta en el estándar *ISO/IEC/IEEE 24765:2010 "*Una de las partes que compone un sistema*"* pero que se considera no hacen parte funcional del mismo*.* Por ejemplo: el uso de una red, una conexión al servidor, etc.

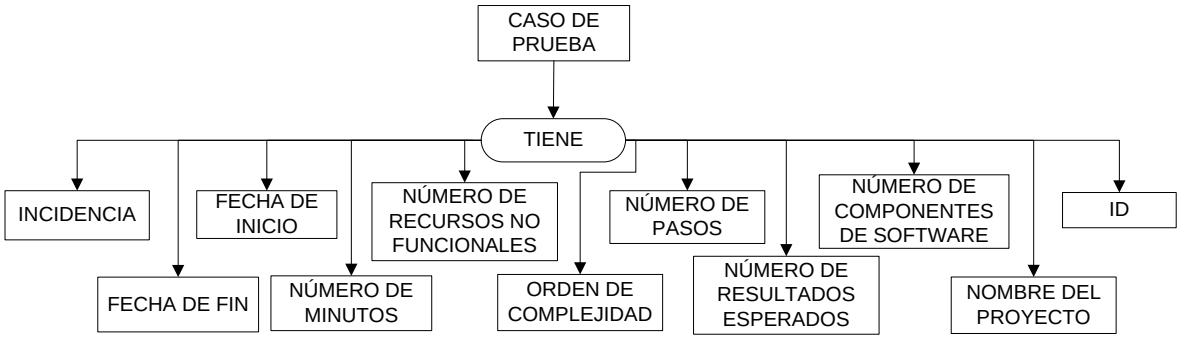


Figura 4: Esquema preconceptual del concepto de caso de prueba

4.3.2 Los probadores

Para el objeto de esta propuesta un probador es la persona encargada de ejecutar el caso de prueba. Los atributos que representan un probador son: el nivel de

educación, la experiencia y el entrenamiento. En la Figura 5 se muestra una representación grafica del concepto de probador usando esquemas preconceptuales.

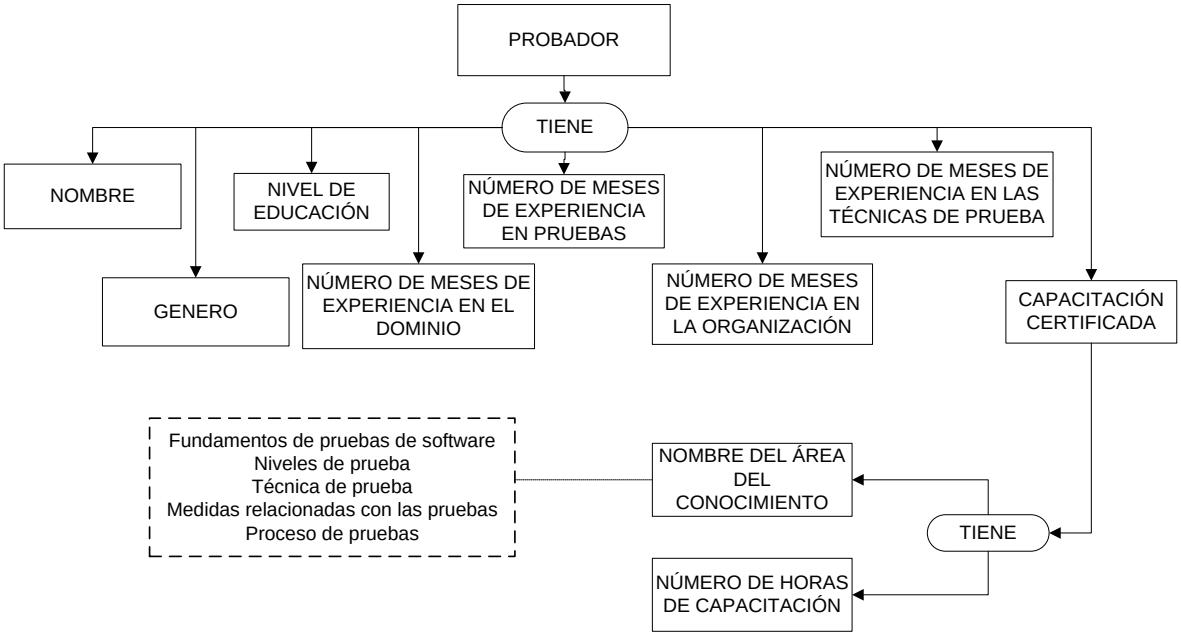


Figura 5: Esquema preconceptual del concepto de probador

Nivel de educación

El nivel de educación se refiere al último grado que obtuvo el probador, que se define de acuerdo con la Ley 30 de 1992 del Congreso de la Republica de Colombia. Por consenso de los expertos, el nivel de educación se organiza según su aporte sobre la productividad del equipo de pruebas de mayor a menor, así:

1. Profesional: son profesionales todas las personas que culminaron un programa de pregrado y tienen un titulo expedido por una institución autorizada. “Los programas de pregrado preparan para el desempeño de ocupaciones, para el ejercicio de una profesión o disciplina determinada, de naturaleza tecnológica o científica o en el área de las humanidades, las artes o la filosofía. También son programas de pregrado aquellos de naturaleza multidisciplinaria conocidos también como estudios de artes liberales” (Congreso de Colombia 30, 1992).

2. Estudiantes de pregrado: son todas las personas que se encuentren desarrollando un estudio sin culminar en algún programa de pregrado.

3. Técnico o tecnólogo: son técnicos todas las personas que culminaron un programa de educación técnica profesional y tienen un título de una institución autorizada. Un programa de educación técnica profesional ofrece formación en ocupaciones de carácter operativo e instrumental y de especialización en su respectivo campo de acción (Congreso de Colombia 30, 1993). Son tecnólogos todas aquellas personas que culminaron un programa de educación tecnológica y tienen un título de una institución autorizada. Un programa de educación tecnológica ofrece formación en ocupaciones, programas de formación académica en profesiones o disciplinas y programas de especialización (Congreso de Colombia 30, 1993).

4. Especialista: Son especialistas todos los profesionales que culminaron un programa de especialización y tienen un título de especialista que expide una institución autorizada. “Los programas de especialización son aquellos que se desarrollan con posterioridad a un programa de pregrado y posibilitan el perfeccionamiento en la misma ocupación, profesión, disciplina o áreas afines o complementarias” (Congreso de Colombia 30, 1992).

5. Magíster: Son todos los profesionales que cursaron un programa de Maestría y que tengan un título de Maestría expedido por una institución autorizada. “Los programas de Maestría tienen a la investigación como fundamento y ámbito necesarios de su actividad. Las Maestrías buscan ampliar y desarrollar los conocimientos para la solución de problemas disciplinarios, interdisciplinarios o profesionales y dotar a la persona de los instrumentos básicos que la habilitan como un investigador en un área especifica de las ciencias y de las tecnologías o que le permitan profundizar teórica y conceptualmente en un campo de la filosofía, de las humanidades y de las artes” (Congreso de Colombia 30, 1992).

Nivel de Experiencia

El nivel de experiencia se refiere a los meses de experiencia que tiene un probador en cuatro categorías: experiencia en pruebas de software, experiencia

en técnicas de prueba específicas, experiencia en la organización y experiencia en el dominio de la aplicación a probar.

1. Experiencia en pruebas de software: abarca la experiencia de un probador realizando las actividades de ejecución de pruebas de software, como: establecer el ambiente de pruebas, verificar la instalación y documentación, coordinar recursos remotos, correr casos de prueba y *scripts*, identificar problemas y probar cambios (Weyuker, Ostrand, Brophy, & Prasad, 2000).

2. Experiencia en técnicas de prueba específicas: abarca la experiencia de un probador en el conocimiento y manejo de las técnicas de prueba que se utilizan en el proyecto de prueba. Puede ser cualquiera de las técnicas que se agrupan en las siguientes categorías: aleatorias, funcionales, de flujo de control, de flujo de datos, mutación, ordenamiento o regresión (Juristo, Moreno, & Vegas, 2002).

3. Experiencia en la organización: abarca el tiempo en meses que el probador tiene trabajando en la empresa.

4. Experiencia en el dominio de la aplicación: abarca el tiempo en meses que el probador tiene trabajando con dominios similares o iguales al del proyecto.

Entrenamiento en las áreas del conocimiento de pruebas

El entrenamiento en las áreas del conocimiento de pruebas se refiere a las lista de las áreas en las que un probador tiene entrenamiento certificado. La áreas del conocimiento se definen en la guía del cuerpo de conocimiento en ingeniería de software (Bourque, Robert, Lavoie, Lee, Trudel, & Lethbridge, 2002), así: fundamentos de pruebas, niveles de prueba, técnicas de prueba, medidas relacionadas con las pruebas y proceso de prueba.

1. Fundamentos de prueba: cubren todos los conceptos básicos en el campo de pruebas de software, la terminología básica, los temas principales y su relación con otras actividades.

2. Niveles de prueba: cubren los criterios de adecuación y de selección de las pruebas, con referencia a los objetivos y alcance específicos.

3. Técnicas de prueba: cubren las técnicas de prueba que acepta la comunidad científica.

4. Medidas relacionadas con las pruebas: cubren las mediciones relacionadas con la evaluación del programa bajo pruebas y la evaluación de la ejecución de las pruebas.

5. Proceso de pruebas: cubren las actividades de las etapas del proceso de pruebas y la administración del proceso de pruebas.

4.3.3 El equipo de pruebas

Los atributos que representan el equipo de pruebas son: el tamaño del equipo de pruebas y la rotación del personal de pruebas. En la Figura 6 se muestra una representación grafica del concepto equipo de pruebas usando esquemas preconceptuales.

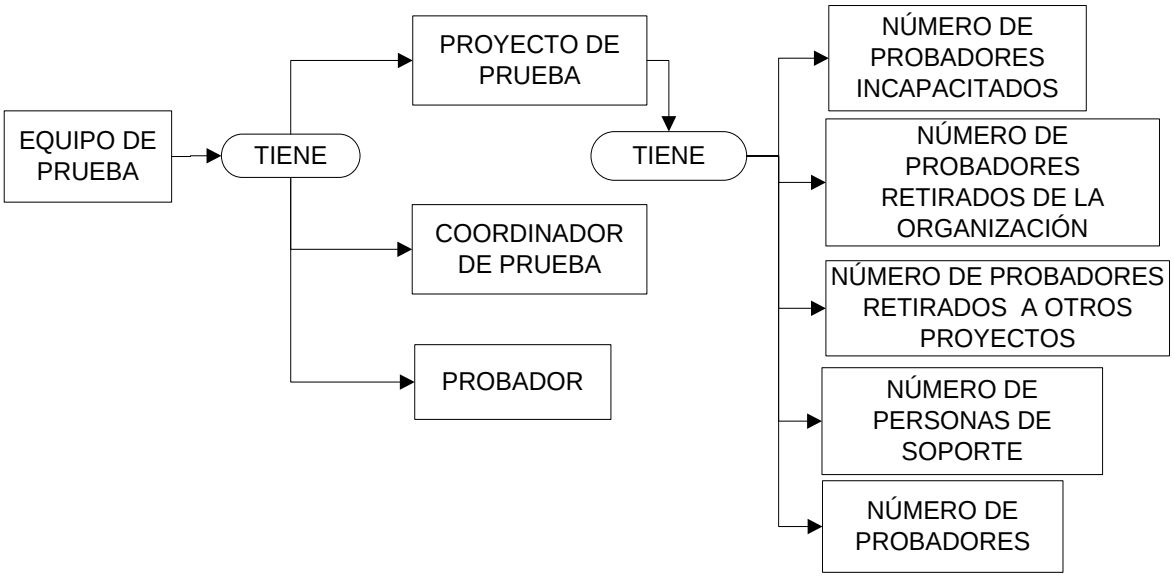


Figura 6: Esquema preconceptual del concepto de equipo de pruebas

Tamaño del equipo de pruebas

Se refiere al número de probadores asignados directamente al equipo de pruebas y al número de personas de soporte implicadas en el proceso de pruebas.
Los probadores asignados directamente al equipo son todos los probadores que ejecutan casos de prueba del proyecto de pruebas.
Las personas de soporte son todos los expertos disponibles para el proceso de pruebas, que pueden asesorar al equipo en caso de surgir algún problema o inquietud de tipo técnico.

Rotación del personal

Se refiere al número de probadores que se retiran de un proyecto de pruebas, sin que terminen de ejecutar los casos de prueba asignados.

4.3.4 El ambiente de pruebas

Se refiere a las herramientas, técnicas y recursos de administración de que se dispone para ejecutar las pruebas. Por consenso de los expertos, las herramientas, técnicas y recursos que afectan la ejecución de las pruebas son: técnicas y herramientas de gestión de la configuración, formato y contenido estándar para la documentación y herramientas automatizadas de prueba.

1. Técnicas y herramientas de gestión de la configuración: incluyen todas las herramientas y técnicas que se usan para controlar los cambios que ocurren en el desarrollo del software.

2. Formato y contenido estándar para la documentación: incluyen todos los formatos que se usan para estandarizar la información de los documentos generados en las etapas del ciclo de vida de desarrollo. Por ejemplo, el estándar ANSI/ANS 10.3-1995 para documentación de software y el estándar IEEE std. 829-1998 para la documentación del proceso de pruebas de software.

3. Herramientas automatizadas de pruebas: incluye todas las herramientas automáticas para la planeación de las pruebas, el diseño de casos de prueba, la ejecución de pruebas, el análisis de los resultados y la regresión de pruebas (Leung, 1998).

En la Figura 7 se muestra una representación grafica del concepto de ambiente de pruebas usando esquemas preconceptuales.

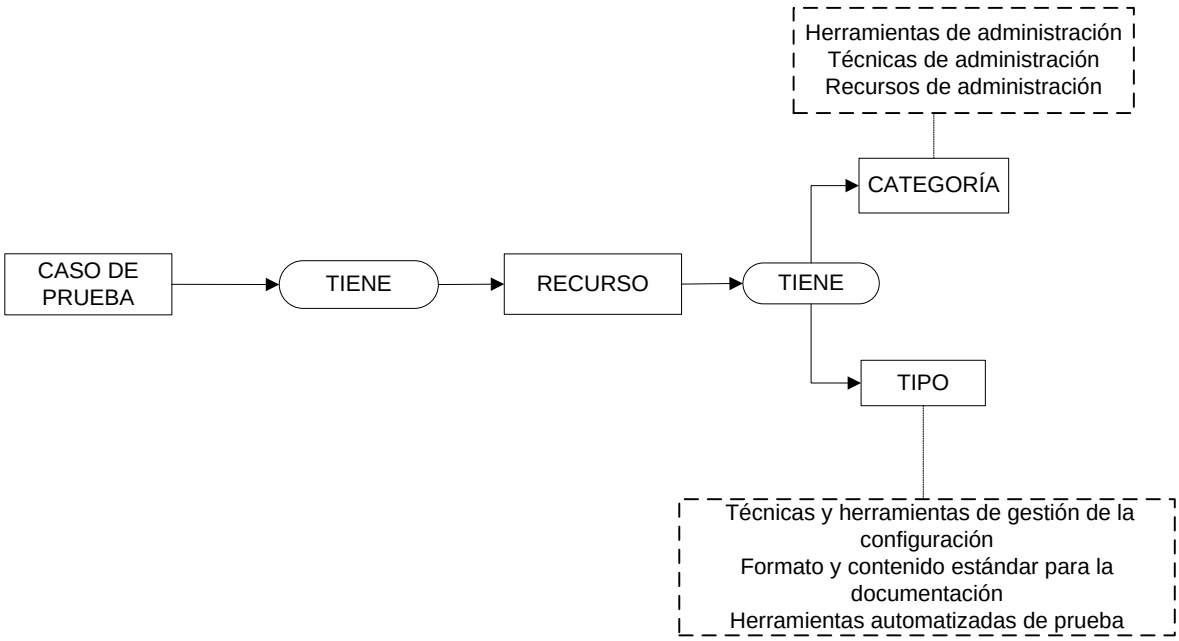


Figura 7: Esquema preconceptual del concepto de ambiente de pruebas

4.3.5 Estabilidad de los requisitos de software

Se refiere al número de requisitos del software que se modifican, se anulan o se eliminan.

