



**UNIVERSIDAD TÉCNICA DE BABAHOYO FACULTAD DE ADMINISTRACIÓN,
FINANZAS E INFORMÁTICA**

PROCESO DE TITULACIÓN

EXAMEN COMPLEXIVO DE GRADO O DE FIN DE CARRERA ESTUDIO DE CASO

INGENIERÍA EN SISTEMAS

PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS

TEMA:

**ESTUDIO COMPARATIVO DE LA METODOLOGÍA DE SOFTWARE RUP Y SCRUM EN
EL DESARROLLO DE SOFTWARE**

EGRESADO:

RONALDO JACOBO RODRIGUEZ VEGA

TUTOR:

ING. CARLOS ALFREDO CEVALLOS MONAR

AÑO 2023

RESUMEN

En la actualidad, existen varias opciones al momento de seleccionar una metodología para el desarrollo de software. Por lo tanto, el presente caso de estudio se enfoca en determinar que metodología resulta ser más recomendable, mostrando ventajas, desventajas y a su vez al tipo de proyecto que se adapta cada una.

Tiempo atrás dominaban las metodologías austeras, también denominadas metodologías tradicionales, las cuales existía bastante documentación, acciones y roles. Después del año 2000 surgen otros tipos de metodologías, más ágiles y eficaces para desarrollo de software.

Estas metodologías fueron desarrolladas para aplicarlas en pequeños y medianos proyectos que tienen requerimientos que cambian constantemente. En ciertas empresas del Ecuador se usa este tipo de metodología (ágiles) con el objetivo de corregir problemas de tiempo, dinero y calidad en el desarrollo de software. Mientras que las metodologías tradicionales, son aplicadas para proyectos de gran tamaño, que tienen requerimientos más estables y no cambian con tanta frecuencia.

Con base en lo antes mencionado, se desea explicar las metodologías: Scrum, Rup. Tomando en cuenta todos los detalles y determinar el uso y enfoque adecuado de cada una.

Palabras claves: Metodología de software, Metodología tradicional, Metodología ágil, Rup, Scrum.

ABSTRACT

Currently, there are several options when selecting a methodology for software development. Therefore, the present case study focuses on determining which methodology turns out to be more recommendable, showing advantages, disadvantages and, in turn, the type of project that each one is adapted to.

Long ago, the austere methodologies, also called traditional methodologies, dominated, which existed a lot of documentation, actions and roles. After the year 2000, other types of methodologies emerged, more agile and effective for software development.

These methodologies were developed to be applied in small and medium projects that have constantly changing requirements. In certain companies in Ecuador, this type of methodology (agile) is used with the aim of correcting problems of time, money and quality in software development. While traditional methodologies are applied to large projects, which have more stable requirements and do not change as frequently.

Based on the aforementioned, it is desired to explain the methodologies: Scrum, Rup. Taking into account all the details and determining the proper use and approach of each one.

Keywords: Software Methodology, Traditional Methodology, Agile Methodology, Rup, Scrum.

INTRODUCCIÓN

Una metodología debe especificar de manera precisa las herramientas, los roles y las actividades. Usualmente el “método” se lo usa para hacer referencia a diversos procedimientos que se aplican a las acciones del proceso de desarrollo del software.

La comparación de las metodologías resulta ser una labor compleja, esto se debe a la gran cantidad de diferencias, la disponibilidad de los datos y el alcance de cada una. A mayor escala y tomando en cuenta sus ideologías de desarrollo, las metodologías con más importancia en la preparación, dominio del proyecto, requerimientos y modelado, se las denomina metodologías tradicionales.

Existen otro tipo de metodologías, a estas se las conocen como metodologías ágiles. Estas se enfocan en la creación de código con pequeños periodos de desarrollo; están dirigidos a pequeños segmentos. Estas se enfocan en el trabajo en equipo y hacen partícipe de manera directa al cliente en el proceso.

Las metodologías de desarrollo de software se usan para la estructuración, planificación y control del proceso de un IS (Information System) o SI (Sistema de información).

