Profundizar en técnicas y herramientas relacionadas con la mejora de la calidad del software

Profesor: Luis Pedraza Gomara

Ingeniería del software avanzada   
actividad grupal

Calidad del software



Marta Barbeito De La Rocha

Ernesto González Pradas

ernesto.gonzalez023@comunidadunir.net

# Índice

Contenido

[Índice 1](#_Toc105009669)

[Introducción 2](#_Toc105009670)

[Desarrollo de una calculadora con TDD 3](#_Toc105009671)

[Representación continua del CMMI 5](#_Toc105009672)

[Gestión de requisitos (REQM) 5](#_Toc105009673)

[Gestión integrada del proyecto (IPM) 8](#_Toc105009674)

[Bibliografía 12](#_Toc105009675)

[Hoja de control 13](#_Toc105009676)

# Introducción

En esta actividad grupal, profundizaremos en las distintas técnicas y herramientas relacionadas con la mejora de la calidad del proceso de software. Esto lo llevaremos a cabo mediante el desarrollo dirigido por pruebas, donde dichas pruebas se desarrollan antes de implementar el código con su funcionalidad.

La actividad la realizaremos en dos partes. La primera, desarrollando una calculadora científica empleando la técnica TDD y la segunda reflexionado y aplicando el modelo CMMI.

## Desarrollo de una calculadora con TDD

Para realizar esta actividad grupal, hemos utilizado Python como lenguaje de programación principal. Para la elaboración de las pruebas unitarias, nos hemos basado en la utilización de la librería unittest de Python y como contorl de versiones hemos utilizado Git.

A continuación, se adjunta el enlace al repositorio donde está alojado el código:

<https://github.com/ErnestoGPradas/ActividadGrupalISA>

La calculadora junto con sus pruebas la hemos estructurado de la siguiente manera:

* En el archivo ***CalculadoraCientifica.py***: tenemos las implementaciones de cada uno de los métodos (suma, resta, multiplicación, división, raíz cuadrada y raíz cuadrada con el método de aproximación babilónico).
* En el archivo ***pruebas\_Unitarias.py***: tenemos todas las pruebas realizadas para cada uno de los métodos propuestos.

Además, en el archivo *CalculadoraCientifica.py* tenemos la calculadora con su interfaz de consola, con la que podremos interactuar por medio de dicha consola (Mendoza, 2020).

De esta parte hemos cogido la idea de preguntarle al usuario de la calculadora que operación quiere realizar y después en función de dicha operación se le solicita que introduzca uno o dos números para poder operar. En el tutorial el flujo de trabajo no es el mismo, pero si nos hemos basado en la estructura.

Por otro lado, hemos elaborado dos métodos para calcular la raíz cuadrada de cualquier número. El primer método es:

**def** **raiz\_cuadrada**(a):

**if** **not** isinstance(a, int) **and** **not** isinstance(a, float):

**raise** **TypeError**

**return** pow(a, **0.5**)

Al final calcular la raíz cuadrada de cualquier número se puede calcular elevando dicho número a 0.5. Esto es exactamente lo que hace la función pow() (DelftStack, 2021). El segundo método elaborado para calcular la raíz cuadrada ha sido:

**def** **raiz\_cuadradaBabilonico**(a):

**if** **not** isinstance(a, int) **and** **not** isinstance(a, float):

**raise** **TypeError**

x = a / **2**

**for** i **in** range(**20**):

**if** x \* x == a:

**return** x

**else**:

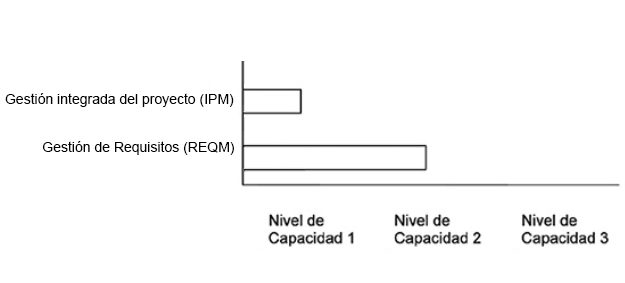
x = (x + (a/x)) / **2**

**return** x

Esta función, está basada en el método de aproximación babilónico para el cálculo de raíces cuadradas (Manuel González, 2018).

