

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
SEDE ESMERALDAS



ESCUELA DE INGENIERÍA EN SISTEMAS Y
COMPUTACIÓN

**Análisis de los procesos de desarrollo de software en el
departamento de TICs de la Pontificia Universidad
Católica del Ecuador Sede Esmeraldas**

**Previo a la obtención del título de
Ingeniero de Sistemas y Computación**

Autor:

Mera Quiroz Junior Williams

Asesor:

Quiñonez Ku Victor Xavier

Noviembre del 2015

Índice de Contenidos

1. Resumen.....	ii
2. Abstract.....	iii
3. Justificación	4
4. Objetivos.....	5
4.1. Objetivo General	5
4.2. Objetivos Específicos.....	5
5. Caso	6
5.1. Antecedentes	6
5.2. Marco teórico	8
5.3. Metodología	13
5.4. Población y Muestra.....	14
5.4.1. Población	14
5.4.2. Muestra	15
5.4.3. Técnica muestral aplicada.....	17
5.5. Análisis e interpretación de los datos.....	18
6. Propuesta de intervención.....	26
7. Referencias Bibliográficas	36
8. Anexos	39

1. Resumen

El estudio de caso realizado analizó los procesos de desarrollo de software que se llevan a cabo en el departamento de TICs de la PUCESE, con la finalidad de identificar una metodología de desarrollo de software, que permitiese optimizar de manera adecuada la creación de sistemas informáticos en el mencionado departamento.

Para el correcto desarrollo de la investigación se utilizó el método analítico-sintético y posteriormente un estudio holístico del problema. Como técnica de recolección de datos se utilizaron encuestas para determinar el nivel de satisfacción que perciben los usuarios finales sobre los sistemas informáticos desarrollados por el Departamento de TICs, además de entrevistas a los miembros del equipo de desarrollo del departamento antes mencionado con la finalidad de describir los procesos que ellos utilizan para llevar a cabo el desarrollo de dichos sistemas informáticos.

Posteriormente se compararon las diferentes metodologías de desarrollo de software en base a los problemas evidenciados en el análisis de los resultados de la investigación. Como resultado de la comparación de las metodologías, SCRUM fue considerada como la metodología más adecuada para el departamento de TICs de la PUCESE.

Y para finalizar, se realizó una propuesta de intervención, en la cual se indica cada uno de los parámetros que necesita cumplir el departamento de TICs de la PUCESE, para implementar SCRUM como metodología de desarrollo de software.

2. Abstract

In this case study, the software development process of the PUCESE's ITC department was analyzed. The goal was to identify an adequate software development methodology for the department in order to optimize the way software is elaborated at the moment.

The investigation followed an analytic-synthetic method and subsequently a holistic study of the problem. Surveys were used, as a data collection technique, to determine the level of satisfaction by the end user of the software applications that were developed in the ITC department of the University. Additionally the staff was interviewed to determine the actual software developing process.

Subsequently different software development methodologies were compared, based on the problems highlighted in the results of the research. As a result of the compared methodologies, SCRUM was considered the most suitable for the ITC Department of the University.

This study concludes with a proposal that indicates the parameters required to implement the SCRUM software development methodology in the ICT department of PUCESE.

3. Justificación

Debido a la necesidad de sistematizar procesos en la Pontificia Universidad Católica del Ecuador Sede Esmeraldas (PUCESE), se requiere que en el Departamento de TICs se apliquen procesos guiados a través de una metodología, de tal forma que faciliten el desarrollo de software, y con ello cumplir con los requerimientos de la institución, obteniendo así mejores productos de software.

Al implementar una metodología de desarrollo de software, se optimiza cada una de las fases del mismo, desde la etapa de requerimientos hasta el mantenimiento de los sistemas, mejorando los tiempos de desarrollo y utilización de recursos que provee la institución.

Vale destacar, que al contar con un desarrollo de software eficiente y organizado, mediante una metodología, permitirá generar sistemas muy productivos, lo cual evitaría que la PUCESE tenga que adquirir dichos sistemas a proveedores externos, lo que conlleva a un ahorro importante en términos de costos.

A continuación se enuncian cada uno de los beneficiarios que tendría la implantación de una metodología de desarrollo de software:

- Primero, los programadores debido a que contarán con lineamientos que les guiarán de una forma clara, a lo largo de cada una de las fases del desarrollo de software.
- Segundo, el jefe del departamento, ya que a través de una metodología podrá llevar un mayor control sobre el avance de los proyectos de software.
- Y por último, los usuarios finales debido a que estos tendrán a su disposición software de gran calidad, resultado de aplicar correctamente una metodología de desarrollo, lo cual les permitirá realizar su trabajo de una manera rápida y eficaz.

4. Objetivos

4.1. Objetivo General

Determinar una metodología de desarrollo de software que establezca los procesos para optimizar la creación de sistemas informáticos aplicando un diagnóstico prospectivo del departamento de TICs de la PUCESE.

4.2. Objetivos Específicos

- Describir los procesos de desarrollo que se aplican en el departamento de TICs de la PUCESE.
- Analizar los criterios de los usuarios finales de los sistemas desarrollados por el departamento de TICs de la PUCESE.
- Comparar las diferentes metodologías ágiles aplicadas en el desarrollo de software para establecer la más adecuada en relación a los procesos de desarrollo generados en el departamento de TICs de la PUCESE.
- Elaborar una propuesta que permita aplicar la metodología seleccionada para optimizar el desarrollo de software en el departamento de TICs de la PUCESE.

5. Caso

5.1. Antecedentes

El desarrollo de software en una institución educativa de nivel superior debe cumplir un rol fundamental dentro de la misma, debido en gran medida a que este tipo de instituciones deben contar con una amplia variedad de sistemas para manejar cada uno de los diferentes procesos de la mejor manera posible y así poder optimizar al máximo los mismos.

Uno de los principales problemas que se presentan al momento de desarrollar cualquier tipo de sistema informático, es la escasa atención que se le presta a criterios tan importantes como el uso de buenas prácticas y parámetros en cada una de las fases de desarrollo del software.

En estudios anteriores (Egas Luis, 2014), se enuncia que debido a la falta de aplicación de herramientas, normas, estándares y metodologías disponibles que proporcionen un soporte tecnológico, conceptual y humano en el Departamento TICs, conlleva a la adquisición de sistemas a otras empresas por la deficiente calidad de software diseñado y tiempos tardíos de entrega.

Otro problema común ocasionado por la falta de una metodología de desarrollo de software, es la poca valoración que se le asigna a la fase de levantamiento de requisitos, lo cual conlleva a obtener como resultado un sistema que no cumple con las expectativas inicialmente planteadas por el cliente, y con una grave deficiencia en la calidad.

Los problemas antes mencionados generan como resultado inconvenientes en el desarrollo normal de un producto de software, así como también una enorme sobrecarga de trabajo y escasa comunicación entre los diferentes involucrados.

De acuerdo con Canós, Letelier y Penadés (s.f), las metodologías ágiles para el desarrollo de software han tenido gran efectividad en proyectos con requisitos muy cambiantes y especialmente cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Y por tanto el uso de este tipo de metodologías resulta beneficioso para instituciones que tengan las necesidades antes mencionadas y que además requieran de la implementación de una metodología de desarrollo de software.

Canós et al. (s.f) indica que toda metodología que se desee implantar en un ambiente de desarrollo de software primeramente debe de ser adaptada de la mejor manera posible al contexto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc.) en el que se encuentra el proyecto que se desea realizar.

5.2. Marco teórico

Es común encontrar personas cuyo criterio acerca del desarrollo de software está ligado únicamente a los programas de computadora, lo cual es bastante erróneo, ya que el desarrollo de software también está relacionado con todos aquellos documentos que recaban información acerca de los requerimientos, el diseño y demás procesos vinculados.

El desarrollo de software debe seguir metodologías que permitan conseguir en la mayor cantidad de aspectos posibles la más alta calidad y satisfacción del cliente. La disciplina que se encarga de determinar, analizar, describir y establecer las diferentes metodologías y lineamientos para mejorar u optimizar el desarrollo de sistemas informáticos se la denomina como Ingeniería de Software.

De acuerdo con Cendejas (2014), la Ingeniería de software busca convertir el desarrollo de software en un proceso formal, con resultados predecibles, que permitan obtener un producto final de alta calidad y que satisfaga las necesidades y expectativas del cliente. En resumen, la ingeniería de software es aquella disciplina que brinda un conjunto de metodologías, métodos y demás lineamientos que permiten agilizar el desarrollo de software, con el objetivo de obtener sistemas robustos de alta calidad que cumplan con los requerimientos del cliente.

