

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**ESTUDIO DE PRÁCTICAS DE CALIDAD EN LOS PROCESOS DE
DESARROLLO DE SOFTWARE EN UNIVERSIDADES DEL
ECUADOR**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL GRADO DE
MAGISTER EN SOFTWARE MENCIÓN CALIDAD**

GRACIELA SUSANA GONZÁLEZ FLORES

graciela.gonzalez@epn.edu.ec

Director: EDISON FERNANDO LOZA AGUIRRE, PhD

edison.loza@epn.edu.ec

QUITO, JUNIO 2019

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Graciela Susana González Flores bajo mi supervisión.

Edison Fernando Loza Aguirre, PhD

DIRECTOR

DECLARACIÓN

Yo, Graciela Susana González Flores, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Ing. Graciela Susana González Flores

DEDICATORIA

Dedico este trabajo a mis queridos y muy incansables padres, Segundo González y María Flores, quienes una vez me dijeron “*no te daremos riquezas, pero eso sí, por nuestra parte no te faltará el estudio*”, sin darse cuenta de que el estudio es la mayor herencia que unos padres amorosos, como ellos, nos lo han entregado a cada uno de mis hermanos y hermanas.

Por ser mi cómplice, mi apoyo incondicional en todo momento desde que nos conocimos, este trabajo, también lo dedico con todo mi corazón a mi amado esposo Edison Taipe, a quién considero la mayor bendición en mi vida por parte de Dios.

Graciela

AGRADECIMIENTO

En primer lugar, agradezco infinitamente a Dios por ser quién me cuida, guía en cada propósito profesional y personal de mi vida.

A Edi (Dr. Edison Loza), quién me acompañó con su conocimiento en la dirección y culminación de este mi proyecto de titulación, pero más aún, por ser una persona sencilla y humilde, cualidades de un docente-amigo.

A cada uno de los directores/ras o jefes del área de desarrollo de las universidades y escuelas politécnicas del Ecuador que participaron en esta investigación (UCE, PUCE, U. CUENCA, ESPE, ESPOL, UASB, UISEK, FLACSO y EPN), los cuales mostraron desde el primer contacto su predisposición y apertura de colaboración.

A Do, Nancita, Cindy, Miriamcita Nicolalde y Xime Rojas, por sus buenas vibras.

A cada uno de los miembros y descendientes de mi gran familia GONZÁLEZ-FLORES por sus palabras, consejos y alientos de continuar preparándome profesionalmente.

Y especialmente, a mi compañero de vida, mi cómplice, mi amado esposo Edison Taipe, quién me impulsó y apoyó incondicionalmente a continuar escalando peldaños personales y profesionales.

A todos muchísimas gracias☺.

Graciela

ÍNDICE DE CONTENIDO

LISTA DE FIGURAS	i
LISTA DE TABLAS	ii
LISTA DE ANEXOS	iii
LISTA DE ABREVIATURAS	iv
RESUMEN.....	vi
<i>ABSTRACT</i>	vii
1 INTRODUCCIÓN.....	1
1.1 PLANTEAMIENTO DEL PROBLEMA.....	1
1.2 OBJETIVO GENERAL.....	4
1.3 OBJETIVOS ESPECÍFICOS.....	4
1.4 MARCO TEÓRICO.....	4
1.4.1 Revisión sistemática de literatura	4
1.4.2 Software	6
1.4.3 Estándares-Norma y Modelos de Calidad en los Procesos de Software	8
1.4.4 Frenos u Obstáculos en Calidad de Software	23
1.4.5 TIC's.....	24
2 METODOLOGÍA.....	29
2.1 MUESTRA.....	29
2.2 DISEÑO DE LA INVESTIGACIÓN	30
2.2.1 Recolección de datos	30
2.2.2 Transcripción.....	32
2.2.3 Confidencialidad	32
2.2.4 Guía o árbol de codificación	33
3 RESULTADOS Y DISCUSIÓN.....	37
3.1 RESULTADOS	37
3.1.1 Definición de la Calidad de Software (QOSdef)	37

3.1.2	Expectativas de la Calidad de Software (QOSexp).....	38
3.1.3	Motivadores de la Calidad de Software (QOSmot)	41
3.1.4	Frenos de la Calidad de Software (QOSfre)	43
3.1.5	Modelos de Calidad de Software (QOSmod).....	44
3.1.6	Prácticas en Calidad de Software (PRAant, PRAact, PRAfut)	45
3.1.7	Resumen	50
3.2	DISCUSIÓN.....	50
3.2.1	Definición de la Calidad de Software	51
3.2.2	Expectativas y Motivadores de la Calidad de Software.....	51
3.2.3	Frenos de la Calidad de Software.....	52
3.2.4	Modelos de la Calidad de Software	52
3.2.5	Prácticas en Calidad de Software	55
4	RECOMENDACIONES PARA LA MEJORA DE CALIDAD EN LOS PROCESOS DE DESARROLLO DE SOFTWARE EN UNIVERSIDADES DEL ECUADOR	56
5	CONCLUSIONES Y RECOMENDACIONES	61
5.1	CONCLUSIONES.....	61
5.2	RECOMENDACIONES.....	63
6	REFERENCIAS BIBLIOGRÁFICAS.....	65
7	ANEXOS.....	68

LISTA DE FIGURAS

Figura 1 - Procesos de Evaluación, Acreditación y Clasificación Sistema de Educación Superior (2009-2018) [3], [4].....	2
Figura 2 - Calidad del Software [15].....	7
Figura 3 - Relación entre el Modelo de Evaluación del Proceso (PAM) y sus entradas [22]	10
Figura 4 - Procesos en el PAM [22].....	11
Figura 5 - Representación de la estructura de la Norma ISO 9001:2015 en el ciclo PDCA [24].	14
Figura 6 - Estructura de Bootstrap V3.0 [27]	18
Figura 7 - Estructura PSP [25]	19
Figura 8 - Componentes del Modelo CMMi [32]	22
Figura 9 - Estructura y áreas de proceso de las representaciones continua y por etapas [32].....	23
Figura 10 - Comparativo de Años de la Calidad de TI – Software [33]	26
Figura 11 - Metodología Formal de Desarrollo de Software [34]	27
Figura 12 - Sistemas de Información [34]	28
Figura 13 – Guía de Entrevista	31
Figura 14 - Guía o árbol de codificación general	34
Figura 15 - Guía o árbol de codificación - Calidad de Software	34
Figura 16 - Clasificación de frenos de Calidad de Software en Universidades Ecuatorianas.....	43
Figura 17 - Frenos de la calidad de software según Universidades Ecuatorianas.....	44
Figura 18 - Modelo/Metodología/Estándar mencionada en el contexto de Calidad de Software en las Universidades Ecuatorianas	45

LISTA DE TABLAS

Tabla 1 - Resultados de la búsqueda inicial en las fuentes	5
Tabla 2 - Estándares-Norma y Modelos de Calidad en los Procesos de Software	9
Tabla 3 - Niveles de capacidad y atributos de proceso	13
Tabla 4 - Lista de frenos u obstáculos en Calidad de Software	24
Tabla 5 - Herramientas para la implantación del Gobierno de las TI [20]	25
Tabla 6 - Listado de universidades ecuatorianas participantes de la entrevista.....	30
Tabla 7 - Codificación de Universidades y su número de participantes en la investigación.....	33
Tabla 8 - Subcategorías Identificadas	35
Tabla 9 - Matriz Genérica de Codificación	36
Tabla 10 - Resultado del análisis de contenido de las intervenciones	37
Tabla 11 - Definiciones de Calidad de Software.....	38
Tabla 12 - Listado de expectativas de Calidad de Software	39
Tabla 13 - Motivadores de la Calidad de Software.....	42
Tabla 14 - Prácticas de Calidad de Software implementadas o por implementar en las Universidades del Ecuador	46
Tabla 15 - Resumen del Estudio.....	50
Tabla 16 - Descripción y ámbito de Aplicación de los términos presentados en el contexto Modelos de Calidad de Software	53
Tabla 17 – Recomendaciones para la mejora de calidad en los procesos de desarrollo de software en universidades del Ecuador – Basado en el modelo CMMiDEV	56

LISTA DE ANEXOS

Anexo I - Información para el participante y su consentimiento.....	68
Anexo II - Anonimizado de la Transcripción.....	68
Anexo III - Matriz de Codificación.....	68

LISTA DE ABREVIATURAS

CACES	Consejo de Aseguramiento de la Calidad de la Educación Superior
CEAACES	Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior
CES	Consejo de Educación Superior
CMMi	<i>Capability Maturity Model Integration</i> , Modelo de Madurez de Capacidad Integrado
EEIG	<i>European Economic Interest Grouping</i> , Grupo de Interés Económico Europeo
ESPRIT	<i>European Strategic Programme for Reserch in Information Technology</i> , Programa Estratégico Europeo de Investigación en Tecnología de la Información
IEC	<i>International Electrotechnical Commission</i> , Comisión Electrotécnica Internacional
IA	Inteligencia Artificial
IEEE	<i>Institute of Electrical and Electronics Engineers</i> , Instituto de Ingenieros Eléctricos y Electrónicos
IES	Instituciones de Educación Superior
ISACA	<i>Information Systems Audit and Control Association</i> , Asociación de Auditoría y Control de Sistemas de Información
ISO	<i>International Organization for Standardization</i> , Organización Internacional de Normalización
LOES	Ley Orgánica de Educación Superior
PA	<i>Process Attributes</i> , Atributos de proceso
PAM	<i>Process Assesment Model</i> , Modelo de Evaluación del Proceso
PRM	<i>Process Reference Model</i> , Modelo de Proceso de Referencia
PSP	<i>Personal Software Process</i> , Proceso de Software Personal
SEI	<i>Software Engineering Institute</i> , Instituto de Ingeniería en Software
SENESCYT	Secretaría de Educación Superior, Ciencia, Tecnología e Innovación
SES	Sistema de Educación Superior
SIAC	Sistema Interinstitucional de Aseguramiento de la Calidad
SP	<i>Specific Práctices</i> , Prácticas Específicas
SPICE	<i>Software Process Improvement and Capability dEtermination</i> , Determinación de la Capacidad de Mejora del Proceso de Software

SPU *Software Producing Unit*, Unidad de Desarrollo Software
TIC's Tecnologías de Información y Comunicación
TSP *Team Software Process*, Proceso de Software de Equipo
UCM Universidad de Carnegie Mellon

RESUMEN

Hoy en día, las universidades y escuelas politécnicas del Ecuador se apoyan en el uso de tecnología de vanguardia para cubrir los parámetros de aseguramiento de la calidad que los organismos de control exigen. Entre estos activos podemos encontrar a los sistemas de información para la gestión de sus procesos Core. Sistemas, cuyo desarrollo de software ha sido tradicionalmente interno y donde se manipula una gran cantidad de información sensible. Por lo tanto, dentro de una perspectiva de mejora continua, es necesario investigar cuáles son las prácticas implementadas o por implementar por las instituciones de educación superior para gestionar y garantizar la calidad en los procesos de desarrollo de software.

Al no existir un estudio previo, en el presente proyecto se siguió una metodología de investigación exploratoria cualitativa, cuya técnica empleada para la recopilación de información fue la entrevista semidirigida a directores y jefes de desarrollo de varias universidades del país (nueve instituciones y 13 participantes), cuyo discurso fue analizado y clasificado.

En el presente trabajo, se reportan los resultados del estudio en torno a las expectativas, los motivadores (proactivos y reactivos), los frenos u obstáculos, los modelos y las prácticas (anteriores, actuales y futuras) en el contexto de la calidad de software, finalizando con las conclusiones y recomendaciones del autor del presente trabajo.

Palabras clave: Calidad de software, Prácticas, Procesos, Universidades y Escuelas Politécnicas Ecuatorianas.

ABSTRACT

Nowadays, the universities and polytechnic schools of Ecuador rely on the use of state-of-the-art technology to cover the parameters of quality demanded by their control organisms. Among these assets we can find information systems for the management of their core processes. These systems requires software, whose development has been traditionally internal and where a large amount of sensitive information is manipulated. Therefore, within a perspective of continuous improvement, it is necessary to investigate which practices are implemented or be implemented by higher education institutions to manage and guarantee quality in software development processes.

In the absence of a previous study, a methodology of qualitative exploratory research was followed, whose technique used for the collection of information was the semi-directive interview with the directors and heads of development of several universities (nine institutions and 13 participants), whose words were analyzed and classified.

In the present work, the results of the study are reported around expectations, motivators (proactive and reactive), barriers, models and practices (previous, current and future) in the context of software quality, ending with the author's conclusions and recommendations.

Keywords: Software quality, Practices, Processes, Universities and Ecuadorian Polytechnic Schools

1 INTRODUCCIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

El Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior (CEAACES) [1], ha ejecutado procesos de evaluación, acreditación y aseguramiento de la calidad a universidades y escuelas politécnicas del Ecuador; con el fin de determinar el grado de cumplimiento de los estándares de calidad habilitantes necesarios para su acreditación como parte del Sistema de Educación Superior (SES), según lo determina la Ley Orgánica de Educación Superior (LOES), [2].

Hasta julio 2018, se definieron cuatro categorías de universidades (A, B, C y D). Las tres primeras categorías correspondían a instituciones de nivel superior que aprobaron satisfactoriamente la evaluación propuesta y, por lo tanto, obtuvieron la acreditación con vigencia quinquenal. Mientras que las instituciones con categoría D, pasaron a un proceso de acreditación, existiendo plazos definidos para terminar dicho proceso según el reglamento respectivo. De no aprobar la evaluación, las instituciones de categoría D pasaban a una categoría de no acreditación que les impedía seguir perteneciendo al SES ecuatoriano [1]. La Figura 1, ilustra el histórico de instituciones de nivel superior acreditadas.

Depuración del Sistema de Educación Superior (2009-2018)



CONSEJO DE ASEGURAMIENTO DE LA CALIDAD DE LA EDUCACIÓN SUPERIOR

Año	Proceso	Total IES	IES Evaluadas	Categorías								
				Acreditadas					Evaluación IES "E"			
				A	B	C	D	E	Acept.	Parcial. Acept.	No Acept.	
2009	Evaluación CONEA	71 ¹¹	68	11	9	13	9	26	-	-	-	
2012	Evaluación Universidades y Escuelas Politécnicas "E"	68	26	-	-	-	-	-	3	8	15	
2013	Evaluación Institucional de Universidades y Escuelas Politécnicas	55	54	5	23	18	8	-	-	-	-	
2015-2016	Recategorización de Universidades y Escuelas Politécnicas	59	12	2	3	7	-	-	-	-	-	
	Evaluación obligatoria ESPE		1	1	-	-	-	-	-	-	-	
2016	Evaluación Universidades y Escuelas Politécnicas "D"		8	0	3	5	-	-	-	-	-	
Total Universidades y Escuelas Politécnicas acreditadas (2018)		59¹²		8	28	19						
				55								

¹¹En este dato no se toma en cuenta a la Universidad Jefferson y a la Universidad Técnica Federico Santa María, por razones expuestas en líneas anteriores.

¹²Actualmente en el sistema de educación superior existen 59 universidades y escuelas politécnicas, de éstas 55 están acreditadas y las 4 restantes son las universidades creadas en 2013 (Yachay, IKIAM, Uartes y UNAE).

Figura 1 - Procesos de Evaluación, Acreditación y Clasificación Sistema de Educación Superior (2009-2018) [3], [4]

En agosto 2018 la LOES es reformada. Dentro de estas reformas se establece que el CEAACES cambia de nombre a Consejo de Aseguramiento de la Calidad de la Educación Superior (CACES), y se crea el Sistema Interinstitucional de Aseguramiento de la Calidad (SIAC), cuyos principales actores son el Consejo de Educación Superior (CES), CACES y las instituciones de educación superior (IES).

El CACES mantiene entre sus funciones la evaluación externa, la acreditación y adicionalmente el apoyo al aseguramiento interno de la calidad y a la evaluación sin fines de acreditación. Estas actividades se aplican a diferentes niveles: IES, carreras y programas, obteniendo como resultado una cualificación académica. Se elimina el procedimiento de categorización académica y toma relevancia el proceso de autoevaluación de las IES sobre el cumplimiento de sus propósitos, orientados a la construcción de una cultura de calidad institucional [4].

Así, para evaluar a cada IES se utiliza un modelo de evaluación, siendo vigente el del año 2015 (una actualización se encuentra en borrador desde el año 2018). El modelo de evaluación vigente se encuentra estructurado jerárquicamente por criterios, subcriterios e indicadores. Entre los criterios se encuentra el indicador “Sistema de Información”, donde se indica que la institución educativa superior “debe contar con un sistema de información que garantice a todos los miembros involucrados, la disponibilidad y accesibilidad de la información; la cual debe ser exacta, veraz, completa y oportuna para la toma de decisiones” [5].

Siendo que un sistema de información está compuesto de un conjunto de elementos organizados como software, información, personas, recursos y procesos [6], de los cuales el software es uno de los componentes fundamentales debido a su aporte en la automatización de procesos, es necesario e importante investigar cuales son las prácticas implementadas o a ser implementadas por las universidades y escuelas politécnicas para gestionar y garantizar la calidad en sus procesos de desarrollo de software.

1.2 OBJETIVO GENERAL

Estudiar las prácticas de calidad en los procesos de desarrollo de software en Universidades del Ecuador.

1.3 OBJETIVOS ESPECÍFICOS

- Elaborar una guía de entrevista a partir de una revisión sistemática de literatura sobre las referencias disponibles para la mejora o gestión de calidad en los procesos de desarrollo de software.
- Establecer el estado actual de las prácticas de calidad en los procesos de desarrollo de software en universidades del Ecuador a partir de entrevistas con los directores de los departamentos de sistemas de información o desarrollo de software de las universidades y escuelas politécnicas del Ecuador.
- Proponer recomendaciones para la mejora de calidad en los procesos de desarrollo de software en universidades del Ecuador.

1.4 MARCO TEÓRICO

Para conocer el tema de calidad de software en universidades ecuatorianas, es necesario realizar una revisión sistemática de literatura la cual permite encontrar los estudios disponibles relevantes a una temática en particular [7].

1.4.1 Revisión sistemática de literatura

La guía para revisiones sistemáticas de literatura propuesta por (J. Biolchini, P. Mian, A. Natali, and G. Travassos) [8], la cual está basada en la metodología propuesta por B. Kitchenham [9], es la guía utilizada en este estudio. Consta de cinco partes generales: formulación de la pregunta, selección de las fuentes, selección de los estudios, extracción de información y resumen de los resultados. En esta subsección se aborda las cuatro primeras partes de manera general, y el resumen de los resultados de la revisión sistemática se presenta en la subsección 1.4.2.

Formulación de la pregunta: se define la pregunta de investigación acorde al objetivo del estudio, dando como resultado, para este caso la siguiente pregunta de investigación:

¿Cuáles son las prácticas de calidad en los procesos de desarrollo de software en las universidades ecuatorianas?

El siguiente paso, es la selección de las fuentes de información.