Como hemos mencionado en párrafos anteriores, la parte de test unitarios de cada una de las funciones, lo hemos realizado con la librería *unittest* (Python, 2022). Hemos utilizado esta librería por que nos parecía la forma más sencilla de implementar TDD con el lenguaje de programación elegido.

## Representación continua del CMMI

A continuación, vamos a construir una representación continua, según CMMI, teniendo en cuenta que partimos de la base de que todas las áreas de proceso de la organización están en nivel de capacidad “incompletos”, para las siguientes áreas de trabajo:

### Gestión de requisitos (REQM)

En el área de proceso de gestión de requisitos se pide llegar al nivel de capacidad “gestionado” alcanzando la meta genérica “GG2”, para ello es necesario primeramente alcanzar el nivel “realizado” y la meta genérica “GG1”, la cual indica que para lograrla hay que lograr las metas específicas del área de proceso. Esta área de proceso tiene una única meta específica:

* **Gestionar los requisitos (SG1)** : cuyo objetivo es gestionar los requisitos y mantener la coherencia entre estos y los planes del proyecto. Esta meta específica cuenta con 5 prácticas específicas desarrolladas a continuación.

SP1.1 Comprender los requisitos: para la subpráctica de establecer criterios para distinguir a proveedores apropiados de requisitos se realizará una matriz de stakeholders para clasificarlos y jerarquizarlos y aquellos con alto poder y, especialmente, alto grado de interés serán los apropiados. Subpráctica: establecer criterios objetivos para la evaluación y aceptación de los requisitos, documentar junto con el cliente la funcionalidad de cada requisito así como aquellos requisitos no funcionales que le apliquen. Subpráctica: analizar los requisitos para asegurar que se cumplen los criterios establecidos, obteniendo los criterios definidos con el cliente y dados los requisitos, el equipo se reunirá para analizar si con esos requisitos se puede alcanzar el cumplimiento de los criterios. Para la subpractica alcanzar una comprensión de los requisitos con los proveedores de requisitos para que los participantes del proyecto puedan comprometerse con ellos: Tras el análisis de los criterios, funcionalidad y requisitos y que todos se alinean, se firmará inicialmente el acuerdo con dichos puntos.

SP1.2 Obtener el compromiso sobre los requisitos: se debe documentar los requisitos iniciales a través del Product Backlog que se acuerda con el Product Owner. Cualquier modificación del Product Backlog será a cargo del PO que podrá añadir, eliminar y realizar cambios a los existentes, siempre y cuanto no sean parte del Sprint en curso. Subpractica negociar y registrar los compromisos: Aunque el PO podría realizar cambios de manera unilateral por interés del negocio, cualquier modificación del PB la tratará previamente con el equipo de desarrollo en la reunión de planificación de cada sprint. Utilizar para ambas subpracticas y la siguiente una herramienta como JIRA

SP1.3 Gestionar los cambios a los requisitos: En esa reunión y después con JIRA

SP1.4 Mantener la trazabilidad bidireccional de los requisitos: a través de la descomposición en PBI y tareas concretas que indicarán tanto clases, como módulos y funciones. Con JIRA se podrá ver la descomposición granular de cada requerimiento desde su idea general como funcionalidad hasta las distintas tareas de programación de clases/funciones/métodos/módulos que permiten obtener dicha funcionalidad.

SP1.5 Asegurar el alineamiento entre el trabajo del proyecto y los requisitos: gracias al PO se podrá implementar esta práctica ya que él con sus conocimientos del negocio y su cercanía al equipo de desarrollo a través de las reuniones podrá asegurarse de que el trabajo sigue lo determinado en los requisitos. A su vez en JIRA al estar determinados los requisitos, sus PBI y sus tareas específicas, y gracias al seguimiento del PO del trabajo en esta herramienta, se podrá asegurar el alineamiento del trabajo con los requisitos establecidos. En cada reunión de inicio de sprint y de revisión de sprint se podrá asegurar lo que se va a hacer/lo que se ha hecho para reasegurarse de que se han cumplido con los requisitos y por lo tanto funcionalidades.