4.3.6 Nivel de complejidad de la programación

Se refiere al grado de complejidad del código fuente de la aplicación de software. El grado de complejidad del código fuente se clasifica en la siguiente escala:

1. Bajo: con líneas de código directas, ciclos, ramas y cálculos simples.

2. Medio: con anidaciones simples, comunicación intermodulo simple y cálculos moderadamente difíciles.

3. Alto: con código recursivo, estructuras de datos de control, cálculos complejos con resultados.

4.3.7 Grado de innovación del proyecto de software

Se refiere al que grado de innovación que tiene el dominio de la aplicación de software para el equipo de desarrollo. El grado de innovación se clasifica en la siguiente escala nominal:

1. Nunca antes hecho.

2. Otros previamente lo hicieron.

3. El equipo desarrollador previamente lo hizo.

En la Figura 8 se muestra una representación grafica con esquemas preconceptuales, donde se recopilan los conceptos de: estabilidad de los requisitos de software, nivel de complejidad de la programación y grado de innovación del proyecto de software.

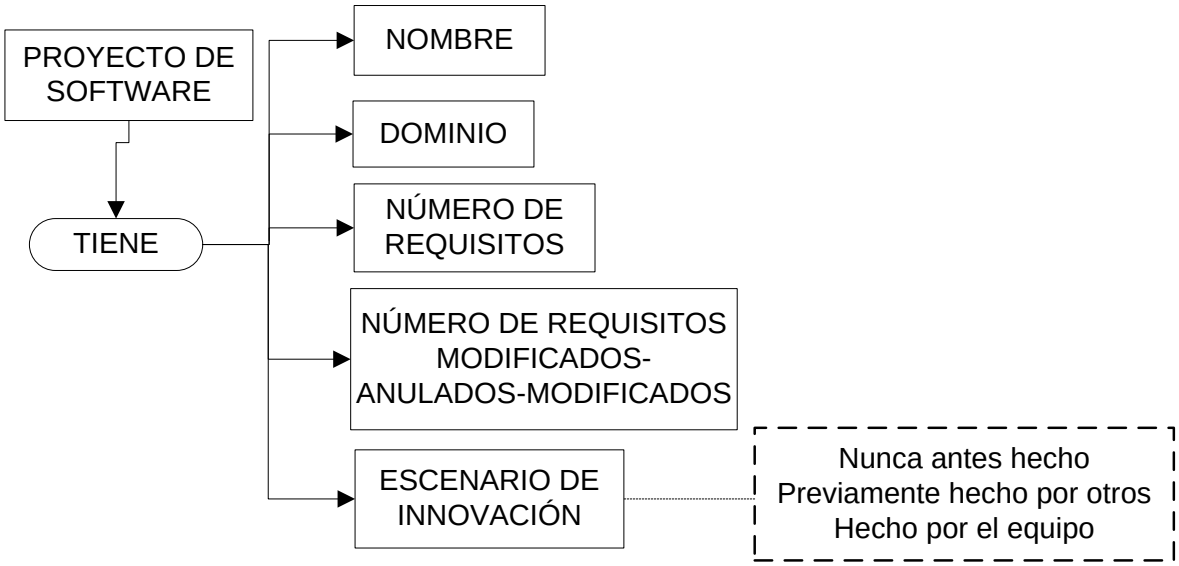


Figura 8: Esquema preconceptual de los conceptos de estabilidad de los requisitos de software, nivel de complejidad de la programación y grado de innovación del proyecto de software

4.3.8 Factores limitantes

Se refiere a los riesgos que suceden durante el proceso de pruebas. Los riesgos pueden ser: de la organización, tecnológicos y ambientales (Zapata *et al.*, 2011).

1. Riesgos de la organización: falta de personal; problemas personales entre equipo o entre miembros del equipo; cooperación insuficiente entre departamentos; estimaciones poco reales.

2. Riesgos tecnológicos: requisitos defectuosos, incompletos o poco reales; tecnologías, métodos, herramientas nuevas o que presentan incertidumbre para el desarrollo de software y pruebas; mala calidad del producto; calidad deficiente o alta complejidad del ambiente de pruebas.

3. Riesgos ambientales: deficiencias de terceros en la provisión de componentes; problemas contractuales; acceso concurrente a recursos externos, cambio de requisitos legales.

En la Figura 9 se muestra una representación grafica del concepto riesgo usando esquemas preconceptuales.

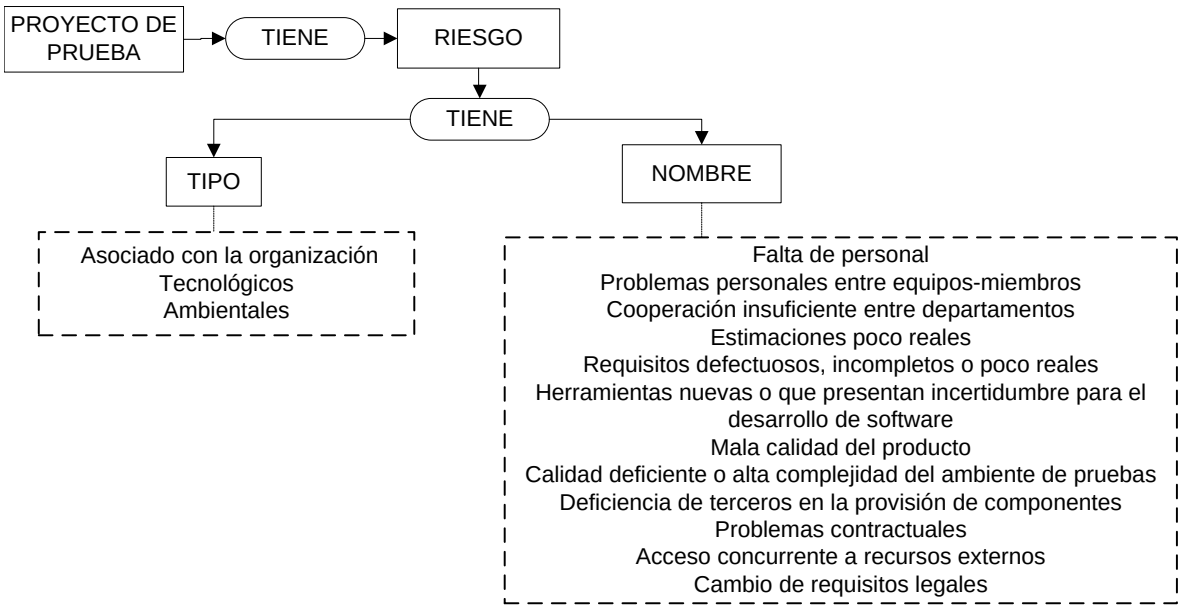


Figura 9: Esquema preconceptual del concepto de riesgo

4.3.9 Participación de los usuarios

Se refiere al nivel de participación que tienen los usuarios finales en la ejecución de los casos de prueba que lo requieran. El nivel de participación de los usuarios se clasifica de acuerdo con la siguiente escala ordinal:

1. Bajo: la participación del usuario es limitada a la etapa posterior a la entrega del proyecto.

2. Medio: la participación del usuario está entre el nivel bajo y alto.

3. Alto: El usuario participa en la ejecución de todos los casos de prueba en los que se requiere.

En la Figura 10 se muestra una representación grafica del concepto de participación de los usuarios usando esquemas preconceptuales.

Figura 10: Esquema preconceptual del concepto de participación de los usuarios

4.3.10 Criticidad

Se refiere a las condiciones de criticidad que se presentan en el proyecto de pruebas. Puede ser: sincronización crítica, memoria crítica y calidad o confiabilidad crítica.

1. Sincronización crítica: sucede cuando el producto se debe probar en ambientes donde el comportamiento en tiempo real, la respuesta del usuario o el rendimiento son críticos.

2. Memoria crítica: sucede cuando el producto se debe probar en un ambiente con memoria limitada.

3. Confiabilidad crítica: sucede cuando el producto se debe probar con criterios de calidad y confiabilidad muy estrictos.

En la Figura 11 se muestra una representación grafica del concepto de criticidad usando esquemas preconceptuales.

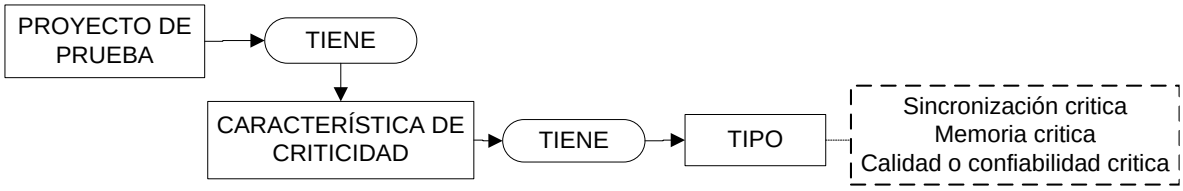


Figura 11: Esquema preconceptual del concepto de criticidad

4.3.11 Grado de innovación del proyecto de pruebas

Se refiere al grado de innovación que tiene el dominio de la aplicación de software para el equipo de pruebas. El grado de innovación se clasifica en la siguiente escala nominal:

1. Nunca antes hecho.

2. Otros previamente lo hicieron.

3. El equipo de pruebas previamente lo hizo.

En la Figura 12 se muestra una representación grafica del concepto de grado de innovación del proyecto de pruebas usando esquemas preconceptuales.

Figura 12: Esquema preconceptual del concepto de grado de innovación del proyecto de pruebas

4.3.12 Complejidad de los datos

Se refiere al nivel de complejidad que tienen los datos de pruebas del proyecto. Los niveles de complejidad organizan en una escala ordinal de la siguiente manera:

1. Bajo: son simples arreglos o archivos en memoria principal, pocas entradas-salidas, datos no reestructurados.

2. Medio: son múltiples entradas-salidas, tipos de datos, datos reestructurados, acceso a otros medios de almacenamiento.

3. Alto: son estructuras de datos relacionales altamente acopladas, algoritmos de búsqueda optimizados, antecedentes de consistencia de datos.

En la Figura 13 se muestra una representación grafica del concepto de complejidad de los datos usando esquemas preconceptuales.

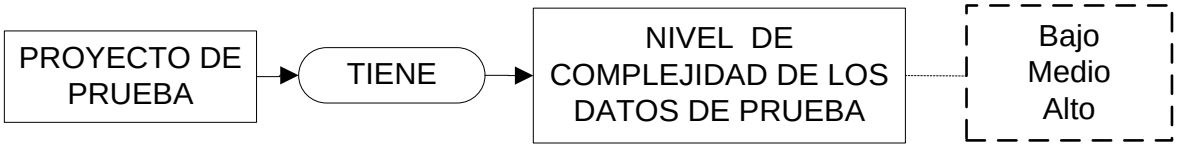


Figura 13: Esquema preconceptual del concepto de complejidad de los datos

4.3.13 Complejidad de la organización

Se refiere a la estructura organizacional del equipo pruebas, enfocado en la comunicación y la coordinación entre los integrantes del mismo. La estructura puede presentar uno o más de los siguientes escenarios:

1. Sólo una persona dedicada a las pruebas.

2. Sólo un departamento.

3. Múltiples departamentos.

4. Múltiples sitios.

5. Subcontratación con terceros.

En la Figura 14 se muestra una representación grafica del concepto de complejidad de la organización usando esquemas preconceptuales.

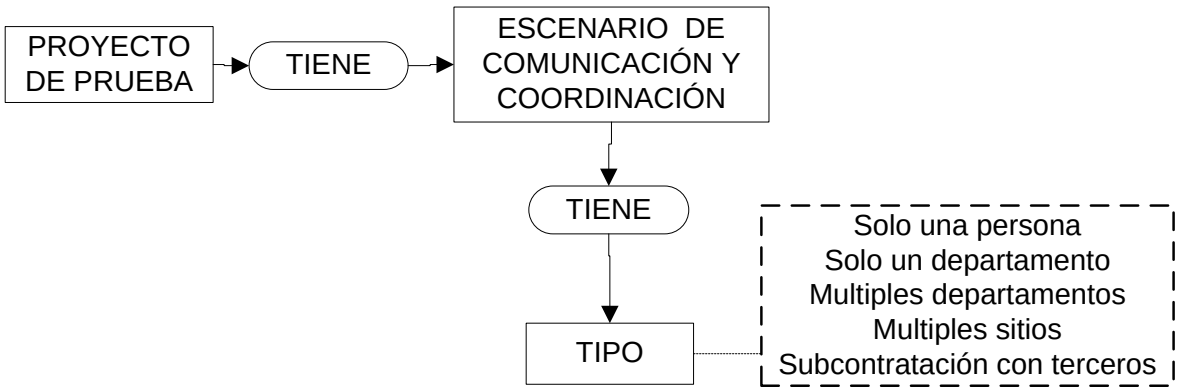


Figura 14: Esquema preconceptual del concepto de complejidad de los datos

4.3.14 Concurrencia de desarrollo

Se refiere al paralelismo de ejecución en diferentes plataformas. Las plataformas paralelas de pruebas se definen de acuerdo con los desarrollos paralelos que involucra un mismo proceso de pruebas. Pueden ser:

1. Hardware y software concurrente: cuando el proyecto de desarrollo involucra desarrollos sobre hardware y desarrollos de software para este hardware. Por ejemplo: programación sobre un PLC y la interfaz gráfica para administrarlo.

2. Hardware y firmware concurrente: cuando el proyecto de software involucra desarrollo de hardware y el firmware asociado con éste. Por ejemplo, controladores de piezas de hardware.

3. Software y firmware concurrente. cuando el proyecto de software involucra desarrollos de software y los controladores de la plataforma hardware.

4. Software y software concurrente: cuando el proyecto de software involucra el desarrollo de otras aplicaciones de software con las cuales comparte información.

5. Hardware, software y firmware concurrente: cuando el proyecto de software involucra desarrollos sobre dispositivos hardware, controladores para la comunicación y desarrollos software para la administración. Por ejemplo: un PLC, con un microcontrolador empotrado y las interfaz gráfica de usuario para administrarlos.

En la Figura 15 se muestra una representación grafica del concepto de concurrencia de desarrollo usando esquemas preconceptuales.