Selección de las fuentes: para lo cual es necesario determinar la cadena de búsqueda, construida en base a la pregunta de investigación y que, a su vez, es usada para resolver la pregunta de investigación, dando como resultado la siguiente cadena de búsqueda:

- AND ((*Quality Software*) AND *Process*)
- AND (*Universities* OR (*High Education*))
- AND (*Ecuadorian* OR *Ecuador*)

Para este estudio, se determinaron las siguientes fuentes:

- Base de Datos: Scopus (<https://www.scopus.com/>)
Web of Science (<https://www.webofknowledge.com/>)
- Buscadores especializados: Google Académico (<https://scholar.google.es/>)

Cabe anotar que fueron adaptadas las cadenas de búsqueda, según la fuente de búsqueda establecida. La Tabla 1, ilustra los resultados obtenidos.

Tabla 1 - Resultados de la búsqueda inicial en las fuentes

Fuente	Resultado (inicial)
Scopus	15
Web of Science	27
Google Académico	67

El siguiente paso, es la selección de los estudios.

Selección de los estudios: permiten obtener sólo las investigaciones relevantes relacionadas a la pregunta de investigación. Siendo necesario aplicar criterios de selección. Los criterios de selección definidos para este estudio son:

- Trabajos pertenecientes al área de Ciencias de la Computación o Ingeniería de Software
- Información de contenido respecto a listado de estándares-normas y modelos de calidad

- Ser escrito en inglés y español

Para el filtrado de los estudios, también se establecieron criterios de exclusión, con relación a: temas asociados con la educación de la asignatura Calidad de Software en Universidades, y áreas que no tengan relación con ciencias de la computación e ingeniería de software.

Además, se aplicó, una estrategia adicional “*Backward snowballing*”, es una estancia de la revisión sistemática de literatura, la misma que permite identificar artículos relevantes utilizando la lista de referencias de un artículo [10].

Para la selección de los estudios, se inició con un análisis del título del artículo para determinar si la propuesta presentada está enfocada con la búsqueda que se está realizando. Posteriormente, con los estudios que pasaron el primer filtro (análisis del título del artículo), se realizó una lectura del *abstract* de los artículos siguiendo el mismo procedimiento.

Extracción de información

Luego del análisis de los estudios, el resultado de la búsqueda de información referente a temas de calidad de software en universidades de nuestro país no reflejó resultado alguno. Sin embargo, se rescata la información concerniente a: software, estándares-norma y modelos de calidad en los procesos de software, frenos u obstáculos en calidad de software y TIC's, cuyos resultados se presentan a continuación.

1.4.2 Software

Con el pasar del tiempo, el software se ha convertido en un elemento cada vez imprescindible en la sociedad a nivel mundial. Prácticamente, no existe sector de negocio o aspecto de nuestra vida cotidiana en el cual el software no brinde su aporte [11]. Es así como hoy en día, los negocios necesitan asegurarse de que el software que utilizan y/o desarrollan cumple con lo necesario para que las partes interesadas puedan desempeñar sus funciones con normalidad. Esta responsabilidad recae en los directores de tecnología y/o software de las empresas e instituciones, los cuales junto con su equipo de trabajo deberán garantizar criterios mínimos de cumplimiento de calidad para el software, por lo que es necesario conocer los términos calidad y calidad de software.

Calidad

La ISO 9000-2005 define a la calidad como el “grado en el que un conjunto de características inherentes cumple con los requisitos” [12]. Según la IEEE, la calidad se refiere al “grado en que un sistema, componente o proceso cumple con los requisitos especificados” o “el grado en que un sistema, componente o proceso satisface las necesidades o expectativas del cliente o usuario” [13], por lo que se deduciría que el cliente o usuario es quién determina lo que es la calidad.

Esta calidad se enfoca a varios temas, entre los cuales se encuentra la calidad de software.

Calidad de Software

Watts S. Humphrey [14] resume que la calidad del software debe enfocarse en satisfacer las necesidades del usuario haciendo que su trabajo sea confiable y consistente, lo cual requiere que el software que se produce tenga pocos defectos o ninguno. En cambio, Pressman [11], define a la calidad de software como el “proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan”, lo que tiene relación con lo establecido en la ISO/IEC 9126-1:2000.

En la Figura 2, se muestra a la calidad del software como una relación de tres enfoques: proceso, producto de software y uso. Se visualiza que la calidad del proceso contribuye a mejorar la calidad del producto de software, y la calidad del producto de software contribuye a mejorar la calidad en el uso de dicho producto; por lo que, evaluar y mejorar un proceso es un medio para mejorar la calidad del producto de software y, evaluar y mejorar la calidad en el uso, teniendo al mismo tiempo una retroalimentación inversa de estos tres enfoques [15].

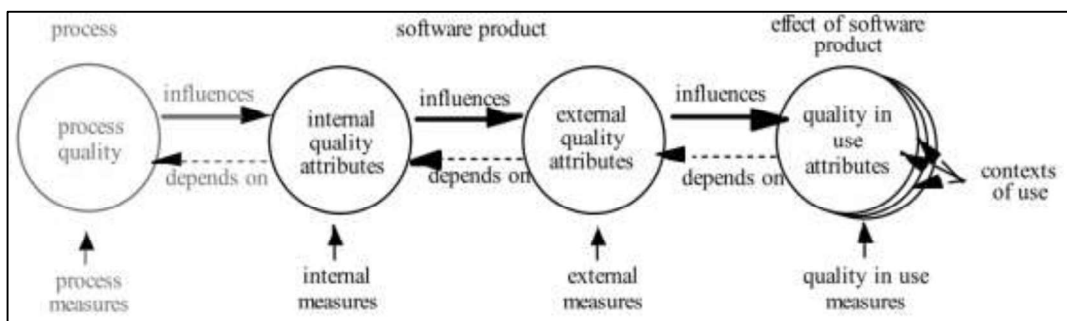


Figura 2 - Calidad del Software [15]

La misma norma ISO/IEC 9126-1:2000, enuncia aspectos relacionados a evaluar y mejorar la calidad del producto de software y su uso. La norma no brinda una explicación acerca de cómo evaluar y mejorar la calidad del proceso de software, lo cual se lo realiza a través de estándares y modelos de calidad para el proceso de software.

1.4.3 Estándares-Norma y Modelos de Calidad en los Procesos de Software

Antes de revisar los estándares-normas y modelos de calidad en los procesos de software, se presenta a continuación una descripción general de su definición.

Metodología es una transformación, implica proceso, modo de llevar a cabo un determinado trabajo, tiene un inicio y un final. Presenta etapas consecutivas que no puede iniciar una sin haber terminado exitosamente la anterior y podría llegar a ser auditable [16].

Estándar-Norma es un documento que proporciona, requisitos, especificaciones, directrices o características de obligatorio cumplimiento emitido por una entidad reguladora, que se puedan usar de manera consistente para garantizar que los materiales, productos, procesos y servicios sean adecuados para su propósito [17]. Estos estándares son auditables, pero a diferencia de la metodología, no son consecutivos ni están establecidos en un orden estricto. Los estándares de procesos de software documentan una definición estandarizada de las prácticas de desarrollo de software [18].

Modelo es la representación en forma esquemática, a menudo de forma simplificada, de un estado o situación existente o futura [17]. Los modelos no constituyen un contrato o una camisa de fuerza, y permiten según la naturaleza del negocio o del proceso que estemos analizando, que se aplique un cierto número de elementos (con criterio profesional) dejando otros de lado.

En la teoría, a un mismo término lo señalan de varias formas, tal es el caso de CMMi donde se menciona como modelo [19], estándar [20], metodología [21]. Por lo que, en base a las definiciones y revisiones, en este estudio se decide presentar el tema de calidad en el proceso de software como una clasificación de dos categorías: estándar-

norma y modelo, listando a los más representativos. La Tabla 2 presenta los resultados nominales de todos los modelos y estándar-norma encontrados.

Tabla 2 - Estándares-Norma y Modelos de Calidad en los Procesos de Software

Estándar-Norma: año última versión	Modelo (año última versión)
1. ISO/IEC TS 33073 :2017 – Reemplaza a ISO 15504 SPICE	1. Bootstrap (1997)
2. ISO/IEC/IEEE 90003:2018	2. PSP (2009)
	3. TSP (2010)
	4. CMMi (2018)

A continuación, se presenta una breve descripción de las características de cada estándar-norma y modelo de calidad en los procesos de software.

ISO/IEC TS 33073: 2017 “Tecnología de la Información – Evaluación del proceso – Modelo de evaluación de la capacidad de proceso para la gestión de la calidad”

ISO/IEC TS 33073: 2017 es una revisión de la anterior norma ISO/IEC 15504 “Determinación de la Capacidad de Mejora del Proceso de Software” (SPICE). Fue la Organización Internacional de Normalización (ISO) quien estableció el proyecto SPICE para llevar a cabo el proceso de estandarización. ISO había publicado previamente los procesos del ciclo de vida del software en la norma ISO 12207 (1995/2002) que formaba una base para el modelo de proceso de referencia para ISO 15504 Parte 2 (1998). El desarrollo de ISO 15504 reunió a expertos de todo el mundo. El objetivo del proyecto era producir un método de evaluación para organizaciones de diferentes tamaños, dominios de aplicación y estilos de administración que pueden tener diferentes prioridades de mejora.

La norma ISO 15504 incluye un modelo de proceso con seis niveles de capacidad y un conjunto de procesos de referencia alineados con la ISO 12207 definición. Los niveles de capacidad de ISO 15504 se aplican a cada proceso individual. No se exige una secuencia predefinida, lo que significa que las prioridades no se fijan para mejorar ciertos procesos. La definición de prioridad se basa en los requisitos y objetivos comerciales de cada organización [18].

La ISO/IEC TS 33073: 2017 [22], propone un Modelo de Proceso de Referencia (PRM) asociado con los atributos del proceso definidos en ISO/IEC 33020, lo que resulta en un Modelo de Evaluación del Proceso (PAM) utilizado como base común para realizar evaluaciones de la capacidad del proceso del sistema de Gestión de la Calidad, (Figura 3). Por un lado, se considera la dimensión del proceso, que los procesos estén definidos; mientras que en la otra dimensión se presenta la capacidad, que es un conjunto de atributos de proceso agrupados en niveles de capacidad definidos.

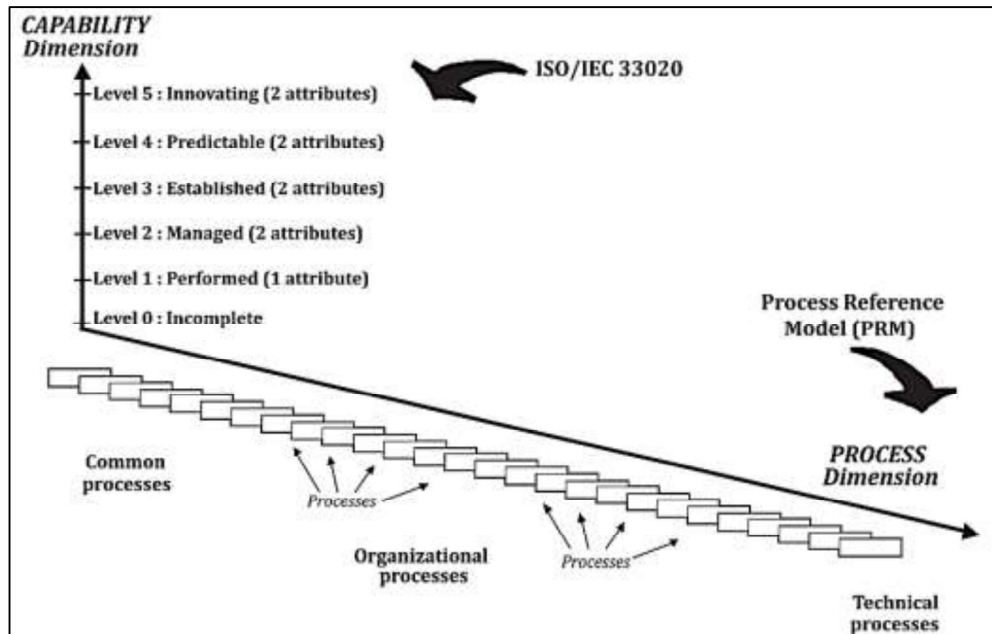


Figura 3 - Relación entre el Modelo de Evaluación del Proceso (PAM) y sus entradas [22]

La Figura 4, muestra los procesos incluidos en la dimensión del proceso del PAM para la gestión de la calidad.

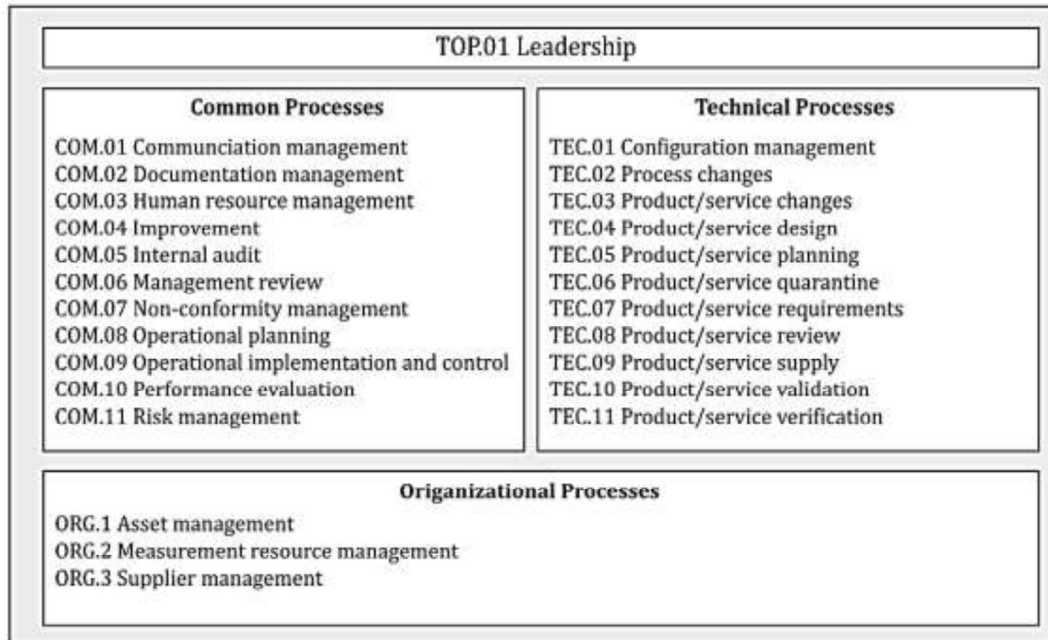


Figura 4 - Procesos en el PAM [22]

Hay seis niveles de capacidad [22], incorporando nueve atributos de proceso.

- Nivel 0: proceso incompleto. El proceso no se implementa o no logra alcanzar su propósito. En este nivel, hay poca o ninguna evidencia de algún logro sistemático del propósito del proceso.
- Nivel 1: Proceso realizado. El proceso implementado logra su propósito de proceso.
- Nivel 2: Proceso gestionado. El proceso realizado anteriormente descrito se implementa ahora de manera administrada (planificado, monitoreado y ajustado) y sus productos de trabajo se establecen, controlan y mantienen de manera adecuada.
- Nivel 3: Proceso establecido. El proceso gestionado descrito anteriormente se implementa ahora utilizando un proceso definido que es capaz de lograr los resultados de su proceso.
- Nivel 4: Proceso predecible. El proceso establecido descrito anteriormente ahora opera predeciblemente dentro de límites definidos para lograr los resultados de su proceso. Se identifican las necesidades de gestión cuantitativas, se recopilan y analizan los datos de medición para identificar las causas de variación asignables. Se toman acciones correctivas para abordar las causas de variación asignables.

- Nivel 5: Proceso innovador. El proceso predecible descrito anteriormente ahora se mejora continuamente para responder al cambio alineado con los objetivos de la organización.

Dentro del PAM, la medida de la capacidad se basa en los nueve atributos de proceso (PA) definidos en ISO/IEC 33020. Los atributos del proceso se utilizan para determinar si un proceso ha alcanzado una capacidad determinada. Cada atributo mide un aspecto particular de la capacidad del proceso.

En cada nivel, no hay orden entre los atributos del proceso; cada atributo aborda un aspecto específico del nivel de capacidad. La lista de atributos de proceso se muestra en la Tabla 3.

Tabla 3 - Niveles de capacidad y atributos de proceso

ID de Atributo de Proceso	Niveles de Capacidad y Atributos de Procesos
	Nivel 0: Proceso Incompleto
	Nivel 1: Proceso Realizado
PA 1.1	Rendimiento de proceso
	Nivel 2: Proceso Gestionado
PA 2.1	Gestión de rendimiento
PA 2.2	Gestión de productos de trabajo
	Nivel 3: Proceso Establecido
PA 3.1	Definición de proceso
PA 3.2	Despliegue del proceso
	Nivel 4: Proceso Predecible
PA 4.1	Análisis cuantitativo
PA 4.2	Control Cuantitativo
	Nivel 5: Proceso Innovador
PA 5.1	Innovación de proceso
PA 5.2	Implementación de proceso de innovación

ISO/IEC/IEEE 90003:2018 - Ingeniería de software - Directrices para la aplicación de la Norma ISO 9001:2015 a los programas informáticos

ISO 9001:2015, Sistema de gestión de calidad – Requisitos [23], especifica los requisitos para un sistema de gestión de calidad (Figura 5) que una organización puede usar cuando ella:

- Debe demostrar su capacidad para proporcionar productos y servicios de manera consistente que cumplan con los requisitos legales y reglamentarios del cliente; y,
- Tiene como objetivo mejorar la satisfacción del cliente, mediante la aplicación efectiva del sistema, incluidos los procesos de mejora del sistema y la garantía de conformidad con el cliente y los requisitos legales y reglamentarios aplicables.

Todos los requisitos de la norma ISO 9001: 2015 son genéricos y están destinados a ser aplicables a cualquier organización, independientemente de su tipo o tamaño, o de los productos y servicios que proporciona.

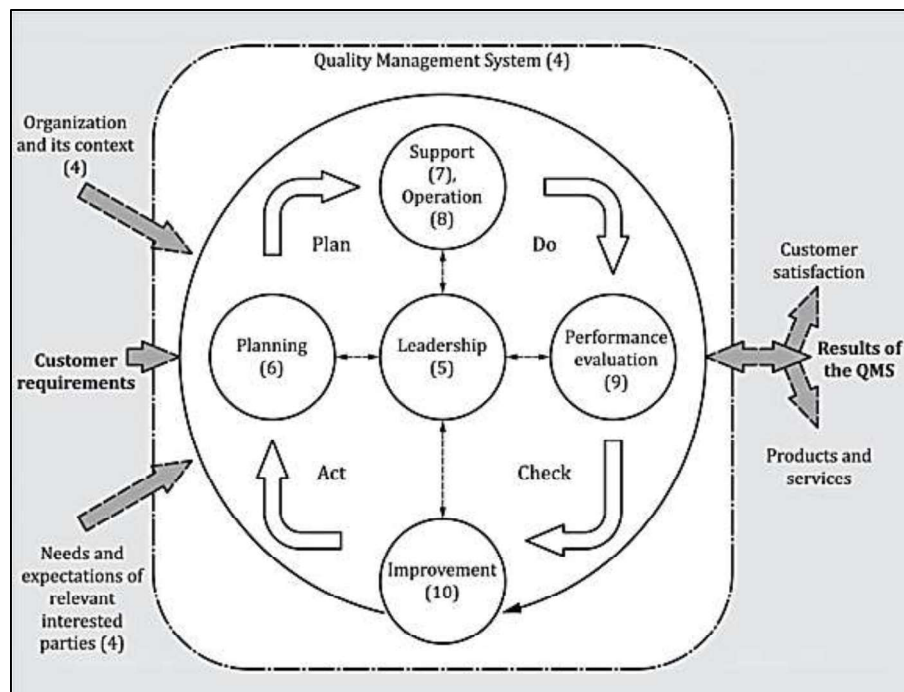


Figura 5 - Representación de la estructura de la Norma ISO 9001:2015 en el ciclo PDCA [24].