En el presente estudio de caso se empleará una metodología de investigación cualitativa, con técnicas de investigación documental. La línea de investigación tiene relación con los Sistemas de información y comunicación, emprendimiento e innovación y la sub línea de investigación con las redes y tecnologías inteligentes de software y hardware.

DESARROLLO

Metodología de Software RUP

Su abreviatura significa Rational Unified Process (Proceso Unificado Racional). Los encargados de crear este término fue la empresa Rational Software Corporation, actualmente perteneciente a IBM. (Mancha, 2021)

Es un proceso de ingeniería de software que define qué es lo que debe hacerse en el proyecto, proporciona una perspectiva para la asignación de roles y obligaciones en el interior de un entorno de desarrollo. Es utilizada principalmente en proyectos que poseen mayor complejidad con equipos de gran tamaño. (BEDOYA, 2018)

La metodología RUP es una metodología tradicional o rígida con enfoque iterativo, se apoya en el modelo UML (Unified Modeling Language o Lenguaje Unificado de Modelado) al momento de tratarse de POO. El Lenguaje Unificado de Modelado por medio de varias clases de diagramas, beneficia y apoya al momento de ejecutar tareas del software que se va a desarrollar. (Patiño, 2018)

Su objetivo es garantizar que se genere un software de excelente calidad, que satisfaga las exigencias de los usuarios dentro de un periodo y costo previsto. Todos los miembros del entorno pueden ingresar información en la base de datos, esto, sin la necesidad de que sea su tarea asignada específica. De esta manera todos están en el mismo “canal”, tienen la misma visión y el mismo objetivo de desarrollar un software. (Maida & Pacienza, 2020)

La metodología RUP está compuesta por las siguientes fases de desarrollo:

- Fase de inicio
- Fase de Elaboración
- Fase de Construcción
- Fase de Transición o cierre

Fase de Inicio

Se analizan los fundamentales casos de uso, el alcance del proyecto y los peligros que pueden existir. Se determinan objetivos.

Modelado del negocio

El grupo se adapta a la empresa a sus funciones y sus procesos.

Visión y atributos

- Analizar cuáles son las necesidades de una empresa y determinar el tipo sistema que se desarrollará.
- Comprender los problemas que se generan dentro de una empresa e implementar posibles soluciones y mejoras.
- Garantizar que los usuarios y desarrolladores tengan la comprensión total de los objetivos de la empresa.

Requerimientos

En esta parte, tratamos las normas que se deben llevar a cabo, de manera que los clientes comprendan y acepten los requerimientos que se indican.

Visión y atributos

- Llevar a cabo un acuerdo de confidencialidad entre usuarios y desarrolladores sobre las funcionalidades del software.
- Brindar total información a los desarrolladores sobre los requerimientos del sistema.
- Determinar el entorno del sistema.
- Analizar y tener en cuenta el periodo y presupuesto del desarrollo del software.
- Realizar una interfaz que sea totalmente intuitiva para los usuarios. (Maida & Pacienza, 2020)

Fase de Elaboración

Esta fase se enfoca en la arquitectura central del sistema, en planear las actividades, el diseño y las características.

Estudio & Diseño

En esta parte se especifican los requisitos y se explica la implementación del sistema.

Visión y atributos

- Convierte los requerimientos a la interfaz del sistema
- Desarrollo de una infraestructura para el sistema (Pérez, 2019)

Fase de Construcción

Se trata de verificar la operatividad del sistema, administrarlo y realizar los cambios que sean necesarios, hasta adaptarlo a las necesidades y requerimientos solicitados. Construir el manual de usuario y cuando esté completo socializarlo con los demás. (Piñeiro, 2022)

Implementación

Se realiza la implementación de clases y objetos con el fin de obtener un software que se pueda ejecutar.

Visión y atributos

- Conformar el plan de integración, determinando el orden de los subsistemas.
- Los desarrolladores deciden el orden de implementación de los componentes de los subsistemas.
- En el caso de que existan errores de interfaz, se los comunica.
- Se completa el software conforme a lo planeado.

Pruebas

En esta sección se evalúa la calidad de lo que se está desarrollando.