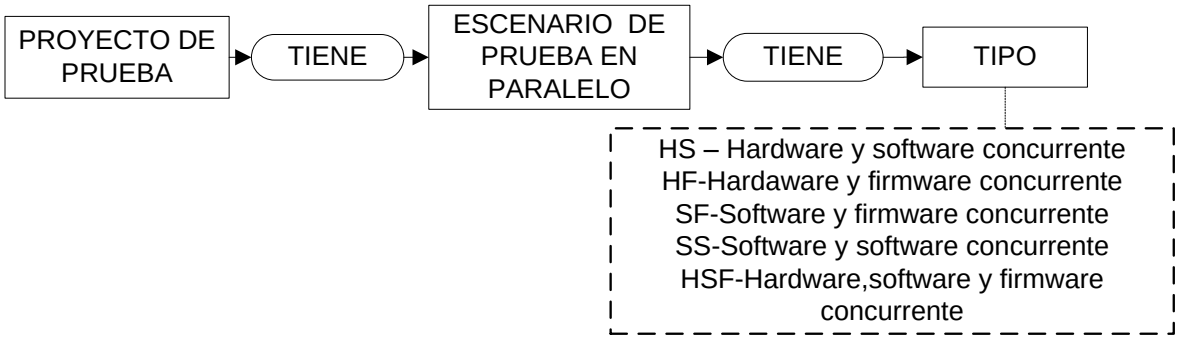


Figura 15: Esquema preconceptual del concepto de complejidad de los datos

4.4 Recopilación de la información acerca de productividad del equipo de pruebas

Para recopilar la información del método de medición se presenta un esquema preconceptual. El esquema preconceptual permite identificar los conceptos de la productividad del equipo de pruebas de software: los actores y las características de la productividad del equipo de pruebas.

Por medio de las relaciones entre los actores y las características se puede identificar donde se origina la información que el método propone recopilar. Los actores se asocian con la información que el método requiere, así:

- El gerente del proyecto conoce la información del proyecto de software
- El coordinador de prueba conoce la información del proyecto de prueba
- El probador conoce su información personal y la información de los casos de prueba que ejecuta.

En la Figura 16 se presenta el esquema preconceptual del método de la productividad del equipo de pruebas de software.

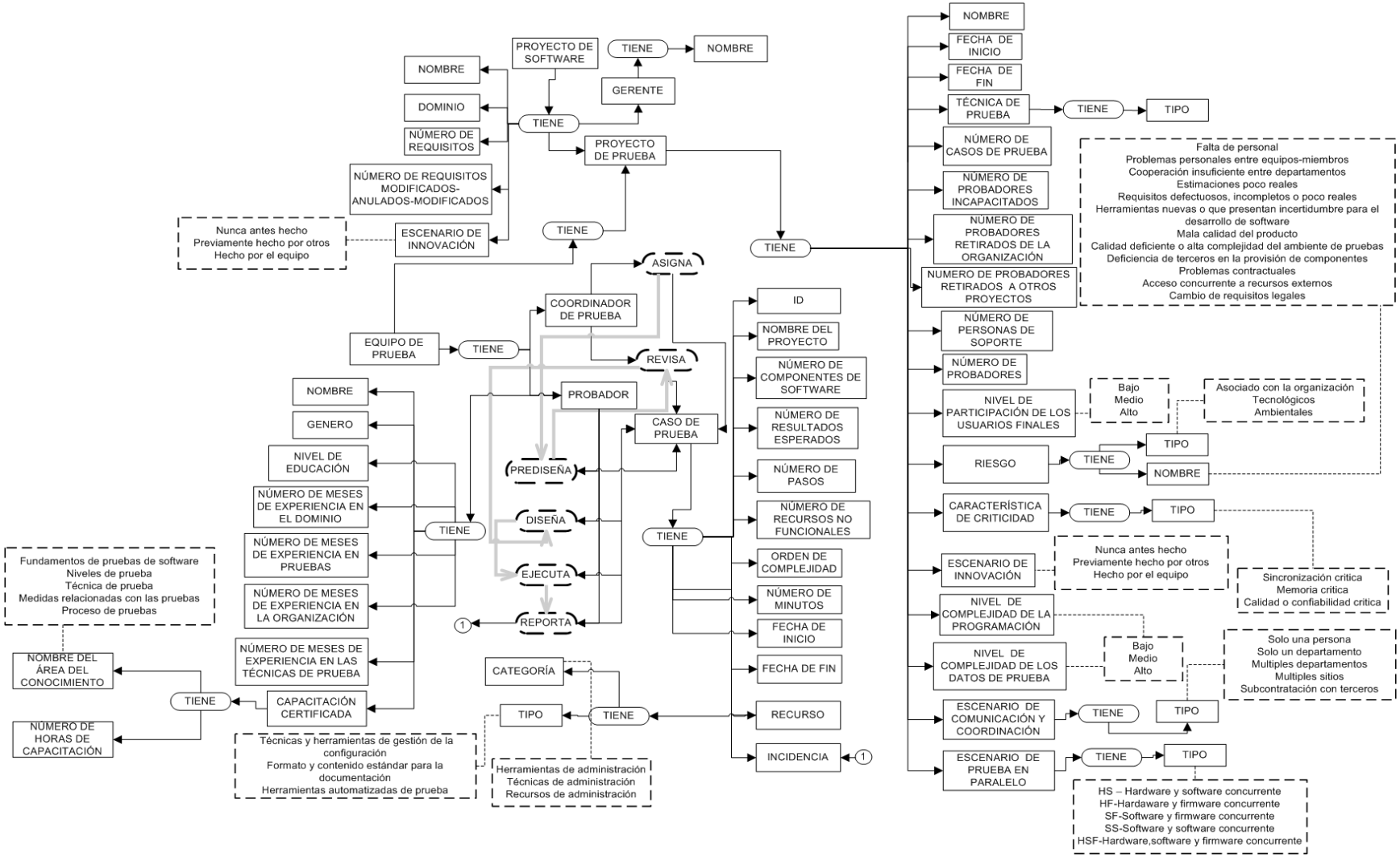


Figura 16: Esquema preconceptual del método de medición de la productividad del equipo de pruebas de software

4.5 Definición de los pasos del método de medición de la productividad del equipo de pruebas.

El método de medición consta de los siguientes pasos:

1. Inicialmente, se registra la información de las características del proyecto de software, entrevistando al gerente del proyecto o al representante del cliente. Se recoge la información para cada uno de los proyectos de software que se encuentren en proceso de ejecución de pruebas con el formato correspondiente al gerente de proyecto, que se presenta en la Tabla 5.

2. Se registra la información de las características de la administración del proyecto de pruebas, alguna información del proceso de pruebas y del proyecto de software, entrevistando al(a los) coordinador(es) que tenga(n) a cargo proyectos de prueba en ejecución. La información se recoge con el formato correspondiente al coordinador de prueba, que se presenta en la Tabla 6.

3. Se registra la información de las características del proyecto de pruebas en dos dimensiones, así: información de los probadores e información de los casos de prueba asignados a cada probador. Para ello se debe entrevistar cada probador que ejecuta pruebas de un proyecto de pruebas en ejecución. Cada probador ejecuta los siguientes pasos:

   - Debe registrar la información personal
   - Debe registrar la información de los casos de prueba que ejecuta
   - Debe dar un orden de complejidad a cada caso de prueba siguiendo los lineamientos que se presentaron en la Sección 4.3.1.

   La información de cada probador se recoge con el formato correspondiente al probador, que se presenta en la Tabla 7.

Tabla 6: Formato del método de medición de la productividad del equipo de pruebas para el gerente de proyecto

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLÍN	**Recolección de información acerca de la productividad de equipos de pruebas de software para los gerentes de proyectos**	**FMT- 003** **Fecha:**
Nombre:		

	1. Nombre de los proyectos de software	**2 . Descripción del dominio de la aplicación**	**3. Número de requisitos del proyecto**
1.			
2.			
3.			
4.			
5.			
6.			

4. Registre el número de requisitos que fueron modificados, eliminados o anulados en el desarrollo para cada proyecto de software

	Nombre de los proyecto de software	**Número de requisitos modificados, eliminados o anulados**
1.	0	
2.	0	
3.	0	
4.	0	
5.	0	
6.	0	

Tabla 6: Formato del método de medición de la productividad del equipo de pruebas para el gerente de proyecto (Continuación)

5. Registre el grado de innovación del dominio del proyecto de software de acuerdo con la siguiente escala:
1. Nunca antes hecho
2. Previamente hecho por otros
3. Previamente hecho por el equipo

	Nombre de los proyecto de software	**Escenarios de innovación del dominio del proyecto de software**
1.	0	
2.	0	
3.	0	
4.	0	
5.	0	
6.	0	

Tabla 7: Formato del método de medición de la productividad del equipo de pruebas para el coordinador de pruebas

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLÍN	**Recolección de información acerca de la productividad de equipos de pruebas de software para los coordinadores de pruebas**								**FMT- 002**	
Nombre:									**Fecha:**	
1. Nombre de los proyectos de pruebas que coordina	**2. Descripción del dominio de la aplicación**	**3. Fecha de Inicio del proyecto**	**4. Fecha de fin del proyecto**	**5. Técnicas de prueba del proyecto**	**6. Número de casos de prueba**	**7. Número de probadores asignados directamente**	**8. Número de personas de soporte disponibles**	**9. Número de probadores retirados a otros proyectos**	**10. Número de probadores que se fueron de la organización**	**11. Número de probadores del proyecto incapacitados**
1.										
2.										
3.										
4.										
5.										
6.										
7.										
8.										
9.										

Tabla 7: Formato del método de medición de la productividad del equipo de pruebas para el coordinador de pruebas (Continuación)

12. Registre el nivel de participación de los usuarios finales en el proyecto de pruebas de acuerdo con la siguiente escala:
1. Bajo: La participación del usuario es limitada a la etapa posterior a la entrega del proyecto
2. Medio: La participación del usuario esta entre el nivel bajo y alto
3. Alto: El usuario participa en la ejecución de todos los casos de prueba en los que se requiere
4. N/A: El usuario no participa

	Nombre de los proyectos de pruebas que coordina	**Nivel de participación del usuario**
1.	0	
2.	0	
3.	0	
4.	0	
5.	0	
6.	0	
7.	0	
8.	0	
9.	0	

Tabla 7: Formato del método de medición de la productividad del equipo de pruebas para el coordinador de pruebas (Continuación)

13. Seleccione los factores limitantes que estuvieron presentes en la ejecución del proyecto de pruebas

Nombre de los proyectos de pruebas que coordina		Materialización de riesgos asociados con la organización				Materialización de los riesgos tecnológicos				Materialización de los riesgos ambientales			
		Falta de personal	Problemas personales entre equipos / miembros del equipo	Cooperación insuficiente entre departamentos	Estimaciones poco reales	Requisitos defectuosos, incompletos o poco reales	Herramientas nuevas que presentan incertidumbre para el desarrollo de software y pruebas	Falta de calidad del producto	Calidad deficiente o alta complejidad del ambiente de pruebas	Deficiencia de terceros en la provisión de componentes	Problemas contractuales	Acceso concurrente a recursos externos	Cambio de requisitos legales
1.	0												
2.	0												
3.	0												
4.	0												
5.	0												
6.	0												
7.	0												
8.	0												
9.	0												

Tabla 7: Formato del método de medición de la productividad del equipo de pruebas para el coordinador de pruebas (Continuación)

14. Registre el nivel de criticidad del proyecto de pruebas marcando con una (x) las características que posee el proyecto de acuerdo con la siguiente descripción:
1. Sincronización crítica: Si el producto se debe probar en ambientes donde el comportamiento en tiempo real, la respuesta del usuario o el rendimiento es crítico
2. Memoria crítica: Si el producto se debe probar en un ambiente con memoria limitada
3. Calidad o confiabilidad crítica: Si el producto se debe probar con criterios de calidad y confiabilidad muy estrictos

Nombre de los proyectos de pruebas que coordina		Sincronización crítica	Memoria crítica	Calidad o confiabilidad crítica
1.	0			
2	0			
3.	0			
4.	0			
5.	0			
6.	0			
7.	0			
8.	0			
9.	0			

Tabla 7: Formato del método de medición de la productividad del equipo de pruebas para el coordinador de pruebas (Continuación)

15. Registre el grado de innovación del dominio del proyecto de pruebas de acuerdo con la siguiente escala:
1. Nunca antes hecho
2. Previamente hecho por otros
3. Previamente hecho por el equipo

Nombre de los proyectos de pruebas que coordina		**Escenarios de innovación del dominio del proyecto de pruebas**
1.	0	
2.	0	
3.	0	
4.	0	
5.	0	
6.	0	
7.	0	
8.	0	
9.	0	

Tabla 7: Formato del método de medición de la productividad del equipo de pruebas para el coordinador de pruebas (Continuación)

16. Registre el nivel de complejidad del flujo de control del producto de software, asociado al proyecto de pruebas, de acuerdo con la siguiente escala:
1. Bajo: con líneas de código directas, ciclos, ramas y cálculos simples
2. Medio: con anidación simples, comunicación intermodulo simple y cálculos moderadamente difíciles
3. Alto: con código recursivo, estructuras de datos de control, cálculos complejos con resultados exactos

	Nombre de los proyectos de pruebas que coordina	Nivel de complejidad de la programación
1.	0	
2.	0	
3.	0	
4.	0	
5.	0	
6.	0	
7.	0	
8.	0	
9.	0	

Tabla 7: Formato del método de medición de la productividad del equipo de pruebas para el coordinador de pruebas (Continuación)

17. Registre el nivel de complejidad de los datos del proyecto de pruebas de acuerdo con la siguiente escala:
1. Bajo: son simples arreglos o archivos en memoria principal, pocas entradas-salidas, datos no reestructurados
2. Medio: son múltiples entradas-salidas, tipos de datos, datos reestructurados, acceso a otros medios de almacenamiento
3. Alto: son estructuras de datos relacionales altamente acopladas, algoritmos de búsqueda optimizados, antecedentes de consistencia de datos

	Nombre de los proyectos de pruebas que coordina	Nivel de complejidad de los datos de prueba
1.	0	
2.	0	
3.	0	
4.	0	
5.	0	
6.	0	
7.	0	
8.	0	
9.	0	

Tabla 7: Formato del método de medición de la productividad del equipo de pruebas para el coordinador de pruebas (Continuación)

18. Seleccione el tipo de escenario para la comunicación y la coordinación del equipo en el que se ejecuta cada proyecto de pruebas

Nombre de los proyectos de pruebas que coordina		Solo una persona	Solo un departamento	Múltiples departamentos	Múltiples sitios	Subcontratación con terceros
1.	0					
2.	0					
3.	0					
4.	0					
5.	0					
6.	0					
7.	0					
8.	0					
9.	0					

Tabla 7: Formato del método de medición de la productividad del equipo de pruebas para el coordinador de pruebas (Continuación)

19. Seleccione el tipo de escenario de prueba en paralelo de cada proyecto de pruebas marcando con una (x) el escenario que corresponda con el proyecto, si aplica

Nombre de los proyectos de pruebas que coordina		Hardware y software concurrente	Hardware y firmware concurrente	Software y firmware concurrente	Software y software concurrente	Hardware, software y firmware concurrente
1.	0					
2.	0					
3.	0					
4.	0					
5.	0					
6.	0					
7.	0					
8.	0					
9.	0					

Tabla 8: Formato del método de medición de la productividad del equipo de pruebas para el probador

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLÍN	**Recolección de información acerca de la productividad de equipos de pruebas de software para los probadores y los analistas de pruebas**				**FMT- 001**	
					Fecha:	
Nombre:		**Genero:**				
1. Nombre de los proyectos asignados para probar	**2. Dominio de la aplicación**	**3. Meses de experiencia en el dominio de la aplicación**	**4. Meses de experiencia en las técnicas de pruebas del proyecto**	**5. Fecha de ingreso(a) al proyecto**	**6. Fecha en que fue retirado(a) del proyecto**	**7. Número de días incapacitado(a)**
1.						
2.						
3.						
4.						
5.						