El ciclo PDCA se puede describir brevemente de la siguiente manera:

- Plan: establecer los objetivos del sistema y sus procesos, y los recursos necesarios para entregar resultados de acuerdo con los requisitos de los clientes y las políticas de la organización, e identificar y abordar los riesgos y oportunidades;
- Hacer: implementar lo planeado;
- Verificar: monitorear y (cuando corresponda) medir los procesos y los productos y servicios resultantes en relación con las políticas, los objetivos, los requisitos y las actividades planificadas, e informar los resultados;
- Actuar: tomar acciones para mejorar el rendimiento, según sea necesario.

Cabe indicar que los números entre paréntesis hacen referencia a las cláusulas de la norma ISO 9001:2015.

Referente a ISO/IEC/IEEE 90003, sus ediciones anteriores son (ISO/IEC 9000-3:2004 e ISO/IEC 9000-3:2014), mientras su versión actual es la del 2018, la cual está diseñada como una lista de verificación para el desarrollo, suministro y mantenimiento de

programas informáticos. La versión actualizada recientemente combina los beneficios comprobados de ISO 9001 con algunos de los documentos de soporte más importantes del mundo en ingeniería de software, lo que permite a una organización beneficiarse de las mejores prácticas internacionales para mejorar la calidad en cada paso del ciclo de vida. Esto incluye el suministro, adquisición, operación y mantenimiento, hasta el proceso circular de mejora continua.

Desarrollado en conjunto por ISO, con la Comisión Electrotécnica Internacional (IEC) y el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), la norma se revisó recientemente para alinearla con la versión más reciente de la norma ISO 9001 (publicada en 2015).

Las directrices proporcionadas por ISO/IEC/IEEE 90003:2018 [24], no están destinadas a ser utilizadas como criterios de evaluación en el registro/certificación de sistemas de gestión de calidad. Sin embargo, algunas organizaciones pueden considerar útil implementar las pautas propuestas en este documento y pueden estar interesadas en saber si el sistema de gestión de calidad resultante cumple o no con este documento.

En este caso, una organización puede usar esta norma y la norma ISO 9001 como criterios de evaluación para los sistemas de gestión de calidad en el dominio del software.

La estructura del documento de la norma ISO/IEC/IEEE 90003: 2018, consta de 10 cláusulas, siendo las cláusulas 4 a la 10 alineadas a la norma ISO 9001:2015.

1. Alcance
2. Referencias de normativas
3. Términos y definiciones
4. Contexto de la organización.
5. Liderazgo
6. Planificación
7. Apoyo
8. Operación
9. Evaluación del desempeño
10. Mejora

BOOTSTRAP

Entre 1990 a 1992, la Comisión Europea dentro del “Programa Estratégico Europeo de Investigación en Tecnología de la Información” (ESPRIT), desarrolló inicialmente el modelo Bootstrap. En 1994 se funda el instituto Bootstrap como un grupo de interés económico europeo (EEIG), para la gestión y desarrollo de la metodología. En 1997 se realiza el lanzamiento de Bootstrap 3.0[18].

Los objetivos que presenta Bootstrap son [25]:

- Proporcionar soporte para la evaluación de la capacidad del proceso frente a un conjunto de mejores prácticas de ingeniería de software reconocidas;
- Incluir estándares de ingeniería de software reconocidos internacionalmente como fuentes para la identificación de mejores prácticas;
- Respaldar la evaluación de cómo se han implementado los estándares de referencia en la organización evaluada;
- Asegurar que la evaluación sea confiable y repetible;
- Identificar las fortalezas y debilidades en los procesos de la organización;
- Proporcionar resultados que forman una base adecuada y confiable para la planificación de mejoras;
- Planificar acciones de mejora que respalden el logro de los objetivos de la organización;
- Ayudar a incrementar la eficacia del proceso al implementar requisitos estándar en la organización.

Bootstrap V3.0 es el resultado del proyecto europeo ESPRIT 5441, integra requisitos de varios estándares reconocidos internacionalmente como CMM, ISO 12207, ISO 9001, ISO 9000-3 e incorpora los estándares aceptados de la Agencia Espacial Europea PSS-05-0 [26]. Dentro de los estándares mencionados anteriormente, los siguientes tienen un rol particular:

- CMM ha sido una de las principales referencias para la versión inicial de la metodología Bootstrap y está destinada a ser utilizada como referencia para una mayor evolución de la metodología.
- ISO 12207 proporciona un marco reconocido para los procesos de software.

- ISO 9001 proporciona el enfoque organizacional, particularmente con respecto al sistema de calidad y los procedimientos de toda la organización.
- ISO 15504 proporciona el marco para la definición de procesos y capacidades.

Los objetivos de Bootstrap 3.0 son, el proporcionar un servicio confiable y un punto de referencia consistente para evaluar las prácticas y procesos existentes, así como definir un conjunto de acciones e iniciativas que mejorarían la calidad, la confiabilidad, los costos de desarrollo y la productividad.

La metodología Bootstrap 3.0, contiene un modelo de proceso y un método de evaluación. El modelo de proceso se basa en el modelo de referencia ISO 15504. Presenta las dimensiones de Proceso y Capacidad.

- Dimensión de Proceso: contiene 33 procesos diferentes organizados en seis grupos: Organización, Dependiente del ciclo de vida, Gestión, Soporte, Cliente Proveedor y Proceso relacionado. Además, se añade el grupo de Tecnología (4 procesos), visualizar Figura 6.
- Dimensión Capacidad: consta de seis niveles, cada nivel consta de uno o más atributos de proceso, adoptados a partir de ISO 15504.

Mientras que en el método de evaluación se realiza una evaluación a nivel de Unidad de Desarrollo Software (SPU) y de proyecto.

El Instituto Bootstrap organiza y coordina el plan de capacitación y registro de evaluadores. Hasta 2003, Bootstrap sólo estaba disponible bajo una licencia, lo que puede haber impedido su uso a gran escala.

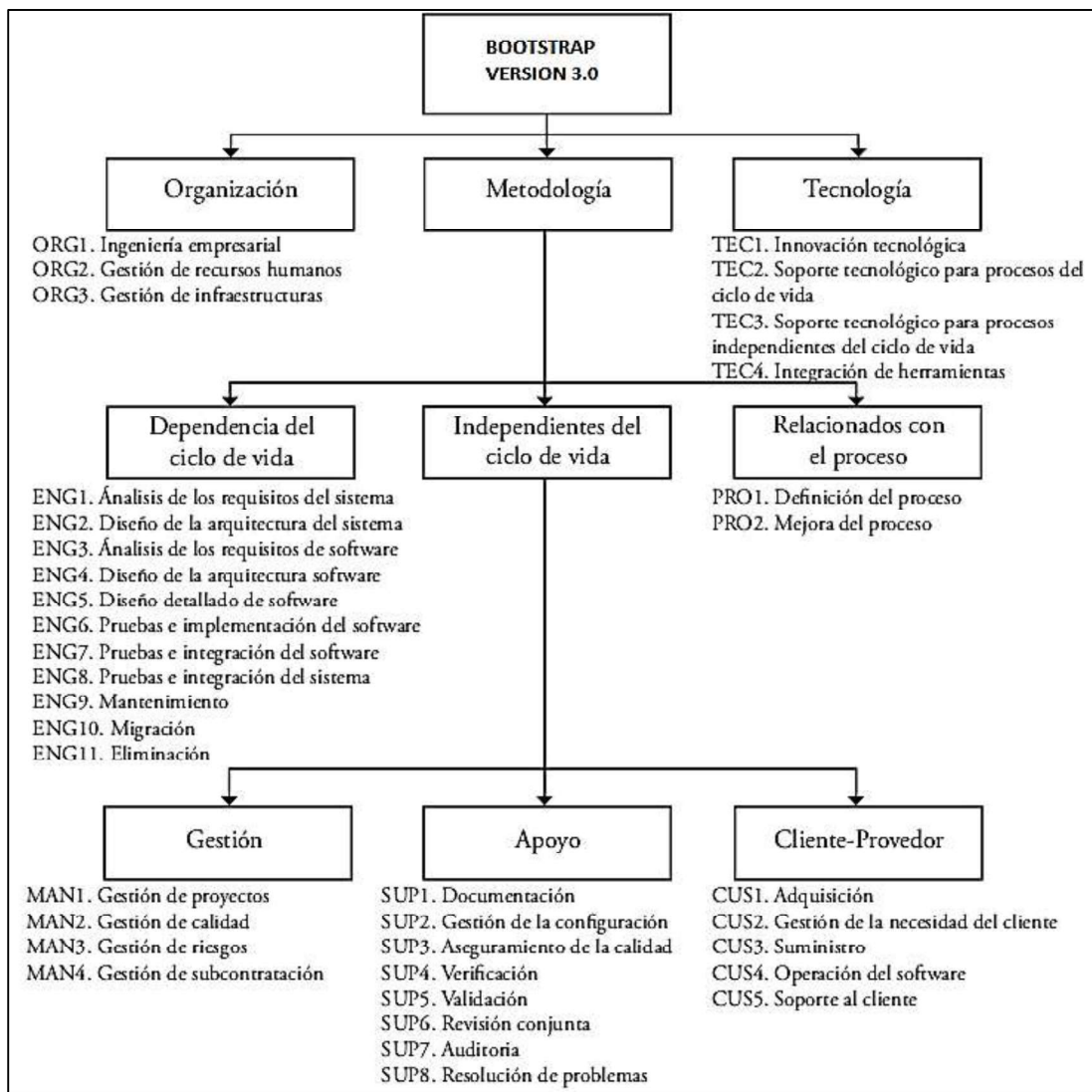


Figura 6 - Estructura de Bootstrap V3.0 [27]

Proceso de Software Personal

En 1995, PSP fue propuesto por Watts Humphrey del Instituto de Ingeniería en Software (SEI) de la Universidad de Carnegie Mellon (UCM) y dirigido para estudiantes de ingeniería de software, para en 1997 destinarlo a ingenieros de software [12]. En 2005, se propone la Guía de Conocimiento para el Proceso de Software Personal (PSP BOK) Versión 1.0 y en 2009 PSP BOK, Versión 2.0 [28].

El PSP, es un modelo que apoya a los ingenieros del software a medir y mejorar su forma de trabajar [14]. Al utilizar los conceptos y métodos de PSP en su trabajo, las

personas en casi cualquier campo técnico pueden mejorar sus habilidades de estimación y planificación, hacer compromisos que puedan cumplir, gestionar la calidad de su trabajo y reducir el número de defectos en sus productos.

Los objetivos que presenta el PSP son [29]:

- Planificar, estimar, medir, seguir y controlar todo el proceso de desarrollo de software.
- Lograr una disciplina de mejora continua en el proceso de desarrollo.
- Mejorar los niveles de calidad del proceso de desarrollo de software.
- En general, PSP provee calidad y productividad a todo el proceso de desarrollo de software.

La estructura del proceso PSP se muestra en la Figura 7. Es posible observar que el proceso comienza con un levantamiento de los requerimientos, y con la primera tarea llamada “Planificación”. Hay un script que sirve de guía para este trabajo y un resumen para registrar los datos de la planificación. Los Ingenieros registran el tiempo y los datos de los defectos. Al final del trabajo, durante la última etapa (postmortem), los ingenieros sumarian tanto los datos de los defectos como los tiempos, miden el tamaño del programa e ingresan estos datos en el resumen de plan. Luego, se entrega el producto terminado con el resumen de la planificación [25].

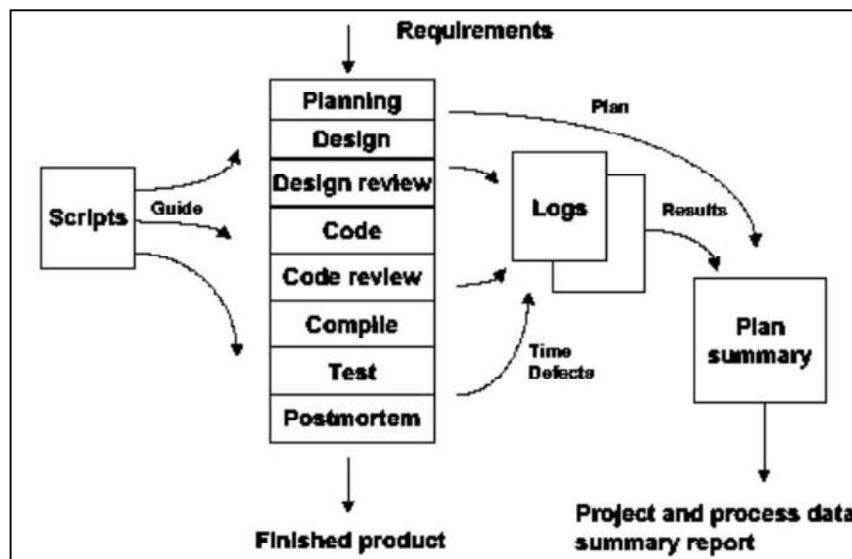


Figura 7 - Estructura PSP [25]

Proceso de Software de Equipo

En 1999 Watts Humphrey del SEI crea el TSP con el fin de proporcionarles a los estudiantes de ingeniería de software de la UCM una visión total del ciclo de vida del software. En 2010 se presenta la guía de fundamentos de TSP [30].

El TSP, proporciona un marco operacional en el que las personas capacitadas en PSP combinan sus habilidades de disciplina de procesos personales con técnicas de gestión de procesos comprobadas para planificar, desarrollar y entregar productos de software de alta calidad dentro del cronograma y los parámetros de costos dados. La formación de equipos, la calidad, los procesos y la mejora continua del proyecto, funciona según la filosofía de un equipo que trabaja en conjunto para alcanzar un objetivo común (sus miembros se ayudan mutuamente para completar el proyecto). Esto también resuelve el problema de la habituación al revisar e inspeccionar el código de cada uno y aprender unos de otros.

Los objetivos que presenta el TSP son [31]:

- Construir rápidamente equipos autodirigidos que puedan entregar el proyecto exitosamente.
- Construir productos de alta calidad dentro del presupuesto y el cronograma.
- Tener un proceso de desarrollo maduro que pueda ser aprovechado por los equipos y, al mismo tiempo, tener un proceso personal para las personas.
- Garantizar la comunicación continua, el seguimiento general del proyecto y el rendimiento optimizado del equipo durante todo el ciclo de vida del proyecto.
- Habilitar el mecanismo de mejora del proceso de software en el proyecto.

Modelo de Madurez de Capacidad Integrado

En 1987, el departamento de Defensa de los Estados Unidos de Norteamérica contrató a la UCM, para desarrollar un modelo de madurez de capacidades (CMM) para evaluar la calidad y la capacidad de sus contratistas de software. En 1991 aparece la primera versión del CMM para desarrollo de software. En 2010 se cambia a Modelo de Madurez de Capacidades Integrado (CMMi) y se expande a otras áreas (Servicios, Adquisición, Personas y Software). En 2012, la UCM fundó el instituto CMMi, para extender los beneficios de CMMi más allá de la ingeniería de software y sistemas, a cualquier compañía de productos y servicios, independientemente del tamaño o la

industria. En 2014, el instituto CMMi presenta el Modelo de Madurez de Gestión de Datos para ayudar a las empresas a construir, mejorar y medir su función y personal de gestión de datos empresariales. En 2016, el instituto CMMi fue adquirido por la Asociación de Auditoría y Control de Sistemas de Información (ISACA). En 2018 se da el lanzamiento de CMMi *Development* V2.0, con mejoras en cuatro áreas principales: enfoque en el rendimiento, integrado con Ágil y Scrum, Seguridad y Seguro [30].

Los modelos CMMi son colecciones de buenas prácticas que ayudan a las organizaciones a mejorar sus procesos. Estos modelos son desarrollados por equipos de producto con miembros procedentes de la industria, del gobierno y el SEI y se listan a continuación:

- CMMi *Development*: CMMiDEV
- CMMi *Acquisition*: CMMiACQ
- CMMi *Services*: CMMiSVR
- *People Capability Maturity Model*: PCMM

En 2010, se publica la versión 1.3 estable de CMMiDEV, CMMiSVC, CMMiACQ, y PCMM, mientras se tiene en estudio la versión 2.0 de CMMiDEV destinada a salir en 2018.

El modelo, denominado CMMi para Desarrollo proporciona un conjunto completo e integrado de guías para desarrollar componentes, productos y servicios. Aborda las prácticas que cubren el ciclo de vida del producto de software desde la concepción hasta la entrega y el mantenimiento. Las buenas prácticas del modelo se centran en las actividades para desarrollar productos y servicios de calidad con el fin de cumplir las necesidades de clientes y usuarios finales.

El modelo presenta un total de 22 áreas de proceso, de las cuales 16 son áreas de proceso base (para todos los modelos CMMi), una es un área de proceso compartida y cinco son áreas de proceso específicas de desarrollo.

“Un área de proceso base es un área de proceso que es común a todos los modelos CMMi. Un área de proceso compartida está presente en al menos dos modelos CMMi,

pero no en todos”. La estructura del modelo CMMi (Figura 8), se basa en tres componentes [32]:

- Componente Requerido, esenciales para lograr la mejora de procesos en un área de proceso dada. Son las metas específicas y genéricas. La satisfacción de las metas se utiliza en las evaluaciones como base para determinar si un área de proceso ha sido satisfecha.
- Componente Esperado, actividades que son importantes para lograr un componente CMMi requerido. Son las prácticas específicas y genéricas.
- Componente Informativo, proporciona información necesaria para lograr la correcta comprensión de las metas y prácticas y, por ello, no se puede ignorar. Son la declaración del propósito, notas introductorias, áreas de proceso relacionadas, ejemplos de productos de trabajo, subprácticas, elaboración de la práctica genérica.