La ingeniería de software según Méndez (2013), requiere llevar a cabo numerosas tareas agrupadas en etapas, al conjunto de estas etapas se le denomina ciclo de vida. Existen una amplia variedad de modelos que especifican las diferentes etapas por las cuales debe pasar un software, pero las etapas más comunes a casi todos los modelos de ciclo de vida son las siguientes:

- La primera etapa es la denominada Análisis de Requerimientos, en esta se especifica por parte del cliente cada uno de los requisitos que se requieren plasmar en el software a desarrollar. Se puede decir que el cliente sabe lo que quiere, pero en realidad no es así, se requiere de real capacidad y experiencia para lograr identificar con claridad los requisitos.

- La segunda etapa es el Diseño y Arquitectura del Software, es aquí donde se diseña la estructura de todos los procesos que realizará el software, los resultados que devolverá y cómo se van a realizar, así como también las herramientas a utilizar.
- La tercera etapa se denomina Programación, esta etapa consiste en programar los diferentes procesos de la manera en que fueron especificados en la etapa de Diseño.
- La cuarta etapa se llama Prueba, es en esta donde se comprueba que todos los procesos desarrollados devuelvan los resultados correctos, tal y como se lo especificó en la etapa de Análisis de los Requerimientos.
- La última etapa es la denominada como Mantenimiento, esta consiste en todo aquel proceso que permite mejorar el software una vez desarrollado, así como también corregir errores descubiertos y adaptar el software ante cambios en el hardware.

De acuerdo con Instituto Nacional de Tecnologías de la Comunicación (2009), una metodología de desarrollo de software se define como un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Estas técnicas y métodos permiten planificar el desarrollo de software de forma adecuada en cada una de sus diferentes etapas, lo que ayuda a llevar un mayor control a lo largo del desarrollo, lo que a su vez se refleja en la calidad de los sistemas resultantes.

Estudios anteriores (Egas Luis, 2014), revelaron que la falta de aplicación de herramientas, normas, estándares y metodologías disponibles que proporcionen un soporte tecnológico, conceptual y humano en el Departamento TIC, ocasiona la necesidad de adquirir sistemas a otras empresas debido a la deficiente calidad del software diseñado y tiempos tardíos de entrega. Es por esto que las metodologías de desarrollo son necesarias en los departamentos que se dediquen al desarrollo de sistemas informáticos, ya que están basadas en una combinación de todos los modelos genéricos del software, tales como el cascada, espiral, evolutivo, incremental entre otros. Adicionalmente se puede enunciar que una metodología de desarrollo define las herramientas a utilizar, los diferentes roles y actividades a realizar, así como también

una serie de técnicas y buenas prácticas, y en algunos casos; guías que permitan adaptar la metodología a un determinado proyecto.

Un proceso es considerado como ágil cuando el desarrollo de software es incremental, cooperativo, sencillo y adaptable (ProcesosdeSoftware, s.f.). Incremental porque se realiza a través de pequeños módulos de software en ciclos rápidos de desarrollo, cooperativo porque tanto el cliente como el desarrollador trabajan juntos en una estrecha comunicación, sencillo debido a que este método es fácil de aprender y adaptable porque los cambios se pueden realizar incluso estando en la última etapa del desarrollo.

El enfoque de las metodologías ágiles está teniendo una amplia efectividad en proyectos donde los requisitos son muy cambiantes, ya que en este tipo de proyectos la comunicación con el cliente debe ser fundamental y precisamente ese es uno de los principios básicos de las metodologías ágiles. En el Departamento de TICs de la PUCESE se reciben solicitudes de sistemas informáticos cuyos requisitos sufren cambios a lo largo del proceso de desarrollo, lo cual conlleva a extender los tiempos de entrega de dichas solicitudes.

En marzo de 2001, 17 críticos de los modelos de producción basados en procesos resumieron en cuatro postulados lo que hoy en día se conoce como el Manifiesto Ágil (Scrummanager, 2014). Los cuatro postulados mencionados en aquella reunión en 2001 mencionaban los siguientes criterios: Individuos e interacciones sobre procesos y herramientas, Software que funciona sobre documentación exhaustiva, Colaboración de clientes sobre la negociación del contrato y Respuestas a cambios sobre seguir un plan. Estos cuatro postulados son considerados como la definición canónica del desarrollo ágil.

En la actualidad existen un sin número de metodologías ágiles, unas más populares que otras, cada una aportando al desarrollo ágil distintos métodos que ayudan a mejorar de una manera eficaz la calidad del software. Entre las metodologías ágiles más populares se encuentran las siguientes: Extreme Programing, Scrum, Dynamic Systems Development Method, Crystal Clear, Agile Unified Process, entre otras.

Scrum es actualmente una de las metodologías ágiles más conocidas. De acuerdo con Schwaber & Sutherland (2013) la mencionada metodología emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo.

La metodología ágil Scrum dentro de sus lineamientos define los siguientes roles:

1. **Product Owner.-** Es el encargado de obtener y organizar de manera clara la lista de requerimientos del producto de software en base a las necesidades descritas por el cliente. Es el encargado de generar el Product Backlog.
2. **Scrum Master.-** Es el responsable de asegurar que todos los lineamientos, modelos, prácticas y procedimientos que define Scrum, sean conocidos y bien utilizados por el equipo de trabajo.
3. **Equipo de Desarrollo.-** Son quienes están encargados de desglosar cada una de las historias de usuario que se encuentran en el Product Backlog en tareas, con la única finalidad de generar el Spring Backlog para su posterior desarrollo y entrega.

Además propone los siguientes eventos:

1. **El Sprint.-** Son aquellos bloques de desarrollo que son planificados para desarrollarse en 15 o 30 días, el mismo que una vez terminado genera un incremento del producto final.
2. **Reunión de Planificación de Sprint (Sprint Planning Meeting).-** Es aquí donde se planifican las actividades a desarrollarse durante el Sprint. Este evento es organizado y coordinado por el Scrum Master, pero es el equipo de desarrollo es quien realiza todo el trabajo.
3. **Objetivo del Sprint (Sprint Goal).-** El Objetivo del Sprint es una meta establecida para el Sprint que puede ser alcanzada mediante la implementación del Product Backlog.

4. **Scrum Diario (Daily Scrum).**- Es una reunión diaria que debe demorar alrededor de 15 minutos, en la cual se planifican las actividades a realizar durante las siguientes 24 horas. Este evento es organizado y controlado por el Scrum Master.
5. **Revisión de Sprint (Sprint Review).**- Esta reunión tiene por objetivo mostrar los resultados alcanzados por el Sprint, a esta reunión asisten los integrantes del Equipo Scrum y los interesados. Este evento es organizado y controlado por el Scrum Master.
6. **Retrospectiva de Sprint (Sprint Retrospective).**- Es una reunión que permite al Equipo Scrum analizarse a sí mismo. Este evento es organizado y controlado por el Scrum Master.

Y utiliza las siguientes herramientas:

1. **Lista de Producto (Product Backlog).**- Es una lista ordenada en la cual se encuentran los requerimientos del producto que necesita el cliente.
2. **Lista de Pendientes del Sprint (Sprint Backlog).**- Es la lista de tareas que se deben resolver para la culminación del Sprint.

5.3. Metodología

En esta investigación se utilizó el método analítico-sintético, ya que a través de este, se logró descomponer el objeto de estudio en diferentes partes, para así poder estudiarlas de una manera individual y en consecuencia poder obtener una descripción del problema más clara y precisa. Para luego finalizar con un estudio holístico del problema, es decir, tomando cada una de las partes antes mencionadas como un todo integral (Raúl, 2009).

La presente investigación fue desarrollada utilizando diferentes fuentes de información tales como sitios web, libros digitales, tesis de grado, artículos científicos y demás fuentes electrónicas, las mismas que fueron utilizadas como base para redactar el marco teórico y los antecedentes, y que además fueron referenciadas haciendo uso de las normas APA 6ta Edición.

Para investigar las metodologías de desarrollo que se están aplicando actualmente en el Departamento de TICs de la PUCESE se procedió a realizar entrevistas a cada uno de los miembros del equipo de desarrollo, y para determinar el nivel de satisfacción se realizaron encuestas tanto a los estudiantes, docentes y administrativos, así como también a los jefes departamentales de la PUCESE.