Visión y atributos

- Encuentra y detalla fallas en el sistema.
- Asesora sobre la calidad del sistema.
- Corrobora las funcionalidades del sistema según lo planteado. (Pérez, 2019)

Fase de transición o cierre

Esta fase se enfoca en garantizar la total disponibilidad del sistema para los usuarios finales. Se instala el software al usuario y se le da su respectiva preparación. (Piñeiro, 2022)

Despliegue

Se trata de distribuir con éxito el sistema a los usuarios. Algunas de las funciones incluidas son:

- Prueba del sistema en su etapa final.
- Preparar el sistema para distribuirlo.
- Distribución del sistema
- Instalación del sistema
- Brindar apoyo a los usuarios

De manera que continua el proyecto, va de una fase a otra. La fase de inicio se enfoca en el modelamiento de negocio y la fase de construcción en la implementación e interfaz. (Pérez, 2019)



Figura n.º 1: Fases de la metodología RUP- Autor: Luis Castellanos

Principios de RUP

Los principios del RUP son:

- Adaptación al Proceso
- Equilibrio de prioridades
- Demostrar valor iterativamente
- Colaborar entre equipos
- Enfoque en la calidad
- Elevación del Nivel de Abstracción

Adaptación al Proceso

El proceso debe adaptarse a lo que necesita el cliente, porque es de suma importancia la interacción con él. La naturaleza, alcance, tipo o normativa aplicable del proyecto afectará a su diseño específico. (Vera, Córdova, & López, 2019)

Equilibrio de prioridades

Los requerimientos de diferentes actores pueden ser diferentes, entrar en conflicto entre sí o competir por recursos limitados. Se debe encontrar un “punto medio” que tenga contentos a todos. Debido a este “punto medio”, las diferencias futuras pueden corregirse. Una vez más, este enfoque sigue el lenguaje de modelado unificado (UML). (Vera, Córdova, & López, 2019)

Demostrar valor iterativamente

Los proyectos se dan en fases iterativas, aunque internamente. En cada iteración se afina el rumbo del proyecto y los peligros asociados al mismo. (Fuentes, 2018)

Colaborar entre equipos

Varios equipos realizan lo que es el desarrollo de software, no solo una persona. Debe existir buena comunicación para alinear requerimientos, desarrollo, evaluación, planificación, etc. (Vera, Córdova, & López, 2019)

Enfoque en la calidad

El control de calidad debe realizarse en todos los aspectos de la producción, no al final de cada iteración. El garantizar la calidad también forma parte del proceso de desarrollo de software. (Vera, Córdova, & López, 2019)

Elevación del Nivel de Abstracción

Esta causa fundamental incentiva la utilización de patrones de interfaces de software, frameworks 4GL, etc. Se pueden complementar con una representación visual de la arquitectura, como UML (Lenguaje Unificado de Modelado). (Vera, Córdova, & López, 2019)

Elementos RUP

Se divide en cuatro elementos:

- Rol
- Actividad
- Producto o artefacto
- Flujo de Trabajo

Rol

Determina las actividades a realizar y responsabilidades que tiene una persona o un grupo de personas. (CHACÓN, 2018)

Actividad

Es el trabajo que es asignado a un rol, para que lo ejecute un usuario ligado a ese rol. (CHACÓN, 2018)

Producto/Artefacto

Los productos/artefactos son datos producidos, modificados y utilizados en el proceso de desarrollo de sistemas. Es lo que se ha creado en el proceso de desarrollo y se va puliendo hasta que se obtenga el producto final. (CHACÓN, 2018)

Flujo de Trabajo

Es la unión entrelazada que existe entre las actividades la que genera buenos resultados. (CHACÓN, 2018)