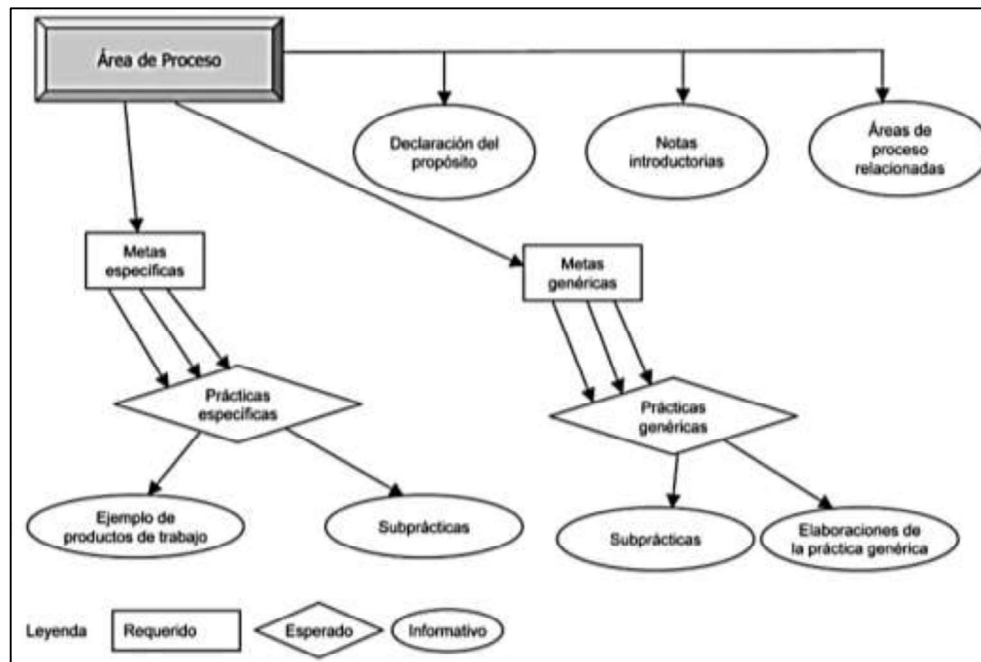


Figura 8 - Componentes del Modelo CMMi [32]

CMMi da soporte a dos rutas de mejora usando niveles, el uno es por uso de la representación continua la cual permite alcanzar niveles de capacidad y la otra ruta es por el uso de la representación por etapas la cual permite alcanzar niveles de madurez

(Figura 9). La representación continua permite a la organización seleccionar el enfoque de sus esfuerzos de mejora de procesos, eligiendo aquellas áreas de proceso o conjunto de áreas de proceso interrelacionados, que más benefician a la organización y sus objetivos de negocios. En la representación por etapas, las áreas de proceso se agrupan por niveles de madurez, indicando que áreas de proceso se deben implementar para alcanzar cada nivel de madurez [32].

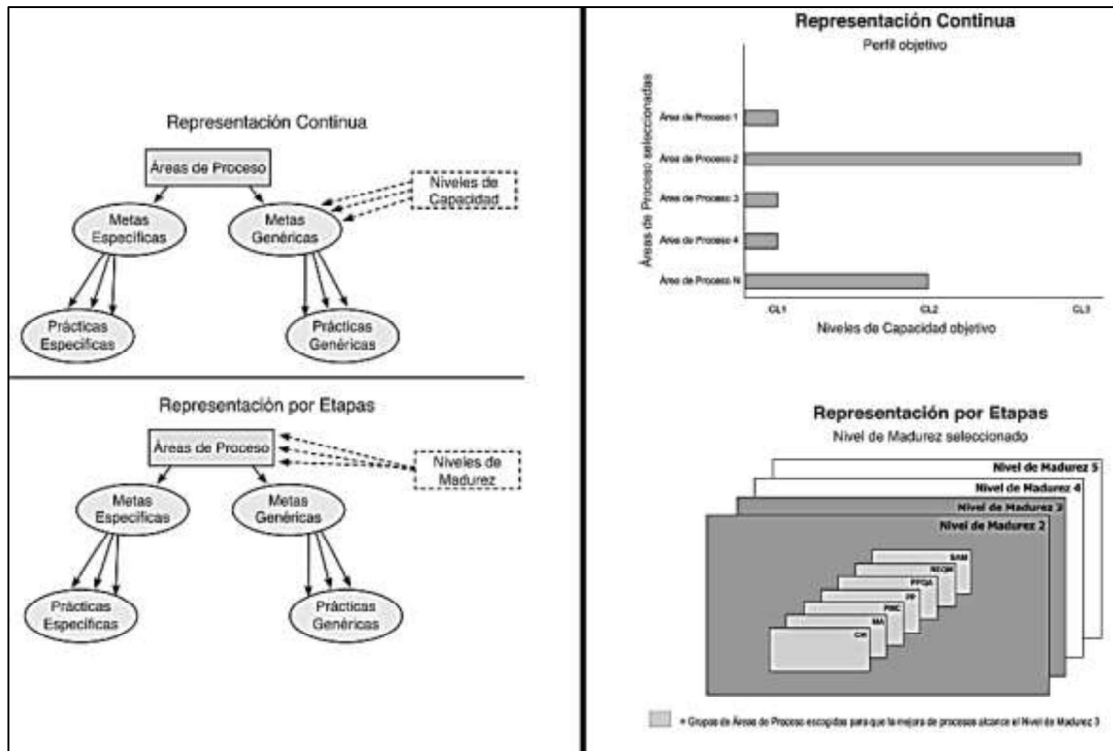


Figura 9 - Estructura y áreas de proceso de las representaciones continua y por etapas [32]

1.4.4 Frenos u Obstáculos en Calidad de Software

De la revisión de literatura (Tabla 4), se pueden citar los siguientes frenos u obstáculos que interfieren en Calidad de Software o de forma general en las Tecnologías de la Información y la Comunicación (TIC's).

Tabla 4 - Lista de frenos u obstáculos en Calidad de Software

Frenos u Obstáculos	Literatura
<ul style="list-style-type: none">• Resistencia al Cambio• Incompatibilidad con aplicaciones existentes• Falta de apoyo de la alta dirección	ANUIES 2018
<ul style="list-style-type: none">• El tiempo• Lo financiero• Necesidad de procesos	B. Fitzgerald, E. Wynn 2004
<ul style="list-style-type: none">• Horarios poco realistas• Recursos inadecuados• Requisitos inestables	W. Humphrey 2005

1.4.5 TIC's

El resultado de la búsqueda de información referente a temas de calidad de software en universidades de otros países y el nuestro, no reflejó resultado alguno. Por esta razón se decide buscar esta información en el término técnico que lo engloba de manera organizacional, es decir el tema de las Tecnologías de Información y Comunicación (TIC's). De esta manera se encontró: el documento UNIVERSITIC 2017 [20], informe que muestra el análisis de las TIC's en universidades españolas publicado por Crue Universidades Españolas; el informe "Estado actual de las Tecnologías de la Información y las Comunicaciones en las Instituciones de Educación Superior en México" 2018, publicado por ANUIES [33]; y, el informe UETIC-2017 [34], el cual analiza el estado de las Tecnologías de la Información y la Comunicación en las Universidades Ecuatorianas, publicado el 14 de marzo de 2018 por CEDIA.

A continuación, se resalta el tema de calidad de software en cada informe.

España

En el informe UNIVERSITIC 2017 (Tabla 5) se presentan los estándares según el ámbito de cobertura de las TIC's, desde el Gobierno de las TI hasta el ámbito de Auditoría, donde el ámbito de Desarrollo de Software presenta los siguientes estándares: ISO 12207/ISO 15504, CMMi, Bootstrap.

Tabla 5 - Herramientas para la implantación del Gobierno de las TI [20]

	Estándar internacional	Estándar nacional	Estándar organización
Gobierno de las TI	ISO 38500	AS 8015 COSO	COBIT
Planificación TI		PSI-Metrica 3	
Valor de las TI			Val IT
Gestión Servicios TI	ISO 20000	BS 15000	COBIT ITIL MOF
Gestión de Proyectos		UNE 15781	PMBOK PRINCE2 APMs IPMA
Desarrollo Software	ISO12207/ISO15504	Ticket Metrica 3	CMMI Bootstrap
Gestión de Riesgos		AS/NZS 4360 COSO Magent UNE 71504	
Gestión de Seguridad	ISO27000 ISO13335 ISO 13569 ISO17799 ISO15408	NIST-800 series BS 7799-2 GAO s FISCAM German BSI	ASCI-33 COBIT ISF ENV12924 SEI s OCTAVE SEI sSw- CMM BPM
Gestión Continuidad	ISO/IEC 25999	PAS-56 AS/NZS 4360 HB221-2004 BS25999	
Gestión de la Calidad	ISO 9001	EFQM BNQP SixSigma	
Auditoría	ISO 19011		COBIT

México

Del Estudio 2018 ANUIES, la información a rescatar son las prácticas de calidad de software y las certificaciones (Figura 10), donde se visualiza que son prácticamente pocas (22%) de las IEs del país azteca que tienen operando sus sistemas con consideraciones de calidad de software. Al contrario, la mayoría de las IEs no tienen prácticas de calidad de software (56%).

		2017	2018	Variación
8.6 Tienes prácticas Calidad de Software	Sí y están operando actualmente	21%	22%	1%
	Sí pero están en etapa de pilotaje y/o implementación	23%	19%	-4%
	No	52%	56%	4%
	No contestó	4%	3%	-1%

		2017	2018	Variación
8.7 Desde que año	Entre el 2014 y el 2017	80%	87%	7%
	Entre el 2010 y el 2013	15%	8%	-7%
	Antes del 2010	5%	5%	0%

		2017	2018	Variación
8.8 Cuenta con certificación en Calidad de Software	Certificados a Nivel Organización con CMMI, Moprosoft, etc.	9%	16%	7%
	Personal certificado en CMMI, SCRUM, TSP/PSP, etc.	18%	23%	5%
	Recibimos capacitación formal en CMMI, Moprosoft, SCRUM, TSP/PSP, etc.	23%	41%	18%
	No contamos con ningún tipo de capacitación ni certificación en Metodología de Calidad de Software	48%	20%	-28%
	No contestó	2%	0%	-2%

Figura 10 - Comparativo de Años de la Calidad de TI – Software [33]

Ecuador

El informe UETIC 2017 analiza el estado de las TIC's en las universidades ecuatorianas, en cuya primera edición 2017 contó con la participación de 37 universidades a nivel nacional (22 públicas y 15 privadas).

Los dos indicadores que se puede resaltar con respecto al Software presentados en el informe son: la metodología formal de desarrollo de software y el otro los sistemas de información.

- Metodología formal de desarrollo de software “Este indicador presenta si las instituciones cuentan con una metodología formal para el desarrollo de software”.

La Figura 11, muestra que el 62% de las universidades que fueron encuestadas tienen una metodología formal de desarrollo de software, mientras que el 35% de las instituciones no la tiene.

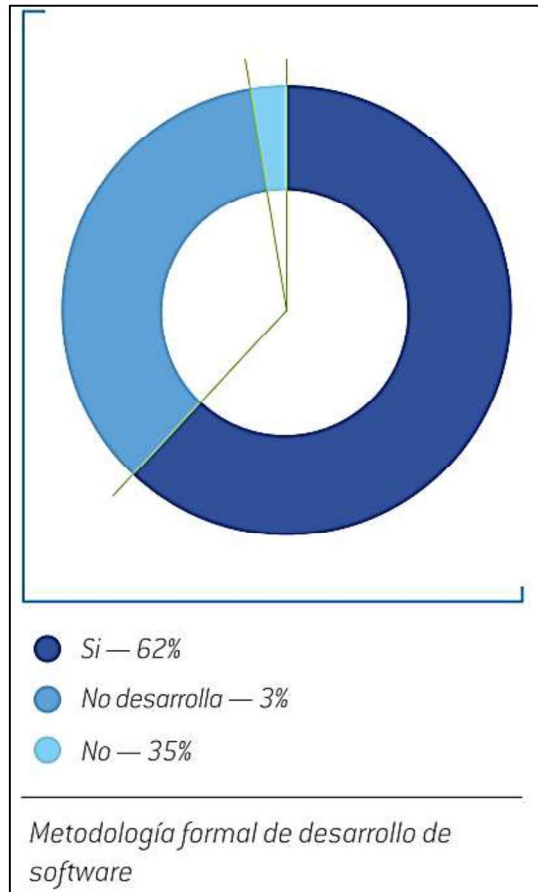


Figura 11 - Metodología Formal de Desarrollo de Software [34]

- De la misma forma, el indicador sistemas de información (Figura 12), muestra los tipos de sistemas de las IES que han sido propiamente desarrollados, comprado, *open source*, SaaS o no dispone, resaltando así que los sistemas Core de las instituciones, en su mayoría han sido desarrollo propio.

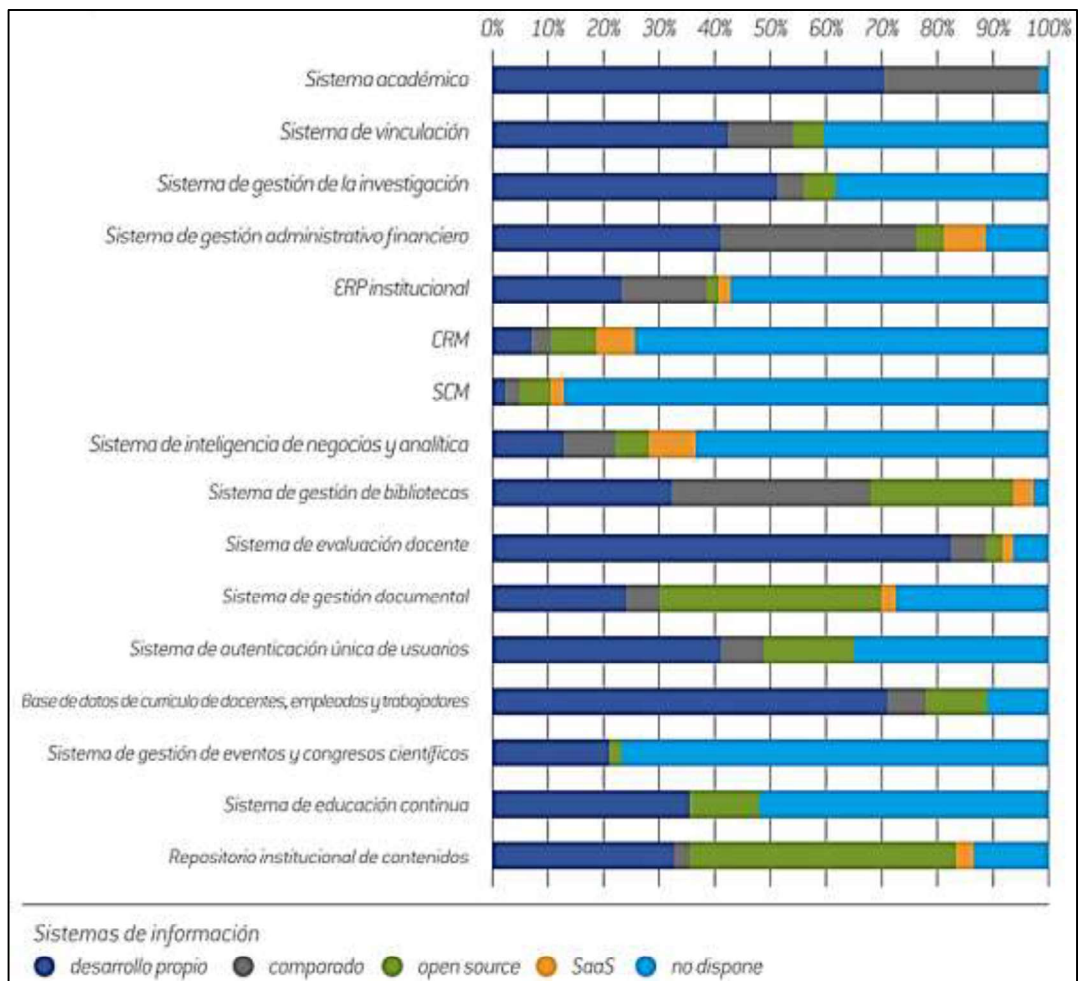


Figura 12 - Sistemas de Información [34]

2 METODOLOGÍA

Debido a que no existen estudios que se hayan realizado de las universidades del Ecuador en cuanto a la temática de calidad de software, se decide aplicar el tipo de investigación exploratorio cualitativo para conocer en primera instancia y de manera general el contexto en el que se encuentran las universidades del país con respecto a dicho tema. Para esto, se realizó la selección de la muestra y diseño de la investigación [35], el cual consiste en aplicar la técnica de recolección de información a través de entrevistas semidirectivas audio-grabadas “donde existe un control sobre los temas a tratar con base en un guion de entrevista” [36]. Estas entrevistas fueron realizadas a directores o directoras de tecnología de varias universidades del país, para posteriormente, proceder a su transcripción y codificación (análisis de contenido).

2.1 MUESTRA

Del listado de las universidades y escuelas politécnicas del Ecuador según su categoría a enero 2019 (portal del CACES [37]), fueron invitados a participar de la entrevista los directores de tecnología de las universidades y escuelas politécnicas presentes en la categoría A y algunas instituciones de categoría B. Aceptaron la invitación las instituciones de educación superior mostradas en la Tabla 6, donde además se visualiza la sigla, la región geográfica de ubicación, así como información estadística referente al registro de matrículas y docentes correspondiente al año 2016.

Tabla 6 - Listado de universidades ecuatorianas participantes de la entrevista

No.	Universidad o Escuela Politécnica	Siglas	Región	2016	
				Registro de matrículas [38]	Registro de Docentes [39]
1.	Universidad Central del Ecuador	UCE	Sierra	45.332	2.564
2.	Pontificia Universidad Católica del Ecuador	PUCE	Sierra	23.458	2.692
3.	Universidad de Cuenca	U. CUENCA	Sierra	20.391	1.199
4.	Universidad de las Fuerza Armadas	ESPE	Sierra	15.110	1.255
5.	Escuela Superior Politécnica del Litoral	ESPOL	Costa	12.199	1.059
6.	Escuela Politécnica Nacional	EPN	Sierra	10.543	779
7.	Universidad Andina Simón Bolívar	UASB	Sierra	2.876	286
8.	Universidad Particular Internacional SEK	UISEK	Sierra	1.882	178
9.	Facultad Latinoamericana de Ciencias Sociales	FLACSO	Sierra	1.128	97

Nota: la información estadística 2016 presentada en esta tabla, a la fecha, es la información más actual recopilada y presentada en el portal de la Secretaría de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT).

2.2 DISEÑO DE LA INVESTIGACIÓN

2.2.1 Recolección de datos

El primer contacto con los directores/ras o jefes del área de desarrollo, se lo realizó con una visita personal, llamada telefónica y/o el envío de un correo electrónico desde la cuenta institucional del investigador, para un acercamiento inicial y solicitud de agendamiento de una cita para la entrevista.

Los materiales utilizados en la recolección de datos fueron:

- Dos hojas de información para el participante y formulario de consentimiento (una para el entrevistado y otra para el entrevistador), ver Anexo I.
- Grabadora.
- Teléfono móvil (equipo de backup para la entrevista audio-grabada).
- La guía de entrevista (basada en la revisión sistemática de literatura), ver Figura 13.

GUÍA DE ENTREVISTA

PRESENTACIÓN

Apreciado/a entrevistado/a a continuación voy a realizarle una serie de preguntas relacionadas con la calidad de software.

El objetivo que persigo al hacérselas, es conocer las prácticas de calidad en los procesos de desarrollo de software en Universidades del Ecuador.

Tenga en cuenta lo siguiente:

- La investigación es del tipo exploratorio
- Ninguna respuesta es buena o mala.
- Se mantendrá el anonimato del/la entrevistado/a y de la institución si lo considera.
- Autorización para grabar la conversación, para a posteriori realizar el respectivo análisis de información
- Desea recibir el resultado del estudio
- Se encontraría interesado en continuar abordando el sujeto de estudio en una futura fase

ENTREVISTA

Tópicos

A. CONCEPTUAL

1. ¿Qué entiende usted por calidad de Software?
2. ¿Cuáles son sus expectativas en términos de calidad de software?