Los datos generados por las diferentes técnicas de investigación utilizadas, fueron procesados haciendo uso de programas de cálculo numérico tales como Microsoft Excel y Google Forms, con los cuales se generaron las diferentes tablas y gráficos, los mismos que permitieron realizar los análisis respectivos.

Las encuestas realizadas fueron diseñadas en base a un análisis crítico del problema planteado y tomando en cuenta las diferentes características de la población objeto de la encuesta. Por otro lado para el diseño de las entrevistas, se procedió inicialmente a realizar un análisis de la población a entrevistar, para así determinar de forma correcta las diferentes preguntas a realizar y con ello obtener datos relevantes que contribuyan a la investigación del problema. Según los estudios realizados por José Supo (2014), la presente investigación se encuentra en el nivel descriptivo.

5.4. Población y Muestra

5.4.1. Población

La presente investigación se realizó en la Pontificia Universidad Católica del Ecuador Sede Esmeraldas, durante el periodo académico 2015-2016. La población está conformada por los jefes departamentales, estudiantes, administrativos, docentes y el equipo de desarrollo del departamento de TICs.

A continuación se detalla el número de personas que pertenecen a cada una de las unidades de muestreo consideradas en esta investigación:

Descripción	Cantidad
Jefes Departamentales	6
Estudiantes	1442
Administrativos	71
Docentes	154
Equipo de desarrollo	3
Total	1676

Tabla I Unidades de muestreo de la investigación

- **Población N1:** A los integrantes de esta población, quienes son los encargados de solicitar nuevos sistemas informáticos, se les realizó una encuesta para determinar si perciben alguna metodología al momento de realizar un requerimiento de sistemas al Departamento de TICs de la PUCESE.
- **Población N2, N3 y N4:** Esta población está conformada por quienes usan los sistemas informáticos que ofrece el Departamento de TICs de la PUCESE, a

estos se les efectuó una encuesta que permita establecer el grado de satisfacción que perciben al utilizar dichos sistemas.

- **Población N5:** La presente población está integrada por quienes desarrollan los diferentes sistemas informáticos de la PUCESE, a estos se les generó una entrevista para determinar si aplican alguna metodología de desarrollo de software al momento de construir un nuevo sistema.

5.4.2. Muestra

Como se mencionó anteriormente el equipo de desarrollo del departamento de TICs de la PUCESE es una población pequeña, en consecuencia no hubo la necesidad de aplicar una muestra, por tal razón se procedió a realizar una entrevista estructurada acorde a los lineamientos del estudio a cada uno de los miembros integrantes de dicha población.

- Muestra N2 (Estudiantes)

La presente muestra se la obtuvo aplicando la siguiente fórmula:

$$n = \frac{N * d^2 * Z^2}{(N - 1)E^2 + d^2 * Z^2}$$

N =	Población	1442	
d ² =		0.25	
N - 1 =		1441	
E =	Error de muestreo	0.05	5%
Z =	Nivel de confiabilidad	1.96	95%

$$n1 = \frac{1442 * 0,25 * 3,8416}{(1441)0,0025 + 0,25 * 3,8416}$$

$$n1 = \frac{1384,8968}{4,5629}$$

$$n1 = 303,51$$

Valor de muestra n1=303

- Muestra N3 (Administrativo)

La presente muestra se la obtuvo aplicando la siguiente fórmula:

$$n = \frac{N * d^2 * Z^2}{(N - 1)E^2 + d^2 * Z^2}$$

N =	Población	71	
d ² =		0.25	
N - 1 =		70	
E =	Error de muestreo	0.05	5%
Z =	Nivel de confiabilidad	1.96	95%

$$n2 = \frac{71 * 0,25 * 3,8416}{(70)0,0025 + 0,25 * 3,8416}$$

$$n2 = \frac{68,1884}{1,1354}$$

$$n2 = 60,06$$

Valor de muestra n2=60

- Muestra N4 (Docentes)

La presente muestra se la obtuvo aplicando la siguiente fórmula:

$$n = \frac{N * d^2 * Z^2}{(N - 1)E^2 + d^2 * Z^2}$$

N =	Población	155	
d ² =		0.25	
N - 1 =		154	
E =	Error de muestreo	0.05	5%
Z =	Nivel de confiabilidad	1.96	95%

$$n3 = \frac{155 * 0,25 * 3,8416}{(154)0,0025 + 0,25 * 3,8416}$$

$$n3 = \frac{148,862}{1,3454}$$

$$n3 = 110,65$$

Valor de muestra n3=111

Una vez obtenido el marco muestral de cada una de las poblaciones se procedió a realizar en el caso de los estudiantes, docentes y administrativos una pre-encuesta, con la finalidad de validar las preguntas y obtener observaciones acerca de las mismas. Posteriormente se realizaron las respectivas entrevistas al equipo de desarrollo del departamento de TICs, y las encuestas definitivas a los jefes departamentales, docentes, administrativos y estudiantes.

5.4.3. Técnica muestral aplicada

Debido a las características de las poblaciones que intervienen en este estudio, la técnica de muestreo aplicada es la aleatoria simple, ya que de acuerdo con Godoy (2015) esta técnica evidencia un sesgo menor entre la muestra y la población, dando mayor representatividad y fiabilidad en la obtención de los datos y por ende al resultado del análisis de los mismos.

5.5. Análisis e interpretación de los datos

Luego de proceder a realizar las diferentes encuestas y entrevistas a las poblaciones anteriormente descritas, se procedió a efectuar la pertinente tabulación de los resultados para posteriormente continuar con su respectivo análisis.

Vale recalcar que para el análisis de los datos obtenidos a través de las encuestas, se utilizó el software estadístico SPSS, a través del cual se realizó el correcto análisis de los resultados que a continuación se empezaran a detallar. Primeramente se analizaron los resultados obtenidos de las encuestas dirigidas a las poblaciones N2 (estudiantes), N3 (administrativos) y N4 (docentes).

De las encuestas mencionadas en el párrafo anterior, en primer lugar se calculó el alfa de Cronbach, que de acuerdo con la Universitat de València (2010), permite estimar la fiabilidad de un instrumento de medida a través de un conjunto de ítems, a continuación se puede observar en las diferentes tablas (Tabla II, III, IV) el alfa de Cronbach respectivo.

Alfa de Cronbach	N de elementos
,710	5

Tabla II Alfa de Cronbach (Estudiantes)

Alfa de Cronbach	N de elementos
,712	5

Tabla III Alfa de Cronbach (Administrativos)

Alfa de Cronbach	N de elementos
,704	5

Tabla IV Alfa de Cronbach (Docentes)

Como se puede apreciar en las tablas anteriores (Tabla II, Tabla III, Tabla IV), el alfa de Cronbach de cada una de las poblaciones es superior a 0.7, esto indica que el nivel de fiabilidad de la encuesta efectuada es aceptable.

Continuando con el análisis de los resultados, se evidenció claramente el bajo nivel de satisfacción que tienen tanto estudiantes, administrativos y docentes respecto a la funcionalidad que prestan los sistemas informáticos de la PUCESE, donde se puede constatar que en promedio el 55.43% de las poblaciones encuestadas califican el funcionamiento de los sistemas antes mencionados entre regular y pésimo, como se puede apreciar en la Tabla V, VI y VII.

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido Pésimo	65	21,5	21,5	21,5
Regular	129	42,6	42,6	<u>64,0</u>
Bueno	81	26,7	26,7	90,8
Muy Bueno	13	4,3	4,3	95,0
Excelente	15	5,0	5,0	100,0
Total	303	100,0	100,0	

Tabla V ¿Cómo calificaría el funcionamiento de los sistemas informáticos de la PUCESE?
(Estudiantes)

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido Pésimo	4	6,7	6,7	6,7
Regular	26	43,3	43,3	<u>50,0</u>
Bueno	15	25,0	25,0	75,0
Muy bueno	15	25,0	25,0	100,0
Total	60	100,0	100,0	

Tabla VI ¿Cómo calificaría el funcionamiento de los sistemas informáticos de la PUCESE?
(Administrativos)

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido Pésimo	3	2,7	2,7	2,7
Regular	55	49,5	49,5	<u>52,3</u>
Bueno	36	32,4	32,4	84,7
Muy bueno	11	9,9	9,9	94,6
Excelente	6	5,4	5,4	100,0
Total	111	100,0	100,0	

Tabla VII ¿Cómo calificaría el funcionamiento de los sistemas informáticos de la PUCESE?
(Docentes)

Otro punto evidenciado en el resultado de las encuestas, es la alta cantidad de problemas que presentan los sistemas informáticos de la PUCESE, ya que en las distintas poblaciones se puede apreciar que en promedio el 57.63% de los encuestados, afirma haber sufrido en alguna ocasión algún tipo de problema relacionado con el uso de los sistemas informáticos, mientras que por otro lado se puede enunciar que en promedio el 22.57% de los encuestados asevera que la frecuencia con la que perciben los problemas varía entre Siempre y Casi siempre, como lo indica la Tabla VIII, IX y X.