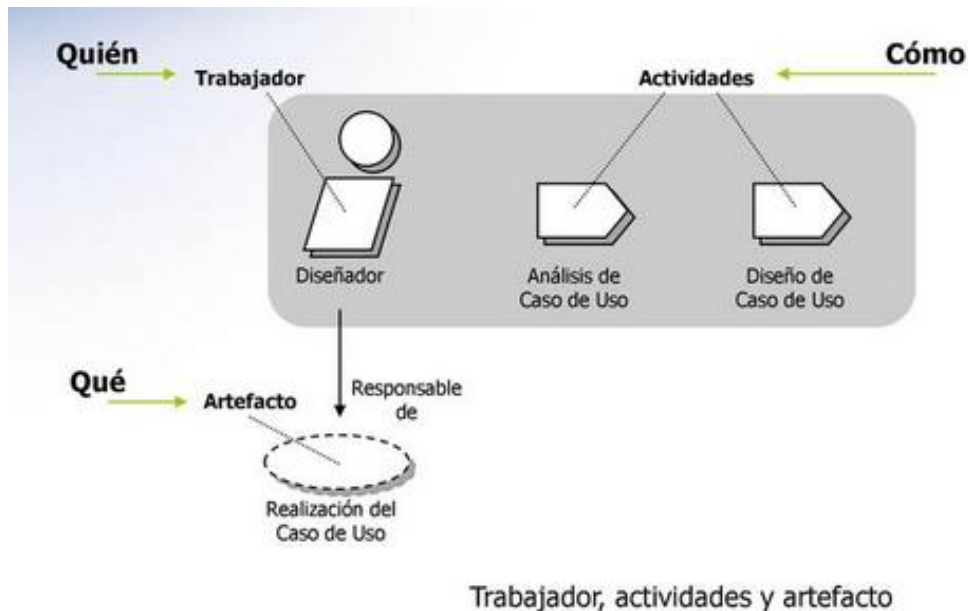


Figura n.º 2: Elementos de la metodología RUP- Autor: Pontificia Universidad Católica de Chile

Ventajas y Desventajas de la Metodología RUP

Ventajas

- Es el proceso de desarrollo más común en la actualidad. En otras palabras, este proceso es uno de los más utilizados por las empresas para el desarrollo de software debido a que está altamente optimizado y es factible para la mayoría de las empresas.
- Es una manera útil para realizar asignaciones de actividades en una organización, debido a que los roles se definen correctamente y se determina quien hace las actividades.
- Fácil de realizar mantenimiento y modificar localmente
- Todo el equipo ligado al proyecto puede acceder al mismo con total facilidad.
- Brinda a cada usuario un filtro personalizado, según el rol que ocupe en el proyecto.

Desventajas

- Debido a la complejidad, puede que no sea muy adecuado. Esto se debe a que es un gran proceso y no se presta a pequeños proyectos.
- Los pequeños proyectos no pueden cubrir los costos. Esto se debe a que es una metodología muy costosa y necesita bastantes roles (personal), esto hace que el proyecto sea inasequible.

- Muchas veces resulta ser pesado.

Metodología de Software SCRUM



Figura n.º 3: Valores de la metodología SCRUM- Autor: Alexander Menzinsky

Su terminología parte de una formación que se hace en el juego rugby, donde los jugadores se ayudan mutuamente y se define como “colaboración”.

Es una metodología ágil que se enfoca en el trabajo en equipo. Su enfoque es totalmente a los valores, prácticas de gestión y no a los requisitos, ni lo técnico. Es una metodología que confía totalmente en el equipo y deja que decidan cual es la mejor manera de trabajar y elevar la productividad al máximo. (López, 2019)

SCRUM aborda cualquier proyecto segmentándolo en partes pequeñas. Es particularmente adecuado para entornos donde los resultados se necesitan rápidamente, donde los resultados cambian o no se definen correctamente. Donde es más importante innovar, competir, ser flexible y productivo. (Estrada, Saltos, & Núñez, 2021)

La metodología se basa en los siguientes principios ágiles:

- Valorar a las personas por encima de los procesos.
- Hacer la entrega del sistema funcional, en poco tiempo.
- Tener voluntad y respuestas al cambio
- Comunicarse de manera verbal y directa con los que conforman el proyecto
- Autoridad completa a los miembros del equipo para cumplir con los requerimientos.
- Formar equipos pequeños entre tres y nueve personas. (Menéndez, 2020)