Tabla 8: Formato del método de medición de la productividad del equipo de pruebas para el probador (Continuación)

8. Clasifique los casos de prueba que tiene asignados de acuerdo con los cuatro atributos siguientes. Por favor, ordene los casos de prueba según la complejidad yendo del caso de prueba de complejidad más alta al caso de prueba de complejidad más baja, de la siguiente manera:
1. Asigne primero los casos de prueba por el número de componentes de software que involucra
2. Luego asigne los casos de prueba por el número de resultados esperados
3. Luego asigne los casos de prueba por el número de pasos
4. Por ultimo, asigne los casos de prueba por el número de recursos no funcionales que involucra

	Id del caso de prueba	Nombre del proyecto	Número de resultados esperados	Número de pasos	Número de recursos no funcionales que involucra	Orden de complejidad del caso de prueba
1.						
2.						
3.						
4.						
5.						
6.						
7.						
8.						
9.						
10.						
11.						
12.						
13.						
14.						
15.						
16.						
17.						

Tabla 8: Formato del método de medición de la productividad del equipo de pruebas para el probador (Continuación)

9. Registre el número de minutos que tardó en ejecutar cada caso de prueba, la fecha de inicio y de finalización de la ejecución y seleccione marcando con una (x) las herramientas, las técnicas y los recursos que usó para ejecutar los casos de prueba

	Id del caso de prueba	Nombre del proyecto	Fecha de Inicio	Fecha de fin	Técnicas y herramientas de gestión de la configuración	Formato y contenido estándar para la documentación	Herramientas automatizadas de pruebas
1.	0	0					
2.	0	0					
3.	0	0					
4.	0	0					
5.	0	0					
6.	0	0					
7.	0	0					
8.	0	0					
9.	0	0					
10.	0	0					
11.	0	0					
12.	0	0					
13.	0	0					
14.	0	0					
15.	0	0					
16.	0	0					
17.	0	0					

Tabla 8: Formato del método de medición de la productividad del equipo de pruebas para el probador (Continuación)

10. Seleccione el nivel de educación que corresponde con su formación

Nivel de educación	
Estudiante de pregrado	
Técnico o tecnólogo	
Profesional	
Especialista	
Magíster	

11. Registre el número de meses de experiencia que tenga en cada unas de las categorías siguientes

Categoría de experiencia	**Número de meses de experiencia**
Experiencia en pruebas de software	
Experiencia en la organización	

Tabla 8: Formato del método de medición de la productividad del equipo de pruebas para el probador (Continuación)

12. Registre el número de horas de capacitación certificada que tiene en las áreas del conocimiento siguientes

Áreas del conocimiento	Horas de capacitación
Fundamentos de pruebas de software: cubre todo los conceptos básicos en el campo de pruebas de software, la terminología básica, los temas principales y su relación con otras actividades.	
Niveles de pruebas: cubre los criterios de adecuación y de selección de las pruebas, con referencia a los objetivos y alcance específicos.	
Técnicas de pruebas: cubre las técnicas aceptadas por la comunidad científica.	
Medidas relacionadas con las pruebas: cubre las mediciones relacionadas con la evaluación del programa bajo pruebas y la evaluación de la ejecución de las pruebas.	
Proceso de pruebas: cubre las actividades de las etapas del proceso de pruebas y la administración del proceso de pruebas.	

En este Capítulo se plantearon los objetivos del método de medición de la productividad del equipo de pruebas, se diseñó el metamodelo de la productividad del equipo de pruebas, se definieron las características y los atributos que representan la productividad. Se definieron formatos, para recolectar la información necesaria para medir la productividad, por cada actor en el proceso de ejecución de un proyecto de pruebas.

5. Validación

En este Capítulo se presenta la validación del método propuesto, mediante la recolección de la información de las características que afectan la productividad, en los formatos propuestos en las Tablas 5, 6 y 7.

La información pertenece a SEQUAL S.A. que es una empresa que cuenta con un grupo de profesionales de ingeniería de software dedicados a ofrecer servicios empresariales outsourcing de mejoramiento del proceso de desarrollo de software con énfasis en la calidad durante todo el ciclo de vida del software. SEQUAL S.A. tiene entre sus clientes diferentes organizaciones de gran envergadura que se desempeñan en diversos sectores de la economía con gran impacto en el ámbito regional, nacional y mundial. Actualmente la empresa tiene proyectos de prueba en ejecución con dos empresas de la ciudad de Medellín. La información que se suministra referente a los proyectos cuenta con aprobación, por parte del cliente, para publicar los datos. Sin embargo, los nombres reales de las empresas, de los proyectos y de los actores no se divulgaran para guardar las condiciones de confidencialidad que debe cumplir SEQUAL S.A. para con sus clientes.

Para aplicar el método de medición de la productividad del equipo de pruebas, se siguen los pasos que propone el método, así:

1. La empresa presenta al investigador con el representante del cliente, en este caso el interventor por parte de la organización en cada uno de los proyectos. Se presentan los objetivos de recolección de la información al interventor. El método presenta esta variante por que la información se va a usar por parte de terceros para hacerla pública en esta Tesis. Sin embargo, en condiciones normales, se espera que cuando la empresa contrata sus servicios, en reuniones iniciales se recoja esta información. Los datos se presentan en las Tablas 8 y 12.

2. Se registra la información de las características de la administración del proyecto de pruebas con el coordinador del proyecto de pruebas de la empresa, la información se recoge por medio de entrevista. Los datos se presentan en las Tablas 9 y 13.

3. Se registra la información de las características del proyecto de pruebas. Inicialmente se entrevista a cada probador que ejecuta pruebas de los proyectos con el objetivo de recoger la información personal de cada probador. Posteriormente, cada probador recibe su formato por correo y se encargan de registrar los datos correspondientes a los casos de prueba que tiene asignados para ejecutar. Los datos que suministran los probadores se presentan en las Tablas 10, 11, 14, 15, 16 y 17.

Tabla 9: Datos del método para medir la productividad del equipo de pruebas-Gerente de proyecto

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLÍN	**Recolección de información acerca de la productividad de equipos de pruebas de software para los gerentes de proyectos**	**FMT- 003**
		Fecha:14-06-2013
Nombre:	Representante del Cliente No.1	

	1. Nombre de los proyectos de software	**2. Descripción del dominio de la aplicación**	**3. Número de requisitos del proyecto**		
1.	Sistema de investigación Universitaria SIIU módulo convocatorias	Investigación	73		
2.					

4. Registre el número de requisitos que fueron modificados, eliminados o anulados en el desarrollo para cada proyecto de software

	Nombre de los proyecto de software	**Número de requisitos modificados, eliminados o anulados**
1.	Sistema de investigación Universitaria SIIU módulo convocatorias	0
2.	0	

Tabla 9: Datos del método para medir la productividad del equipo de pruebas-Gerente de proyecto (Continuación)

5. Registre el grado de innovación del dominio del proyecto de software de acuerdo con la siguiente escala: 1. Nunca antes hecho 2. Previamente hecho por otros 3. Previamente hecho por el equipo		
Nombre de los proyecto de software		**Escenarios de innovación del dominio del proyecto de software**
1.	Sistema de investigación Universitaria SIIU módulo convocatorias	1
2.	0	

Tabla 10: Datos del método para medir la productividad del equipo de pruebas-Coordinador de pruebas

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLÍN	**Recolección de información acerca de la productividad de equipos de pruebas de software para los coordinadores de pruebas**	**FMT- 002** **Fecha:**14-06-2013
Nombre:	Coordinador de pruebas No.1	

	1. Nombre de los proyectos de pruebas que coordina	**2. Descripción del dominio de la aplicación**	**3. Fecha de Inicio del proyecto**	**4. Fecha de fin del proyecto**	**5. Técnicas de prueba del proyecto**	**6. Número de casos de prueba**	**7. Número de probadores asignados directamente**	**8. Número de personas de soporte disponibles**	**9. Número de probadores retirados a otros proyectos**	**10. Número de probadores que se fueron de la organización**	**11. Número de probadores del proyecto incapacitados**
1.	Sistema de convocatorias de investigación	Investigación universitaria: publicación convocatoria, registro de convocatorias, evaluación de convocatorias, selección de proyectos	04/02/2013	12/04/2013	Técnicas de caja negra-Técnicas no funcionales de desempeño	128	2	2	1	0	2
2.											

12. Registre el nivel de participación de los usuarios finales en el proyecto de pruebas de acuerdo con la siguiente escala:
1. Bajo: La participación del usuario es limitada a la etapa posterior a la entrega del proyecto
2. Medio: La participación del usuario esta entre el nivel bajo y alto
3. Alto: El usuario participa en la ejecución de todos los casos de prueba en los que se requiere
4. N/A: El usuario no participa

Nombre de los proyectos de pruebas que coordina		**Nivel de participación del usuario**
1.	Sistema de convocatorias de investigación	2
2.	0	

Tabla 10: Datos del método para medir la productividad del equipo de pruebas-Coordinador de pruebas (Continuación)

13. Seleccione los factores limitantes que estuvieron presentes en la ejecución del proyecto de pruebas

Nombre de los proyectos de pruebas que coordina		Materialización de riesgos asociados con la organización				Materialización de los riesgos tecnológicos				Materialización de los riesgos ambientales			
		Falta de personal	Problemas personales entre equipos / miembros del equipo	Cooperación insuficiente entre departamentos	Estimaciones poco reales	Requisitos defectuosos, incompletos o poco reales	Herramientas nuevas que presentan incertidumbre para el desarrollo de software y pruebas	Falta de calidad del producto	Calidad deficiente o alta complejidad del ambiente de pruebas	Deficiencia de terceros en la provisión de componentes	Problemas contractuales	Acceso concurrente a recursos externos	Cambio de requisitos legales
1.	Sistema de convocatorias de investigación				x	x		x	x		x	x	x
2.	0												
3.	0												

14. Registre el nivel de criticidad del proyecto de pruebas marcando con una (x) las características que posee el proyecto de acuerdo con la siguiente descripción:

1. Sincronización crítica: Si el producto se debe probar en ambientes donde el comportamiento en tiempo real, la respuesta del usuario o el rendimiento es critico

2. Memoria crítica: Si el producto se debe probar en un ambiente con memoria limitada

3. Calidad o confiabilidad crítica: Si el producto se debe probar con criterios de calidad y confiabilidad muy estrictos

Nombre de los proyectos de pruebas que coordina		Sincronización crítica	Memoria crítica	Calidad o confiabilidad crítica
1.	Sistema de convocatorias de investigación		x	
2	0			

Tabla 10: Datos del método para medir la productividad del equipo de pruebas-Coordinador de pruebas (Continuación)

15. Registre el grado de innovación del dominio del proyecto de pruebas de acuerdo con la siguiente escala
1. Nunca antes hecho
2. Previamente hecho por otros
3. Previamente hecho por el equipo

	Nombre de los proyectos de pruebas que coordina	Escenarios de innovación del dominio del proyecto de pruebas
1.	Sistema de convocatorias de investigación	1
2.	0	

16. Registre el nivel de complejidad del flujo de control del producto de software, asociado al proyecto de pruebas, de acuerdo con la siguiente escala:
1. Bajo: con líneas de código directas, ciclos, ramas y cálculos simples
2. Medio: con anidación simples, comunicación intermodulo simple y cálculos moderadamente difíciles
3. Alto: con código recursivo, estructuras de datos de control, cálculos complejos con resultados exactos

	Nombre de los proyectos de pruebas que coordina	Nivel de complejidad de la programación
1.	Sistema de convocatorias de investigación	2
2.	0	

Tabla 10: Datos del método para medir la productividad del equipo de pruebas-Coordinador de pruebas (Continuación)

17. Registre el nivel de complejidad de los datos del proyecto de pruebas de acuerdo con la siguiente escala:
1. Bajo: son simples arreglos o archivos en memoria principal, pocas entradas-salidas, datos no reestructurados
2. Medio: son múltiples entradas-salidas, tipos de datos, datos reestructurados, acceso a otros medios de almacenamiento
3. Alto: son estructuras de datos relacionales altamente acopladas, algoritmos de búsqueda optimizados, antecedentes de consistencia de datos

	Nombre de los proyectos de pruebas que coordina	Nivel de complejidad de los datos de prueba
1.	Sistema de convocatorias de investigación	1
2.	0	

18. Seleccione el tipo de escenario para la comunicación y la coordinación del equipo en el que se ejecuta cada proyecto de pruebas

	Nombre de los proyectos de pruebas que coordina	Solo una persona	Solo un departamento	Múltiples departamentos	Múltiples sitios	Subcontratación con terceros
1.	Sistema de convocatorias de investigación		x			
2.	0					

Tabla 10: Datos del método para medir la productividad del equipo de pruebas-Coordinador de pruebas (Continuación)

19. Seleccione el tipo de escenario de prueba en paralelo de cada proyecto de pruebas marcando con una (x) el escenario que corresponda con el proyecto, si aplica						
Nombre de los proyectos de pruebas que coordina		**Hardware y software concurrente**	**Hardware y firmware concurrente**	**Software y firmware concurrente**	**Software y software concurrente**	**Hardware, software y firmware concurrente**
1.	Sistema de convocatorias de investigación				x	
2.	0					

Tabla 11: Datos del método para medir la productividad del equipo de pruebas- Probador No.1

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLÍN	**Recolección de información acerca de la productividad de equipos de pruebas de software para los probadores y los analistas de pruebas**					**FMT- 001** **Fecha**: 13-06-2013
Nombre:	Probador No.1	**Genero:**	Masculino			
1. Nombre de los proyectos asignados para probar	**2. Dominio de la aplicación**	**3. Meses de experiencia en el dominio de la aplicación**	**4. Meses de experiencia en las técnicas de pruebas del proyecto**	**5. Fecha en que fue retirado(a) del proyecto**	**6. Número de días incapacitado(a)**	**7. Fecha en que ingresó al proyecto**
1. SIIU - Sistema de Información para la Investigación universitaria	Matricula de Proyectos de Investigación	3,0			1	4-II-2013
2.						