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido <u>Siempre</u>	<u>34</u>	<u>11,2</u>	<u>11,2</u>	<u>11,2</u>
<u>Casi siempre</u>	<u>61</u>	<u>20,1</u>	<u>20,1</u>	<u>31,4</u>
A veces	140	46,2	46,2	77,6
Casi nunca	50	16,5	16,5	94,1
Nunca	18	5,9	5,9	100,0
Total	303	100,0	100,0	

Tabla VIII ¿Cuán a menudo se le han presentado dificultades al utilizar un sistema informático de la PUCESE? (Estudiantes)

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido <u>Siempre</u>	<u>4</u>	<u>6,7</u>	<u>6,7</u>	<u>6,7</u>
<u>Casi siempre</u>	<u>14</u>	<u>23,3</u>	<u>23,3</u>	<u>30,0</u>
A veces	36	60,0	60,0	90,0
Casi nunca	6	10,0	10,0	100,0
Total	60	100,0	100,0	

Tabla IX ¿Cuán a menudo se le han presentado dificultades al utilizar un sistema informático de la PUCESE? (Administrativos)

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido <u>Siempre</u>	<u>6</u>	<u>5,4</u>	<u>5,4</u>	<u>5,4</u>
<u>Casi siempre</u>	<u>1</u>	<u>,9</u>	<u>,9</u>	<u>6,3</u>
A veces	74	66,7	66,7	73,0
Casi nunca	29	26,1	26,1	99,1
Nunca	1	,9	,9	100,0
Total	111	100,0	100,0	

Tabla X ¿Cuán a menudo se le han presentado dificultades al utilizar un sistema informático de la PUCESE? (Docentes)

Otro factor igual de importante, es la escasa documentación (Manuales de usuario) que presentan los sistemas informáticos de la PUCESE, dicha apreciación se puede constatar en los resultados de las encuestas, ya que en promedio los resultados reflejan que el 74% de los encuestados asegura no percibir la existencia de documentación que ayude o contribuya a aprender el funcionamiento de los sistemas informáticos de la PUCESE, como se puede apreciar en la Tabla XI, XII y XII.

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido Si	112	37,0	37,0	37,0
No	191	63,0	63,0	100,0
Total	303	100,0	100,0	

Tabla XI ¿Los sistemas informáticos de la PUCESE que usted ha utilizado cuentan con documentación apropiada para aprender a manejarlos? (Estudiantes)

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido Si	13	21,7	21,7	21,7
No	47	78,3	78,3	100,0
Total	60	100,0	100,0	

Tabla XII ¿Los sistemas informáticos de la PUCESE que usted ha utilizado cuentan con documentación apropiada para aprender a manejarlos? (Administrativos)

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido Si	21	18,9	18,9	18,9
No	90	81,1	81,1	100,0
Total	111	100,0	100,0	

Tabla XIII ¿Los sistemas informáticos de la PUCESE que usted ha utilizado cuentan con documentación apropiada para aprender a manejarlos? (Docentes)

De acuerdo con los resultados de la investigación, evidentemente existen graves problemas en lo que se refiere a funcionalidad y documentación de los sistemas informáticos de la PUCESE, lo que conlleva a que dichos sistemas presenten problemas o dificultades a los usuarios finales (estudiantes, administrativos y docentes). Y es aquí donde nace la siguiente pregunta ¿Dónde surge el problema de la funcionalidad y la documentación?

De acuerdo con las entrevistas realizadas a los integrantes del equipo de desarrollo del departamento de TICs de la PUCESE, el proceso que actualmente se sigue para el desarrollo de un sistema, es el conocido modelo lineal, donde se pueden evidenciar las siguientes etapas: análisis, diseño, programación y pruebas. Según los desarrolladores estas etapas antes mencionadas no se las realiza con la debida planificación y sin tanta profundidad y detalle como lo requiere el desarrollo del sistema.

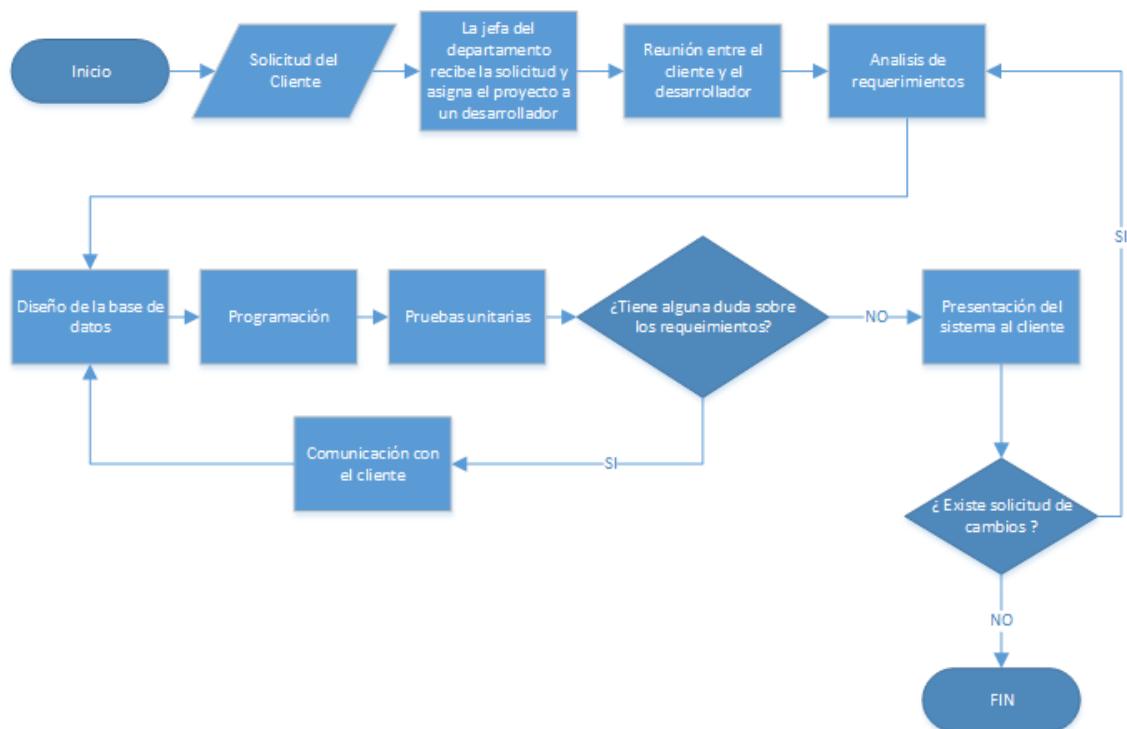


Fig. 1 Diagrama de flujo sobre el actual proceso que se lleva para el desarrollo de un sistema en el Departamento de TICs

En la imagen anterior (Fig. 1), se pueda apreciar el diagrama de flujo del proceso que actualmente se sigue en el Departamento de TICs de la PUCESE para llevar a cabo el desarrollo de un sistema. En este diagrama se puede evidenciar problemas tales como la presentación del sistema al cliente una vez concluido por completo el desarrollo del mismo, la solicitud de cambios al final de todo el proceso y las constantes dudas que surgen sobre los requerimientos a lo largo del desarrollo.

De acuerdo a lo anteriormente expuesto y en base a las entrevistas realizadas al equipo de desarrollo del Departamento de TICs, se puede enunciar que no se evidencia una buena ingeniería de requerimientos en la etapa de análisis del proceso de desarrollo (Fig. 1), lo que deriva en un diseño poco claro que no refleja los requisitos del cliente, por consiguiente la programación de un diseño que no contiene las funcionalidades requeridas, que en conjunto con una fase de pruebas poco eficiente, un proceso carente de documentación, presentación del sistema desarrollado y solicitud de cambios al final de todo el proceso de desarrollo, conlleva a tener los problemas que se vieron reflejados en las encuestas realizadas a los estudiantes, administrativos y docentes.