Características de SCRUM

- Flexible, personalizado, mínimo riesgo, productivo y de alta calidad.
- Se utiliza equipos autoorganizados.
- Las reuniones de Scrum se realizan diariamente y son reuniones de progreso de 15 minutos o menos con el objetivo de recibir comentarios sobre las actividades del equipo y las dificultades encontradas.
- Contiene bases de valores, principios. (Pérez, 2019)

Elementos de SCRUM

Son varios los elementos de SCRUM y a continuación se mencionan algunos:

- **Roles:** el total del equipo no deberá pasar de las 20 personas, si pasa esto, lo mejor es dividir en varios equipos. La estructura de Scrum es muy simple. Las actividades del proyecto se dividen en 3 roles:
 - **Dueño del producto:** es la persona responsable de gestionar y controlar el proyecto. Es el responsable de las áreas de financiamiento del proyecto, requerimientos del software, retorno del dinero invertido, presentación del proyecto. (González, 2021)
 - **Scrum Master (facilitador):** es el encargado de que la meta se cumpla y se resuelvan los problemas.
 - **Desarrollador:** es el encargado de convertir el product backlog en un sistema entregable. (Dimes, 2021)
- **Poda de requerimientos:** consiste en realizar todo lo que el usuario necesita.
- **Product Backlog:** es una manera de llevar el registro y la organización del trabajo. Permanece vigente en todo el ciclo de vida y el dueño del producto es el responsable. (Trigas, 2019)
- **Sprint:** son los ciclos y/o iteraciones en un proyecto Scrum.
 - Tiempo máximo: 30 días.

- **Planificación:** se realiza la planificación a detalle de cada ciclo; esto se hace con el fin de disminuir el riesgo.
 - **Rol del Scrum Master:** lidera, relaciona al equipo con el dueño del proyecto, registra los problemas que se detectaron en la planificación, registra las actividades. (Trigas, 2019)
- **Sprint Backlog:** Convierte las asignaciones al producto ya en funcionamiento.
- **Scrum diario:** Scrum comprende que el proceso es de suma importancia y hay que realizar inspecciones frecuentemente.
 - **Rol del Scrum Master en el Scrum:** dirige reuniones y se enfoca en la atención, aclara dudas, registra y documenta los riesgos para resolverlos luego de la reunión. (Pérez, 2019)
- **Revisión del Sprint:** es sobre la presentación del producto que se ha desarrollado por el equipo de software. Su enfoque es la detección de inconformidades del producto; a esta presentación asiste el Scrum Master, el dueño del producto y los usuarios. (Pérez, 2019)
- **Retrospectiva:** última reunión en la que concluyen el proyecto.



Figura n.º 4: Procesos de la metodología SCRUM- Autor: Adrián Salazar

Ventajas y Desventajas de la Metodología SCRUM

Ventajas

- Promueve el trabajo en equipo
- Logra rápidos resultados
- Es una metodología flexible
- Fija metas medibles, de esta manera mantiene a su equipo enfocados todo el tiempo y beneficia a la productividad.
- Presenta total visión del proyecto

Desventajas

- Al usar SCRUM el equipo debe tener una profunda comprensión de los principios, marco teórico, roles. De lo contrario, se pueden producir conflictos y roles divididos.
- Las actividades y los periodos siempre deben estar bien definidos.
- No existe espacio para tareas sin finalizar
- No es recomendable tener gente sin experiencia
- Funciona bien cuando se labora con grupos pequeños.