Tras alcanzar el nivel “realizado” a través de la meta genérica GG1 del área de proceso de gestión de requisitos, se continúa con la implementación de la meta genérica GG2: Institucionalizar un proceso gestionado, que permitirá alcanzar el nivel “gestionado” en esta área de proceso. Esta meta genérica consiste en establecer las siguientes prácticas:

GP2.1 Establecer una política de la organización: determinar el uso de prácticas y herramientas en todo proyecto que ayuden a descomponer los requerimientos acordados con todo cliente en tareas específicas que permitan el seguimiento. Desde el uso de herramientas de gestión de equipos y requerimientos como JIRA hasta el uso de hojas de Excel en un SharePoint común donde se van descomponiendo los requisitos y asignado a personas las tareas específicas.

GP2.2 Planificar el proceso: Se tendrá un documento que describa el marco de trabajo Scrum y sus características así como variantes que se puedan llevar a cabo en la organización. Dado que cada proyecto podrá adaptar Scrum a sus necesidades, se establecerá como parte de su plan de proyecto cómo gestionarán los requisitos, su comprensión, acuerdo con los distintos stakeholders, gestión de cambios y trazabilidad.

GP2.3 Proporcionar recursos: Los equipos contarán con herramientas de gestión de requisitos adaptadas a sus preferencias: desde JIRA y Excel serán las dos principales herramientas que permitirán tener una lista de los requerimientos, su descomposición en funcionalidades concretas y determinar las tareas exactas que cada uno realizará para implementarlo, *Algunos ejemplos de recursos son: Herramientas para el seguimiento de los requisitos. Herramientas de trazabilidad.*

GP2.4: En el plan de proyecto se determinará el nombre del Product Owner que ejercerá de responsable del éxito del mismo, y en la realización de cada tarea específica constará en la herramienta de gestión que se utilice el nombre de la persona asignada a la realización de dicha tarea.

GP2.5 Formar al personal: Cuando un nuevo integrante se incorpore a un proyecto recibirá la documentación explicativa del mismo, en concreto en lo referente a gestión de requisitos, contará con las explicaciones que deben plasmarse en el plan de proyecto y contará con cursos de formación de la herramienta o herramientas específicas que se haya escogido en el proyecto. Un miembro del equipo tendrá también sesiones iniciales de formación de la persona recién incorporada para asegurarse de que comprende el alcance y metodología/marco de trabajo del proyecto y resolver posibles dudas.

GP2.6 Controlar los productos de trabajo: A través de las historias de usuario que estarán materializadas en requisitos, y enlazadas con los PBI’s y sus tareas específicas. Los requerimientos en scrum los controla el PO, haciendo seguimiento junto con el equipo en el inicio de cada iteración, así como en las reuniones diarias, para asegurarse de que los requerimientos que se tienen son los necesarios para cumplir con las necesidades del cliente, y que los trabajos que se realizan para cumplir con los requerimientos son los adecuados. Al iniciar cada sprint y determinar los PBI’s. Cada PBI con sus tareas se implementará en el sprint seleccionado, donde se realizarán los test y las pruebas con el cliente, permitiendo un seguimiento exhaustivo. El uso de herramientas digitales que establezcan en forma piramidal las subdivisiones de cada requerimiento desde la historia de usuario hasta sus tareas más específicas.

GP2.7 Identificar e involucrar a las partes interesadas relevantes: En un proceso scrum se tendrá al PO que es el encargado de representar los intereses de los stakeholders comprendiendo sus puntos de vista y necesidades de cara al producto, él será quien les priorice y represente sus intereses y sus opiniones en las reuniones de planificación y revisión del sprint así como las diarias para poder garantizar la comprensión de requisitos y los trabajos que efectivamente se realizan para cumplir con ellos (PBI’s y tares específicas). También en las reuniones de revisión de sprint se podrán hacer partícipes con sus asistencia los diferentes stakeholders que el PO considere que deben asistir.

GP2.8 Monitorizar y controlar el proceso:Herramientas como el Epic Burndown Chart en Jira. Calendario. *Algunos ejemplos de medidas y productos de trabajo utilizados en la monitorización y control son: Volatilidad de los requisitos (porcentaje de requisitos modificados). Calendario de coordinación de los requisitos. Calendario para el análisis de un cambio propuesto a los requisitos.*

GP2.9 Evaluar objetivamente la adherencia: *Algunos ejemplos de actividades revisadas son: Gestionar los requisitos. Asegurar que los planes, los productos de trabajo y los requisitos del proyecto estan alineados.*

GP 2.10: Revisar el estado con el nivel directivo: ---

### Gestión integrada del proyecto (IPM)

Para esta área, se desea alcanzar un nivel de capacidad “realizado”. Para alcanzar este nivel de capacidad, tenemos que lograr la meta genérica “GG1”, la cual nos dice que tenemos que llevar a cabo todo el trabajo necesario para satisfacer las metas específicas asociadas a esta área de proceso, e implementar las prácticas específicas correspondientes.