Otros problemas tales como la falta de control y seguimiento en los proyectos de software, tiempos de entrega tardíos, requisitos cambiantes, comunicaciones poco eficientes con los clientes y entre los miembros del equipo de desarrollo también fueron evidenciados en la entrevista que se le realizó a la persona encargada de la Jefatura del Departamento de TICs.

De acuerdo con lo descrito en el párrafo anterior, el problema raíz se encuentra en el proceso que se sigue para el desarrollo de los sistemas, en pocas palabras, la carencia de una metodología de desarrollo de software. Entonces habiendo llegado a esta conclusión surge una incógnita más ¿Qué tipo de metodología de desarrollo de software utilizar? Para responder esta pregunta es indispensable evaluar los conceptos de las metodologías tradicionales y las ágiles.

Por un lado se tiene a las metodologías tradicionales que son usadas en ambientes donde el problema es claramente conocido y además la solución está bien definida, algo que de acuerdo con las entrevistas no ocurre en la PUCESE, por otro lado se encuentran las metodologías ágiles que son usadas comúnmente en ambientes donde los requisitos son sumamente cambiantes, ya que estas incorporan mecanismos que permiten gestionar el cambio de una manera adecuada y eficiente.

De acuerdo con lo anteriormente expuesto, es necesario concluir que resulta conveniente la implantación de una metodología ágil para el desarrollo de software, la misma que proporcione lineamientos bien estructurados que permitan manejar de manera eficiente la comunicación tanto con el cliente como entre los miembros del

equipo de desarrollo, así como también los requisitos cambiantes, una buena documentación, tiempos de entrega correctos, facilidad para llevar el control sobre el proyecto y una bien planificada ingeniería de requerimientos.

Esta metodología ágil, contribuirá enormemente a obtener como resultado sistemas informáticos con funcionalidades adecuadas acorde a los requisitos del cliente, con una fase de pruebas bien definida que permita detectar problemas de forma rápida y así evitar que a futuro los usuarios finales (estudiantes, administrativos y docentes) sufran problemas con los sistemas informáticos.

6. Propuesta de intervención

Una vez expuesto en el apartado anterior la necesidad de implementar una metodología ágil para el desarrollo de software en el departamento de TICs de la PUCESE, se determinó aplicar un proceso que permita encontrar la metodología ágil más adecuada que cumpla con los requerimientos del departamento antes indicado.

Actualmente existe una gran variedad de metodologías ágiles, cada una de ellas con cualidades y enfoques diferentes, pero siempre manteniendo la agilidad como su principal característica, de acuerdo con investigaciones previas (Oscar Tinoco Gómez, 2010) entre las metodologías ágiles más relevantes se pueden encontrar las siguientes:

- Agile Project Management (APM)
- Dynamic Systems Development Methods (DSDM)
- Scrum
- Test Driven Development (TDD)
- Extreme Programming (XP)

Para determinar la metodología ágil que se acople a cubrir los requerimientos del departamento de TICs de la PUCESE, en primer lugar se deben establecer ciertos criterios de presencia con la única finalidad de reducir la lista de las metodologías anteriormente expuestas a un número de dos, para posteriormente proceder con un análisis comparativo de las dos metodologías ágiles resultantes y así lograr establecer la más adecuada para el departamento de TICs de la PUCESE.

Los criterios de presencia seleccionados para la evaluación de las metodologías ágiles son los siguientes:

- Presencia en internet
- Mejor documentación
- Presencia empresarial
- Utilización en proyectos de software

Una vez expuestos los criterios de presencia, se procedió con la evaluación de las metodologías ágiles tal y como lo describe Oscar Tinoco Gómez (2010) en su investigación, obteniendo como resultado la siguiente tabla:

Metodología / Criterio	Presencia en internet	Mejor documentación	Presencia empresarial	Utilización en proyectos de software	Total
APM	2	1	1	1	5
DSDM	1	3	4	4	12
Scrum	5	2	5	5	17
TDD	3	4	2	2	11
XP	4	5	3	3	15

Tabla XIV Valoración de las diferentes metodologías ágiles disponibles (Oscar Tinoco Gómez, 2010)

Como se puede apreciar en la tabla anterior (Tabla XIV), las metodologías ágiles con mejor puntuación en base a los criterios de presencia anteriormente expuestos son las denominadas SCRUM y XP. Una vez seleccionadas las dos metodologías ágiles, se procedió a establecer un análisis comparativo entre ambas, en base a las necesidades encontradas en el departamento de TICs, las mismas que fueron evidenciadas en la investigación realizada en la PUCESE, las necesidades más relevantes fueron las siguientes:

- Comunicación con el cliente y entre los miembros del equipo de desarrollo
- Adaptabilidad a cambios
- Documentación ágil
- Ciclos de entrega rápidos
- Facilidad para llevar el control sobre el proyecto
- Ingeniería de requerimientos
- Pruebas de software

Para realizar la evaluación se estableció una escala del 1 al 5 para calificar el grado con que cada una de las metodologías satisface las necesidades del departamento de TICs anteriormente expuestas, dicha escala se encuentra especificada en la rúbrica 1 (Anexos).

A continuación, se mostrará en la siguiente tabla el resultado de la evaluación y posteriormente se presentará una lista numerada en relación a cada una de las necesidades evaluadas indicando el ¿Por qué? de cada calificación:

<i>Necesidad / Metodología</i>	<i>SCRUM</i>	<i>XP</i>
1. Comunicación con el cliente y entre los miembros del equipo de desarrollo.	5	5
2. Adaptabilidad a cambios.	5	4
3. Documentación ágil.	5	3
4. Ciclos de entrega rápidos.	4	5
5. Facilidad para llevar el control sobre el proyecto.	5	5
6. Ingeniería de requerimientos.	5	5
7. Pruebas de software.	5	5
TOTAL	34	32

Tabla XV Evaluación de las metodologías SCRUM y XP en base a las necesidades del departamento de TICs de la PUCESE

Scrum

1. Establece una comunicación continua con el cliente, además de reuniones diarias de sincronización del equipo (proyectosagiles.org, s.f.).
2. Gran capacidad de reacción ante los requisitos cambiantes, además incorpora una gestión del cambio e incluso define roles dedicados únicamente a eso (Scrummanager, 2011).

3. Define una serie de documentos ágiles tales como el Product backlog, Sprint backlog y Burn down chart, además de manuales de usuario, de sistema y de mantenimiento.
4. Las iteraciones de entrega se realizan de dos a cuatro semanas.
5. Define roles para llevar el seguimiento y control del proyecto.
6. Se realiza ingeniería de requerimientos al inicio de cada interacción.
7. Las pruebas se hacen efectivas al final de cada iteración.

XP

1. Los programadores se comunican constantemente gracias a la programación por parejas, además la comunicación con el cliente es fluida ya que el cliente forma parte del equipo de desarrollo (Ingeniería de Software, 2012).
2. Para XP los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos, sin embargo no establece roles dedicados a la gestión del cambio (Wikipedia, s.f.).
3. Documentación centrada en el proceso de desarrollo más no en el cliente.
4. Las iteraciones de entrega son de una a tres semanas.
5. Define roles para llevar el seguimiento y control del proyecto.
6. Se realiza ingeniería de requerimientos al inicio de cada interacción (sourceforge, s.f.)
7. Se deben realizar programas de prueba automática y deben ejecutarse con mucha frecuencia. Cuantas más pruebas se hagan, mejor.

De acuerdo con la evaluación realizada, la metodología ágil que se acopla de manera adecuada a las necesidades del departamento de TICs de la PUCESE es la denominada SCRUM.

Una vez determinada la metodología ágil más adecuada para el departamento de TICs de la PUCESE, el siguiente paso es organizar al equipo de trabajo del departamento en base a los lineamientos que define la metodología SCRUM.

De acuerdo con Schwaber & Sutherland (2013), el equipo SCRUM debe estar conformado por tres roles principales: el Dueño del producto (Product Owner), un

Scrum Master y el Equipo de desarrollo. Además del cliente, cuya participación es un factor fundamental para el éxito de la metodología.

En base a las investigaciones realizadas en el departamento de TICs, actualmente el área de desarrollo está conformada por la jefatura del departamento, dos programadores, además de uno o dos estudiantes que por motivo de convenios con la PUCESE se encuentran desarrollando sistemas en el departamento de TICs.