Cuadro comparativo de la Metodología SCRUM y RUP

CRITERIOS	RUP	SCRUM
Clasificación	Metodología Ágil	Metodología Tradicional o Rígida
Enfoque	Iterativa	Iterativa
Ciclo	Está conformado por cuatro fases: inicio, elaboración, construcción, transición o cierre. Ciertos flujos de trabajo se realizan de manera concurrente.	Cada iteración es un ciclo completo
Plan	Está basada en una planificación de proyecto formal, asociado a varios sprints. Hay fecha final e hito intermedio.	Por cada iteración hay una fecha de entrega. Cada planeación del siguiente sprint se fija al final del sprint actual.
Alcance	Lo predefine antes de iniciar el proyecto y se registra en el documento de alcance.	Sustituye el alcance por una cartera de proyectos que se reevalúa al finalizar cada iteración.
Elementos	Roles: Analistas, Desarrolladores, Gestores, Apoyo, Especialista en Pruebas. Actividades, productos/Artefacto, Flujos de Trabajo.	Roles: Dueño del producto Poda de requerimientos, Product Backlog, Sprint, Valores.
Tipo de proyectos	Dirigida especialmente para grandes proyectos, que se ejecutan a largo plazo. A nivel empresarial se enfoca en proyectos con mediana y alta complejidad.	Dirigida a proyectos que requieren rápidas mejoras y para empresas que no tengan límite de tiempo. Se ajustan a proyectos con requisitos flexibles y con tendencia a cambios.
Documentación	Bastante Documentación	Poca Documentación
Énfasis	Su enfoque es hacia los procesos	Su enfoque es hacia las personas
Artefactos	Paquete formal de requerimientos funcionales, documento de arquitectura del software, planeación de desarrollo, pruebas, ect.	Su producto formal es el SO
Esquema	Estudia, Diseña, implementa y documenta sistemas orientados a objetos.	Gestiona y desarrolla sistemas basados en procesos iterativos e incrementales.
Tipos de Desarrollo	Proceso iterativo incremental por fases: inicio, elaboración, construcción, transición o cierre.	Desarrollo simple, complementado con trabajo duro. Se maneja empíricamente y se adapta a conforme avanza el proyecto.
Tipos de Revisión	Por cada fase se realiza uno o varios sprints, con el fin de perfeccionar los objetivos. Si no termina una fase, no se avanza con la siguiente.	Se realiza revisiones diarias: 1. Se revisa el trabajo realizado un día antes. 2. Se revisa el trabajo previsto a realizar. 3. Se revisa todo lo que puede suceder: riesgos, impedimentos.
Objetivos	Dirigida a todos los entornos y fases de desarrollo.	Dirigida a proyectos complejos: Resultados Rápidos, requerimientos cambiantes, innovación y competitividad.

Figura n.º 5: Cuadro Comparativo de las Metodologías RUP y SCRUM- Autor:

Ronaldo Rodríguez

CONCLUSIÓN

Con base en las características de las metodologías de Software que sirvieron de objeto en el presente estudio de comparación, se ha llegado a la conclusión que la metodología RUP (Rational Unified Process) es una metodología que posee excelentes prácticas en el entorno de desarrollo de Software; y esto se debe a su adaptación a grandes y complejos proyectos. Sin embargo, no cuenta con la misma adaptación a pequeños y medianos proyectos. Es ahí donde se involucra la metodología SCRUM y de alguna manera saca ventaja y se convierte en una excelente opción para implementar en proyectos, debido a su sencilla adaptación en cualquier parte del proceso. También permite que los resultados se visualicen más rápido, mientras que la metodología RUP solo lo permite cuando se finaliza el proyecto.

Las dos metodologías de Software: RUP y SCRUM son de gran importancia al momento de implementarlas en un proyecto, pero cada una tiene su enfoque y cumple funciones diferentes, sin embargo, eventualmente podrían “trabajar” juntas en el mismo proyecto.

