

Universidad Católica San Pablo
Facultad de Ingeniería y Computación
Escuela Profesional de
Ciencia de la Computación
SILABO



CS391. Calidad de Software (Obligatorio)

2018-I

1. DATOS GENERALES

1.1 CARRERA PROFESIONAL	:	Ciencia de la Computación
1.2 ASIGNATURA	:	CS391. Calidad de Software
1.3 SEMESTRE ACADÉMICO	:	8 ^{vo} Semestre.
1.4 PREREQUISITO(S)	:	CS390. Ingeniería de Software II. (6 ^{to} Sem)
1.5 CARÁCTER	:	Obligatorio
1.6 HORAS	:	2 HT; 2 HP;
1.7 CRÉDITOS	:	3

2. DOCENTE

Prof. Jhon Humberto Cano Chávez

- Prof. Ingeniería de Sistemas, Universidad Católica Santa María, Perú, 2000.

3. FUNDAMENTACIÓN DEL CURSO

Calidad: cómo asegurar y verificar la calidad, y la necesidad de una cultura de calidad. Como proveer patrones de calidad por medio de los estándares y métricas como CMMI, PSP/TSP e ISO. Técnicas de prueba, verificación y validación. Aseguramiento de proceso contra aseguramiento del producto. Estándares de proceso de calidad. Producto y aseguramiento del proceso. Análisis y divulgación del problema. Acercamientos estadísticos al control de calidad.

4. SUMILLA

1. CMMI v 1.2 2. People Software Process & Team Software Process 3. Estándares ISO/IEC 4. Técnicas de Prueba de Software

5. OBJETIVO GENERAL

- Los alumnos deben describir los conceptos fundamentales y comprender la terminología del CMMI.
- Los alumnos discutirán acerca de las 22 áreas de proceso CMMI así como reconocer el valor de este modelo en diferentes casos de estudio.
- Los alumnos deben comprender los conceptos fundamentales CMMI para que sean adoptados en los proyectos de software.
- Describir y comprender los conceptos de calidad, las normas de la familia ISO en sus diferentes versiones.
- El alumno debe comprender y aplicar el proceso de pruebas de en software desarrollado así como las estadísticas aplicadas a este proceso.
- El alumno establecerá una metodología de pruebas para el software realizado.

6. CONTRIBUCIÓN A LA FORMACIÓN PROFESIONAL Y FORMACIÓN GENERAL

Esta disciplina contribuye al logro de los siguientes resultados de la carrera:

- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. [Nivel Bloom: 4]
- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. [Nivel Bloom: 4]
- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. [Nivel Bloom: 3]
- f) Comunicarse efectivamente con audiencias diversas. [Nivel Bloom: 3]
- g) Analizar el impacto local y global de la computación sobre los individuos, organizaciones y sociedad. [Nivel Bloom: 4]
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. [Nivel Bloom: 3]
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. [Nivel Bloom: 3]
- k) Aplicar los principios de desarrollo y diseño en la construcción de sistemas de software de complejidad variable. [Nivel Bloom: 3]

7. CONTENIDOS

UNIDAD 1: CMMI v 1.2 (18 horas)

Nivel Bloom: 3

OBJETIVO GENERAL

- Describir los componentes y el contenido del modelo CMMI-DEV y sus relaciones.
- Discutir las 22 áreas de procesos que conforman el modelo.
- Ubicar información relevante en el modelo.

CONTENIDO

- Introducción.
- Conceptos de mejora de procesos y CMMI.
- Visión general a los componentes del modelo CMMI.
- Representaciones del modelo e institucionalización.
- Desarrollo del producto parte 1.
- Gestionando el proyecto.
- Soporte al proyecto y a la organización.
- Desarrollo del producto parte 2.
- Infraestructura de mejora.
- Gestionando cuantitativamente.
- Soportando ambientes complejos.
- Integrando los temas tratados.
- Siguiendo pasos.
- Resumen.

Lecturas: [Team, 2006], [Chrissis et al., 2007], [Kulpa and Johnson, 2008]

UNIDAD 2: People Software Process & Team Software Process (12 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ En esta unidad se revisará el PSP como una herramienta de mejora del desempeño personal de los desarrolladores de software y cómo éstos pueden convertirse en un equipo de alto desempeño usando TSP. ▪ Se explicará la relación que existe entre PSP/TSP y CMMI. 	<ul style="list-style-type: none"> ▪ Fundamentos. ▪ Conceptos básicos de PSP. ▪ Medición de tamaño y estimación. ▪ Creación y seguimiento de planes de proyecto. ▪ Planificación y seguimiento de calidad de software. ▪ Diseño de software. ▪ Extensiones de proceso y personalizaciones. ▪ Conceptos básicos de TSP. ▪ Relaciones entre PSP/TSP y CMMI.
Lecturas: [Pomeroy-Huff et al., 2005], [Humphrey, 1995], [Humphrey, 1997], [Humphrey, 2000], [Humphrey, 2001], [Humphrey, 2005], [Humphrey, 2006b], [Humphrey, 2006a]	