En el caso de la gestión integrada del proyecto (IPM), las metas específicas son dos:

* **Utilizar el proceso definido del proyecto (SG1)**: el proyecto se lleva a cabo utilizando un proceso definido adapta a partir del conjunto de procesos estándar de la organización.
* **Coordinar y colaborar con las partes interesadas relevantes (SG2):** la coordinación y la colaboración entre el proyecto y las partes interesadas relevantes se llevan a cabo.

Dentro de le meta específica “utilizar el proceso definido del proyecto” tenemos que implementar prácticas específicas y lo vamos a realizar desde el punto de vista de Scrum.

La primera práctica específica sería Establecer el proceso definido del proyecto (SP1.1) y lo haríamos de la siguiente manera:

Primero debemos tener una lista de requisitos gestionada por el Product Owner, ya que él es responsable último del éxito del producto y comprende los puntos de vista de los interesados.

A continuación, debemos conocer el nivel de compromiso de los interesados para saber cuánta retroalimentación vamos a obtener cada vez que se realicen los Sprint Reviews mostrando el incremento realizado por el equipo de desarrollo.

Estableceremos que durante los Sprint no se pueden meter modificaciones y estos serán de cuatro semanas. Utilizaremos artefactos como la pila del Sprint (Sprint Backlog), siendo este el conjunto de PBIs que se descompondrán en tareas específicas y estarán gestionadas por el equipo de desarrollo estimando su esfuerzo en horas.

Todo esto se realizará en el Sprint Planning entre el PO (Product Owner) y el equipo de desarrollo (Team).

La segunda práctica sería Utilizar los activos de proceso de la organización para planificar las actividades del proyecto (SP1.2):

El equipo de desarrollo (Team) acuerdan con el PO el dominio de la aplicación y se establecen los enfoques de diseño y desarrollo. Para llevar a cabo esto se tiene en cuenta la experiencia del Team que suelen ser profesionales autoorganizados, multifuncionales y no hay sub-equipos.

El Scrum Master (SM) ayuda tanto al Team como al PO a maximizar el valor del negocio y planificar la implantación del Scrum junto con la organización.

La tercera práctica sería establecer el entorno de trabajo del proyecto (SP1.3):

Ya que estamos en siguiendo Scrum y uno de sus valores principales es los individuos y sus interacciones frente a procesos y herramientas. El éxito del proyecto depende del equipo siendo la comunicación básica y existiendo flujos de información dentro del equipo y con el entorno dejando que las herramientas concretas pasen a un segundo plano.

La cuarta práctica Integrar los planes (SP1.4):

Se realizan los disintos eventos de Scrum para asegurar que tanto planes como estrategias se cumplen. El Sprint Planning para el arranque de cada iteración, se negocia con el compromiso con el PO y se define el alcance de dicha iteración. Las dailys de 15 min para para monitorizar el proceso de desarrollo, Sprint Review para mostrar las funcionalidades incrementadas en cada iteración con todos los stakeholders y finalmente el Sprint Retrospective para analizar problemas y proponer mejoras entre el equipo y el SM.

Quinta práctica Gestionar el proyecto utilizando planes integrados (SP1.5):

Como se ha visto en la práctica anterior con los distintos eventos de Scrum se pueden monitorizar los riesgos de la interfaz del producto y del proyecto, así como seguir los parámetros de la planificación del proyecto.

También podemos cambiar el calendario con ajustes necesarios después de cada reunión de revisión del Sprint entregado.

Sexta práctica Establecer los equipos (SP1.6):

Como se ha mencionado anteriormente los roles principales serían el Product Owner, el equipo de desarrollo y el Scrum Master. Con estos roles podemos establecer y mantener la visión compartida del proyecto la estructura del equipo.