Durante el desarrollo de esta investigación, el departamento de TICs de la PUCESE está sufriendo una serie de cambios tanto estructurales como de personal, es por esto, que vale recalcar que la presente propuesta se realiza tomando como base la estructura y el personal con el que actualmente cuenta el departamento.

La metodología SCRUM (proyectosagiles.org, 2009) recomienda que el equipo de trabajo debe estar conformado entre 5 y 9 personas, ya que por debajo de las 5 personas cualquier imprevisto sobre algún miembro del equipo puede complicar el resultado que se debe entregar al cliente al finalizar la interacción, mientras que por otro lado una cantidad superior a las 9 personas provoca que la comunicación y la colaboración entre los miembros del equipo resulte más complicada.

Otro punto importante que enuncia la metodología SCRUM (proyectosagiles.org, 2009), es la dedicación a tiempo completo que le deben brindar los miembros del equipo a los proyectos de desarrollo. En base a lo anteriormente expuesto y tomando en cuenta el personal con el que actualmente cuenta el departamento, se debería contratar los servicios de un nuevo programador, que en conjunto con los dos programadores, dos estudiantes y la jefatura del departamento formarían un equipo SCRUM que cumple con los requisitos necesarios que se requieren para la implementación de la metodología.

De acuerdo con bolsas de empleo tales como multitrabajos.com, computrabajo.com y similares (Anunico, 2015), el salario que percibe un programador semi-senior se encuentra actualmente valorado entre \$800 y \$900. Vale destacar que este salario puede variar dependiendo de la experiencia del programador y la cantidad de lenguajes de programación que maneje.

A continuación en la siguiente tabla (Tabla XVI), se detallan en la primera columna los diferentes roles que define la metodología SCRUM asignados a los cargos (Segunda columna) que actualmente se encuentran disponibles en el área de desarrollo del departamento de TICs:

Roles de Scrum	Cargo
Product Owner	Jefatura del departamento de TICs
Scrum Master	Líder del Equipo de desarrollo
Equipo de desarrollo	Estudiantes con convenios y programadores del área de desarrollo.

Tabla XVI Asignación de los Roles que define SCRUM

Para comenzar a utilizar SCRUM como metodología de desarrollo de software en el departamento de TICs, es necesario que previamente todos los miembros del equipo de trabajo reciban una capacitación sobre los conceptos y prácticas que ofrece la metodología SCRUM, para que así tengan bien en claro las diferentes funciones que deben realizar una vez que se les asignen sus respectivos roles.

Para cumplir con lo expuesto en el párrafo anterior, es necesario que en primer lugar se capacite únicamente a la persona que será encargada de realizar las funciones de Scrum Master, ya que este será el responsable de asegurar que la metodología SCRUM sea entendida y adoptada de forma correcta, ajustándose a las reglas, prácticas y teoría que ofrece la metodología SCRUM. De acuerdo con Banda (2015), en Ecuador la certificación para un Scrum Master se encuentra valorada en \$650 más impuestos de ley, la misma que incluye todos los conceptos que un Scrum Master debe conocer para implementar con éxito la metodología SCRUM en un equipo de trabajo.

Una vez que la persona encargada de realizar las funciones de Scrum Master reciba la capacitación anteriormente mencionada, deberá impartir a los miembros del equipo SCRUM todo el conocimiento adquirido en dicha capacitación, una vez concluido esto,

todos los miembros del equipo contarán con los conocimientos necesarios para comenzar la implementación de la metodología.

Como se puede apreciar en la Tabla XVI, la persona encargada de la Jefatura del departamento de TICs será quien desempeñe las funciones de Product Owner, debido a que es quien establece el lazo de comunicación entre el equipo de desarrollo y el cliente, por tal razón tiene la función de recoger a través de una ingeniería de requerimientos las necesidades del cliente, y a su vez organizar dichas necesidades las veces que sea necesaria en base a las prioridades del cliente y dividir las en historias de usuarios, para finalmente generar el Product Backlog.

El líder del equipo de desarrollo del departamento de TICs deberá desempeñar el rol de Scrum Master, será el encargado de asegurar que el equipo trabaje siguiendo los lineamientos, prácticas, teoría y reglas que propone SCRUM, debido a esto el mencionado rol es fundamental dentro de la metodología. En otras palabras, tiene como función coordinar los diferentes tipos de reuniones, así como también vigilar el progreso normal del trabajo que desempeña el equipo de desarrollo.

Los estudiantes que se encuentran con convenio y los programadores, formarán el equipo de desarrollo, estos tendrán la capacidad de ser auto-organizados y multifuncionales, son quienes están encargados de planificar los Sprint Backlog, además de desarrollar los diferentes Spring que requieran los proyectos de software, para así generar un desarrollo ágil.

Vale destacar que el Scrum Master es el encargado de verificar y controlar que la metodología SCRUM se siga en todo momento, creando diferentes herramientas y capacitando a su equipo de trabajo para que gradualmente se vayan acoplando a las diferentes técnicas y ajustándose cada vez más al desarrollo de proyectos de manera ágil a través de SCRUM.

A continuación en el siguiente diagrama de flujo de trabajo (Figura 2) se puede apreciar de manera clara las diferentes funciones que deben desempeñar cada uno de los integrantes del Equipo Scrum para llevar a cabo el correcto desarrollo de sistemas informáticos en el departamento de TICs de la PUCESE.

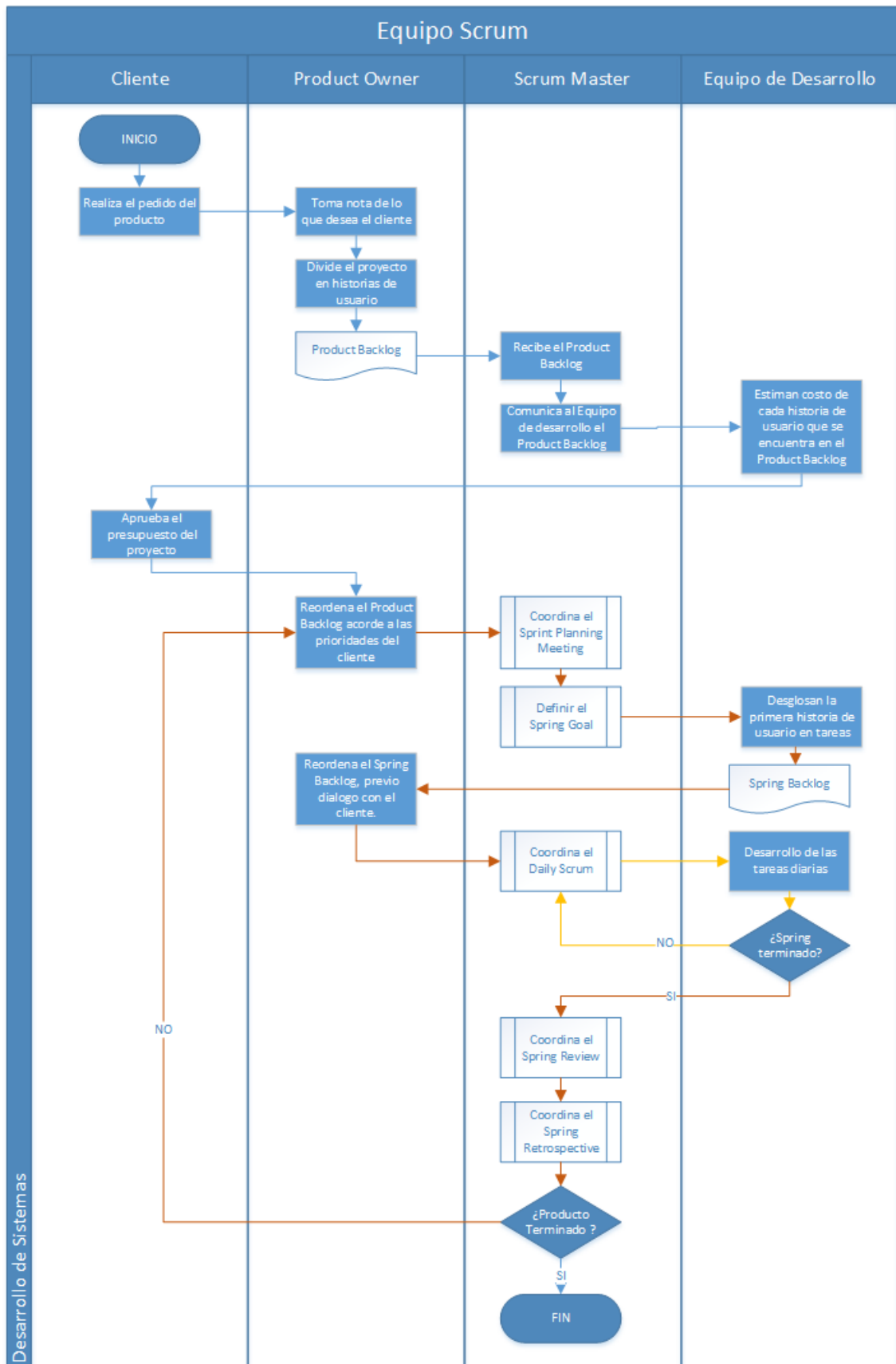


Fig. 2 Flujo de trabajo del Equipo SCRUM

El éxito de la implementación de esta metodología de desarrollo ágil depende en gran medida del completo compromiso de los integrantes del área de desarrollo del departamento de TICs para seguir los lineamientos que define SCRUM, además de la enorme predisposición que deben tener los clientes y sobre todo la alta gerencia en lo que se refiere a participar de manera continua de las actividades del equipo SCRUM.