B. CRONOLOGÍA DE PRÁCTICAS

3. ¿Cuáles han sido los principales esfuerzos realizados en su institución para mejorar la calidad de software?
4. Actualmente, ¿qué proyecto o iniciativa de proyecto están llevando en términos de calidad de software?
5. ¿Cuáles son los proyectos o iniciativas de proyectos a corto o mediano plazo para mejorar la calidad de software?

C. MOTIVACIONES Y FRENOS

6. ¿Cuáles han sido los motivadores para mejorar la calidad de software?
7. ¿Cuáles son los frenos que usted ha encontrado para mejorar la calidad de software?

D. MODELOS

8. ¿Tienen o han pensado en implementar un modelo o certificación de calidad de software?
¿Cuál?

VARIABLES SOCIODEMOGRÁFICAS

1. Nivel de estudios alcanzados o que esté cursando el entrevistado
2. Años de experiencia en el cargo
3. Tiene formación con relación a calidad de software

Figura 13 – Guía de Entrevista

Previo a la entrevista presencial, se le entrega al director/a de la Dirección de Tecnología de la Universidad o jefe del área de desarrollo, la hoja de información para el participante y el formulario de consentimiento, documento que muestra la formalidad del caso y que ayuda a la decisión de participación en el estudio. Al término de la lectura de dicho documento, se procede a la firma conjunta de los participantes en dos documentos originales, siempre que el director/directora o jefe de desarrollo haya aceptado su participar en la investigación.

Para las entrevistas a distancia, la firma de consentimiento a la entrevista audio-grabada es sustituida por la respuesta favorable del director/a enviado por correo electrónico. Se usaron aplicativos que permitan videollamadas como Skype, WhatsApp, o simplemente el teléfono móvil para realizar la llamada.

La entrevista audio-grabada inicia con el uso de la grabadora, el teléfono móvil y la guía de entrevista.

2.2.2 Transcripción

El siguiente paso es la transcripción en texto de todas las audio-grabaciones, para lo cual, se procedió a consultar herramientas online y/o gratuitas que permitan realizar esta tarea. De las pruebas realizadas a esas herramientas como:

Speech to Text Demo (<https://speech-to-text-demo.ng.bluemix.net/>),

Express Scribe (<https://www.nch.com.au/scribe/index.html>), ninguna fue fiable al 100% por lo que se optó en realizar la transcripción de manera manual utilizando como editor de texto a Microsoft Word.

El cuerpo de la transcripción consta de una parte inicial donde se identifican a las etiquetas: entrevistador, entrevistado/s - cargo y el nombre de la universidad o escuela politécnica, donde se asignó un color a cada una de las etiquetas de los participantes de la entrevista, lo cual, permitió identificar de mejor manera la intervención de cada uno de ellos.

2.2.3 Confidencialidad

Por temas de confidencialidad de la información del participante e institución, se procede a generar el documento anonimizado de la transcripción (Anexo II), eliminando para ello,

los nombres de los participantes y dejando sólo las etiquetas que los representan en todo el texto que los mencionen. Así también, el nombre del documento es sustituido por el código de la institución asignado sin que tenga relación alguna con la lista de universidades y escuelas politécnicas participantes.

La Tabla 7, muestra la codificación asignada a cada universidad participante, así como el número de participantes en cada entrevista.

Tabla 7 - Codificación de Universidades y su número de participantes en la investigación

Universidad o Escuela Politécnica	No. de Participantes
UNI1	3
UNI2	1
UNI3	1
UNI4	1
UNI5	2
UNI6	1
UNI7	1
UNI8	2
UNI9	1
TOTAL	13

2.2.4 Guía o árbol de codificación

Para extraer los datos o información esencial de cada documento transcrito-anonimizado, es necesario aplicar una guía o árbol de codificación, donde se determina las categorías, subcategorías y definiciones [35], ver Figura 14.

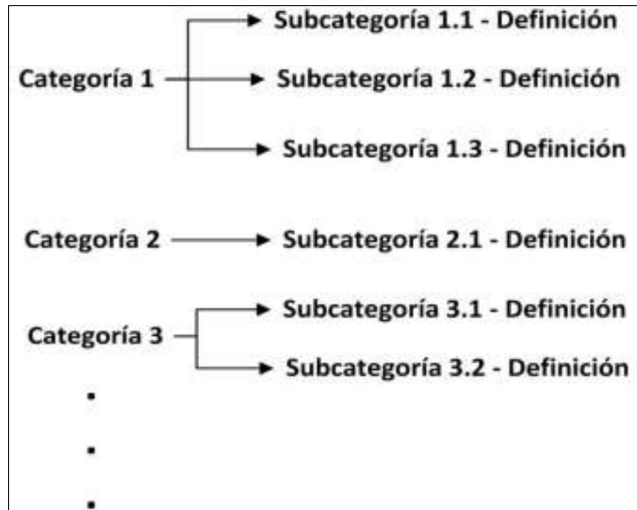


Figura 14 - Guía o árbol de codificación general

- Las categorías son los temas macro generales y diferenciales que engloba un tema en especial.
- Las subcategorías vienen a representar la clasificación de una categoría.
- Mientras que las definiciones son los significados o descripciones de las subcategorías.

Para determinar las categorías y subcategorías de esta investigación, se utilizó como base la guía de entrevista. El resultado se presenta en la Figura 15.

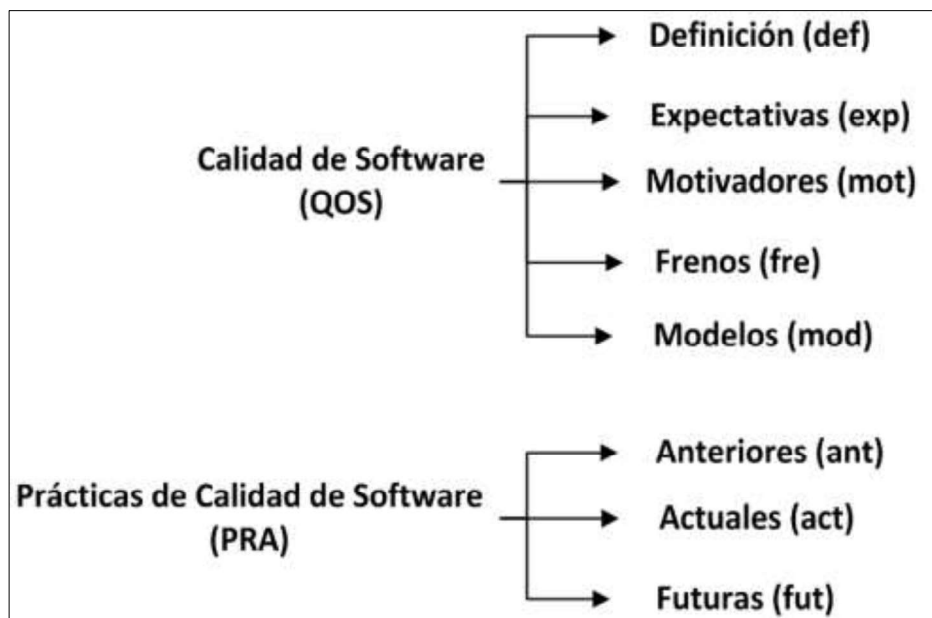


Figura 15 - Guía o árbol de codificación - Calidad de Software

La Tabla 8, presenta las subcategorías identificadas, su correspondiente definición según la literatura y la pregunta asociada según el estudio propuesto.

Tabla 8 - Subcategorías Identificadas

Subcategoría	Según la Real Académica de la Lengua	Según el estudio propuesto
Definición	f. Proposición que expone con claridad y exactitud los caracteres genéricos y diferenciales de algo material o inmaterial.	¿Qué es lo que entienden por calidad de software? ¿Qué no más involucra?
Expectativas	1. f. Esperanza de realizar o conseguir algo. 2. f. Posibilidad razonable de que algo suceda.	¿Qué es lo que ellos esperan de la calidad de software? Lo que puede aportar en la institución
Motivadores	Adj. Que motiva. Motiva: m. Causa o razón que mueve para algo.	¿Qué les impulsa a trabajar en calidad de software?
Frenos	m. Sujeción que se pone a alguien para moderar sus acciones.	¿Qué es lo que limita, detiene, entorpece el trabajar en calidad de software?
Modelos	Arquetipo o punto de referencia para imitarlo o reproducirlo.	¿Qué certificaciones, estándares, marcos de referencia mencionan respecto a calidad de software?
Anteriores	Adj. Que precede en lugar o tiempo.	¿Qué inicios llevaron a cabo en calidad de software?
Actuales	Adj. Dicho del tiempo en que se está: presente. Adj. Perteneciente o relativo al tiempo actual	¿Qué proyecto o iniciativa de proyecto están llevando a cabo para mejorar la calidad de software?
Futuras	Adj. Que está por venir y ha de suceder con el tiempo. Adj. Que todavía no es, pero va a ser.	¿Qué proyecto o iniciativa de proyecto a corto o mediano plazo van a llevar a cabo para mejorar la calidad de software?

Para completar la guía o árbol de codificación de esta investigación, falta por presentar las definiciones e información que se pretende encontrar en los documentos transcritos-anonimizados.

La identificación en el texto de una respectiva “definición”, podrá ser una palabra, conjunto de palabras o un párrafo (unidad de codificación variable). Su codificación estará representada por tres iniciales en mayúsculas de la categoría y las tres iniciales en minúsculas de la subcategoría. Por ejemplo: QOSexp (significa que se ha encontrado una definición de expectativas con relación a calidad de software).

Para ejecutar la codificación de las definiciones en todos los documentos transcritos-anonimizados, se utilizó la herramienta Atlas.ti versión 7 [40], lo que facilitó la obtención de la información a través de la generación de reportes según el tipo de definición.

Al final, se genera una matriz en Excel (Tabla 9), donde se presentan los campos: entrevista (código de la universidad), código (identificación de la definición), verbatim (descripción de la definición).

Tabla 9 - Matriz Genérica de Codificación

entrevista	Código	Verbatim
UNI1	Código 1	
UNI2	Código 2	
...	...	
UNIn	Código n	

3 RESULTADOS Y DISCUSIÓN

En este apartado, se presenta los resultados del estudio, para finalizar con el análisis cualitativo de la información de tipo análisis de contenido.

3.1 RESULTADOS

En base a la matriz en Excel (Anexo III), se procede a cuantificar los comentarios o aportaciones de cada universidad según el tipo de código (Tabla 10), observando que existen un total de ocho códigos identificados, nueve universidades participantes y 275 verbatimines.

Tabla 10 - Resultado del análisis de contenido de las intervenciones

Código	UNI1	UNI2	UNI3	UNI4	UNI5	UNI6	UNI7	UNI8	UNI9	Total
QOSdef	3	1	1	1	4	0	1	1	1	13
QOSexp	11	2	5	3	1	1	7	7	3	40
QOSmot	5	4	3	2	6	2	2	5	3	32
QOSfre	22	1	7	3	0	3	9	1	13	59
QOSmod	3	0	0	4	1	2	1	4	3	18
PRAant	10	3	1	2	3	0	0	2	0	21
PRAact	14	3	4	5	10	4	5	5	4	54
PRAfut	8	1	5	1	4	3	5	1	10	38
Totales	76	15	26	21	29	15	30	26	37	275

3.1.1 Definición de la Calidad de Software (QOSdef)

Ocho de nueve universidades indican definiciones de calidad de software. En algunos casos se tiene más detalle e indican características como: fases, tecnología e información. En la Tabla 11, se presentan los verbatimines más representativos.

Tabla 11 - Definiciones de Calidad de Software

Definición Calidad de Software según la Teoría	Ejemplos de definiciones de Calidad de Software indicadas por los participantes de las Universidades
<p>La calidad del software se refiere a satisfacer las necesidades del usuario haciendo que su trabajo sea confiable y consistente, lo cual requiere que el software que se produce tenga pocos defectos o ninguno (W. S. Humphrey)</p>	<p><i>“La calidad de software está relacionada con que el producto de software que nosotros estamos desarrollando, construyendo cumpla primero con todos los requisitos que han sido solicitados por el cliente lo podemos llamar así, y que también cumpla con todas las características que trabajamos como usabilidad, seguridad, eficiencia, en la parte de rendimiento también de software hay algunos criterios de diferentes autores, diferentes técnicas y estándares que se van incluyendo en los procesos cuando hablamos de calidad de software, pero básicamente es eso que cumpla con los requerimientos del usuario” (UNI1).</i></p>
<p>Proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan (R. Pressman)</p>	<p><i>“Calidad de software primero que cumpla las necesidades del usuario final y a la vez que también en la parte de desarrollo a nivel técnico pues cumpla con todos los estándares o las metodologías, que el desarrollador elija teniendo en cuenta no sólo la parte de desarrollo sino también algunos otros factores como la parte de seguridad, de acceso, de privilegios, de roles o se haga algo integral no solamente de desarrollo no, si no se desarrolla obviamente tiene una gran responsabilidad porque el desarrollo al fin y al cabo entrega un producto para ser utilizado por usuarios de diferente índole y diferente criticidad como no se pues, un sistema educativo financiero sistema académico que debe guardar concordancia no solamente, a nivel funcional sino también con todos los otros como digo variables que deben tomarse en cuenta, sobre todo la parte de accesos, seguridad, integridad, disponibilidad de esa información” (UNI3)</i></p>
	<p><i>“La calidad de software, la calidad de software depende del servicio que brinde esa aplicación al usuario, que tenga plena satisfacción, que se encuentre sin fallas, se encuentre sin errores que sea, que esté al tiempo, que esté disponible” (UNI5)</i></p>

Una de las universidades no da una definición de calidad de software sólo menciona *"de momento lo estamos viendo más como unas etapas de pruebas"* (UNI6).

3.1.2 Expectativas de la Calidad de Software (QOSexp)

Las nueve universidades presentan expectativas en términos de calidad de software, las cuales fueron agrupadas según las siguientes categorías: información, usuario, desarrollador/área, producto/software y universidad. La Tabla 12, muestra el detalle de cada categoría y un ejemplo de verbatim.

Tabla 12 - Listado de expectativas de Calidad de Software

Expectativas de Calidad de Software	
Esperamos que la Información sea	
<ul style="list-style-type: none"> • Íntegra <p><i>“que se asegure también la integridad de la información que se va a manejar dentro de estos sistemas” (UNI1)</i></p>	<ul style="list-style-type: none"> • Consolidada <p><i>“queremos llegar a lo general ... que se pueda cumplir donde la información se encuentre consolidada” (UNI3)</i></p>
<ul style="list-style-type: none"> • Esté disponible <p><i>“permita tener información en el momento que se necesite y que éste integra y disponible” (UNI3)</i></p>	<ul style="list-style-type: none"> • Pertinente/Adecuada <p><i>“que los resultados que emita el software sean lo que esperamos” (UNI9)</i></p>
<ul style="list-style-type: none"> • Esté segura <p><i>“que ese software no permita ..., robo de información, algún otro tipo de atentados que vayan más allá de lo del desarrollo en sí” (UNI7)</i></p>	
Esperamos que en relación al usuario, nuestro software	
<ul style="list-style-type: none"> • Sea del agrado <p><i>“sea de gusto del usuario” (UNI1)</i></p>	<ul style="list-style-type: none"> • Permita el máximo de provecho <p><i>“es que el software que nosotros desarrollamos sea utilizado al ciento por ciento por los usuarios finales” (UNI5)</i></p>
<ul style="list-style-type: none"> • Responda a sus necesidades <p><i>“mejorar, ... para que pues estos puedan responder justamente a las necesidades del usuario” (UNI2)</i></p>	<ul style="list-style-type: none"> • Satisfaga sus expectativas <p><i>“el usuario este satisfecho” (UNI8)</i></p>
<ul style="list-style-type: none"> • Genere valor <p><i>“también lo que se requiere a nivel tecnológico para que la solución sea realmente de calidad, aporte a la institución, a los usuarios” (UNI1)</i></p>	<ul style="list-style-type: none"> • Sea entendible <p><i>“software fiable para el usuario y para el desarrollador, entendible también tanto para la una persona y la otra,” (UNI8)</i></p>
<ul style="list-style-type: none"> • Presente un entorno amigable <p><i>“que tenga un entorno amigable sobre los sistemas que está utilizando” (UNI8)</i></p>	
Las expectativas del desarrollador/área de desarrollo	
<ul style="list-style-type: none"> • Que el software sea entendible <p><i>“software fiable para el usuario y para el desarrollador, entendible también tanto para la una persona y la otra,” (UNI8)</i></p>	<ul style="list-style-type: none"> • Tener un equipo calificado <p><i>“integrantes que sepan y que tengan experiencia, para poder tener y desarrollar una arquitectura de tecnología de desarrollo y que esté acorde a esta institución” (UNI8)</i></p>
<ul style="list-style-type: none"> • Se segreguen funciones <p><i>“aspiro poco para segreguar funciones para que los desarrolladores no estén en la data de producción sino segregados” (UNI1)</i></p>	<ul style="list-style-type: none"> • Se defina un modelo/metodología de calidad de software para su institución <p><i>“llegar a definir un modelo de calidad de software para la UNI6 en la parte de TIC’s” (UNI6)</i></p>

Expectativas de Calidad de Software	
Las expectativas del desarrollador/área de desarrollo	
<ul style="list-style-type: none"> • Se separen ambientes <p><i>"en la parte de desarrollo todavía nos falta separar todos los ambientes, en desarrollo y código"</i> (UNI1)</p>	<ul style="list-style-type: none"> • Se forme/eduque a la gente <p><i>"queremos formar un poco a la gente, queremos educar un poco,"</i> (UNI7)</p>
<ul style="list-style-type: none"> • Se tenga poder de decisión para dimensionar sus implicaciones <p><i>"que siempre nos permitan como tecnología dimensionar no, este cambio, tiene estos impactos y tiene estos tiempos para hacer el desarrollo, para hacer un testing, QA, ... también lo que se requiere a nivel tecnológico para que la solución sea realmente de calidad"</i> (UNI1)</p>	
Características del Producto/Software como tal	
<ul style="list-style-type: none"> • Confiable <p><i>"producto es confiable"</i> (UNI1)</p>	<ul style="list-style-type: none"> • Fiable • Entendible <p><i>"software fiable para el usuario y para el desarrollador, entendible también tanto para la una persona y la otra"</i> (UNI8)</p>
<ul style="list-style-type: none"> • Seguro <p><i>"... podrían ellos implementar los procesos de software de desarrollo y siempre asegurando el tema de calidad"</i> (UNI1)</p>	<ul style="list-style-type: none"> • Cumple las funciones requeridas • Tenga consistencia • Genere errores no contemplados <p><i>"es que el software que se pone en ambientes productivos cumple las funciones requeridas ... tengan consistencia si, y que generen errores no contemplados"</i> (UNI9)</p>
<ul style="list-style-type: none"> • Identificar tipo de software que se utiliza <p><i>"identificar probablemente qué tipo de software se está ocupando, quizás en a un futuro"</i> (UNI2)</p>	<ul style="list-style-type: none"> • Garantice utilizar buenas prácticas de desarrollo <p><i>"principalmente que la calidad del software no esté solamente para cumplir con su tarea, sino que garantice otras buenas prácticas si en el desarrollo"</i> (UNI7)</p>
<ul style="list-style-type: none"> • No permita vulnerabilidades <p><i>"que ese software no permita vulnerabilidades, robo de información, algún otro tipo de atentados que vayan más allá de lo del desarrollo en sí"</i> (UNI7)</p>	<ul style="list-style-type: none"> • Tenga calidad <p><i>"si nosotros realizamos un adecuado control de calidad en el software podemos tener un producto de calidad"</i> (UNI4)</p>
Se espera que la universidad pueda	
<ul style="list-style-type: none"> • Crear un grupo investigador <p><i>"sería bueno instaurar en la academia o a las universidades un grupo que se dedique al tema de calidad, ósea, me refiero como por ejemplo tenemos el grupo de los repositorios de CEDIA,"</i> (UNI4)</p>	<ul style="list-style-type: none"> • Unifique la información <p><i>"queremos llegar a lo general ... que pueda interactuar esa información con los diferentes sistemas"</i> (UNI3)</p>

Expectativas de Calidad de Software	
Se espera que la universidad pueda	
<ul style="list-style-type: none"> Reducción de costos <p><i>“porque finalmente si ahondamos esfuerzos entre las universidades públicas el costo va a ser menos a que lo haga una sola persona, una sola institución” (UNI4)</i></p>	<ul style="list-style-type: none"> Posea sistemas centralizados <p><i>“lo que queremos es tener un, un sistema centralizado” (UNI3)</i></p>
<ul style="list-style-type: none"> Intentar estandarizar una metodología para todas las universidades <p><i>“todo esto que se recabe de las universidades llegue al órgano rector para que sus peticiones, su software a largo plazo intentemos estandarizar y tener una metodología clara para todos poder alinearnos a eso” (UNI7)</i></p>	<ul style="list-style-type: none"> Cliente interno y externo satisfechos <p><i>“garantizar que los dos clientes tanto el interno como el externo estén satisfechos” (UNI7)</i></p>
<ul style="list-style-type: none"> Automatizar sus procesos <p><i>“mejorar, automatizando con cada uno de los procesos institucionales para que pues estos puedan responder justamente a las necesidades del usuario” (UNI2)</i></p>	<ul style="list-style-type: none"> Buena comunicación entre todos <p><i>“el alcance real de lo que se va a poner antes y que todos estén claros, que exista una buena comunicación para que todos estén claros de lo que se espera al final” (UNI9)</i></p>
<ul style="list-style-type: none"> Trabajo colectivo por procesos <p><i>“la idea es hacerlo colectivamente y los procesos nos ayudan” (UNI3)</i></p>	

3.1.3 Motivadores de la Calidad de Software (QOSmot)

Las nueve universidades indican motivadores en términos de calidad de software, los cuales fueron agrupados en motivadores proactivos y motivadores reactivos en función del comentario realizado por el participante.