9. Seleccione el nivel de educación que corresponde con su formación

Nivel de educación	
Estudiante de pregrado	
Técnico o tecnólogo	
Profesional	X
Especialista	
Magíster	

Tabla 11: Datos del método para medir la productividad del equipo de pruebas- Probador No.1 (Continuación)

10. Registre el número de meses de experiencia que tenga en cada unas de las categorías siguientes

Categoría de experiencia	Número de meses de experiencia
Experiencia en pruebas de software	8
Experiencia en la organización	8

11. Registre el número de horas de capacitación certificada que tiene en las áreas del conocimiento siguientes

Áreas del conocimiento	Horas de capacitación
Fundamentos de pruebas de software: cubre todo los conceptos básicos en el campo de pruebas de software, la terminología básica, los temas principales y su relación con otras actividades	6
Niveles de pruebas: cubre los criterios de adecuación y de selección de las pruebas, con referencia a los objetivos y alcance específicos.	5
Técnicas de pruebas: cubre las técnicas aceptadas por la comunidad científica	5
Medidas relacionadas con las pruebas: cubre las mediciones relacionadas con la evaluación del programa bajo pruebas y la evaluación de la ejecución de las pruebas	5
Proceso de pruebas: cubre las actividades de las etapas del proceso de pruebas y la administración del proceso de pruebas.	6

Tabla 11: Datos del método para medir la productividad del equipo de pruebas-Probador No.1 (Continuación)

8. Clasifique los casos de prueba que tiene asignados de acuerdo con los cuatro atributos siguientes. Por favor, ordene los casos de prueba según la complejidad yendo del caso de prueba de complejidad mas alta al caso de prueba de complejidad más baja, de la siguiente manera:
1. Asigne primero los casos de prueba por el número de componentes de software que involucra
2. Luego asigne los casos de prueba por el número de resultados esperados
3. Luego asigne los casos de prueba por el número de pasos
4. Por ultimo, asigne los casos de prueba por el número de recursos no funcionales que involucra

Id del caso de prueba	Nombre del proyecto	Número de componentes de software que involucra	Número de resultados esperados	Número de pasos	Número de recursos no funcionales que involucra	Orden de complejidad del caso de prueba	Número de minutos	Fecha de Inicio	Fecha de fin	Técnicas y herramientas de gestión de la configuración	Formato y contenido estándar para la documentación
CNVCT-108	SIIU	1	2	10	1	1	7	27-III-2013	27-III-2013	x	x
CNVCT-95	SIIU	1	4	5	3	2	6	26-III-2013	26-III-2013	x	x
CNVCT-14	SIIU	1	3	6	3	3	6	25-III-2013	25-III-2013	x	x
CNVCT-13	SIIU	1	3	7	1	4	6	25-III-2013	25-III-2013	x	x
CNVCT-64	SIIU	1	2	6	3	5	6	18-III-2013	18-III-2013	x	x
CNVCT-10	SIIU	1	1	6	3	6	5	20-III-2013	20-III-2013	x	x
CNVCT-11	SIIU	1	1	6	3	7	5	20-III-2013	20-III-2013	x	x
CNVCT-27	SIIU	1	1	5	2	8	3	19-III-2013	19-III-2013	x	x
CNVCT-65	SIIU	1	3	2	3	9	3	18-III-2013	18-III-2013	x	x
CNVCT-96	SIIU	1	2	3	3	10	3	26-III-2013	26-III-2013	x	x
CNVCT-117	SIIU	1	2	2	3	11	4	27-III-2013	27-III-2013	x	x
CNVCT-124	SIIU	1	3	3	1	12	4	20-III-2013	20-III-2013	x	x
CNVCT-28	SIIU	1	2	4	1	13	4	19-III-2013	19-III-2013	x	x
CNVCT-51	SIIU	1	1	5	1	14	4	20-III-2013	20-III-2013	x	x
CNVCT-52	SIIU	1	3	1	2	15	3	20-III-2013	20-III-2013	x	x
CNVCT-118	SIIU	1	1	1	3	16	3	28-III-2013	28-III-2013	x	x
CNVCT-125	SIIU	1	2	2	1	17	3	20-III-2013	20-III-2013	x	x

Tabla 11: Datos del método para medir la productividad del equipo de pruebas-Probador No.1 (Continuación)

Id del caso de prueba	Nombre del proyecto	Número de componentes de software que involucra	Número de resultados esperados	Número de pasos	Número de recursos no funcionales que involucra	Orden de complejidad del caso de prueba	Número de minutos	Fecha de Inicio	Fecha de fin	Técnicas y herramientas de gestión de la configuración	Formato y contenido estándar para la documentación
CNVCT-109	SIIU	1	2	1	2	18	3	27-III-2013	27-III-2013	x	X
CNVCT-106	SIIU	1	1	2	1	19	2	26-III-2013	26-III-2013	x	X
CNVCT-107	SIIU	1	1	2	1	20	2	27-III-2013	27-III-2013	x	x
CNVCT-120	SIIU	1	1	1	1	21	2	28-III-2013	28-III-2013	x	x
CNVCT-121	SIIU	1	1	1	1	22	2	20-III-2013	20-III-2013	x	x
CNVCT-126	SIIU	1	1	1	1	23	2	20-III-2013	20-III-2013	x	x

Tabla 12: Datos del método para medir la productividad del equipo de pruebas-Probador No.2

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLÍN	**Recolección de información acerca de la productividad de equipos de pruebas de software para los probadores y los analistas de pruebas**				**FMT- 001**	
					Fecha: 13-06-2013	
Nombre:	Probador No.2	**Genero:**	Masculino			
1. Nombre de los proyectos asignados para probar	**2. Dominio de la aplicación**	**3. Meses de experiencia en el dominio de la aplicación**	**4. Meses de experiencia en las técnicas de pruebas del proyecto**	**5. Fecha en que fue retirado(a) del proyecto**	**6. Número de días incapacitado(a)**	**7. Fecha en que ingreso en el proyecto**
1.Sistema de Información para la Investigación Universitaria(SIIU)	Matricula de proyectos de investigación	2,0				08-04-2013
2.						

9. Seleccione el nivel de educación que corresponde con su formación

Nivel de educación	
Estudiante de pregrado	
Técnico o tecnólogo	
Profesional	x
Especialista	
Magíster	

Tabla 12: Datos del método para medir la productividad del equipo de pruebas-Probador No.2 (Continuación)

10. Registre el número de meses de experiencia que tenga en cada unas de las categorías siguientes

Categoría de experiencia	Número de meses de experiencia
Experiencia en pruebas de software	5
Experiencia en la organización	5

11. Registre el número de horas de capacitación certificada que tiene en las áreas del conocimiento siguientes

Áreas del conocimiento	Horas de capacitación
Fundamentos de pruebas de software: cubre todo los conceptos básicos en el campo de pruebas de software, la terminología básica, los temas principales y su relación con otras actividades	7
Niveles de pruebas: cubre los criterios de adecuación y de selección de las pruebas, con referencia a los objetivos y alcance específicos.	6
Técnicas de pruebas: cubre las técnicas aceptadas por la comunidad científica	6
Medidas relacionadas con las pruebas: cubre las mediciones relacionadas con la evaluación del programa bajo pruebas y la evaluación de la ejecución de las pruebas	6
Proceso de pruebas: cubre las actividades de las etapas del proceso de pruebas y la administración del proceso de pruebas.	7

Tabla 12: Datos del método para medir la productividad del equipo de pruebas-Probador No.2 (Continuación)

8. Clasifique los casos de prueba que tiene asignados de acuerdo con los cuatro atributos siguientes. Por favor, ordene los casos de prueba según la complejidad yendo del caso de prueba de complejidad mas alta al caso de prueba de complejidad más baja, de la siguiente manera:
1. Asigne primero los casos de prueba por el número de componentes de software que involucra
2. Luego asigne los casos de prueba por el número de resultados esperados
3. Luego asigne los casos de prueba por el número de pasos
4. Por ultimo, asigne los casos de prueba por el número de recursos no funcionales que involucra

Id del caso de prueba	Nombre del proyecto	Número de componentes de software que involucra	Número de resultados esperados	Número de pasos	Número de recursos no funcionales que involucra	Orden de complejidad del caso de prueba	Número de minutos	Fecha de Inicio	Fecha de fin	Técnicas y herramientas de gestión de la configuración	Formato y contenido estándar para la documentación
CNVCT-87	SIIU-Convoctorias	5	4	5	7	1	20	22 de marzo	22 de marzo	X	X
CNVCT-90	SIIU-Convoctorias	5	1	7	3	2	18	22 de marzo	22 de marzo	X	X
CNVCT-88	SIIU-Convoctorias	4	8	5	1	3	18	22 de marzo	22 de marzo	X	X
CNVCT-86	SIIU-Convoctorias	4	3	3	3	4	18	22 de marzo	22 de marzo	X	X
CNVCT-89	SIIU-Convoctorias	4	1	7	2	5	18	22 de marzo	22 de marzo	X	X
CNVCT-14	SIIU-Convoctorias	3	3	6	3	6	10	26 de marzo	27 de marzo	X	X
CNVCT-83	SIIU-Convoctorias	3	2	5	3	7	12	21 de marzo	21 de marzo	X	X
CNVCT-92	SIIU-Convoctorias	3	2	1	3	8	10	22 de marzo	22 de marzo	X	X
CNVCT-03	SIIU-Convoctorias	3	1	7	3	9	8	20 de marzo	20 de marzo	X	X
CNVCT-79	SIIU-Convoctorias	3	1	3	1	10	8	21 de marzo	21 de marzo	X	X
CNVCT-82	SIIU-Convoctorias	2	8	10	2	11	12	21 de marzo	21 de marzo	X	X
CNVCT-85	SIIU-Convoctorias	2	4	2	3	12	10	21 de marzo	21 de marzo	X	X
CNVCT-84	SIIU-Convoctorias	2	3	3	3	13	10	21 de marzo	21 de marzo	X	X

Tabla 12: Datos del método para medir la productividad del equipo de pruebas- Probador No.2 (Continuación)

Id del caso de prueba	Nombre del proyecto	Número de componentes de software que involucra	Número de resultados esperados	Número de pasos	Número de recursos no funcionales que involucra	Orden de complejidad del caso de prueba	Número de minutos	Fecha de Inicio	Fecha de fin	Técnicas y herramientas de gestión de la configuración	Formato y contenido estándar para la documentación
CNVCT-76	SIIU-Convoctorias	2	3	3	2	14	10	21 de marzo	21 de marzo	X	X
CNVCT-09	SIIU-Convoctorias	2	2	6	1	15	8	20 de marzo	20 de marzo	X	X
CNVCT-77	SIIU-Convoctorias	2	2	2	2	16	8	21 de marzo	21 de marzo	X	X
CNVCT-11	SIIU-Convoctorias	2	1	6	3	17	8	20 de marzo	20 de marzo	X	X
CNVCT-10	SIIU-Convoctorias	2	1	6	3	18	8	20 de marzo	20 de marzo	X	X
CNVCT-01	SIIU-Convoctorias	2	1	6	2	19	8	20 de marzo	20 de marzo	X	X
CNVCT-71	SIIU-Convoctorias	2	1	5	2	20	10	21 de marzo	21 de marzo	X	X
CNVCT-05	SIIU-Convoctorias	2	1	5	1	21	8	20 de marzo	20 de marzo	X	X
CNVCT-07	SIIU-Convoctorias	2	1	4	1	22	8	20 de marzo	20 de marzo	X	X
CNVCT-91	SIIU-Convoctorias	2	1	2	2	23	5	22 de marzo	22 de marzo	X	X

Una vez recolectada la información acerca de la productividad en los formatos propuestos, se pueden realizar diferentes análisis con base en los datos que se muestran en las Tablas 8-11. El análisis que se realiza de la información proporciona al coordinador de pruebas herramientas para diagnosticar como las características propias del proyecto de pruebas y del equipo que se encarga de ejecutar las pruebas impactan la productividad del equipo de pruebas. En el caso de los datos suministrados por SEQUAL S.A. se observan las siguientes características:

1. Las estimaciones que realizó el coordinador de pruebas fueron poco reales (véase Tabla 9), lo cual se puede deber a cuatro razones: porque se presentaron cambios de los requisitos legales durante la ejecución de las pruebas, porque el producto del desarrollo presenta falta de calidad, porque el ambiente de pruebas presenta deficiencia de calidad o porque se presentaron problemas contractuales.
2. La realización de cambios en los requisitos legales o la baja calidad del producto pueden ser consecuencia del grado de innovación del proyecto, ya que el equipo de desarrollo nunca antes hizo desarrollos en dominios semejantes (véase Tabla 9).
3. La baja calidad del producto se refleja en que los requisitos presentan defectos, están incompletos o son poco reales (véase Tabla 9).
4. La deficiencia en la calidad de los requisitos provoca que los probadores tarden más tiempo en entender las necesidades que plantea el cliente en el documento, lo que conlleva a que los probadores deban dedicar mayor tiempo para alcanzar la curva de aprendizaje, o en el peor de los casos deban completar el documento original.
5. Los problemas contractuales entre el cliente y la empresa de calidad pueden ocasionar que la ejecución de las pruebas se detenga, o que las fechas programadas se posterguen.

6. La necesidad de acceder concurrentemente a recursos externos para ejecutar las pruebas, puede generar retrasos en los compromisos planteados si los recursos no se encuentren disponibles en el momento de la solicitud.
7. El ambiente de calidad presenta deficiencias debido a que las pruebas se deben ejecutar con restricciones de memoria (véase Tabla 9).
8. Los probadores encargados de ejecutar las pruebas son profesionales que tienen nivel de experiencia bajo en el dominio de la aplicación y en pruebas de software, su nivel de experiencia en la organización se encuentra entre bajo y medio. Las áreas del conocimiento en que más presentan entrenamiento son: fundamentos de pruebas y proceso de pruebas (véanse Tablas 10-11).
9. La complejidad del caso de prueba presenta una relación directa con el número de recursos funcionales, el número de resultados esperados, el número de pasos y el número de recursos no funcionales. Esto se puede concluir, ya qué según la clasificación que los probadores realizan de los casos de prueba con respecto a estos atributos, los casos de prueba de complejidad más baja tienen menores valores en estos atributos y los casos de prueba de complejidad más alta presentan mayores valores en estos atributos. Asimismo, los casos de prueba de complejidad más baja requieren un menor tiempo de ejecución que los de complejidad más alta (véanse Tablas 10-11).
10. La relación del tiempo de ejecución entre los casos de prueba de complejidad más baja y los casos de prueba de complejidad más alta es del orden de 3 a 4 veces (véanse Tablas 10-11).
11. Los casos de prueba asignados al Probador No.1 (véase Tabla 10), quien tiene más meses de experiencia en la organización, en el dominio de la aplicación y en el proceso de pruebas, son de complejidad mas baja (tienen menor número en los atributos que representan el caso de prueba) comparados con los casos de prueba asignados al Probador No.2 (véase Tabla 11). La diferencia entre el tiempo de ejecución de los casos de prueba de complejidad mas alta asignados al Probador No.1 y los asignados al Probador No.2 es del orden de 2,8 veces.

12. El Probador No.2 tiene experiencia más baja en la organización, en el proceso de prueba y como su vinculación al proyecto se realiza dos meses mas tarde de la fecha de inicio del mismo, tiene menor experiencia en el dominio de la aplicación (véase Tabla 11). Lo anterior sugiere que el proceso de adaptación en un proyecto cuyo dominio no es de conocimiento del probador puede provocar retrasos en el tiempo de ejecución de los casos de prueba.

En las Tablas 12-17 se presentan los datos de los proyectos de prueba que contrató la organización No.2 con SEQUAL S.A.