BIBLIOGRAFÍA

- BEDOYA, L. F. (2018). *PERSONALIZACIÓN DE RUP PARA PROYECTOS ACADÉMICOS DE*. Obtenido de https://repository.eafit.edu.co/bitstream/handle/10784/2754/TabaresBedoya_LuisFelipe_2011.pdf?sequence=1
- CHACÓN, J. C. (2018). *APLICACIÓN DE LA METODOLOGÍA RUP PARA EL DESARROLLO RÁPIDO DE APLICACIONES BASADO EN EL ESTÁNDAR J2EE*. Obtenido de http://clasescun.pbworks.com/f/Metodolog%C3%ADa_RUP.pdf
- Dimes, T. (2021). *Conceptos básicos de Scrum: Desarrollo de software Agile y manejo de proyectos*. Obtenido de https://www.google.com.ec/books/edition/Conceptos_B%C3%A1sicos_De_Scrum_Desarrollo_D/ETuXBgAAQBAJ?hl=es&gbpv=1&dq=roles+de+SCRUM&printsec=frontcover
- Estrada, M. V., Saltos, P. R., & Núñez, J. A. (2021). *Revisión Sistemática de la Metodología Scrum para el Desarrollo de Software*. Obtenido de <https://dominiodelasciencias.com/ojs/index.php/es/article/view/2429/0>
- Fuentes, J. R. (2018). *Desarrollo de Software Ágil*. Obtenido de https://www.google.com.ec/books/edition/Desarrollo_de_Software_%C3%81gil/TxRpCwAAQBAJ?hl=es&gbpv=1&dq=METODOLOGIA+RUP+Y+SCRUM&printsec=frontcover
- González, P. C. (2021). *ESTUDIO SOBRE LA ADMINISTRACIÓN EFECTIVA DE PROYECTOS DE SOFTWARE, UTILIZANDO LA METODOLOGÍA SCRUM,*

DIRIGIDO A EMPRESAS DE SISTEMAS. Obtenido de <https://biblioteca-farmacia.usac.edu.gt/tesis/MAIES232.pdf>

López, N. B. (2019). *Monografía análisis e identificación de metodologías ágiles y procedimientos específicos para la construcción del software*. Obtenido de https://www.google.com.ec/books/edition/Monograf%C3%ADa_an%C3%A1lisis_e_identificaci%C3%B3n/xm1-tAEACAAJ?hl=es

Maida, E., & Pacienza, J. (2020). *Metodologías de desarrollo de software*. Obtenido de <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>

Mancha, J. V. (2021). *Desarrollo e implementación de una aplicación web para la mejora de los procesos de compra y venta de la Distribuidora Plasduit*. Obtenido de <https://repositorio.uncp.edu.pe/handle/20.500.12894/6748>

Menéndez, R. E. (2020). *Metodologías Ágiles de Desarrollo de Software Aplicadas a la Gestión de Proyectos Empresariales*. Obtenido de <https://core.ac.uk/download/pdf/80296686.pdf>

Patiño, E. (2018). *Juguemos con regletas Cuisenaire*. Obtenido de <http://funes.uniandes.edu.co/10902/1/Patino2012JURECU.pdf>

Pérez, A. (2019). *Cuatro enfoques metodológicos para el desarrollo de Software RUP – MSF – XP - SCRUM*. Obtenido de <https://revistas.uniminuto.edu/index.php/Inventum/article/view/9>

Piñero, J. M. (2022). *ENTORNOS DE DESARROLLO*. Obtenido de https://www.google.com.ec/books/edition/Entornos_de_desarrollo/WYd3EAAA

QBAJ?hl=es&gbpv=1&dq=METODOLOGIA+RUP+Y+SCRUM&pg=PA43&printsec=frontcover

Trigas, M. (2019). *Metodología Scrum*. Obtenido de

<https://openaccess.uoc.edu/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>

Vera, D., Córdova, L., & López, R. (2019). *Análisis de la metodología RUP en el*

desarrollo de software académico mediante la herramienta DJANGO. Obtenido de <https://recimundo.com/index.php/es/article/view/486>

ANEXOS

Metodologías con las que trabaja la empresa desarrolladora de software SIPECOM en la ciudad de Guayaquil