Motivadores reactivos son renombrados como motivadores resultantes de un problema, mientras que los motivadores proactivos son los motivadores resultantes de iniciativas pro-mejora.

La Tabla 13, muestra los verbatimés más representativos en el tema de motivadores de la calidad de software.

Tabla 13 - Motivadores de la Calidad de Software

Motivadores Proactivos	Motivadores Reactivos
<ul style="list-style-type: none"> Tener buenas soluciones tecnológicas e informáticas, productos de software de calidad que les aporta a su trabajo, a la institución para procesos de mejora continua (UNI1, UNI5, UNI6, UNI8) 	<ul style="list-style-type: none"> Software o sistemas que no se adaptan a la realidad de lo que necesita el usuario lo cual necesita mejoras. Pérdida de tiempo en la calidad de software (UNI1, UNI7, UNI8)
<ul style="list-style-type: none"> Mejorar la calidad del servicio (UNI2, UNI5) 	<ul style="list-style-type: none"> Tener duplicidad, registros triplicados de la misma persona. (UNI3)
<ul style="list-style-type: none"> Un público bien exigente, usuarios, estudiantes, docentes, profesionales de alto nivel exigen siempre los mejores productos con la mejor calidad, entonces esa constante retroalimentación, conversaciones, acercamiento que se tiene con ellos, es uno de los principales motivadores que tiene para seguir evolucionando rápido (UNI1) 	<ul style="list-style-type: none"> Hacerle entender al usuario que tenemos que trabajar bajo un proceso, sí, porque si fue un poco complicado, que ellos entiendan que para que un software salga debidamente con las características que ellos quieren teníamos que seguir un proceso, un requerimiento (UNI8)
<ul style="list-style-type: none"> Apoyo institucional, trabajo en conjunto desde las autoridades, el apoyo de las autoridades, el trabajo conjunto con cada una de las unidades, en cada una de las etapas del proceso y eso es lo que justamente garantiza que el software sea de calidad (UNI2, UNI4) 	<ul style="list-style-type: none"> Tema reputación del área no, porque cuando los softwares se ponen y fallan pues la reputación del área se ve afectada, muchas veces, la afectación del software no es porque se caiga el programa, sino porque no hace lo que se solicita (UNI9)
<ul style="list-style-type: none"> Permite tomar decisiones al momento (UNI3) 	<ul style="list-style-type: none"> El problema general del desarrollo es los resultados (UNI7)
<ul style="list-style-type: none"> Información centralizada, consolidada, real, transparente, válida (UNI3) 	<ul style="list-style-type: none"> La crítica de productos que tienen novedades el momento de salir a producción (UNI6)
<ul style="list-style-type: none"> Apoyo económico (UNI2) 	<ul style="list-style-type: none"> El álgido o el alto número de soporte y mantenimiento que se da a las aplicaciones (UNI4)
<ul style="list-style-type: none"> Prestigio institucional (UNI5) 	
<ul style="list-style-type: none"> Arquitectura y tecnología que se utilice. Avances en tecnología (UNI8) 	
<ul style="list-style-type: none"> El tema de seguridad (UNI9) 	

3.1.4 Frenos de la Calidad de Software (QOSfre)

Ocho de las nueve universidades indicaron frenos u obstáculos en cuanto a calidad de software, las cuales se agrupan en categorías. La Figura 16, muestra el detalle de cada una de las doce categorías: falta de tiempo, requerimientos, personal, presupuesto, usuario/funcionales, procesos/procedimientos, legislación, autoridades, unidades como entes aislados, versionamiento, falta de metodologías y falta de documentación formal escrita.

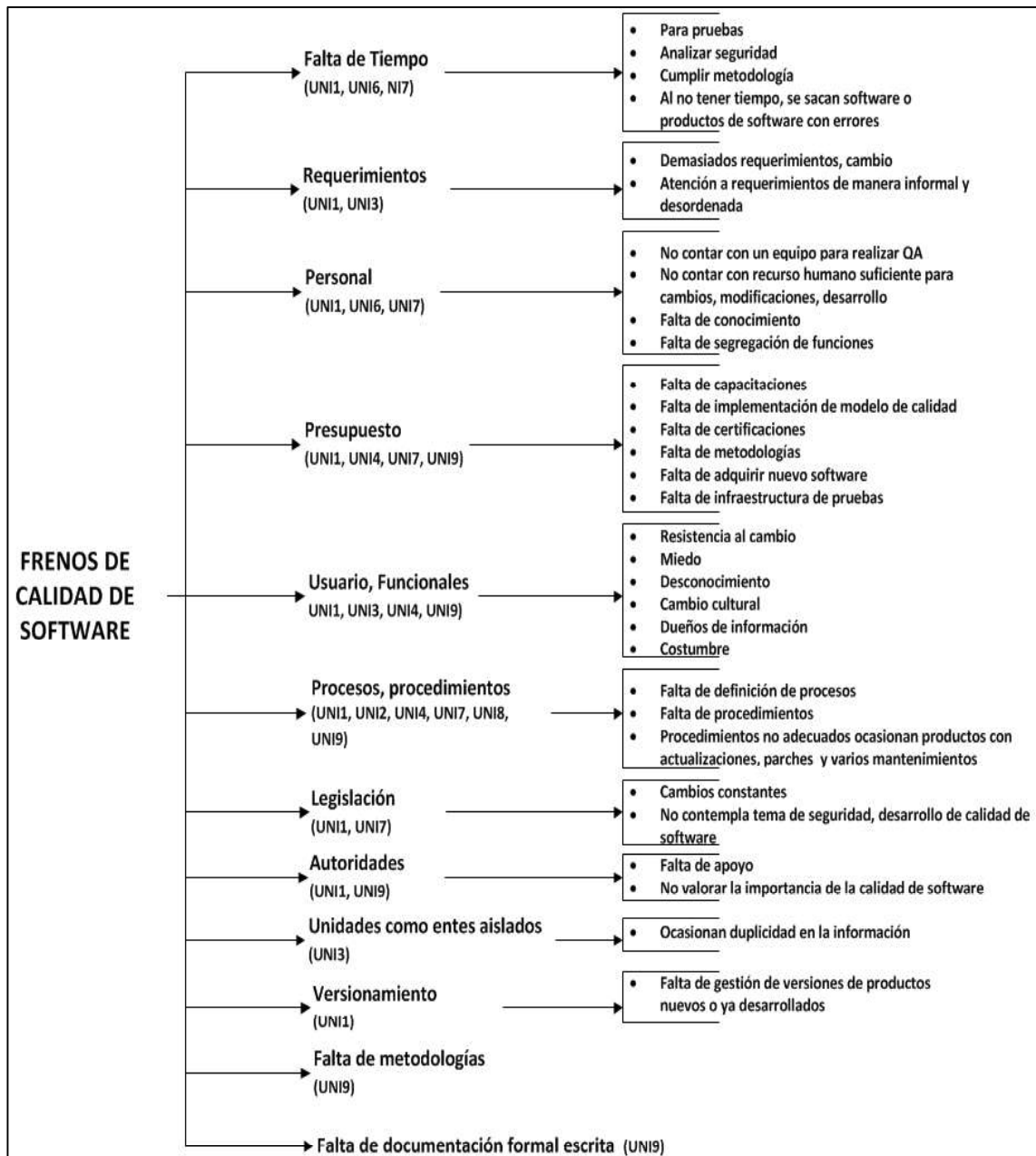


Figura 16 - Clasificación de frenos de Calidad de Software en Universidades Ecuatorianas

La Figura 17, resalta que seis de las universidades indican frenos relacionados con los procesos/procedimientos, cuatro universidades identifican frenos relacionados con los usuarios/funcionales y presupuesto y tres universidades identifican frenos con temas de personal y falta de tiempo.

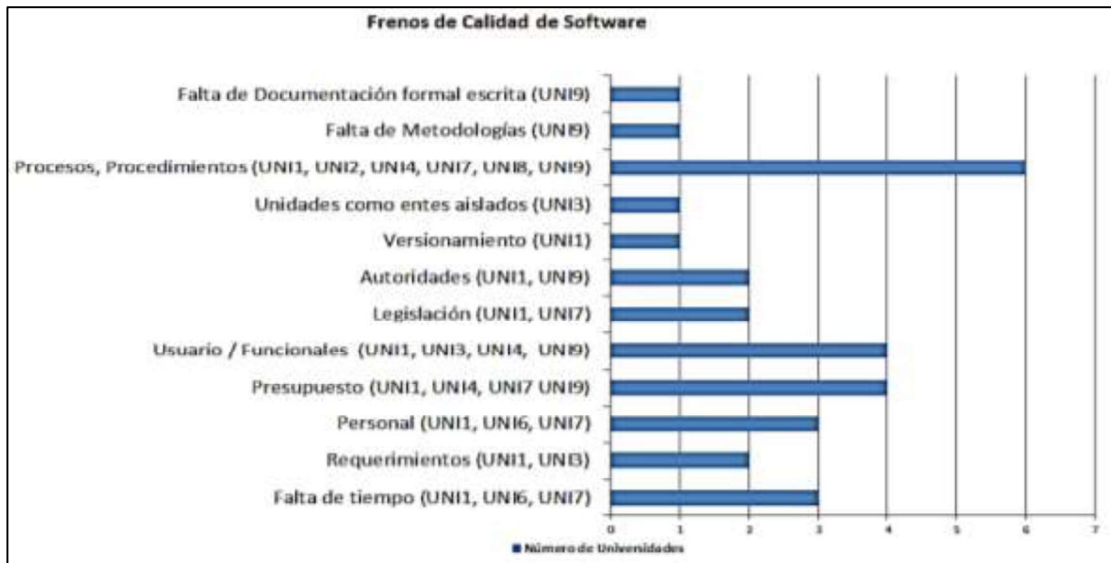


Figura 17 - Frenos de la calidad de software según Universidades Ecuatorianas

La UNI5, mencionó no tener ningún freno en la entrevista.

3.1.5 Modelos de Calidad de Software (QOSmod)

Siete de las nueve universidades mencionaron en alguna parte de la entrevista, términos con relación a metodología, modelo o estándar relacionado con la calidad de software, lo cual se puede visualizar en la Figura 18, donde también se visualiza que dos universidades mencionaron la metodología XP y Srum. Mientras que las universidades UNI2 y UNI3 no mencionaron algún término relacionado.

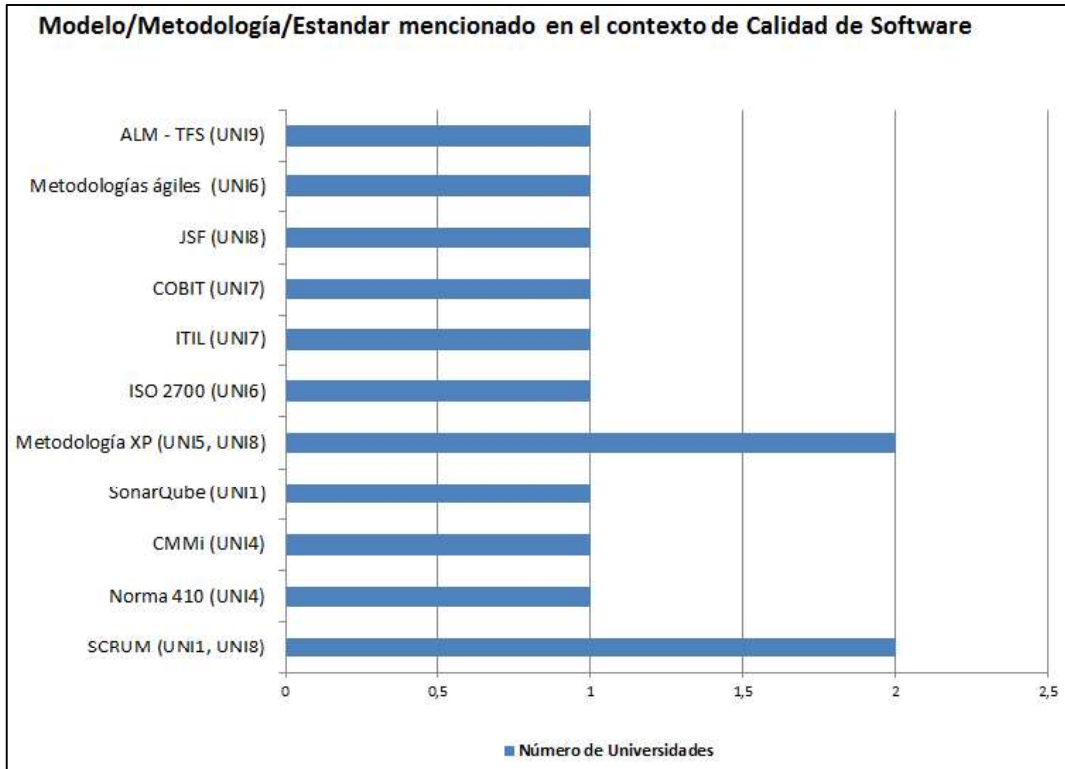


Figura 18 - Modelo/Metodología/Estándar mencionada en el contexto de Calidad de Software en las Universidades Ecuatorianas

Las universidades UNI1, UNI2, UNI3, UNI6 Y UNI7 indican que el tema de definir un modelo de calidad de software lo tendrían como un proyecto a futuro ya que en este momento se encuentran cubriendo otros temas.

3.1.6 Prácticas en Calidad de Software (PRAant, PRAact, PRAfut)

Seis de las nueve universidades identificaron prácticas anteriores, mientras que las nueve universidades identificaron prácticas actuales y futuras (Tabla 14). Una vez agrupadas según su relación, se presenta nueve categorías las cuales son formación profesional y capacitación, invertir en tecnología, evaluaciones de calidad, proyectos, determinar normativas, involucramiento de los stakeholders, procesos, tecnología e información.

Tabla 14 - Prácticas de Calidad de Software implementadas o por implementar en las Universidades del Ecuador

PRÁCTICAS DE CALIDAD DE SOFTWARE	ANTERIORES	ACTUALES	FUTURAS
Formación profesional y Capacitación (UNI1, UNI2, UNI3, UNI5, UNI6, UNI7, UNI8)	Al personal técnico del área		En nuevas tecnologías
	--	En temas técnicos y de procesos	
Invertir en Tecnología (UNI1, UNI3, UNI2, UNI8, UNI9)	Interoperabilidad	--	Interoperabilidad
	Infraestructura tecnológica	Contratación de una empresa externa para apoyo	
	Entornos de desarrollo (preproducción, producción, pruebas)		--
	Arquitectura, middleware		--
	Herramientas tecnológicas	--	--
	Compras de licencias	--	--
	Seguridad	Certificación del usuario	Certificación de calidad general
Evaluaciones de calidad (UNI1, UNI2, UNI3, UNI5, UNI6, UNI7, UNI8, UNI9)	Participación de grupos de estudiantes de las carreras de los últimos niveles para evaluación de calidad de software	Uso de herramientas de evaluación como SonarQube	--
	--	Testing (pruebas unitarias, pruebas de estrés, pruebas de carga, pruebas de usuarios)	--
	--	Revisión de métodos, diseños, análisis	--

PRÁCTICAS DE CALIDAD DE SOFTWARE	ANTERIORES	ACTUALES	FUTURAS
Evaluaciones de calidad (UNI1, UNI2, UNI3, UNI5, UNI6, UNI7, UNI8, UNI9)	--	Creación de plantillas (análisis, de especificaciones de requerimientos, desarrollo, construcción)	--
	--	Apoyo en software para registrar toda la documentación y para ir pasando cada una de las etapas de desarrollo	--
Proyectos (UNI2, UNI4, UNI5, UNI8, UNI9)	Creación de una herramienta/sistema que cubre todos los procesos institucionales	Actualizar el corre académico	--
	--	Levantamiento de nuevos requerimientos	--
	--	Implementar la metodología del ciclo de vida de aplicaciones	--
	Cubrir todas las etapas de desarrollo todo el tiempo		--
	Creación de nuevos sistemas		--
Determinar normativas (UNI1, UNI3, UNI4)	--	Políticas	
		--	Establecer procedimientos técnicos y de documentación

PRÁCTICAS DE CALIDAD DE SOFTWARE	ANTERIORES	ACTUALES	FUTURAS
Determinar normativas (UNI1, UNI3, UNI4)	--	--	Establecer formalmente tiempos
Involucramiento de los stakeholders (UNI5, UNI8)	Participación	Segregación de funciones	--
Procesos (UNI1, UNI3, UNI7, UNI9)	Entender todo el o los procesos antes de comenzar el desarrollo	--	Establecer metodologías, modelo
	--	--	Calidad de software
	--	--	Crear una unidad de calidad
	--	Aseguramiento de la calidad	--
	--	Conocimiento de todas las afectaciones cuando se involucra un cambio	--
Tecnología (UNI1, UNI3, UNI4, UNI5, UNI6, UNI8, UNI9)	--	--	Control de versiones
	--	--	Integración continua
	--	--	Minería de Datos

PRÁCTICAS DE CALIDAD DE SOFTWARE	ANTERIORES	ACTUALES	FUTURAS
Tecnología (UNI1, UNI3, UNI4, UNI5, UNI6, UNI8, UNI9)	--	--	Herramientas para realizar pruebas automáticamente
	--	Mantener actualizado el software que se utiliza para el desarrollo, administración, etc	CMMi
	--	JSF	modernización del tipo de herramienta que utilizamos
	--	--	IA
	--	--	Metodologías ágiles
	--	--	ISO 27000
	--	--	ALM-TFS
	Información (UNI2, UNI3, UNI5)	Estándares de seguridad de información	
		Integrar y centralizar la información	

3.1.7 Resumen

A continuación, se presenta en la Tabla 15 un resumen de las aportaciones de las universidades en cuanto a los temas planteados en este estudio, los casilleros pintados en rojo representan que la universidad no ha entregado aporte alguno, mientras que, el color verde representa que dicha universidad ha entregado uno o más aportes a este estudio.

