

PROGRAMA de Calidad del Software

Carreras: Tecnicatura Universitaria en Programación Informática / Licenciatura en Informática

Asignatura: Calidad del Software

Núcleo al que pertenece: Orientación

Profesor: Pablo Tobia

Asignaturas Correlativas: Programación con Objetos I, Estructuras de Datos, Redes de Computadoras, Sistemas Operativos, Elementos de Ingeniería de Software

Carga horaria: 4 horas

Objetivos

Se espera que quien curse la materia:

- Entienda el concepto de calidad en el desarrollo de software y pueda evaluar su calidad.
- Pueda analizar funcionalidades y diseñar un plan de pruebas a partir de una especificación funcional.
- Pueda definir atributos de calidad, clasificarlos y medirlos en un determinado software para establecer expectativas de calidad.
- Conozca las diferentes estrategias y herramientas tecnológicas para aportar calidad al desarrollo.
- Comprenda la ejecución de los diferentes tipos de test de acuerdo a los atributos de calidad definidos y a la etapa del desarrollo en la que se ejecuta.
- Pueda trabajar en equipo utilizando diferentes canales de comunicación y metodologías de trabajo.
- Reconozca como encontrar errores a través de:
 - El plan de pruebas diseñado
 - La comunicación efectiva con superiores y pares
 - El seguimiento de incidencias
 - Investigación del error para encontrar el defecto
- Sea capaz de implementar un plan de pruebas en diferentes metodologías de trabajo y tecnologías de desarrollo.
- Entienda la diferencia entre tareas de Testing, Control de Calidad (QC) y Aseguramiento de la Calidad (QA) y cómo se complementan en el desarrollo de software.
- Conozca diferentes herramientas que pueden utilizarse para realizar pruebas, reporte de incidencias y automatización de tareas.

Contenidos Mínimos

- Concepto de calidad de procesos y productos, en relación al desarrollo de software

- Necesidad, valor y planes de prueba. Validación y verificación de Sistemas.
- Metodologías de desarrollo y la implementación de las tareas de calidad en cada una.
- Integración Continua y Entrega Continúa
- Modelos de calidad, estándares y certificación.
- Administración de la calidad, roles, tareas y relación con el resto de las actividades del proyecto.

Programa Analítico

- **Unidad 1: Definición del significado de Calidad en el desarrollo de software.**
Calidad de procesos y productos. Conceptos de Error, Falla, Defecto.
- **Unidad 2: Validación y verificación de Sistemas.**
Pruebas funcionales. Procedimientos de prueba. Estilo, dependencias, tamaño, complejidad, cubrimiento de las pruebas. Herramientas
- **Unidad 3: Tipos de prueba.**
Casos de prueba. Reporte de errores. Herramientas para cada uno.
- **Unidad 4: Metodologías de desarrollo.**
UP (Unified Process), XP (Extreme Programming), Lean Programming, Ágiles, TDD, BDD. Frameworks de tipo xUnit. Las pruebas como parte del proceso de desarrollo.
- **Unidad 5: Plan de pruebas y Administración de la calidad.**
Alcance el plan, roles, tareas y su relación con el resto de las actividades del proyecto. Implementación, problemas, ambientes. Análisis de casos, seguimiento de incidencias.
- **Unidad 6: Integración Continua y Entrega Continua.**
Administración del código compartido y su relación con el proceso de desarrollo. Construcción e instalación del software. Pruebas continuas. Herramientas
- **Unidad 7: Modelos marco de referencia.**
Modelos de tipo CMMI y tareas de calidad relacionadas. Certificaciones de calidad.

Bibliografía Obligatoria

- Lisa Crispin, Janet Gregory, Agile Testing, Addison-Wesley Professional, 2009. ISBN-13: 978-0-321-53446-0
- Rex Black, Managing Testing Process, Addison-Wiley; 3rd edition, 2009. ISBN 0-471-22398-0
- Kshirasagas Naik, Priyadarshi Tripathy, Software Testing & Quality Assurance, Wiley-Spektrum; 1st edition, 2002. ISBN 978-0-471-78911-6

Bibliografía de Consulta

- Gerard Meszaros, xUnit Test Patterns: Refactoring Test Code, Addison-Wesley, 2007.

