SLEEK SOFTWARE

Universidad Simón Bolívar

Sistema de Gestión de Programa de Servicio Comunitario, SIGESC.
Plan de Iteración.
Versión 7.0.

Integrantes:
Andrea Díaz 09-10236
Iranid Pérez 08-11369
Hetsy Rodríguez 06-40232
David Goudet 08-10479
Daniel Zeait 08-11216
Joel Rivas 11-10866
Jesus Sánchez 10-10898
Nilver Viera 07-41654

Historial de Revisiones

Fecha	Versión	Descripción	Autor
10/02/16	1.0	Unificación del Plan de Iteración (FC) de todos los equipos.	Auyantepui-Quantum-Syscorp-RupDev
20/02/16	2.0	Unificación del Plan de Iteración (FC) de todos los equipos.	Auyantepui-Quantum-Syscorp-RupDev
14/03/16	3.0	Unificación del Plan de Iteración (FC) de todos los equipos.	Auyantepui-Quantum-Syscorp-RupDev
28/04/16	4.0	Plan de iteración de la primera fase de construcción.	SLEEK Software
23/05/16	5.0	Plan de iteración de la segunda fase de construcción.	SLEEK Software
02/06/16	6.0	Plan de iteración de la tercera fase de construcción.	SLEEK Software
06/06/16	7.0	Plan de iteración de la cuarta fase de construcción.	SLEEK Software

Tabla de Contenido

Plan de Iteración

1. Introducción

1.1 Propósito

El plan de iteración que a continuación se desarrolla busca mostrar de forma clara, precisa y detallada la secuencia de actividades que es necesario llevar a cabo en la primera iteración de la fase de Construcción del Sistema de Gestión de Servicio Comunitario (SIGESC).

Todo acorde a lo indicado por la dinámica de fases iterativas de la metodología RUP para el desarrollo del SIGESC de la Universidad Simón Bolívar. La idea que se pretende concretar con la redacción de este documento es entonces, la distribución de las responsabilidades y cargas de trabajo entre los integrantes del equipo de desarrollo.

1.2 Alcance

El plan de iteración es un entregable cuya elaboración está centrada en la descripción de un desarrollo de proyecto iterativo e incremental, y se enfoca en la división de responsabilidades. Por este motivo, para cada actividad a llevar a cabo en las iteraciones, se especifica su duración (tanto en ciclos internos a cada iteración como en semanas reales), el(los) encargado(s) de la entrega y realización de la misma y el producto o hito final que se busca producir para dicha actividad.

Para la distribución de actividades, se ha utilizado el estudio de las prioridades establecidas por el modelo de casos de uso del SIGESC, las especificaciones encontradas en los documentos de Arquitectura del Software y de Especificación de Requerimientos del Sistema.

1.3 Definiciones, Siglas y Abreviaciones

Se presentan las siglas a emplear a continuación, con su significado y definiciones:

- **Actores:** Algo o alguien externo al módulo que interactúa con el sistema.
- **Casos de Uso:** Es una secuencia de acciones que el sistema realiza, el cual proporciona un resultado de acción observable para un actor en particular.
- **DEx**: Decanato de Extensión.
- **ERE:** Entidad Relación Extendido.
- **RUP:** Son las siglas de Rational Unified Process. Se trata de una metodología para describir el proceso de desarrollo de software.
- **SIGESC:** Sistema de Gestión de Servicio Comunitario.
- **USB:** Universidad Simón Bolívar.
- **CFCG**: Coordinación de Formación Complementaria General.
- **ERS:** Especificación de Requerimientos de Software.
- **DAS:** Documento de Arquitectura de Software.

1.4 Referencias

- Plan de Iteración. Versión 5.0.
- Documento de la Arquitectura de Software. Versión 1.2
- Especificación de Requerimientos del Sistema. Versión 1.0
- Modelo de Casos de Uso refinado.

1.5 Descripción

El Plan de Iteración que se desarrolla a continuación muestra de forma clara el desarrollo temporal y secuencial de las actividades del sistema correspondientes a la primera iteración de la fase de Construcción del proyecto. Esto se logra mediante una tabla que muestra de forma organizada y concisa la información que concierne a cada actividad planificada para alcanzar los objetivos (responsables, tiempo de duración, el hito que se espera alcanzar).

En el punto posterior se muestran los recursos necesarios y disponibles para que el desarrollo de las actividades planteadas sea adecuado y exitoso. Por último se aclaran los casos de uso implementados o concebidos en las iteraciones analizadas, esto junto con los criterios de evaluación tomados en cuenta para dicha planificación de actividades.

2. Plan

Actividad	Trimestre	Duración (Semanas)	Iteración	Responsable(s)	Hito
Actualizar casos de prueba	2	1	4	Hetsy Rodríguez, Daniel Zeait Iranid Pérez,	
Actualizar casos de uso	2	1	4	Hetsy Rodríguez, Daniel Zeait Iranid Pérez	
Desarrollo de funcionalidad para el 70%.	2	1	4	Andrea Díaz, David Goudet Joel Rivas, Nilver Viera Jesús Sánchez	
Actualización de Minutas y Matriz de Seguimiento.	2	1	4	Todo el Equipo.	
Actualizar documentos del sistema.	2	1	4	Hetsy Rodríguez, Daniel Zeait Iranid Pérez,	

3. Recursos

- Humanos:

    1. Estudiantes en el equipo de trabajo que cumplan con la entrega y las correcciones de los documentos necesarios para cumplir con la iteración, siguiendo los lineamientos de la metodología de RUP.