Tabla 15 - Resumen del Estudio

Calidad de Software	UNI1	UNI2	UNI3	UNI4	UNI5	UNI6	UNI7	UNI8	UNI9
• Definición	Verde	Verde	Verde	Verde	Verde	Rojo	Verde	Verde	Verde
• Expectativas	Verde	Verde	Verde	Verde	Verde	Verde	Verde	Verde	Verde
• Motivadores	Verde	Verde	Verde	Verde	Verde	Verde	Verde	Verde	Verde
• Frenos	Verde	Verde	Verde	Verde	Rojo	Verde	Verde	Verde	Verde
• Modelos	Verde	Rojo	Rojo	Verde	Verde	Verde	Verde	Verde	Verde
• Prácticas anteriores	Verde	Verde	Verde	Verde	Verde	Rojo	Rojo	Verde	Rojo
• Prácticas actuales	Verde	Verde	Verde	Verde	Verde	Verde	Verde	Verde	Verde
• Prácticas futuras	Verde	Verde	Verde	Verde	Verde	Verde	Verde	Verde	Verde

3.2 DISCUSIÓN

Para realizar el estudio, se invitaron a ocho universidades o escuelas politécnicas del país cuya categoría es A y 3 IES cuya categoría es B. La invitación se realizó a través de llamada telefónica o email, de las cuales nueve universidades aceptaron la invitación a la entrevista. Una universidad respondió el email indicando su deseo de participar del estudio; sin embargo, por asuntos internos de trabajo, esta universidad no pudo realizar la entrevista. Mientras que la otra universidad faltante no dio respuesta alguna a la solicitud de entrevista por ninguno de los medios de comunicación utilizados.

Es así como en base a los resultados presentados en el apartado 3.1, a continuación, se listan las interpretaciones realizadas por el entrevistador en base a cada subcategoría identificada en términos de calidad de software.

3.2.1 Definición de la Calidad de Software

Ocho de las nueve universidades indicaron una definición, de las cuales cuatro instituciones coinciden con las definiciones de calidad de software indicado en la teoría y cuatro definiciones corresponden a definiciones de calidad en general. La diferencia entre una y otra definición, es que la definición de calidad de software explícitamente menciona el término producto de software, proceso de software, aplicación y/o desarrollo, mientras que la definición general menciona producto, proceso, componente o sistema.

Lo indicado en la teoría y lo expuesto por las universidades coinciden en velar por que el software que se haya diseñado cumpla con los requisitos que fueron levantados juntamente con los usuarios; asegurar que el producto/sistema que se haya desarrollado cumpla las necesidades del usuario; que tenga plena satisfacción el usuario; y en; que éste se encuentre sin fallas, sin errores.

Así también, las universidades listan funcionalidades internas y externas en el marco de la definición de calidad de software, como es: usabilidad, seguridad, eficiencia, rendimiento, privilegios, roles, integridad, disponibilidad de la información y optimización de los recursos.

Por último, la universidad faltante, aunque no menciona una definición de calidad de software, realiza una relación directa de la etapa de pruebas como uno de los puntos de partida en el tema de calidad de software.

3.2.2 Expectativas y Motivadores de la Calidad de Software

El haber presentado el significado de expectativa y motivador fue el determinante para poder identificar los correspondientes verbatines, ya que la expectativa es un deseo razonable de que algo suceda mientras que los motivadores son la causa/origen de algo y; cualquiera de los dos términos provoca que se piense en tomar alguna acción. Siendo por ellos que las nueve universidades mencionaron expectativas relacionadas a mejorar o contribuir en temas relacionados con la universidad, producto/software, desarrollador/área, usuario e información, mientras que sus principales motivadores identificados entre proactivos y reactivos provocan la acción de brindar un mejor servicio al cliente interno y externo con soluciones tecnológicas que realmente aporten al trabajo diario.

3.2.3 Frenos de la Calidad de Software

El listado de los frenos u obstáculos fue contribución de ocho de las nueve universidades, identificadas según condiciones antes, pasadas o que actualmente se encuentran atravesando las IES. La mayor contribución que se tiene a lo largo de las entrevistas es justamente los verbatines de frenos con 59 comentarios agrupados en 12 categorías para un mejor entendimiento del tema.

Se resalta las falencias internas de la institución como el tema de falta de tiempo, requerimientos, personal, presupuesto, usuario/funcionales, autoridades, versionamiento, unidades como entes aislados, los procesos/procedimientos, la falta de metodologías y la falta de documentación formal escrita, mientras que como falencia interna/externa se tiene el tema de la legislación que al ser muy cambiante no termina de aterrizar.

El freno indicado en la literatura “Incompatibilidad con aplicaciones existentes” no fue identificado por ninguna IES, sin embargo, se lo puede incluir como una categoría más debido al interés del investigador en listar la mayor cantidad de frenos.

También, fue muy interesante encontrar una IES que mencionó el no tener ningún freno y de lo conversado, se pudo apreciar que el apoyo en todos los sentidos de todas las partes interesadas es el determinante para obtener dicha respuesta.

3.2.4 Modelos de la Calidad de Software

Las aportaciones de las IES respecto a metodología/modelos/estándar mencionados en alguna parte de la entrevista, necesitaron de una descripción general de cada uno de ellos para determinar si corresponden o no a modelos reconocidos como modelos de calidad de software, siendo necesario añadir una identificación de su ámbito de aplicación lo cual se puede visualizar en la Tabla 16.

Tabla 16 - Descripción y ámbito de Aplicación de los términos presentados en el contexto Modelos de Calidad de Software

Modelos relacionados al tema de Calidad de Software	Descripción	Ámbito de Aplicación
<ul style="list-style-type: none"> SCRUM (UNI1, UNI8) 	<ul style="list-style-type: none"> Metodología del ciclo de vida para Desarrollo Ágil de Software [41]. 	<ul style="list-style-type: none"> Ciclo de vida de Desarrollo de Software
<ul style="list-style-type: none"> Norma 410 (UNI4) 	<ul style="list-style-type: none"> Norma de la Contraloría General del Estado ecuatoriano, dirigida para uso de Tecnologías de la Información para entidades del sector público y personas jurídicas de derecho privada, que dispongan de recursos públicos [42]. 	<ul style="list-style-type: none"> Gestión de TI
<ul style="list-style-type: none"> CMMi (UNI4) 	<ul style="list-style-type: none"> "Modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software" [43]. 	<ul style="list-style-type: none"> Evaluación de procesos de calidad para el desarrollo de software
<ul style="list-style-type: none"> SonarQube (UNI1) 	<ul style="list-style-type: none"> Herramienta para el análisis y evaluación de código fuente que proveen indicadores y métricas para la calidad de software a desarrollar [44]. 	<ul style="list-style-type: none"> Herramienta de análisis de la calidad de código fuente
<ul style="list-style-type: none"> Metodología XP (UNI5, UNI8) 	<ul style="list-style-type: none"> Metodología del ciclo de vida para Desarrollo Ágil de Software [41]. 	<ul style="list-style-type: none"> Ciclo de vida de Desarrollo de Software
<ul style="list-style-type: none"> ISO 2700 (UNI6) 	<ul style="list-style-type: none"> "La serie de normas ISO 27000 contiene las mejores prácticas recomendadas en seguridad de la información para desarrollar, implementar y mantener Especificaciones para los Sistemas de Gestión de la Seguridad de la Información (SGSI)" [20]. 	<ul style="list-style-type: none"> Gestión de la Seguridad
<ul style="list-style-type: none"> ITIL (UNI7) 	<ul style="list-style-type: none"> "Conjunto de conceptos y prácticas para la gestión de servicios de tecnologías de la información" [20]. 	<ul style="list-style-type: none"> Gestión Servicios de TI
<ul style="list-style-type: none"> COBIT (UNI7) 	<ul style="list-style-type: none"> "Estándar propuesto por ISACA para la Auditoría y el control de las TI, su última versión incluye elementos de gobierno de las TI" [20]. 	<ul style="list-style-type: none"> Auditoría, Control y Gobierno de TI
<ul style="list-style-type: none"> JSF (UNI8) 	<ul style="list-style-type: none"> Java Server Faces (JSF) es un marco de aplicación web basado en Java destinado a simplificar la integración del desarrollo de interfaces de usuario basadas en web [45]. 	<ul style="list-style-type: none"> Desarrollo de Software WEB

Modelos relacionados al tema de Calidad de Software	Descripción	Ámbito de Aplicación
<ul style="list-style-type: none"> • Metodologías ágiles (UNI6) 	<ul style="list-style-type: none"> • Nombre general de todas las metodologías del ciclo de vida para desarrollo ágil de software que engloba por ejemplo a XP, SCRUM y otras más [41]. 	<ul style="list-style-type: none"> • Ciclo de vida de Desarrollo de Software
<ul style="list-style-type: none"> • ALM - TFS (UNI9) 	<ul style="list-style-type: none"> • ALM Application Lifecycle Management fue creado para admitir una administración superior para entregar la aplicación basada en la línea de tiempo con facilidad y perfección. ALM monitorea y controla de cerca todas las etapas del ciclo de vida de la aplicación, como la definición de requisitos, diseño, desarrollo, control de calidad, implementación y, finalmente, mantenimiento y asistencia. • TFS, Team Foundation Server es una herramienta ALM que es un servidor independiente que admite a todos los involucrados en el ciclo de vida de las aplicaciones, como desarrolladores, profesionales de control de calidad, arquitectos de tecnología, gerentes de proyectos, analistas de sistemas y equipo de soporte. Proporciona control de versiones, sistema de compilación, herramientas y métricas necesarias para la gestión y organización de proyectos.[46] 	<ul style="list-style-type: none"> • Ciclo de vida de Desarrollo de Software

De lo revisado, es notorio que sólo una IES (UNI4) pudo identificar un modelo relacionado a la calidad de procesos de software. Las demás aportaciones tienen alguna relación con la calidad de software y se refieren al ámbito de ciclo de vida de desarrollo de software, gestión de TI y gestión de seguridad de forma general.

El tener un modelo o certificación de calidad de procesos de software como proyecto futuro en las nueve IES, resalta el interés, importancia y deseo de mejorar sus procesos internos lo que contribuirá a mejorar la calidad de forma general en la institución.

3.2.5 Prácticas en Calidad de Software

La Tabla 14, permite observar que todas las nueve universidades se encuentran interesadas en cubrir o mejorar el tema de calidad de manera general en su institución, por lo que han ejecutado o se encuentran ejecutando una serie de actividades las cuales fueron agrupadas o clasificadas obteniendo nueve categorías. Se visualiza que la categoría de Evaluaciones de calidad (ocho IES) y; Formación profesional y capacitación (siete IES) son las prácticas más utilizadas en el tema de calidad de software en la mayoría de las universidades, y; que junto con Invertir en tecnología (cinco IES), Proyecto (cinco IES), Procesos (cuatro IES), Determinar Normativas (tres IES), Información (tres IES) e Involucramiento de los stakeholders (dos IES) son las prácticas identificadas como un punto de partida y/o continuidad para el tema de calidad general y/o software, mientras que la categoría Tecnología (siete IES) y algunos ítems de la categoría de Procesos son considerados como prácticas futuras.

La categoría “Tecnología” presenta el mayor número de aportaciones identificadas (12 ítems) a nivel general, resaltando que la mayoría de estos ítems se han programado o pensado implementar como prácticas futuras, entre los cuales se encuentra el control de versiones, integración continua, minería de datos, herramientas para realizar pruebas automáticamente, modernización del tipo de herramientas que usan e IA (Inteligencia Artificial). La segunda categoría en presentar más aportaciones (nueve ítems) es “Evaluaciones de calidad”, entre las cuales se resalta el tema de seguridad, certificación del usuario, uso de herramientas de evaluación y documentación; y la certificación de calidad general.

Las prácticas implementadas o por implementar son las acciones que permitirán contrarrestar los frenos u obstáculos identificados por las universidades ecuatorianas (Figura 16), así también ayudarán a contrarrestar los motivadores reactivos identificados en la Tabla 13, mientras que, dichas prácticas son las acciones que permitirán cumplir y cubrir las expectativas de calidad de software identificadas en la Tabla 12.

4 RECOMENDACIONES PARA LA MEJORA DE CALIDAD EN LOS PROCESOS DE DESARROLLO DE SOFTWARE EN UNIVERSIDADES DEL ECUADOR

A continuación, se presentan breves recomendaciones para la mejora de la calidad en los procesos de desarrollo de software en universidades del Ecuador Tabla 17; en base a la identificación de la práctica específica (SP) y/o área de proceso del modelo CMMiDEV (modelo identificado en la entrevista y seleccionado para la comparación), el cual mitiga o eliminará la correspondiente categoría de freno u obstáculo mostrado en el apartado 3.1.4 de este estudio.

Tabla 17 – Recomendaciones para la mejora de calidad en los procesos de desarrollo de software en universidades del Ecuador – Basado en el modelo CMMiDEV

Frenos u Obstáculos	CMMI Dev [31]	Recomendación
Falta de Tiempo	<p>Gestión del Rendimiento de la Organización – Un área de proceso de gestión de procesos en el nivel de madurez 5</p> <p>SP 1.1 Mantener los objetivos de negocio</p> <p>SP 1.2 Analizar los datos de rendimiento de proceso</p> <p>SP 1.3 Identificar las áreas potenciales para la mejora</p> <p>SP 3.1 Planificar el despliegue</p> <p>Rendimiento de Procesos de la Organización – Un área de proceso de gestión de procesos en el nivel de madurez 4</p> <p>SP 1.1 Establecer los objetivos de calidad y de rendimiento de proceso</p> <p>Gestión cuantitativa del proyecto-Un área de proceso de gestión de proyectos en el nivel de madurez 4</p> <p>SP1.1 Establecer los objetivos del proyecto</p> <p>SP1.2 Componer el proceso definido</p> <p>Desarrollo de Requisitos – Un área de proceso de</p>	<p>Se sugiere revisar y ejecutar las prácticas identificadas en el Modelo CMMiDEV, para la reducción del tiempo de desarrollo o de producción, cambiar la funcionalidad, añadir nuevas características o adaptarse a las nuevas tecnologías.</p>

Frenos u Obstáculos	CMMI Dev [31]	Recomendación
	ingeniería en el nivel de madurez 3 SP 2.1 Establecer los requisitos de producto y de componente de producto.	
Requerimientos	Gestión de Requisitos- Un área de proceso de gestión de proyectos en el nivel de madurez 2 (5 SP) Desarrollo de Requisitos- Un área de proceso de ingeniería en el nivel de madurez 3 (10 SP)	Se sugiere revisar y ejecutar las prácticas identificadas en las áreas de proceso del Modelo CMMiDEV, para identificar las necesidades del cliente y transformarlas en requisitos del producto.
Personal	Formación en la Organización – Un área de proceso de gestión de procesos en el nivel de madurez 3 (7 SP)	Se sugiere revisar y ejecutar las prácticas identificadas en el área de proceso del Modelo CMMiDEV, para desarrollar las habilidades y los conocimientos de las personas, así podrán desempeñar sus roles eficaz y eficientemente.
Presupuesto	Planificación del Proyecto – Un área de proceso de Gestión de Proyectos en el nivel de madurez 2 SP 2.1 Establecer el presupuesto y el calendario	Se sugiere revisar y ejecutar la práctica identificada en el Modelo CMMiDEV, para asegurar que adecuadamente se realicen las asignaciones de presupuesto en los temas de infraestructura, personal, materiales, etc., según el calendario establecido.
Usuarios, Funcionales	Gestión integrada del Proyecto-Un área de proceso de Gestión de Proyectos en el nivel de madurez 3 SP 2.1 Gestionar la involucración de las partes interesadas Monitorización y control de proyecto-Un área de proceso de Gestión de Proyectos en el nivel de madurez 2 SP1.5 Monitorizar la involucración de las partes interesadas	Se sugiere revisar y ejecutar las prácticas identificadas en el Modelo CMMiDEV, para la gestión y monitorización del involucramiento de los stakeholders o partes interesadas, lo que ayudará a mejorar la gestión integrada de los proyectos.
Procesos, procedimientos	Gestión integrada del Proyecto-Un área de proceso de Gestión de Proyectos en el nivel de madurez 3	Se sugiere revisar y ejecutar las prácticas identificadas en el Modelo CMMiDEV, para la

Frenos u Obstáculos	CMMI Dev [31]	Recomendación
	<p>SP1.1 Establecer el proceso definido del proyecto</p> <p>Definición de Procesos de la Organización – un área de procesos de gestión de procesos en el nivel de madurez 3.</p> <p>SP 1.1 Establecer los procesos estándar</p> <p>Enfoque en Procesos de la Organización – Un área de proceso de Gestión de Procesos en el nivel de madurez 3 (9 SP)</p> <p>Rendimiento de Procesos de la Organización- Un área de procesos de Gestión de procesos en el nivel de madurez 4 (5 SP)</p> <p>Aseguramiento de la calidad del proceso y del producto- Un área de proceso de Soporte en el nivel de madurez 2</p> <p>SP 1.1 Evaluar objetivamente los procesos</p>	<p>gestión, enfoque, rendimiento y aseguramiento de la calidad en los procesos y procedimientos.</p>
Legislación	<p>Definición de procesos de la organización – Un área de procesos de Gestión de Procesos en el nivel de madurez 3.</p> <p>SP 1.7 Establecer las reglas y guías para los equipos</p> <p>Gestión cuantitativa del Proyecto-Un área de proceso de Gestión de Proyectos en el nivel de madurez 4.</p> <p>SP 1.2 Componer el proceso definido</p>	<p>Se sugiere revisar y ejecutar las prácticas identificadas en el Modelo CMMiDEV, para cumplir con las normativas o leyes locales cuando se establecen las reglas y guías para los equipos.</p>
Autoridades	<p>Gestión integrada del Proyecto-Un área de proceso de Gestión de Proyectos en el nivel de madurez 3</p> <p>SP 1.6 Establecer los equipos</p> <p>GP 2.10 Revisar el estado con el nivel directivo</p>	<p>Se recomienda involucrar a las autoridades o directivos en los equipos de los proyectos para: apoyo, toma de decisiones, integración e información.</p>
Unidades como entes aislados	<p>Gestión Integrada del Proyecto-Un área de proceso de Gestión de proyectos en el nivel de madurez 3</p> <p>SP 1.4 Integrar los planes</p> <p>Integración del Producto- Un área de proceso de Ingeniería en el nivel de madurez 3</p>	<p>Se sugiere revisar y ejecutar las prácticas identificadas en el Modelo CMMiDEV, para mantener integrada la información, permitiendo que las unidades de las IES visualicen la misma información en sus</p>