Vale destacar que la metodología SCRUM cumple con todos los requerimientos propuestos en la auditoría interna realizada en el departamento de TICs en el año 2014, la cual indicaba que la metodología a implementar en el mencionado departamento debía de contar como mínimo con las siguientes fases:

- Definición de las necesidades
- Análisis y diseño del sistema
- Desarrollo
- Pruebas
- Implementación
- Mantenimiento

La auditoría interna - 2014 también sugería que la metodología a utilizar debía de contar con procedimientos que permitieran la aprobación del proyecto por parte de los usuarios finales y el responsable del proyecto. Adicionalmente enunciaba que el proceso de desarrollo implementado a través de la metodología debía cumplir con estándares de programación, pruebas, control y verificación.

Para culminar la propuesta, a continuación se expone un diagrama jerárquico (Figura 3) donde se especifica la relación organizacional con la que debe contar el departamento de TICs de la PUCESE una vez implementada la metodología SCRUM.

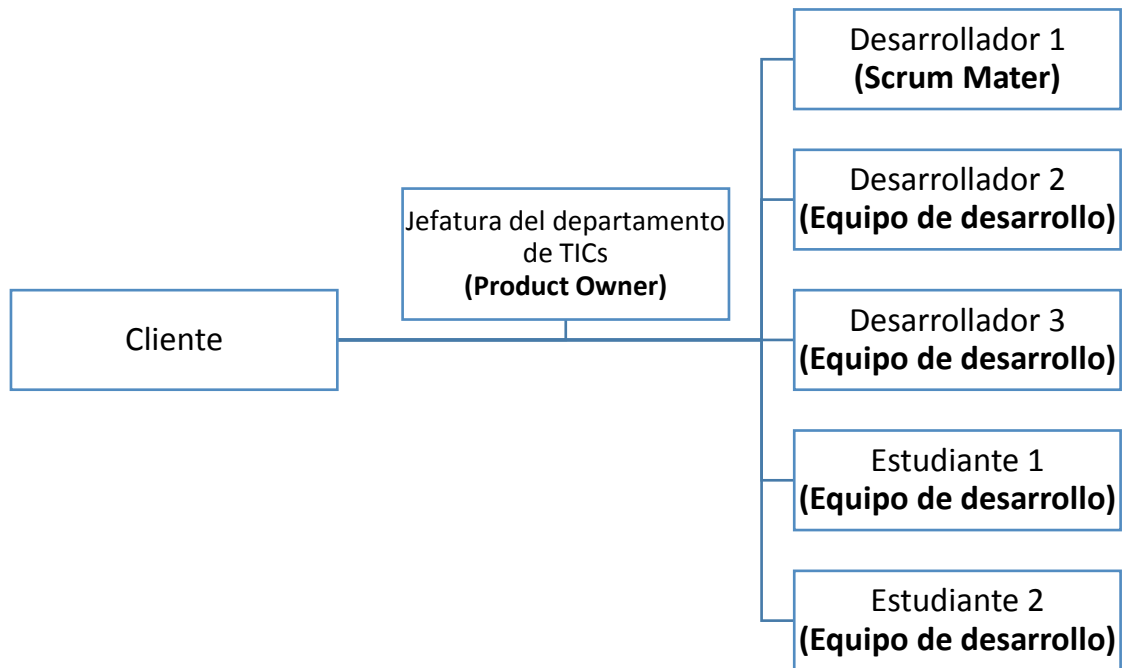


Fig. 3 Estructura jerárquica del departamento de TICs de la PUCESE una vez implementada la metodología SCRUM.

7. Referencias Bibliográficas

Anunico. (2015). *Salario de un desarrollador Semi Senior*. Obtenido de Anunico:
http://quito.anunico.ec/anuncio-de/software_ti_telecomunica_es/desarrollador_sistemas_semi_senior-11871153.html

Banda, J. (2015). *Certified ScrumMaster*. Obtenido de Scrum Alliance:
<https://www.scrumalliance.org/courses-events/courses/csm/ecuador/quito/2015/october/20154619-certified-scrummaster>

Canós, J. H., Letelier, P., & Penadés, M. C. (s.f.). *Métodologías Ágiles en el Desarrollo de Software*. Obtenido de Universidad Politécnica de Valencia:
http://www.carlosfau.com.ar/nqi/nqifiles/XP_Agil.pdf

Cendejas, J. (2014). *Implementación de un modelo integral colaborativo (MDSIC) como fuente de innovación para el desarrollo ágil de software en las empresas de la zona centro - occidente en México*. Obtenido de Eumed:
<http://www.eumed.net/tesis-doctorales/2014/jlcv/software.htm>

Egas Luis, J. J. (2014). *Implementación de una metodología de desarrollo ágil para el análisis, diseño e implementación del sistema de software y aplicación de un caso de estudio práctico, en el departamento de tecnologías de información y comunicación*. Obtenido de ESPE:
<http://repositorio.espe.edu.ec/bitstream/21000/8770/2/T-ESPEL-SOF-0004-P.pdf>

Godoy, C. (06 de 2014). Participación de docentes universitarios en entornos virtuales de aprendizaje. *Participación de docentes universitarios en entornos virtuales de aprendizaje*. Esmeraldas, Esmeraldas, Ecuador.

- Ingenieria de Software. (2012). *Programación extrema XP*. Obtenido de Ingenieria de Software: http://m.ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html
- INTECO. (03 de 2009). *Ingeniería del software: Metodologías y Ciclos de vida*. Obtenido de Incibe: https://www.incibe.es/file/N85W1ZWfHifRgUc_oY8_Xg
- Mendez, I. G. (13 de 02 de 2013). *Etapas de desarrollo de software*. Obtenido de Fundamentos de Software: http://ithuejutlaisabelgarciamendez.blogspot.com/2013/02/13-etapas-de-desarrollo-de-software_19.html
- Oscar Tinoco Gómez, P. P. (19 de 11 de 2010). *Criterios de selección de metodologías de*. Obtenido de Redalyc: <http://www.redalyc.org/articulo.oa?id=81619984009>
- ProcesosdeSoftware. (s.f.). *Metodologías para el desarrollo de software*. Obtenido de ProcesosdeSoftware: <http://procesosdesoftware.wikispaces.com/METODOLOGIAS+PARA+DESARROLLO+DE+SOFTWARE>
- proyectosagiles.org. (12 de 01 de 2009). *Requisitos para poder utilizar Scrum*. Obtenido de proyectosagiles.org: <http://proyectosagiles.org/requisitos-de-scrum/>
- proyectosagiles.org. (s.f.). *Colaboración y comunicación entre el equipo y con el cliente*. Recuperado el 10 de 09 de 2015, de Proyectos Agiles: <http://www.proyectosagiles.org/colaboracion-comunicacion-equipo-cliente>
- Raúl, E. (30 de 06 de 2009). *Método Análítico Sintético*. Obtenido de Técnicas de Investigación de Campo: <http://niveldostic.blogspot.com/2009/06/metodo-analitico-sintetico.html>
- Schwaber, K., & Sutherland, J. (07 de 2013). *La Guía Definitiva de Scrum : Las reglas del juego*. Obtenido de scrumguides: www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf

Scrummanager. (20 de 04 de 2011). *Gestión de cambios en Scrum*. Obtenido de Scrum Managers Blog: <http://www.scrummanager.net/blog/2011/04/gestion-de-cambios-en-scrum/>