Tabla 13: Datos del método para medir la productividad del equipo de pruebas-Gerente de proyecto No.2

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLÍN	**Recolección de información acerca de la productividad de equipos de pruebas de software para los gerentes de proyectos**	**FMT- 003**
		Fecha: 27/06/2013
Nombre:	Representante del Cliente No.2	

1. Nombre de los proyectos de software		**2. Descripción del dominio de la aplicación**	**3. Número de requisitos del proyecto**		
1.	HP	Dominio de datos desde diferentes fuentes	1		
2.	Ping	Recolección de ordenes y selección de artículos	1		
3.	Seis	Abastecimiento	4		
4.	Bodegas	Gestor de almacenamiento	4		
5.	Reemplazos	Trazabilidad correspondencia de artículos	1		
6.	Inventario	Existencias	3		
7.	Registro	Transacciones Pago	1		

4. Registre el número de requisitos que fueron modificados, eliminados o anulados en el desarrollo para cada proyecto de software

Nombre de los proyecto de software		**Número de requisitos modificados, eliminados o anulados**
1.	HP	1
2.	Ping	1
3.	Seis	3
4.	Bodegas	2
5.	Reemplazos	4

Tabla 13: Datos del método para medir la productividad del equipo de pruebas-Gerente de proyecto No.2 (Continuación)

Nombre de los proyecto de software		Número de requisitos modificados, eliminados o anulados
6.	Inventario	3
7.	Registro	2

5. Registre el grado de innovación del dominio del proyecto de software de acuerdo con la siguiente escala:
1. Nunca antes hecho
2. Previamente hecho por otros
3. Previamente hecho por el equipo

Nombre de los proyecto de software		Escenarios de innovación del dominio del proyecto de software
1.	HP	1
2.	Ping	2
3.	Seis	2
4.	Bodegas	1
5.	Reemplazos	1
6.	Inventario	1

Tabla 13: Datos del método para medir la productividad del equipo de pruebas-Coordinador de pruebas

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLÍN	**Recolección de información acerca de la productividad de equipos de pruebas de software para los coordinadores de pruebas**	**FMT- 002** **Fecha:**08-06-2013

Nombre:	Coordinador de pruebas No.1									
1. Nombre de los proyectos de pruebas que coordina		**2. Descripción del dominio de la aplicación**	**3. Fecha de Inicio del proyecto**	**4. Fecha de fin del proyecto**	**5. Técnicas de prueba del proyecto**	**6. Número de casos de prueba**	**7. Número de probadores asignados directamente**	**8. Número de personas de soporte disponibles**	**9. Número de probadores retirados a otros proyectos**	**10. Número de probadores que se fueron de la organización**	**11. Número de probadores del proyecto incapacitados**
1.	HP	equivalencia de datos desde diferentes fuentes	abr-03	en curso	Caja negra - Tabla decisiones, Clases de equivalencia, valores límite, basadas en experiencia. Caja gris.	62	1	1	0	0	0
2.	Ping	recolección de artículos	abr-03	en curso	Caja negra - Tabla decisiones, Clases de equivalencia, valores límite, basadas en experiencia. Caja gris.	21	1	1	0	0	0

Tabla 14: Datos del método para medir la productividad del equipo de pruebas-Coordinador de pruebas (Continuación)

1. Nombre de los proyectos de pruebas que coordina		2. Descripción del dominio de la aplicación	3. Fecha de Inicio del proyecto	4. Fecha de fin del proyecto	5. Técnicas de prueba del proyecto	6. Número de casos de prueba	7. Número de probadores asignados directamente	8. Número de personas de soporte disponibles	9. Número de probadores retirados a otros proyectos	10. Número de probadores que se fueron de la organización	11. Número de probadores del proyecto incapacitados
3.	SEIS	Abastecimiento	abr-03	en curso	Caja negra - Tabla decisiones, Clases de equivalencia, valores límite, basadas en experiencia.	124	1	2	0	0	0
4.	Bodegas	Administración de bodegaje	abr-03	en curso	Caja negra - Tabla decisiones, Clases de equivalencia, valores límite, experiencia de usuario. Caja gris.	15	1	2	0	0	0
5.	Reemplazos	Trazabilidad y correspondencia de artículos	abr-03	en curso	Caja negra - Tabla decisiones, Clases de equivalencia, valores límite, basadas en experiencia. Caja gris.	21	1	1	0	0	0

Tabla 14: Datos del método para medir la productividad del equipo de pruebas-Coordinador de pruebas (Continuación)

1. Nombre de los proyectos de pruebas que coordina		2. Descripción del dominio de la aplicación	3. Fecha de Inicio del proyecto	4. Fecha de fin del proyecto	5. Técnicas de prueba del proyecto	6. Número de casos de prueba	7. Número de probadores asignados directamente	8. Número de personas de soporte disponibles	9. Número de probadores retirados a otros proyectos	10. Número de probadores que se fueron de la organización	11. Número de probadores del proyecto incapacitados
6.	Inventario	Existencias	abr-03	en curso	Caja negra - Tabla decisiones, Clases de equivalencia, valores límite, basadas en experiencia. Caja gris.	70	1	1	1	0	0

12. Registre el nivel de participación de los usuarios finales en el proyecto de pruebas de acuerdo con la siguiente escala:
1. Bajo: La participación del usuario es limitada a la etapa posterior a la entrega del proyecto
2. Medio: La participación del usuario esta entre el nivel bajo y alto
3. Alto: El usuario participa en la ejecución de todos los casos de prueba en los que se requiere
4. N/A: El usuario no participa

Nombre de los proyectos de pruebas que coordina		Nivel de participación del usuario
1.	HP	2
2.	Ping	2
3.	SEIS	2
4.	Bodegas	2
5.	Reemplazos	2
6.	Inventario	2

Tabla 14: Datos del método para medir la productividad del equipo de pruebas-Coordinador de pruebas (Continuación)

13. Seleccione los factores limitantes que estuvieron presentes en la ejecución del proyecto de pruebas

Nombre de los proyectos de pruebas que coordina		Materialización de riesgos asociados con la organización				Materialización de los riesgos tecnológicos				Materialización de los riesgos ambientales			
		Falta de personal	Problemas personales entre equipos / miembros del equipo	Cooperación insuficiente entre departamentos	Estimaciones poco reales	Requisitos defectuosos, incompletos o poco reales	Herramientas nuevas que presentan incertidumbre para el desarrollo de software y pruebas	Falta de calidad del producto	Calidad deficiente o alta complejidad del ambiente de pruebas	Deficiencia de terceros en la provisión de componentes	Problemas contractuales	Acceso concurrente a recursos externos	Cambio de requisitos legales
1.	HP												
2.	Ping									x		x	
3.	SEIS	**x**							x				
4.	Bodegas					**X**			x	x		x	
5.	Reemplazos												
6.	Inventario								x				

14. Registre el nivel de criticidad del proyecto de pruebas marcando con una (x) las características que posee el proyecto de acuerdo con la siguiente descripción:

1. Sincronización crítica: Si el producto se debe probar en ambientes donde el comportamiento en tiempo real, la respuesta del usuario o el rendimiento es critico

2. Memoria crítica: Si el producto se debe probar en un ambiente con memoria limitada

3. Calidad o confiabilidad crítica: Si el producto se debe probar con criterios de calidad y confiabilidad muy estrictos

Nombre de los proyectos de pruebas que coordina		Sincronización crítica	Memoria crítica	Calidad o confiabilidad crítica
1	HP	X		X
2.	Ping	X		X
3.	SEIS			X

Tabla 14: Datos del método para medir la productividad del equipo de pruebas-Coordinador de pruebas (Continuación)

Nombre de los proyectos de pruebas que coordina		Sincronización crítica	Memoria crítica	Calidad o confiabilidad crítica
4.	Bodegas			X
5.	Reemplazos			X
6.	Inventario	X		X

15. Registre el grado de innovación del dominio del proyecto de pruebas de acuerdo con la siguiente escala
1. Nunca antes hecho
2. Previamente hecho por otros
3. Previamente hecho por el equipo

	Nombre de los proyectos de pruebas que coordina	**Escenarios de innovación del dominio del proyecto de pruebas**
1.	HP	1
2.	Ping	2
3.	SEIS	2
4.	Bodegas	1
5.	Reemplazos	1
6.	Inventario	1

16. Registre el nivel de complejidad del flujo de control del producto de software, asociado al proyecto de pruebas, de acuerdo con la siguiente escala:
1. Bajo: con líneas de código directas, ciclos, ramas y cálculos simples
2. Medio: con anidación simples, comunicación intermodulo simple y cálculos moderadamente difíciles
3. Alto: con código recursivo, estructuras de datos de control, cálculos complejos con resultados exactos

Tabla 14: Datos del método para medir la productividad del equipo de pruebas-Coordinador de pruebas (Continuación)

	Nombre de los proyectos de pruebas que coordina	Nivel de complejidad de la programación
1.	HP	2
2.	Ping	2
3.	SEIS	2
4.	Bodegas	2
5.	Reemplazos	2
6.	Inventario	2

17. Registre el nivel de complejidad de los datos del proyecto de pruebas de acuerdo con la siguiente escala:
1. Bajo: son simples arreglos o archivos en memoria principal, pocas entradas-salidas, datos no reestructurados
2. Medio: son múltiples entradas-salidas, tipos de datos, datos reestructurados, acceso a otros medios de almacenamiento
3. Alto: son estructuras de datos relacionales altamente acopladas, algoritmos de búsqueda optimizados, antecedentes de consistencia de datos

	Nombre de los proyectos de pruebas que coordina	Nivel de complejidad de los datos de prueba
1.	HP	1
2.	Ping	1
3.	SEIS	1
4.	Bodegas	1
5.	Reemplazos	1
6.	Inventario	1

Tabla 14: Datos del método para medir la productividad del equipo de pruebas-Coordinador de pruebas (Continuación)

18. Seleccione el tipo de escenario para la comunicación y la coordinación del equipo en el que se ejecuta cada proyecto de pruebas

Nombre de los proyectos de pruebas que coordina		Solo una persona	Solo un departamento	Múltiples departamentos	Múltiples sitios	Subcontratación con terceros
1.	HP			x	x	
2.	Ping			x	x	X
3.	SEIS			x		x
4.	Bodegas			x		
5.	Inventario			x		

19. Seleccione el tipo de escenario de prueba en paralelo de cada proyecto de pruebas marcando con una (x) el escenario que corresponda con el proyecto, si aplica

Nombre de los proyectos de pruebas que coordina		Hardware y software concurrente	Hardware y firmware concurrente	Software y firmware concurrente	Software y software concurrente	Hardware, software y firmware concurrente
1.	SIIU				x	
2.	HP				x	
3.	Ping				x	
4.	SEIS				x	
5.	Bodegas				x	
9.	Inventario				x	

Tabla 15: Datos del método para medir la productividad del equipo de pruebas-Probador No.3

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLÍN	**Recolección de información acerca de la productividad de equipos de software para los probadores y analistas de pruebas**				**FMT- 001**	
					Fecha: 28 de junio de 2013	
Nombre:	Probador No.3	**Genero:**	Femenino			
1. Nombre de los proyectos asignados para probar	**2. Dominio de la aplicación**	**3. Meses de experiencia en el dominio de la aplicación**	**4. Meses de experiencia en las técnicas de pruebas del proyecto**	**5. Fecha en que fue retirado(a) del proyecto**	**6. Número de días incapacitado(a)**	**7. Fecha en que ingresó al proyecto**
1. HP	Equivalencia de datos desde diferentes fuentes	2,0	5,0	NA	2	08-04-2013
2.						

9. Seleccione el nivel de educación que corresponde con su formación

Nivel de educación	
Estudiante de pregrado	
Técnico o tecnólogo	
Profesional	X
Especialista	
Magíster	

Tabla 15: Datos del método para medir la productividad del equipo de pruebas-Probador No.3 (Continuación)

10. Registre el número de meses de experiencia que tenga en cada unas de las categorías siguientes

Categoría de experiencia	Número de meses de experiencia
Experiencia en pruebas de software	7
Experiencia en la organización	7

11. Registre el número de horas de capacitación certificada que tiene en las áreas del conocimiento siguientes

Áreas del conocimiento	Horas de capacitación
Fundamentos de pruebas de software: cubre todo los conceptos básicos en el campo de pruebas de software, la terminología básica, los temas principales y su relación con otras actividades	5
Niveles de pruebas: cubre los criterios de adecuación y de selección de las pruebas, con referencia a los objetivos y alcance específicos.	2
Técnicas de pruebas: cubre las técnicas aceptadas por la comunidad científica	6.5
Medidas relacionadas con las pruebas: cubre las mediciones relacionadas con la evaluación del programa bajo pruebas y la evaluación de la ejecución de las pruebas	1
Proceso de pruebas: cubre las actividades de las etapas del proceso de pruebas y la administración del proceso de pruebas.	4.5

Tabla 15: Datos del método para medir la productividad del equipo de pruebas-Probador No.3 (Continuación)

8. Clasifique los casos de prueba que tiene asignados de acuerdo con los cuatro atributos siguientes. Por favor, ordene los casos de prueba según la complejidad yendo del caso de prueba de complejidad mas alta al caso de prueba de complejidad mas baja, de la siguiente manera:
1. Asigne primero los casos de prueba por el número de componentes de software que involucra
2. Luego asigne los casos de prueba por el número de resultados esperados
3. Luego asigne los casos de prueba por el número de pasos
4. Por ultimo, asigne los casos de prueba por el número de recursos no funcionales que involucra

Id del caso de prueba	Nombre del proyecto	Número de componentes de software que involucra	Número de resultados esperados	Número de pasos	Número de recursos no funcionales que involucra	Orden de complejidad del caso de prueba	Número de minutos	Fecha de Inicio	Fecha de fin	Formato y contenido estándar para la documentación
HP-46	HP	2	4	13	7	1	25	28 de mayo 2013	28 de mayo 2013	X
HP-55	HP	2	4	13	7	1	20	5 de junio 2013	5 de junio 2013	X
HP-10	HP	2	4	7	4	2	70	20 de mayo 2013	20 de mayo 2013	X
HP-52	HP	2	3	18	8	3	20	29 de mayo 2013	29 de mayo 2013	X
HP-51	HP	2	3	16	8	4	20	29 de mayo 2013	29 de mayo 2013	X
HP-45	HP	2	3	11	7	5	25	27 de mayo 2013	27 de mayo 2013	X
HP-48	HP	2	3	11	7	5	25	28 de mayo 2013	28 de mayo 2013	X
HP-49	HP	2	3	11	7	5	25	28 de mayo 2013	28 de mayo 2013	X
HP-56	HP	2	3	11	7	5	25	6 de junio 2013	6 de junio 2013	X
HP-06	HP	2	2	13	8	6	30	17 de mayo 2013	17 de mayo 2013	X
HP-07	HP	2	2	13	8	6	30	17 de mayo 2013	17 de mayo 2013	X
HP-11	HP	2	2	13	8	6	30	20 de mayo 2013	20 de mayo 2013	X
HP-12	HP	2	2	13	8	6	30	20 de mayo 2013	20 de mayo 2013	X
HP-20	HP	2	2	13	8	6	120	21 de mayo 2013	24 de mayo 2013	X
HP-22	HP	2	2	13	8	6	30	21 de mayo 2013	21 de mayo 2013	X
HP-05	HP	2	2	9	7	7	45	17 de mayo 2013	17 de mayo 2013	X
HP-21	HP	2	2	9	7	7	30	21 de mayo 2013	21 de mayo 2013	X

Tabla 15: Datos del método para medir la productividad del equipo de pruebas-Probador No.3 (Continuación)

Id del caso de prueba	Nombre del proyecto	Número de componentes de software que involucra	Número de resultados esperados	Número de pasos	Número de recursos no funcionales que involucra	Orden de complejidad del caso de prueba	Número de minutos	Fecha de Inicio	Fecha de fin	Formato y contenido estándar para la documentación
HP-19	HP	2	2	9	5	8	50	20 de mayo 2013	20 de mayo 2013	X
HP-09	HP	2	2	8	4	9	70	20 de mayo 2013	20 de mayo 2013	X
HP-54	HP	2	2	8	4	9	20	5 de junio 2013	5 de junio 2013	X
HP-47	HP	2	2	7	5	10	25	28 de mayo 2013	28 de mayo 2013	X
HP-33	HP	2	1	12	8	11	15	23 de mayo 2013	23 de mayo 2013	X
HP-34	HP	2	1	12	8	11	15	24 de mayo 2013	24 de mayo 2013	X
HP-37	HP	2	1	12	8	11	20	24 de mayo 2013	24 de mayo 2013	x
HP-02	HP	2	1	11	8	12	30	16 de mayo 2013	16 de mayo 2013	x
HP-03	HP	2	1	11	8	12	30	16 de mayo 2013	16 de mayo 2013	x
HP-32	HP	2	1	7	7	13	15	22 de mayo 2013	22 de mayo 2013	x
HP-01	HP	2	1	6	7	14	45	16 de mayo 2013	16 de mayo 2013	x
HP-38	HP	2	1	6	7	14	20	24 de mayo 2013	24 de mayo 2013	x
HP-31	HP	2	1	6	4	15	15	22 de mayo 2013	22 de mayo 2013	x
HP-53	HP	2	1	6	4	15	20	5 de junio 2013	5 de junio 2013	x
HP-39	HP	2	1	5	7	16	20	24 de mayo 2013	24 de mayo 2013	x
HP-40	HP	2	1	5	7	16	20	24 de mayo 2013	24 de mayo 2013	x
HP-41	HP	2	1	5	7	16	20	27 de mayo 2013	27 de mayo 2013	x
HP-42	HP	2	1	5	7	16	20	27 de mayo 2013	27 de mayo 2013	x
HP-04	HP	2	1	5	4	17	45	16 de mayo 2013	16 de mayo 2013	x
HP-50	HP	1	2	7	4	18	25	29 de mayo 2013	29 de mayo 2013	x
HP-28	HP	1	1	12	8	19	45	22 de mayo 2013	22 de mayo 2013	x