Frenos u Obstáculos	CMMI Dev [31]	Recomendación
	<p>SP 2.2 Gestionar las interfaces</p> <p>Aseguramiento de la calidad del proceso y del producto- Un área de proceso de Soporte en el nivel de madurez 2</p> <p>SP 1.2 Evaluar objetivamente los productos de trabajo</p> <p>Solución técnica- Un área de proceso de ingeniería en el nivel de madurez 3 (8 SP)</p> <p>Validación – Un área de proceso de ingeniería en el nivel de madurez 3 (5 SP)</p> <p>Verificación – Un área de proceso de ingeniería en el nivel de madurez 3 (8 SP)</p>	<p>diferentes procesos de trabajo, ya sea en lo académico, talento humano, financiero, investigación, etc.</p>
Versionamiento	<p>Solución Técnica – Un área de proceso de Ingeniería en el nivel de madurez 3</p> <p>Verificación – Un área de proceso de ingeniería en el nivel de madurez 3</p>	<p>Se sugiere revisar y ejecutar las prácticas identificadas en el área de proceso del Modelo CMMiDEV, para que la versión del producto de trabajo en uso en un momento dado, pasado o presente sea conocida, y los cambios sean incorporados de una forma controlada.</p>
Falta de metodologías	<p>Definición de Procesos de la Organización- Un área de proceso de Gestión de Procesos en el nivel de madurez 3</p> <p>SP 1.1 Establecer los procesos estándar</p> <p>Gestión del Rendimiento de la Organización-Un área de proceso de gestión de Procesos en el nivel de madurez 5</p> <p>SP 2.1 Educar las sugerencias de mejora</p>	<p>Se sugiere aplicar las prácticas del modelo CMMiDEV, lo que ayudará a las IES a mejorar la forma de hacer su negocio.</p>
Falta de documentación formal escrita	<p>Verificación – Un área de proceso de ingeniería en el nivel de madurez 3</p> <p>Subpráctica</p> <p>Definir y documentar el plan para realizar el proceso</p> <p>Solución Técnica- Un área de proceso de</p>	<p>Se sugiere aplicar las prácticas del modelo CMMiDEV, lo que permitirá la gestión documental de un proceso, proyecto, etc, dependiendo de su criticidad.</p>

Frenos u Obstáculos	CMMI Dev [31]	Recomendación
	ingeniería en el nivel de madurez 3 SP 3.2 Desarrollar la documentación de soporte del producto	
-	Gestión de riesgos-un área de proceso de Gestión de Proyectos en el nivel de madurez 3	Se sugiere tratar el tema de gestión de riesgos en las IES, para con antelación identificar problemas potenciales, realizar una planificación de riesgos para su mitigación.

5 CONCLUSIONES Y RECOMENDACIONES

En base a los resultados y discusiones presentados en este estudio, se plantean las siguientes conclusiones y recomendaciones.

5.1 CONCLUSIONES

- La revisión sistemática de literatura permitió identificar los principales estándares-norma y modelos con relación al tema de calidad en los procesos de desarrollo de software, resaltando a ISO/IEC TS 33073, ISO/IEC/IEEE 9003, Bootstrap, PSP, TSP y CMMi, de los cuales no se encontró indicio alguno de ser implementado en alguna universidad del Ecuador, por lo que se justifica la realización de este estudio.
- Diez de las 11 universidades o escuelas politécnicas del Ecuador, invitadas a la entrevista, proporcionaron respuesta de participación favorable, haciendo notorio desde el primer contacto su predisposición y apertura, signos de colaboración para realizar estudios, desarrollar y presentar soluciones conjuntas en beneficio del ámbito académico superior, pero por circunstancias de índole laboral, a la final sólo nueve universidades participaron del estudio.
- En relación con las definiciones expresadas por los participantes, cuatro de nueve IES dieron una definición de calidad de software en relación con lo establecido en la teoría, cuatro IES indicaron una definición de calidad a manera general según se presentó en la teoría y una IES no mencionó definición alguna.
- Las nueve universidades, en cuanto a expectativas mencionaron sus deseos razonables de mejorar o contribuir en temas relacionados con las categorías identificadas de: universidad, producto/software, desarrollador/área, usuario e información, mientras que sus principales motivadores (proactivos y reactivos) están enfocados en el desarrollo de soluciones tecnológicas que realmente aporten al trabajo diario del cliente interno y externo.
- El tema de frenos u obstáculos de calidad de software tuvo una mayor contribución por parte de ocho de las nueve universidades participantes (59 verbatines), identificando 13 categorías en total, de las cuales se presentaron en

su mayoría falencias internas con 12 categorías (falta de tiempo, requerimientos, personal, presupuesto, usuario/funcionales, autoridades, versionamiento, unidades como entes aislados, procesos/procedimientos, la falta de metodologías, la falta de documentación formal escrita e incompatibilidad con aplicaciones existentes) y en relación a falencia interna/externa con la categoría de legislación. El participante de la universidad faltante no mencionó freno alguno en la conversación.

- Con relación a modelos de calidad de procesos de software, sólo una IES mencionó un modelo (CMMi), seis IES mencionaron modelos, estándares o términos que tienen relación con la calidad de software, refiriéndose al ámbito de ciclo de vida de desarrollo de software, gestión de TI y gestión de seguridad, mientras que dos IES no mencionaron ningún término relacionado con este tema.
- Se identificaron nueve categorías de prácticas en el tema de calidad de software aporte de las nueve IES, las cuales son prácticas desarrolladas o se encuentran en desarrollo. Siendo las Evaluaciones de calidad (ocho IES) y Formación profesional – Capacitación (siete IES) las prácticas punteras más utilizada en las universidades y que junto con las cinco categorías siguientes (invertir en tecnología, proyecto, procesos, determinar normativas, información e involucramiento de los stakeholders) son las prácticas a ser consideradas como punto de partida en el tema de calidad general y/o software. Como prácticas futuras se consideran: la categoría Tecnología y continuar trabajando con el tema de Procesos.
- Las prácticas de calidad de software implementadas o por implementar, son las acciones que permitirán contrarrestar: los frenos/obstáculos y los motivadores reactivos identificados por las universidades, a su vez, dichas prácticas, serán las acciones que permitirán cumplir y cubrir las expectativas de calidad de software aquí identificadas.

5.2 RECOMENDACIONES

- Cuando no se encuentra información respecto al estado actual de un tema en particular, se sugiere aplicar el tipo de investigación exploratorio cualitativo para conocer en primera instancia y de manera general las opiniones y comentarios de sus actores. En este caso, para el estudio de las prácticas de calidad en los procesos de desarrollo de software en universidades del Ecuador, los actores principales fueron los directores o jefes de área de desarrollo de las IES y que, con sus respuestas, ayudaron al investigador a identificar y detallar lo concerniente a expectativas y motivadores, frenos u obstáculos, modelos y prácticas en el contexto de la calidad de software.
- Los resultados presentados en este estudio fue la contribución de nueve de las once universidades invitadas inicialmente a ser parte de este estudio, sin embargo, se recomienda para futuros trabajos relacionados con el tema “Calidad en los Procesos de Desarrollo de Software”, se realice una invitación abierta a todas las universidades y escuelas politécnicas del Ecuador, para presentar la realidad en la cual se encuentra todo el sistema de educación superior del país.
- Antes de finalizar las entrevistas, a los participantes se les consultó sobre tres variables sociodemográficas (nivel de estudios, años de experiencia en el cargo y si tienen formación relacionada con la calidad de software), lo cual está vez no fue incluido en el estudio, por presentar y dar más énfasis al tema de las prácticas de calidad en los procesos de desarrollo de software en las universidades del Ecuador, por lo que posiblemente esta información podría ser relevante para futuros estudios relacionados a este tema.

Recomendaciones generales para la mejora de calidad en los procesos de desarrollo de software en universidades del Ecuador

- Para aquellas universidades que recién se encuentran incursionando en implementar sistemas de información como apoyo al cumplimiento del aseguramiento de la calidad que exige el CACES para la acreditación de ser parte del Sistema de Educación Superior, se sugiere que revisen y observen las practicas que han sido implementadas o se encuentran implementando las universidades que participaron de este estudio, partiendo desde el punto de vista de una evaluación inicial de la situación actual para posteriormente elaborar un

plan de acción de cambio o mejoramiento en lo que a calidad de software o calidad a manera general se refiere.

- Se sugiere conformar grupos de trabajo e investigación con los directores de TIC's de las universidades y escuelas politécnicas, los entes reguladores y demás partes interesadas, donde puedan abordar el tema de calidad de proceso de desarrollo de software, para compartir experiencias, inquietudes y por qué no, fomentar el desarrollo conjunto de sistemas de información core universitarios como apoyo al aseguramiento de calidad que exigen los entes reguladores del país.
- Se sugiere que las IES inviertan en capacitación de calidad para el personal técnico del área de Desarrollo, lo que permitirá tener un conocimiento previo para: proponer nuevos proyectos, plantear propuestas de mejora ante las falencias de calidad de productos de software identificadas en la IES, e incluso, puede considerarse como incentivo de crecimiento y fortalecimiento profesional para los miembros del área técnica.
- Se sugiere el desarrollo de un portafolio interno de proyectos orientados a la mejora de la calidad en cada universidad, lo que permitirá una mejor gestión de la organización de proyectos anteriores, en ejecución y futuros (control de versiones), así como la asignación de roles y controles de acceso para todos los involucrados.
- Se recomienda que las universidades seleccionen o desarrollen un framework de calidad de software para su implementación, lo que permitirá establecer un entorno de trabajo normalizado como base para organización y desarrollo de software.
- Se recomienda asegurar el apoyo e involucramiento de todos los stakeholders y/o áreas de las IES en los proyectos, ante todo por conocimiento, participación en toma de decisiones y para que juntos identifiquen todas las afectaciones posibles ante un cambio, lo que permitirá o eliminará el obstáculo de oposición de los usuarios y/o clientes finales.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] CACES, “Evaluación Institucional,” 2018. [Online]. Available: <https://www.caces.gob.ec/web/ceaaces/institucional>. [Accessed: 28-Aug-2018].
- [2] ASAMBLEA NACIONAL, “Ley Orgánica de Educación Superior,” *Regist. Of. órgano del Gob. del Ecuador*, vol. 298, p. 40, 2010.
- [3] CACES, “Presentación-Política de Evaluación Institucional de Universidades y Escuelas Politécnicas en el marco del Sistema de Aseguramiento de la Calidad de la Educación Superior,” 2018.
- [4] CACES, *Política de Evaluación Institucional de Universidades y Escuelas Politécnicas en el marco del Sistema de Aseguramiento de la Calidad de la Educación Superior*. Ecuador, 2018, pp. 1–52.
- [5] CEAACES, “Modelo de Evaluación Institucional de Universidades y Escuelas Politécnicas,” p. 58, 2015.
- [6] R. Reix, B. Fallery, K. Michel, and F. Rowe, *Systèmes d’information et management*, 7e éditio. Paris: Print book - French, 2016.
- [7] C. Costa, C. Cunha, R. Rocha, a. C. C. França, F. Q. B. da Silva, and R. Prikladnicki, “Models and Tools for Managing Distributed Software Development : A Systematic Literature Review,” *Int. Conf. Eval. Assess. Softw. Eng. (EASE 2010)*, pp. 1–4, 2010.
- [8] J. Biolchini, P. Mian, A. C. Natali, and G. H. Travassos, “Systematic Review in Software Engineering,” Río de Janeiro, 2005.
- [9] B. Kitchenham, “Procedures for Performing Systematic Literature Reviews,” *Keele Univ. Durham Univ. UK*, 2004.
- [10] M. en Software, “Revisión sistemática de literatura,” 2017.
- [11] R. Pressman, *Ingeniería del software*, Séptima. 2010.
- [12] ISO 9000, “sistema de gestión de la calidad - Fundamentos y vocabulario,” *Norm. ISO*, vol. 2005, p. 42, 2005.
- [13] IEEE, “IEEE_SoftwareEngGlossary.pdf.” p. 84, 1990.
- [14] W. S. Humphrey, *Introduction to the Personal Software Process SM*. 1997.
- [15] ISO 9126, “Information technology — Software product quality,” *Iso/lec Fdis 9126-1*, vol. 2000, pp. 1–26, 2000.
- [16] Alejandro, “La diferencia entre Norma – Metodología y Estándar,” 2013. [Online]. Available: <https://projectandprocessmanager.wordpress.com/2013/01/10/la-diferencia-entre-norma-metodologia-y-estandar/>. [Accessed: 06-Mar-2019].
- [17] B. Verbrugge, “Best Practice, Model, Framework, Method, Guidance, Standard:

- towards a consistent use of terminology,” *Van Haren Publishing*, 2016. [Online]. Available: <https://www.vanharen.net/blog/van-haren-publishing/best-practice-model-framework-method-guidance-standard-towards-consistent-use-terminology/>. [Accessed: 31-Aug-2018].
- [18] S. Komi-Sirviö, *Development and Evaluation of Software Process Improvement*. Finland: Technical Research Centre of Finland, 2004.
- [19] B. Curtis, B. Hefley, and S. Miller, “People Capability Maturity Model (P-CMM) Version 2.0,” *Carnegie Mellon Softw. Eng. Inst.*, no. July, pp. 1–601, 2009.
- [20] crue, *UNIVERSITIC 2017- Análisis de las TIC en las Universidades Españolas*, Doce. Madrid-España, 2017.
- [21] D. F. Córdova Báez, “ANÁLISIS COMPARATIVO DE LOS MODELOS Y ESTÁNDARES DE CALIDAD DE SOFTWARE Y APLICACIÓN DE LAS MEJORES PRACTICAS PARA EL LEVANTAMIENTO DEL PROCESO DE GESTIÓN DE CALIDAD DE PRODUCTOS DE SOFTWARE,” UNIVERSIDAD CENTRAL DEL ECUADOR FACULTAD DE INGENIERIA CIENCIAS FÍSICAS Y MATEMÁTICA, 2015.
- [22] I. E. C. Ts, “TECHNICAL SPECIFICATION ISO / IEC TS Information technology — Process assessment — Process capability assessment model for information,” vol. 2016, 2016.
- [23] ISO, “ISO 9001:2015, Quality management systems-Requirements,” 2015.
- [24] Software and Systems Engineering Standards Committee, “ISO/IEC/IEEE 90003:2018, Software engineering-Guidelines for the application of ISO 9001:2015 to computer software,” 2018.
- [25] F. SCALONE, “ESTUDIO COMPARATIVO DE LOS MODELOS Y ESTANDARES DE CALIDAD DEL SOFTWARE,” 2006.
- [26] CORDIS, “Bootstrap assessments improve software quality,” 1994. [Online]. Available: <https://cordis.europa.eu/project/rcn/8768/reporting/en>. [Accessed: 08-Mar-2019].
- [27] A. Herrera, “BootsTrap,” 2012. [Online]. Available: <http://procesossoftwareai.blogspot.com/2012/10/bootstrap.html>.
- [28] M. Pomeroy-Huff and T. A. Chick, “The Personal Software Process SM (PSP SM) Body of Knowledge , Version 2 . 0,” *Area*, vol. 2009, no. August 2009, p. 92, 2010.
- [29] J. Urrego, “Aseguramiento de calidad en el desarrollo de software,” *Rev. Technol.*, vol. 2, no. 7, pp. 68–71, 2007.
- [30] W. S. Humphrey, T. A. Chick, and M. Pomeroy-huff, “The Team Software Process SM (TSP SM) Body of Knowledge (BOK) Draft,” no. July, 2010.

- [31] M. Jain, *Delivering Successful Projects with TSP SM and Six Sigma A Practical Guide to Implementing Team Software Process SM*. 2009.
- [32] SEI, "CMMI ® para Desarrollo, Versión 1.3," *C. Para Desarrollo, Version 1.3*, p. 23, 2010.
- [33] ANUIES, *Estado actual de las Tecnologías de la Información y las Comunicaciones en las Instituciones de Educación Superior en México*, vol. 3. 2018.
- [34] R. Padilla, S. Cadena, R. Enríquez, J. Córdova, and F. Llorens, "UETIC_2017," 2017.
- [35] N. Lesca, M.-L. Caron-Fasan, E. Loza Aguirre, and M.-C. Chalus-Sauvannet, "Drivers and barriers to pre-adoption of strategic scanning information systems in the context of sustainable supply chain," *Systèmes d'information Manag.*, vol. 20, no. 3, p. 9, 2016.
- [36] J. A. Taguenca and M. del R. Vega, "Técnicas de investigación social Las entrevistas abierta y semidirectiva," *Rev. Investig. en Ciencias Soc. y Humanidades, Nueva Época*, vol. 23, no. 7, pp. 789–792, 2012.
- [37] CACES, "CATEGORÍA VIGENTE DE UNIVERSIDADES Y ESCUELAS POLITÉCNICAS(1)," 2018.
- [38] DEYA-CGI-SENESCYT, "Estudiantes_UES_Indice_de_Tabulados_Diciembre_2018," 2018. [Online]. Available: <https://www.educacionsuperior.gob.ec/cuadros-estadisticos-indice-de-tabulados-sobre-los-datos-historicos-de-educacion-superior-a-nivel-nacional-incluye-registro-de-titulos-oferta-academica-matriculados-docentes-becas-y-cupos/>. [Accessed: 07-May-2019].
- [39] DEYA-CGI-SENESCYT, "Docentes_UES_Indice_de_Tabulados_Diciembre_2018," 2018. [Online]. Available: <https://www.educacionsuperior.gob.ec/cuadros-estadisticos-indice-de-tabulados-sobre-los-datos-historicos-de-educacion-superior-a-nivel-nacional-incluye-registro-de-titulos-oferta-academica-matriculados-docentes-becas-y-cupos/>. [Accessed: 07-May-2019].
- [40] Scientific Software Development GmbH, "ATLAS.ti," 2002. [Online]. Available: <https://atlasti.com/>. [Accessed: 29-Mar-2019].
- [41] M. Segura, "1 . Software Management," 2017.
- [42] Norma 410, "Normas De Control Interno De La Contraloria General Del Estado," *Ultima*, pp. 16–2014, 2009.
- [43] CMMI Institute, "CMMI Institute - Company," 2018. [Online]. Available: <https://cmmiinstitute.com/company>. [Accessed: 09-Sep-2018].
- [44] A. Fuggetta, C. Vía, M. Italia, E. Di Nitto, and M. Piazza, "Proceso de Software,"

2000.

- [45] T. P. I. P. Limited, “JSF Tutorial,” 2019. [Online]. Available: <https://www.tutorialspoint.com/jsf>. [Accessed: 07-May-2019].
- [46] B. Infotech, “ALM with TFS,” 2019. [Online]. Available: <https://www.brainvire.com/alm-with-tfs/>. [Accessed: 26-Feb-2019].

7 ANEXOS

Los anexos de este trabajo se encuentran en el CD digital y se los detalla a continuación:

Anexo I - Información para el participante y su consentimiento

Anexo II - Anonimizado de la Transcripción

Anexo III - Matriz de Codificación