Scrummanager. (27 de 04 de 2014). *El Manifiesto Ágil*. Obtenido de Scrum Manager Body of Knowledge: www.srummanager.cnet/bok/index.php?title=El_manifiesto_ágil

sourceforge. (s.f.). *Ciclo de vida de un proyecto XP*. Recuperado el 10 de 09 de 2015, de Sourceforge: <http://oness.sourceforge.net/proyecto/html/ch05s02.html>

Supo, J. (02 de 01 de 2014). *Niveles de Investigación*. Obtenido de Seminarios de la Investigación: <http://seminariosdeinvestigacion.com/niveles-de-investigacion/>

Wikipedia. (s.f.). *Programación extrema*. Recuperado el 10 de 09 de 2015, de Wikipedia: https://es.m.wikipedia.org/wiki/Programaci%C3%B3n_extrema

8. Anexos

ENTREVISTA DIRIGIDA A LA JEFA DEL DEPARTAMENTO DE TICS DE LA PUCESE

1. ¿Utilizan alguna metodología de desarrollo de software para el diseño y programación de los sistemas informáticos?
2. ¿Qué tipo de proceso siguen para llevar a cabo el desarrollo de un software?
3. ¿Cómo realiza el seguimiento y control a los proyectos de software?
4. ¿Cuáles son los problemas comunes que se identifican al momento de llevar a cabo el desarrollo de software?
5. ¿Cómo llevan a cabo la documentación de los sistemas desarrollados?
6. ¿Cumplen de manera adecuada con los tiempos de entrega de los sistemas desarrollados?

ENTREVISTA DIRIGIDA A LOS INTEGRANTES DEL EQUIPO DE DESARROLLO DEL DEPARTAMENTO DE TICS DE LA PUCESE (DESARROLLADORES)

1. **¿Utiliza alguna metodología de desarrollo de software para el diseño y programación de sistemas informáticos?**
 - a. Si
 1. ¿Qué metodología de desarrollo de software está usando actualmente?
 2. ¿Se siente cómodo usando la mencionada metodología de desarrollo de software?
 3. ¿Le resulta complicado seguir los procesos de la mencionada metodología de desarrollo de software?
 4. ¿El uso de la mencionada metodología de desarrollo de software ha contribuido a mejorar los sistemas informáticos resultantes?
 5. ¿Estaría dispuesto a seguir los lineamientos de una nueva metodología de desarrollo de software con la única finalidad de mejorar los sistemas informáticos resultantes?
 - b. No
 1. ¿Por qué razón no usa una metodología de desarrollo de software?
 2. ¿Le parece innecesario el uso de una metodología de desarrollo de software?
 3. ¿Ha contribuido en algo, el no uso de una metodología de desarrollo de software a los sistemas resultantes?
 4. ¿Sigue algún tipo de lineamiento o proceso para llevar a cabo el desarrollo de un software?
 5. ¿Estaría dispuesto a seguir los lineamientos de una metodología de desarrollo de software con la única finalidad de mejorar los sistemas informáticos resultantes?

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
SEDES ESMERALDAS**

Encuesta dirigida a Estudiantes, Docentes y Personal Administrativo

Estimado/a estudiante, administrativo/a o docente de la Pontificia Universidad Católica Sede Esmeraldas (PUCESE). La presente encuesta tiene la finalidad de recabar información acerca del nivel de satisfacción de la comunidad universitaria respecto a los sistemas informáticos de la PUCESE, por tal razón le solicito contestar con absoluta claridad y con la mayor veracidad posible.

Marque con una X la respuesta que estime conveniente:

1. ¿Con que frecuencia utiliza los sistemas informáticos de la PUCESE?

- Siempre ()
- Casi siempre ()
- A veces ()
- Casi nunca ()
- Nunca ()

2. ¿Con qué nivel de dificultad usted calificaría a los sistemas informáticos de la PUCESE en relación con el uso?

- Muy Difícil ()
- Difícil ()
- Normal ()
- Fácil ()
- Muy Fácil ()

3. ¿Cómo calificaría el funcionamiento de los sistemas informáticos de la PUCESE?

- Pésimo ()
- Regular ()
- Bueno ()
- Muy bueno ()
- Excelente ()

4. ¿Los sistemas informáticos de la PUCESE que usted ha utilizado cuentan con documentación apropiada para aprender a manejarlos?

- Si ()
- No ()

5. ¿Cuán a menudo se le han presentado dificultades al utilizar un sistema informático de la PUCESE?

- Siempre ()
- Casi siempre ()
- A veces ()
- Casi nunca ()
- Nunca ()

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
SEDE ESMERALDAS**

La presente encuesta tiene como finalidad recabar información acerca del nivel de satisfacción en cuanto al proceso que se lleva a cabo al solicitar algún tipo de sistema informático al Departamento de TICs de la PUCESE.

Marque con una X la respuesta que estime conveniente:

- 1. ¿Con qué frecuencia solicita el desarrollo de sistemas informáticos al departamento de TICs de la PUCESE?**
Diaria ()
Semanal ()
Mensual ()
Trimestral ()
Semestral ()

- 2. ¿El personal del departamento de TICs encargado de desarrollar el sistema informático que usted ha solicitado sigue algún tipo de lineamiento para recoger sus requerimientos?**
Si ()
No ()

- 3. ¿Cómo calificaría la comunicación entre usted y los desarrolladores del Departamento de TICs durante el periodo de diseño y programación del sistema solicitado?**
Excelente ()
Muy Buena ()
Buena ()
Regular ()
Pésima ()

- 4. ¿Con qué frecuencia los desarrolladores del Departamento de TICs le presentan documentación adecuada acerca de los avances que está teniendo el sistema solicitado?**
Siempre ()
Casi siempre ()
A veces ()
Casi nunca ()
Nunca ()

- 5. ¿Con qué frecuencia el sistema informático solicitado al Departamento de TICs se entrega en los tiempos acordados sin ningún tipo de retrasos?**
Siempre ()
Casi siempre ()
A veces ()
Casi nunca ()
Nunca ()

- 6. ¿El sistema informático desarrollado por el Departamento de TICs cumple de forma adecuada con todos los requerimientos solicitados?**
Si ()
No ()

RÚBRICA 1: COMPARACIÓN DE METODOLOGÍAS ÁGILES

Criterios	5	4	3	2	1
Comunicación con el cliente y entre los miembros del equipo de desarrollo.	Muy Buena comunicación tanto con el cliente como entre los miembros del equipo de desarrollo.	Escasa comunicación con el cliente y buena comunicación entre los miembros del equipo de desarrollo	Escasa comunicación tanto con el cliente como entre los miembros del equipo de desarrollo.	No existe comunicación con el cliente y escasa comunicación entre los miembros del equipo de desarrollo.	No existe comunicación de ningún tipo.
Adaptabilidad a cambios.	Capacidad para gestionar el cambio de forma adecuada e incorpora roles dedicados a dicha gestión.	Capacidad para gestionar el cambio de forma adecuada, pero no incorpora roles dedicados a dicha gestión.	Poca capacidad para gestionar el cambio.	Casi inexistente capacidad de gestionar el cambio.	No existe gestión del cambio
Documentación ágil.	Promueve la documentación ágil basada en el proceso de desarrollo y en el cliente (Manuales).	Promueve la documentación ágil basada en el proceso de desarrollo y escasa documentación ágil para el cliente.	Promueve únicamente documentación ágil basada en el proceso de desarrollo.	Escasa documentación ágil basada en el proceso de desarrollo.	No existe documentación ágil de ningún tipo.
Ciclos de entrega rápidos.	Ciclos de entrega con duración de una a dos semanas.	Ciclos de entrega con duración de tres a cuatro semanas.	Ciclos de entrega mensuales.	Ciclos de entrega trimestrales.	Ciclos de entrega semestrales.
Facilidad para llevar el control sobre el proyecto.	Define roles bien definidos para llevar a cabo el seguimiento y control del proyecto.		Define roles poco definidos para llevar a cabo el seguimiento y control del proyecto.		No existe ningún tipo de herramienta para llevar control sobre el proyecto.
Ingeniería de requerimientos.	Realiza ingeniería de requerimientos a lo largo del proyecto.		Realiza ingeniería de requerimientos únicamente al inicio del proyecto.		No realiza ningún tipo de ingeniería de requerimientos.
Pruebas de software.	Realiza frecuentemente pruebas de software a lo largo del proyecto.		En ciertas ocasiones realiza pruebas de software.		No incorpora ningún tipo de pruebas de software.