    2. Estudiantes en el equipo de trabajo con suficiente dominio par para poder programar en Python, JavaScript, Bootstrap, HTML, CSS, así como manejar Web2py, PostgreSQL y Github.

    3. Personal académico que asesore en la metodóloga del RUP para esta iteración.

    4. Stakeholders que indiquen el nivel de calidad del producto según sus expectativas.

- Financieros:

Este en un proyecto con fines académicos, por lo que no se cuenta con recursos financieros durante el desarrollo de este proyecto. Por este motivo, el proceso de desarrollo del sistema se lleva a cabo, en su mayoría, haciendo uso de herramientas Open Source.

- Hardware:

Computadoras que soporten los requisitos de software involucrados, de manera que la aplicación se pueda ejecutar y mantener de forma eficiente. Por ahora se cuenta con un servidor provisto por el DEx.

- Software:

Herramientas aprobadas en los lineamientos de la CFCG para el desarrollo de software para la USB, entre las cuales están:

1. Leguajes de programación: Python, JavaScript, HTML,CSS.
2. Servidor Web: Apache.
3. El framework a utilizar es Web2py.
4. Manejador de base de datos PostgreSQL.
5. El repositorio Bitbucket.
6. Software colaborativo Google Docs

4. Casos de Uso

A continuación los casos de usos correspondientes al 50% del prototipo funcional del Sistema de Gestión de Servicio Comunitario de la Universidad Simón Bolívar (SIGESC-USB) en el cual se aplicarán las mejoras discutidas con el cliente:

- **Verificar Estado de SC**: Un Estudiante puede verificar el estado de su inscripción de servicio comunitario.
    - Verificar Actividades del Proyecto: Un estudiante puede visualizar todas las actividades que debe realizar para poder culminar el proyecto de servicio comunitario que tiene inscrito actualmente.

- Culminar Proyecto: Un estudiante puede culminar su proyecto una vez ha realizado todas las actividades que el mismo tiene programadas por parte del tutor académico.
    - Solicitar Aprobación de Culminación: Un estudiante al culminar un proyecto puede solicitar la aprobación de culminación, la cual será validada por el coordinador de la CFCG.
- **Ver Cronograma**: Un estudiante puede visualizar su cronograma de actividades de un proyecto de Servicio Comunitario que ya inscribió.
    - Actualizar Horas Cumplidas: Un estudiante puede actualizar sus horas a medida que vaya cumpliendo las actividades que tiene programadas.
- **Certificar Inscripción del Estudiante**: El Asistente de la CFCG puede validar la inscripción del servicio comunitario de un estudiante que cumpla con tener aprobado más de la mitad de los créditos de su carrera.
- **Consultar Proyectos A Cargo**: Un Tutor Académico puede consultar todos los proyectos de los cuales es tutor.
    - Crear Plan de Actividades Del Estudiante: Un Tutor Académico puede crear el plan de actividades de los proyectos que tiene a cargo para un estudiante que haya inscrito dicho proyecto.

5. Criterios de Evaluación.

Para evaluar el desarrollo del sistema junto con la distribución de actividades aquí propuesta, se plantean a continuación unos criterios de calidad y desempeño:

- **Funcionalidad:** Se plantea como meta la implementación de los casos de uso y/o requerimientos funcionales del sistema mencionados en el punto anterior.

- **Desempeño y Disponibilidad:** El sistema debe estar alojado en algún servidor con disponibilidad continua y debe poder hacer un uso óptimo de los recursos de hardware del servidor, de tal forma que responda a las solicitudes de usuarios en un tiempo corto (no más de 15 segundos).

- **Capacidad:** Según las estimaciones sobre los requerimientos del cliente, se pretende que el sistema tenga la capacidad de manejar varios usuarios realizando solicitudes y/o consultas concurrentemente.

- **Objetivos de Calidad:** La seguridad del sistema es de vital importancia; éste debe garantizar la anonimidad de sus usuarios registrados así como la confidencialidad e integridad de la información que se maneja. Además, se debe poder administrar un esquema de roles o “vistas” en el sistema en función de la iniciación de sesiones, de tal forma que cada usuario sólo tenga autorización de acceso a aquella información que le corresponda. Sumado a eso, se debe diseñar y concebir una interfaz intuitiva y sencilla, cuya principal fortaleza sea la rapidez de acceso a los datos o información requerida y la facilidad de aprendizaje por parte de sus usuarios. La probabilidad de que el sistema falle debe ser muy baja, para permitir así que el sistema sea confiable y cumpla con su función. Finalmente el sistema debe ser tolerante a posibles fallas de software, debe ser confiable al momento que ocurra un incidente.