Tabla 15: Datos del método para medir la productividad del equipo de pruebas-Probador No.3 (Continuación)

Id del caso de prueba	Nombre del proyecto	Número de componentes de software que involucra	Número de resultados esperados	Número de pasos	Número de recursos no funcionales que involucra	Orden de complejidad del caso de prueba	Número de minutos	Fecha de Inicio	Fecha de fin	Formato y contenido estándar para la documentación
HP-27	HP	1	1	12	8	20	45	21 de mayo 2013	21 de mayo 2013	x
HP-26	HP	1	1	7	7	21	45	21 de mayo 2013	21 de mayo 2013	x
HP-63	HP	1	1	7	4	22	25	6 de junio 2013	6 de junio 2013	X
HP-25	HP	1	1	6	4	23	45	21 de mayo 2013	21 de mayo 2013	X

Tabla 16: Datos del método para medir la productividad del equipo de pruebas-Probador No.4

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLÍN	**Recolección de información acerca de la productividad de equipos de software para los probadores y analistas de pruebas**				**FMT- 001**	
					Fecha: 12-06-2013	
Nombre:	Probador No.4	**Genero:**	Femenino			
1. Nombre de los proyectos asignados para probar	**2. Dominio de la aplicación**	**3. Meses de experiencia en el dominio de la aplicación**	**4. Meses de experiencia en las técnicas de pruebas del proyecto**	**5. Fecha en que fue retirado(a) del proyecto**	**6. Número de días incapacitado(a)**	**7. Fecha en que ingresó al proyecto**
1. PA	Gestión de clientes y pedidos	10,0	37,0	N/A	0	03/04/2013
2. AyR	Costeo de subl. 88	8,0	37,0	N/A	0	03/04/2013
3. PCK	Entrega de pedidos	5,0	37,0	N/A	0	03/04/2013
4. SEIS	Abastecimiento	12,0	37,0	N/A	0	03/04/2013

9. Seleccione el nivel de educación que corresponde con su formación

Nivel de educación	
Estudiante de pregrado	
Técnico o tecnólogo	
Profesional	X
Especialista	
Magíster	

Tabla 16: Datos del método para medir la productividad del equipo de pruebas-Probador No.4 (Continuación)

10. Registre el número de meses de experiencia que tenga en cada unas de las categorías siguientes

Categoría de experiencia	Número de meses de experiencia
Experiencia en pruebas de software	37
Experiencia en la organización	37

11. Registre el número de horas de capacitación certificada que tiene en las áreas del conocimiento siguientes

Áreas del conocimiento	Horas de capacitación
Fundamentos de pruebas de software: cubre todo los conceptos básicos en el campo de pruebas de software, la terminología básica, los temas principales y su relación con otras actividades	16
Niveles de pruebas: cubre los criterios de adecuación y de selección de las pruebas, con referencia a los objetivos y alcance específicos.	16
Técnicas de pruebas: cubre las técnicas aceptadas por la comunidad científica	32
Medidas relacionadas con las pruebas: cubre las mediciones relacionadas con la evaluación del programa bajo pruebas y la evaluación de la ejecución de las pruebas	16
Proceso de pruebas: cubre las actividades de las etapas del proceso de pruebas y la administración del proceso de pruebas.	16

Tabla 16: Datos del método para medir la productividad del equipo de pruebas-Probador No.4 (Continuación)

8. Clasifique los casos de prueba que tiene asignados de acuerdo con los cuatro atributos siguientes. Por favor, ordene los casos de prueba según la complejidad yendo del caso de prueba de complejidad mas alta al caso de prueba de complejidad mas baja, de la siguiente manera:
1. Asigne primero los casos de prueba por el número de componentes de software que involucra
2. Luego asigne los casos de prueba por el número de resultados esperados
3. Luego asigne los casos de prueba por el número de pasos
4. Por ultimo, asigne los casos de prueba por el número de recursos no funcionales que involucra

Id del caso de prueba	Nombre del proyecto	Número de componentes de software que involucra	Número de resultados esperados	Número de pasos	Número de recursos no funcionales que involucra	Orden de complejidad del caso de prueba	Número de minutos	Fecha de Inicio	Fecha de fin	Técnicas y herramientas de gestión de la configuración
6	SEIS	1	1	4	1	1	3	06/12/2013	06/12/2013	X
7	SEIS	1	1	4	1	1	3	06/12/2013	06/12/2013	X
2	SEIS	1	1	3	1	2	5	06/12/2013	06/12/2013	X
4	SEIS	1	1	3	1	2	4	06/12/2013	06/12/2013	X
5	SEIS	1	1	3	1	2	4	06/12/2013	06/12/2013	X
8	SEIS	1	1	3	1	2	2	06/12/2013	06/12/2013	X
9	SEIS	1	1	3	1	2	2	06/12/2013	06/12/2013	X
10	SEIS	1	1	3	1	2	4	06/12/2013	06/12/2013	X
1	SEIS	1	1	2	1	3	1	06/12/2013	06/12/2013	X
3	SEIS	1	1	2	1	3	1	06/12/2013	06/12/2013	X
11	SEIS	1	1	2	1	3	1	06/12/2013	06/12/2013	X

Tabla 17: Datos del método para medir la productividad del equipo de pruebas-Probador No.5

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLÍN	**Recolección de información acerca de la productividad de equipos de software para los probadores y analistas de pruebas**	**FMT- 001** **Fecha:** 10 Junio 2013

Nombre:	Probador No.5	**Genero:**	Masculino

1. Nombre de los proyectos asignados para probar	**2. Dominio de la aplicación**	**3. Meses de experiencia en el dominio de la aplicación**	**4. Meses de experiencia en las técnicas de pruebas del proyecto**	**5. Fecha en que fue retirado(a) del proyecto**	**6. Número de días incapacitado(a)**	**7. Fecha en que ingresó al proyecto**
1. Reemplazos	reemplazos de mercancías agotadas compras detal	2,0	120,0	En proceso	0	03/04/2013
2.						

9. Seleccione el nivel de educación que corresponde con su formación

Nivel de educación	
Estudiante de pregrado	
Técnico o tecnólogo	
Profesional	
Especialista	X
Magíster	

Tabla 17: Datos del método para medir la productividad del equipo de pruebas-Probador No.5

10. Registre el número de meses de experiencia que tenga en cada unas de las categorías siguientes

Categoría de experiencia	Número de meses de experiencia
Experiencia en pruebas de software	120
Experiencia en la organización	24

11. Registre el número de horas de capacitación certificada que tiene en las áreas del conocimiento siguientes

Áreas del conocimiento	Horas de capacitación
Fundamentos de pruebas de software: cubre todo los conceptos básicos en el campo de pruebas de software, la terminología básica, los temas principales y su relación con otras actividades	Mas 500
Niveles de pruebas: cubre los criterios de adecuación y de selección de las pruebas, con referencia a los objetivos y alcance específicos.	120
Técnicas de pruebas: cubre las técnicas aceptadas por la comunidad científica	120
Medidas relacionadas con las pruebas: cubre las mediciones relacionadas con la evaluación del programa bajo pruebas y la evaluación de la ejecución de las pruebas	Cerca de 200
Proceso de pruebas: cubre las actividades de las etapas del proceso de pruebas y la administración del proceso de pruebas.	Mas de 300

Tabla 17: Datos del método para medir la productividad del equipo de pruebas- Probador No.5 (Continuación)

8. Clasifique los casos de prueba que tiene asignados de acuerdo con los cuatro atributos siguientes. Por favor, ordene los casos de prueba según la complejidad yendo del caso de prueba de complejidad mas alta al caso de prueba de complejidad mas baja, de la siguiente manera:

1. Asigne primero los casos de prueba por el número de componentes de software que involucra
2. Luego asigne los casos de prueba por el número de resultados esperados
3. Luego asigne los casos de prueba por el número de pasos
4. Por ultimo, asigne los casos de prueba por el número de recursos no funcionales que involucra

Id del caso de prueba	Nombre del proyecto	Número de componentes de software que involucra	Número de resultados esperados	Número de pasos	Número de recursos no funcionales que involucra	Orden de complejidad del caso de prueba	Número de minutos	Fecha de Inicio	Fecha de fin	Técnicas y herramientas de gestión de la configuración	Formato y contenido estándar para la documentación
RE-01	Reemplazos	3	1	8	0	1	60	20/05/2013	20/05/2013	x	X
RE-02	Reemplazos	3	1	4	0	2	45	20/05/2013	20/05/2013	x	X
RE-03	Reemplazos	2	1	9	0	3	45	23/05/2013	23/05/2013	x	X
RE-04	Reemplazos	2	1	8	0	4	45	20/05/2013	20/05/2013	x	X
RE-05	Reemplazos	2	1	8	0	5	45	20/05/2013	23/05/2013	x	X
RE-06	Reemplazos	2	1	8	0	6	30	23/05/2013	23/05/2013	x	X
RE-07	Reemplazos	2	1	8	0	7	40	23/05/2013	23/05/2013	x	X
RE-08	Reemplazos	2	1	8	0	8	60	23/05/2013	23/05/2013	x	X
RE-09	Reemplazos	2	1	7	0	9	40	20/05/2013	20/05/2013	x	X
RE-10	Reemplazos	2	1	7	0	10	40	20/05/2013	20/05/2013	x	X
RE-11	Reemplazos	2	1	4	0	11	25	20/05/2013	20/05/2013	x	X
RE-12	Reemplazos	2	1	4	0	12	20	23/05/2013	23/05/2013	x	X
RE-13	Reemplazos	2	1	4	0	13	20	23/05/2013	23/05/2013	x	X
RE-14	Reemplazos	2	1	4	0	14	30	16/05/2013	16/05/2013	x	X
RE-15	Reemplazos	2	1	3	0	15	20	20/05/2013	20/05/2013	x	X
RE-16	Reemplazos	2	1	3	0	16	20	20/05/2013	20/05/2013	x	X
RE-17	Reemplazos	2	1	3	0	17	20	20/05/2013	20/05/2013	x	X
RE-18	Reemplazos	2	1	3	0	18	20	20/05/2013	20/05/2013	x	X

Tabla 17: Datos del método para medir la productividad del equipo de pruebas- Probador No.5 (Continuación)

Id del caso de prueba	Nombre del proyecto	Número de componentes de software que involucra	Número de resultados esperados	Número de pasos	Número de recursos no funcionales que involucra	Orden de complejidad del caso de prueba	Número de minutos	Fecha de Inicio	Fecha de fin	Técnicas y herramientas de gestión de la configuración	Formato y contenido estándar para la documentación
RE-19	Reemplazos	2	1	3	0	19	20	16/05/2013	16/05/2013	x	X
RE-20	Reemplazos	2	1	2	0	20	15	20/05/2013	20/05/2013	x	X
RE-21	Reemplazos	2	1	2	0	21	15	20/05/2013	20/05/2013	x	X
RE-GUI	Reemplazos	1	32	64	0	22	120	17/05/2013	22/05/2013	x	X

Tabla 18: Datos del método para medir la productividad del equipo de pruebas- Probador No.6

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLIN	**Recolección de información acerca de la productividad de equipos de software para los probadores y analistas de pruebas**	**FMT- 001** **Fecha:** 10-06-2013

Nombre:	Probador No.6	**Genero:**	Femenino			
1. Nombre de los proyectos asignados para probar	**2. Dominio de la aplicación**	**3. Meses de experiencia en el dominio de la aplicación**	**4. Meses de experiencia en las técnicas de pruebas del proyecto**	**5. Fecha en que fue retirado(a) del proyecto**	**6. Número de días incapacitado(a)**	**7. Fecha en que ingresó al proyecto**
1. Ping	Recolección de artículos	2,0	7,0		0	03-04-2013
2.						

9. Seleccione el nivel de educación que corresponde con su formación

Nivel de educación	
Estudiante de pregrado	
Técnico o tecnólogo	X
Profesional	
Especialista	
Magíster	

Tabla 18: Datos del método para medir la productividad del equipo de pruebas- Probador No.6 (Continuación)

10. Registre el número de meses de experiencia que tenga en cada unas de las categorías siguientes

Categoría de experiencia	Número de meses de experiencia
Experiencia en pruebas de software	27
Experiencia en la organización	2

11. Registre el número de horas de capacitación certificada que tiene en las áreas del conocimiento siguientes

Áreas del conocimiento	Horas de capacitación
Fundamentos de pruebas de software: cubre todo los conceptos básicos en el campo de pruebas de software, la terminología básica, los temas principales y su relación con otras actividades	27
Niveles de pruebas: cubre los criterios de adecuación y de selección de las pruebas, con referencia a los objetivos y alcance específicos.	27
Técnicas de pruebas: cubre las técnicas aceptadas por la comunidad científica	27
Medidas relacionadas con las pruebas: cubre las mediciones relacionadas con la evaluación del programa bajo pruebas y la evaluación de la ejecución de las pruebas	27
Proceso de pruebas: cubre las actividades de las etapas del proceso de pruebas y la administración del proceso de pruebas.	27

Tabla 18: Datos del método para medir la productividad del equipo de pruebas- Probador No.6 (Continuación)

8. Clasifique los casos de prueba que tiene asignados de acuerdo con los cuatro atributos siguientes. Por favor, ordene los casos de prueba según la complejidad yendo del caso de prueba de complejidad mas alta al caso de prueba de complejidad mas baja, de la siguiente manera:

1. Asigne primero los casos de prueba por el número de componentes de software que involucra
2. Luego asigne los casos de prueba por el número de resultados esperados
3. Luego asigne los casos de prueba por el número de pasos
4. Por ultimo, asigne los casos de prueba por el número de recursos no funcionales que involucra

Id del caso de prueba	Nombre del proyecto	Número de componentes de software que involucra	Número de resultados esperados	Número de pasos	Número de recursos no funcionales que involucra	Orden de complejidad del caso de prueba	Número de minutos	Fecha de Inicio	Fecha de fin
P-08	Ping	1	1	6	3	1	15	26/06/2013	26/06/2013
P-09	Ping	1	1	5	3	2	40	26/06/2013	26/06/2013
P-12	Ping	1	1	4	2	3	20	05/06/2013	05/06/2013
P-13	Ping	1	1	5	3	4	5	26/06/2013	26/06/2013
P-14	Ping	1	1	5	3	5	185	26/06/2013	26/06/2013
P-15	Ping	1	1	4	2	6	20	05/06/2013	05/06/2013
P-18	Ping	1	1	12	2	7	180	27/06/2013	27/06/2013
P-19	Ping	1	1	11	2	8	180	27/06/2013	27/06/2013
P-05	Ping	1	1	2	3	9	60	26/06/2013	26/06/2013
P-06	Ping	1	1	2	3	10	60	26/06/2013	26/06/2013
P-07	Ping	1	1	2	2	11	20	05/06/2013	05/06/2013
P-17	Ping	1	1	2	1	12	5	26/06/2013	26/06/2013
P-01	Ping	1	1	6	2	13	30	04/06/2013	04/06/2013
P-02	Ping	1	1	6	2	14	30	04/06/2013	04/06/2013
P-03	Ping	1	1	3	2	15	30	04/06/2013	04/06/2013
P-04	Ping	1	1	2	1	16	-	-	-
P-10	Ping	1	1	2	2	17	5	26/06/2013	26/06/2013
P-11	Ping	1	1	4	1	18	5	05/06/2013	05/06/2013
P-16	Ping	1	1	2	1	19	10	26/06/2013	26/06/2013