Séptima práctica Contribuir a los activos de proceso de la organización (SP1.7):

El equipo de desarrollo planifica y gestiona en horas las tareas descompuesto de los PBIs, previamente aceptadas por el PO el cual gestiona, como hemos mencionado más arriba la pila del producto.

Las prácticas específicas que tenemos dentro de “coordinar y colaborar con las partes interesadas” son:

Gestionar la involucración de las partes interesadas relevantes (SP2.1):

El Product Owner conoce los intereses del cliente final y en general de los stakeholders, gestiona la pila de producto y acuerda con el equipo de desarrollo que va a entrar en cada Sprint para que este pueda mostrarlo al final de dicho Sprint a todos los interesados. Entre medias tenemos al Scrum Master que se encargará de ayudar al PO a entender la agilidad y maximizar el valor de negocio y ayudar a la organización a entender que iteraciones tiene con el equipo aportan valor.

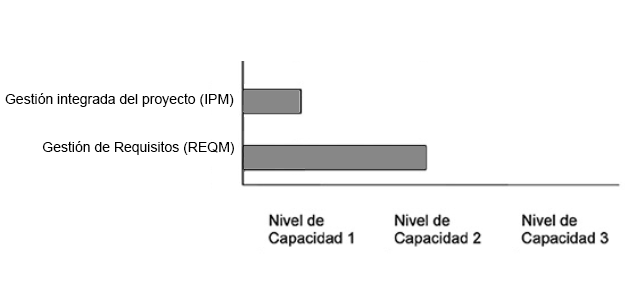
Gestionar las dependencias (SP2.2):

Participar con las partes interesadas relevante para identificar, negociar y seguir las dependencias críticas. Esto se realiza de forma exhaustiva en los cuatro eventos de Scrum definidos con anterioridad.

Resolver las cuestiones de coordinación (SP2.3):

Esta práctica específica sería muy parecida a la anterior ya que intervendrían los mismos roles en los mismos eventos de Scrum resolviendo las cuestiones con las partes interesadas relevantes en cada una de dichas reuniones.

Finalmente quedaría el gráfico de la siguiente forma:



# Bibliografía

DelftStack. (11 de Marzo de 2021). *DelftStack*. Obtenido de https://www.delftstack.com/es/howto/python/square-root-in-python/

Manuel González. (19 de Noviembre de 2018). Python - Nivel 18 - Reto 4 - Método babilónico para calcular raices cuadradas [vídeo]. Youtube. Obtenido de https://www.youtube.com/watch?v=8J4E6xptivk

Mendoza, J. (6 de Octubre de 2020). *Estadísticamente*. Obtenido de https://estadisticamente.com/programar-calculadora-python/

Python. (01 de Enero de 2022). *docs.python*. Obtenido de https://docs.python.org/3/library/unittest.html

# Hoja de control

|  |  |  |  |
| --- | --- | --- | --- |
| **HOJA DE CONTROL ACTIVIDAD GRUPAL**  Nombre y apellidos del primer miembro del equipo: Marta | | | |
| Asistencia a reuniones de equipo por cada integrante  (se incluirá una línea por cada miembro del equipo recogiendo su nombre) | **Marcar con una X lo que proceda** | | |
| Asistencia a una reunión o ninguna  xxx | Asistencia a dos reuniones  X | Asistencia a tres reuniones  xxx |
| Tareas o entregas a realizadas por cada integrante (se incluirá una línea por cada miembro del equipo recogiendo su nombre) | Ninguna o una tarea  xxx | Dos tareas  Xxxxxxx | Tres tareas  xxx |
| **OBSERVACIONES** |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **HOJA DE CONTROL ACTIVIDAD GRUPAL**  Nombre y apellidos del primer miembro del equipo: Ernesto | | | |
| Asistencia a reuniones de equipo por cada integrante  (se incluirá una línea por cada miembro del equipo recogiendo su nombre) | **Marcar con una X lo que proceda** | | |
| Asistencia a una reunión o ninguna  xxx | Asistencia a dos reuniones  Xxxx | Asistencia a tres reuniones  xxx |
| Tareas o entregas a realizadas por cada integrante (se incluirá una línea por cada miembro del equipo recogiendo su nombre) | Ninguna o una tarea  xxx | Dos tareas  Xxxx | Tres tareas  xxx |
| **OBSERVACIONES** |  |  |  |