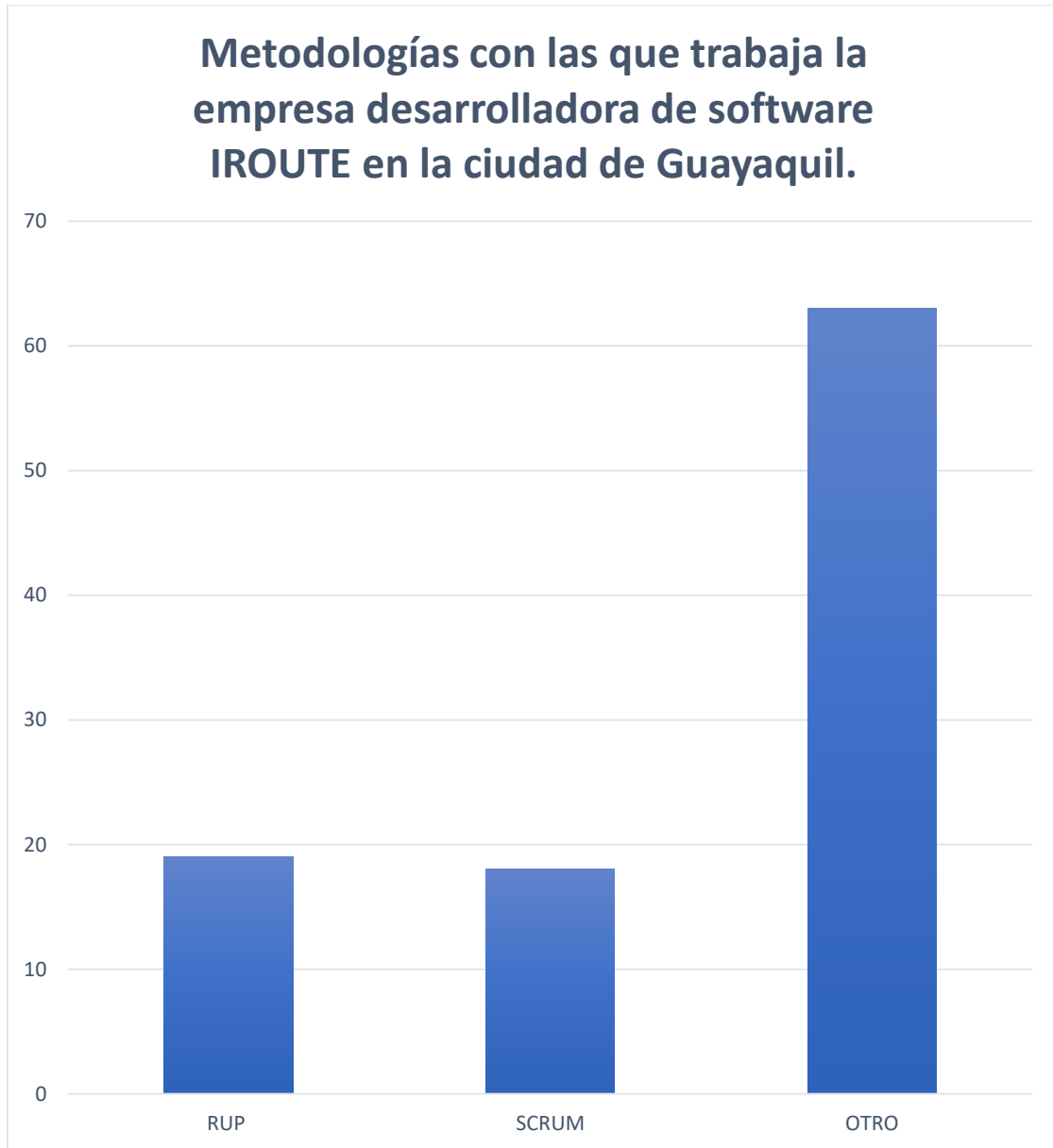
La metodología SCRUM es la más usada para el desarrollo de software en la empresa SIPECOM ubicada en la ciudad de Guayaquil, con un porcentaje de 57%.

Metodologías con las que trabaja la empresa desarrolladora de software SEVEN MARKETING DIGITAL en la provincia de SANTA ELENA.



La metodología SCRUM es la más usada para el desarrollo de software en la empresa SEVEN MARKETING DIGITAL ubicada en la provincia de Santa Elena, con un porcentaje de 46%.

Metodologías con las que trabaja la empresa desarrolladora de software IROUTE en la ciudad de Guayaquil.



Otras son las metodologías utilizadas por la empresa desarrolladora de software IROUTE ubicada en la ciudad de Guayaquil, con un porcentaje de 63%.