Mediante la evaluación de los datos que se muestran en las Tablas 12-17 es posible identificar las siguientes características acerca de la productividad del equipo de pruebas:

1. Los proyectos de pruebas los contrató una empresa de gran envergadura que tiene estándares de calidad y confiabilidad muy estrictos. Por este motivo se dedica la mayor parte del personal de la empresa a la ejecución y el soporte de las pruebas.
2. Los casos de prueba que se diseñan para los proyectos de prueba equivalen a necesidades generales que expresa el representante del cliente, quien conoce muy bien la lógica del negocio pero que no es parte del grupo desarrollador (véase Tabla 12). Por ello, se entiende que los probadores asignados tienen altas capacidades en lo que se refiere a la fundamentación de las pruebas, los niveles de prueba y el proceso de pruebas.
3. Las técnicas de prueba aplicadas en los proyectos demandan un alto nivel de experiencia en el proceso de pruebas, en las técnicas de prueba y en la organización (véase Tabla 13).
4. En el proyecto que SEQUAL S.A. denomina como SEIS se presenta falta de personal (véase Tabla 13). Esta situación puede tener las siguientes causas: el número de casos de prueba es el más alto de todos los proyectos, el proceso presenta calidad deficiente del ambiente de pruebas y no se tiene experiencia en procesos de prueba en el mismo dominio. Adicionalmente, la ejecución de la pruebas se realiza en conjunto con terceros, lo cual demanda que el probador asignado tenga capacidades de trabajo en equipo para alcanzar una comunicación eficiente y cumplir los objetivos de las pruebas.
5. Las técnicas de caja gris que se aplican en el 80% de los proyectos requieren la comunicación permanente con el desarrollador, o el representante del cliente, con los conocimientos técnicos necesarios (véase Tabla 13). Por este motivo, la ejecución de las pruebas depende de la disponibilidad de horario, los conocimientos y las capacidades de terceros.
6. El proyecto que SEQUAL S.A. denomina como Ping presenta las siguientes características (véase Tabla 13): las pruebas se ejecutan entre múltiples

departamentos, múltiples sitios y con subcontratación con terceros; se presenta sincronización crítica en la ejecución de las pruebas, la provisión de los componentes por parte de terceros es deficiente y se requiere acceso concurrente a recursos externos. Estas características hacen que este proyecto se pueda denominar como crítico en cuanto a demanda de tiempo en la ejecución de sus casos de prueba. En el seguimiento que el Probador No.6, quien se asignó a este proyecto, realiza de los casos de prueba se observa que presenta los casos de prueba con tiempos de ejecución más altos, algunos con duración de más de 3 horas (véase Tabla 17).

7. Por los valores de los atributos de los casos de prueba del proyecto que SEQUAL S.A. denomina como HP (véase Tablas 14), se puede concluir que el proyecto accede a una cantidad alta de recursos no funcionales, por lo que cabe suponer que es un proyecto en que frecuentemente se deben verificar la disponibilidad o solicitar permisos de acceso a los recursos. Particularmente, en la ejecución de los casos de prueba de este proyecto se observa una relación diferente entre los tiempos de ejecución de sus casos de prueba, por ejemplo, existen casos de prueba con los mismos valores en sus atributos que presentan diferencias de mas del doble de tiempo de ejecución, lo que permite suponer que estos retrasos se presentan por la necesidad de acceso a los recursos.
8. En los casos de prueba que ejecutó el Probador No.5 (véase Tabla 16), el tiempo de ejecución del caso de prueba clasificado como de complejidad más alta (que contempla tres recursos funcionales, un resultado esperado y ocho pasos), es la mitad del tiempo de ejecución del caso de prueba de complejidad más baja (que incluye un recurso funcional, treinta y dos resultados esperados y sesenta y cuatro pasos).
9. Los probadores con experiencia alta en las cuatro categorías y mayor capacitación en las áreas del conocimiento presentan en general resultados más estables y predecibles entre los diferentes niveles de complejidad de los casos de prueba (véanse Tablas 14-17). Por ejemplo, los casos de prueba con valores más altos en sus atributos tienen tiempos de ejecución más altos. De

igual manera, el tiempo de ejecución del caso de prueba decrece cuando disminuyen los valores de sus atributos.

Las diferencias que presentan las dos organizaciones dueñas de los productos de software tienen impacto sobre los tiempos de ejecución de las pruebas. La primera organización tiene plazos más flexibles y niveles de confidencialidad y calidad menos estrictos, por lo que se presentan situaciones como: estimaciones poco reales, retraso en las entregas, cambio en los requisitos legales, rotación de personal y baja calidad del producto, entre otras. Sin embargo, los proyectos de prueba presentan condiciones más estables, las cuales se evidencian en la relación entre la complejidad de los casos de prueba y los tiempos de ejecución. Por otra parte, la segunda organización tiene altos estándares de calidad y confidencialidad, los proyectos de software pertenecen a diferentes frentes de la organización, por lo tanto, tienen diferentes coordinadores de desarrollo. Las pruebas involucran diferentes tipos de recursos cuya disponibilidad es baja. Finalmente, los tiempos de entrega son críticos para la organización, por lo que las estimaciones deben ser precisas, las pruebas deben tener el alcance y resultado esperado.

6. CONCLUSIONES Y TRABAJO FUTURO

6.1 Conclusiones

Medir la productividad del equipo de pruebas es una necesidad que se evidencia en los métodos de estimación del esfuerzo de pruebas. Estos métodos usan factores de productividad que estiman mediante el uso de técnicas que, en su mayoría, requieren bases de datos históricas. Por esta razón, en esta Tesis se propuso un método que permite:

- Medir la productividad del equipo de pruebas desde las características del proceso de pruebas que tienen impacto sobre la productividad
- Identificar y definir sin ambigüedades las características que afectan la productividad del equipo de pruebas
- Caracterizar numéricamente mediante una unidad y una escala cada característica identificada
- Seguir una secuencia estructurada para recolectar la información de cada característica desde los actores que administran dicha información

Para validar las características que se proponen se usó una herramienta conocida como panel *Delphi*. Con los resultados obtenidos en el estudio, se puede determinar cuáles características impactan la productividad y en qué sentido. Adicionalmente, se obtienen estimaciones de la proporción de impacto sobre la productividad del equipo de pruebas.

Para validar la aplicabilidad del método propuesto se aplicó el método de medición en el equipo de pruebas de una empresa del sector privado que se dedica a prestar servicios de calidad en *outsourcing*, los cuales se enfocan en pruebas de software. Cada actor del proceso de ejecución de pruebas se entrevistó mediante el uso de un formato que recoge la información que representa las características que el método propone. Mediante la información recolectada se pudieron identificar las condiciones actuales del equipo de pruebas, información que es de

utilidad para las estimaciones que realizan los coordinadores de los equipos de prueba. Adicionalmente, el método aportó la información de las características de la productividad del equipo de pruebas, que las estimaciones del esfuerzo de pruebas necesitan.

6.2 Trabajo Futuro

En esta Tesis se dejó a disposición de los coordinadores de pruebas un conjunto de datos que representan las características que impactan la productividad del equipo de pruebas. Esta información requiere un análisis, que se puede realizar partiendo de los resultados obtenidos en el panel *Delphi*, para producir índices que el coordinador de pruebas pueda usar para mejorar las condiciones del equipo de pruebas. Se propone, también, como trabajo futuro validar el método mediante experimentación controlada, con el objetivo de generar modelos predictivos.

El estudio Delphi arrojó unos valores de impacto sobre la productividad del equipo de pruebas que se pueden ampliar mediante el uso de otros estudios empíricos o experimentales que se complementen con el objetivo de obtener una visión más amplia de la productividad en pruebas.

Otro trabajo que se puede derivar de este estudio es la definición de un método que permita la estimación de la productividad en etapas tempranas del ciclo de vida del desarrollo, para lograr una planeación y asignación de recursos cercana a las necesidades reales.

REFERENCIAS

- Abhishek, C., Kumar, V. P., Vitta, H., & Srivastava, P. R. (2010). Test Effort Estimation Using Neural Network. *Journal of Software Engineering and Applications*, *3*, 331–340.
- Aranha, E., & Borba, P. (2007). An Estimation Model for Test Execution Effort. *First International Symposium on Empirical Software Engineering and Measurement*, 107–116.
- Aranha, E., & Borba, P. (2009). Estimating manual test execution effort and capacity based on execution points. *International Journal of Computers & Applications*, *31*(3), 167–172.
- Aranha, E., Borba, P., & Lima, J. (2006). Model Simulation for Test Execution Capacity Estimation. *17th International Symposium on Software Reliability Engineering*, 231–236.
- Bourque, P., Robert, F., Lavoie, J. M., Lee, A., Trudel, S., & Lethbridge, T. C. (2002). Guide to the Software Engineering Body of Knowledge (SWEBOK) and the Software Engineering Education Knowledge (SEEK) - a preliminary mapping. *10th International Workshop on Software Technology and Engineering Practice*, 8–23.
- Coelli, T. J., Rao, D. P., O'Donnell, C. J., & Battese, G. E. (2005). *An introduction to efficiency and productivity analysis*. New York, USA: Springer Science + Business Media.
- De Almeida, E. R. C., De Abreu, B. T., & Moraes, R. (2009). An Alternative Approach to Test Effort Estimation Based on Use Cases. *International Conference on Software Testing Verification and Validation*, 279–288.
- Everett, G. D., & McLeod, R. (2007). *Software testing: testing across the entire software development life cycle*. New Jersey, USA: Wiley-IEEE Computer Society Press.
- Farooq, S. U., Quadri, S. M. K., & Ahmad, N. (2011). Software Measurements And Metrics: Role In Effective Software Testing. *International Journal of Engineering Science and Technology*, *3*, 671–680.

- Ferreira, M., Garcia, F., Ruiz, F., Bertoa, M. F., Vallecillo, A., Piattini, M., & Mora, B. (2006). *Medición del Software Ontología y Metamodelo* (Report UCLM-TSI-001*)*. Retrieved from Universidad de Castilla la Mancha, Departamento de Tecnologías y Sistemas de la Información website: http://www.uclm.net/dep/tsi/pdf/UCLM-TSI-001.pdf
- García, F., Bertoa, M. F., Calero, C., Vallecillo, A., Ruíz, F., Piattini, M., & Genero, M. (2006). Towards a consistent terminology for software measurement. *Information and Software Technology*, *48*, 631–644.
- Habra, N., Abran, A., Lopez, M., & Sellami, A. (2008). A framework for the design and verification of software measurement methods. *Journal of Systems and Software*, *81*, 633–648.
- Harrold, M. J. (2000). Testing: A Roadmap. *22nd International Conference on Software Engineering Future of Software Engineering,* 61–72.
- IEEE Standard for Software Productivity Metrics. (1993). *IEEE Std 1045-1992*.
- IEEE Standard for a software quality metrics (1998). *IEEE Std 1061-1998.*
- Jacquet, J. P., & Abran, A. (1997). From software metrics to software measurement methods: a process model. *Third IEEE International Software Engineering Standards Symposium and Forum*, 128–135.
- Jones, C. (1998). Conflict and litigation between software clients and developers. *IEEE Engineering Management Review*, *26*, 46–54.
- Jones, C. (2008). *Applied Software Measurement. Global Analysis of Productivity and Quality*. New York, USA: McGraw-Hill.
- Juristo, N., Moreno, A. M., & Vegas, S. (2002). A survey on testing technique empirical studies: how limited is our knowledge. *International Symposium on Empirical Software Engineering*, 161–172.
- Kitchenham, B., & Mendes, E. (2004). Software productivity measurement using multiple size measures. *Software Engineering, IEEE Transactions on*, *30*(12), 1023–1035.
- Leung, H. K. N. (1998). Test tools for the Year 2000 challenges. *24th Euromicro Conference,* 830-837.

- Ley de educación superior, Ley 30 de Diciembre 28, Cong. (1992) (promulgada).
- Ley 30 , Cong. (1993) (promulgada).
- Listone, H. A., & Turoff, M. (1975). *The Delphi method: Techniques and applications*. Boston, USA: Addison-Wesley Publishing Company.
- Mizuno, O., Shigematsu, E., Takagi, Y., & Kikuno, T. (2002). On estimating testing effort needed to assure field quality in software development. *13th International Symposium on Software Reliability Engineering,* 139–146.
- Myers, G. J., Sandler, C., Badgett, T., & Thomas, T. M. (2004). *The Art of Software Testing*. New Jersey, USA: Wiley.
- Nageswaran, S. (2001). Test effort estimation using use case points. *Quality Week*, 1–6.
- Silva, D., De Abreu, B. T., & Jino, M. (2009). A simple approach for estimation of execution effort of functional test cases. *International Conference on Software Testing Verification and Validation*, 289–298.
- Skulmoski, G. J., Hartman, F. T., & Krahn, J. (2007). The Delphi method for graduate research. *Journal of information technology education*, *6*(1), 1–21.
- Systems and software engineering--Vocabulary. (2010). *ISO/IEC/IEEE 24765:2010*, 1–418.
- Tian, J. (2005). *Software Quality Engineering:Testing, Quality Assurance, and Quantifiable Improvement*. Chichester, England: Wiley-IEEE Press.
- Veenendaal, E. van, & Dekkers, T. (1999). Test point analysis: a method for test estimation. In R. J. Kusters, A. Cowderoy, F. J. Heemstra, & E. V. Veenendaal (Eds.), *Project control for software quality* (pp. 45–59). Maastricht, the Netherlands: Shaker Publishing BV.
- Veenendaal, E. van, & Pol, M. (1997). A test management approach for structured testing. In E. V. Veenendaal, & J. McMullan (Eds.), *Achieving Software Product Quality* (pp. 145- 158). Den Bosch, the Netherlands: UTN publishers.
- Villavicencio, M., & Abran, A. (2012). Software Measurement in Software Engineering Education: A Delphi Study to Develop a List of Teaching Topics

and Related Levels of Learning. *38th Euromicro Conference on Software Engineering and Advanced Applications*, 357–362.

- Weyuker, E. J., Ostrand, T. J., Brophy, J., & Prasad, B. (2000). Clearing a career path for software testers. *Software, IEEE*, *17*(2), 76–82.
- Xiaochun, Z., Bo, Z., Fan, W., Yi, Q., & Lu, C. (2008). Estimate Test Execution Effort at an Early Stage: An Empirical Study. *International Conference on Cyberworlds*, 195–200.
- Xiaochun, Z., Bo, Z., Li, H., Junbo, C., & Lu, C. (2008). An Experience-Based Approach for Test Execution Effort Estimation. *The 9th International Conference for Young Computer Scientists*, 1193–1198.
- Yi, Q., Bo, Z., & Xiaochun, Z. (2008). Early Estimate the Size of Test Suites from Use Cases. *15th Asia-Pacific Software Engineering Conference*, 487–492.
- Zapata, C. M., Giraldo, G. L., & Londoño, S. (2011). Esquemas preconceptuales ejecutables. *Revista Avances en Sistemas e Informática*, *8*(1), 15–23.
- Zapata, C. M., Ochoa, O. A., & Cardona, C. de J. (2011). *CICLO-P: Un Metodo para el Acoplamiento de Pruebas al Ciclo de Vida del Software*. Medellin, Colombia: Carlos Mario Zapata (Ed.).