- Paul M. Duvall with Andrew Glover and Steve Matyas, Continuous Integration: Improving Software Quality and Reducing Risk, Addison-Wesley Professional, 2008.

Organización de las clases

Las clases serán en modalidades teórico-práctico, dependiendo del tema a desarrollar, será el volumen de cada modalidad para cada tema.

Desde el inicio de la cursada quienes cursen deberán elegir cuál será el software sobre el que van a trabajar y durante el transcurso de la materia deberán aplicar lo aprendido sobre este, para generar la entrega final.

Se propondrán las herramientas necesarias para poder hacer el seguimiento de su progreso y feedback continuo de sus avances.

Trabajo Práctico

Durante la cursada será elaborado un plan de test completo para un sistema en particular. Los objetivos serán que el plan sea elaborado siguiendo los conceptos teóricos que se aprenden en cada unidad. Sean capaces de evaluar ventajas y desventajas del plan elaborado, de reconocer qué modificar para adaptarlo a una metodología diferente de trabajo para la cual lo pensaron, de cómo incide en la calidad del software y comprender que la calidad es una característica importante y medible de un desarrollo de software.

TP 1. Verificación y validación.

Los objetivos son que comprendan la diferencia entre estos conceptos y desarrollen ejemplos sobre el SUT elegido, teniendo en cuenta quienes serán las entidades/equipos/áreas que interactúan, el soporte y los registros de interacción.

TP 2. Tipos de prueba.

Los objetivos son que identifique los tipos de prueba a realizar de acuerdo a las características de cada uno, del SUT elegido y de los atributos de calidad acordados. Investigar, analizar y seleccionar las herramientas para la implementación de los tipos de prueba elegidos, el soporte para el seguimiento y reportes.

TP 3. Metodologías de desarrollo.

Los objetivos son analizar la implementación de las pruebas en las diferentes metodologías y en particular describir aspectos de la implementación en una metodología acordada. Investigar alternativas de herramientas y proponer una para el soporte y seguimiento de dicha metodología.

TP 4. Plan de Pruebas.

Los objetivos son que se documente y de soporte con las herramientas elegidas al plan de test propuesto, contemplando las propuestas de prácticas anteriores, sumando aspectos de infraestructura necesarios, metodología de entrega y herramientas adicionales para dar soporte al registro, monitoreo y seguimiento.

Modalidad de evaluación

Los mecanismos de evaluación en modalidades libre y presencial de esta asignatura están reglamentados según los siguientes artículos del Régimen de estudios de la UNQ (Res. CS 201/18). En la modalidad de libre se evaluarán los contenidos de la asignatura con un examen escrito, un examen oral, un desarrollo de algunas pruebas automáticas en una tecnología acordada e instancias de evaluación similares a las realizadas en la modalidad presencial.

Cronograma Tentativo

Sem	Tema/Unidad	Actividad Práctica				Evaluación
		Teórico	Res. de prob.	Lab	Otros (especificar)	
1	Conceptos de calidad	x			Debates, casos de ejemplo	
2	Validación y Verificación	x	x		Propuestas de SUT*	
3	Validación y Verificación, enfoque práctico		x		Ejercitación y debate	
4	Tipos de Pruebas	x	x		Elección del SUT.	
5	Tipos de pruebas, enfoque práctico		x		Ejercitación y debate	
6	Metodologías de desarrollo	x	x			
7	Metodologías de desarrollo, enfoque práctico		x		Criterios de elección de la metodología para su trabajo	
8	Plan de pruebas	x				
9	Plan de pruebas.	x	x		Elaboración de propuesta. Alcance	
10	Desarrollo de pruebas automáticas	x	x		Feedback y avances sobre su TP	
11	Automatización de tareas	x	x		Uso de herramientas y avances del TP	
12	Entrega y correcciones					x
13	Continuous Integration	x			Integración del SUT a las herramientas a usar	
14	Continuos Delivery	x			Integración del SUT a las herramientas a	



					usar	
15	Modelos de referencia	x	x		Análisis de su trabajo desde cada modelo	
16	Examen escrito					x
17	Guía de cierre del trabajo		x		Se trabajará con los grupos para generar modelos de entrega de TP.	
18	Entrega final de trabajos					x

*SUT: System Under Test