UNIDAD 3: Estándares ISO/IEC (18 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Brindar a los participantes comprensión de los conceptos relacionados con la calidad, y con las normas de la familia ISO 9000, en sus diferentes versiones (la normas ISO 9001:2001, especificidades de la norma ISO 9000-3 para el caso del diseño, desarrollo, suministro, instalación y mantenimiento de software de computación y aplicación de estos conceptos y técnicas; las normas ISO/IEC 9126, ISO/IEC 12207, ISO/IEC 15939, ISO/IEC 14598, ISO/IEC 15504-SPICE, IT Mark, SCRUM, SQuaRE y CISQ, su utilización, etc.) 	<ul style="list-style-type: none"> ▪ ISO 9001:2001. ▪ ISO 9000-3. ▪ ISO/IEC 9126. ▪ ISO/IEC 12207. ▪ ISO/IEC 15939. ▪ ISO/IEC 14598. ▪ ISO/IEC 15504-SPICE. ▪ IT Mark. ▪ SCRUM. ▪ SQuaRE. ▪ CISQ.
Lecturas: [Peach, 2002], [Carroll and Daughtrey, 2007], [Schulmeyer, 2008], [Khan et al., 2006]	

UNIDAD 4: Técnicas de Prueba de Software (12 horas)	
Nivel Bloom: 4	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Elaborar planes de prueba y planes de calidad en sus proyectos de desarrollo. ▪ Aplicar técnicas de pruebas formales para la generación de casos de prueba. ▪ Definir las técnicas de prueba a aplicar, según los requerimientos de cada aplicación. ▪ Desarrollar un plan para implantar una metodología de pruebas en la organización. 	<ul style="list-style-type: none"> ▪ Introducción ▪ Estadísticas relativas al proceso de pruebas. ▪ Estándares relativos a la prueba de software. ▪ El proceso de pruebas. 1. Principios de prueba. 2. El plan de calidad. 3. El plan de pruebas. 4. Técnicas de Verificación. ▪ Software CAST (<i>Computer Aided Software Testing</i>). ▪ Una metodología de pruebas.
Lecturas: [Wang and King, 2000], [Farrell-Vinay, 2008]	

8. METODOLOGÍA
<p>El profesor del curso presentará clases teóricas de los temas señalados en el programa propiciando la intervención de los alumnos.</p> <p>El profesor del curso presentará demostraciones para fundamentar clases teóricas.</p> <p>El profesor y los alumnos realizarán prácticas</p> <p>Los alumnos deberán asistir a clase habiendo leído lo que el profesor va a presentar. De esta manera se facilitará la comprensión y los estudiantes estarán en mejores condiciones de hacer consultas en clase.</p>

9. EVALUACIONES
<p>Evaluación Permanente 1 : 20 %</p> <p>Examen Parcial : 30 %</p> <p>Evaluación Permanente 2 : 20 %</p> <p>Examen Final : 30 %</p>

Referencias

- [Carroll and Daughtrey, 2007] Carroll, S. and Daughtrey, T. (2007). *Fundamental Concepts for the Software Quality Engineering Volume 2*. American Society for Quality Press, 2nd edition.
- [Chrissis et al., 2007] Chrissis, M. B., Konrad, M., and Shrum, S. (2007). *CMMI Guidelines for Process Integration and Product Improvement*. Addison-Wesley, 2nd edition.
- [Farrell-Vinay, 2008] Farrell-Vinay, P. (2008). *Manage Software Testing*. Auerbach Publications, Taylor & Francis Group.
- [Humphrey, 1995] Humphrey, W. S. (1995). *A Discipline for Software Engineering*. Addison-Wesley, 1st edition.
- [Humphrey, 1997] Humphrey, W. S. (1997). *Introduction to the Personal Software Process*. Addison-Wesley, 1st edition.
- [Humphrey, 2000] Humphrey, W. S. (2000). *Introduction to the Team Software Process*. Addison-Wesley, 1st edition.
- [Humphrey, 2001] Humphrey, W. S. (2001). *Winning with Software: An Executive Summary*. Addison-Wesley, 1st edition.

- [Humphrey, 2005] Humphrey, W. S. (2005). *PSP: A Self-Improvement Process for Software Engineers*. Addison-Wesley, 1st edition.
- [Humphrey, 2006a] Humphrey, W. S. (2006a). *TSP: Coaching Development Teams*. Addison-Wesley, 1st edition.
- [Humphrey, 2006b] Humphrey, W. S. (2006b). *TSP: Leading a Development Team*. Addison-Wesley, 1st edition.
- [Khan et al., 2006] Khan, R., Mustafa, K., and Ahson, S. (2006). *Software Quality: Concepts and Practice*. Alpha Science Intl Ltd.
- [Kulpa and Johnson, 2008] Kulpa, M. K. and Johnson, K. A. (2008). *Interpreting the CMMMI a Process Improvement Approach*. CRC Press Taylor & Francis Group, 2nd edition.
- [Peach, 2002] Peach, R. W. (2002). *The ISO 9000 Handbook*. QSU Publishing Company, 4th edition.
- [Pomeroy-Huff et al., 2005] Pomeroy-Huff, M., Mullaney, J., Cannon, R., and Sebum, M. (2005). *The Personal Software Process PSP Body of Knowledge*. CMU/SEI-2005-SR-003, 1st edition.
- [Schulmeyer, 2008] Schulmeyer, G. G. (2008). *Handbook of Software Quality Assurance*. Artech House Inc., 4th edition.
- [Team, 2006] Team, C. P. (2006). *CMMI for Development Version 1.2*. CMU/SEI-2006-TR-2006-008.
- [Wang and King, 2000] Wang, Y. and King, G. (2000). *Software Engineering Processes: Principles and Applications*. CRC Press.