

**SCHEDA DATI PER OFFERTA FORMATIVA PUBBLICA DI CUI AL PUNTO 1.2 DELLA
CIRCOLARE MINISTERIALE N° 187 DELL'11 GIUGNO 2008**

Insegnamento: Docente titolare: Qualifica SSD di appartenenza Struttura di afferenza Telefono e-mail Orario di ricevimento Sito web docente	Ingegneria del Software Michele Marchesi Professore 1° fascia ING-INF/05 – Sistemi di elaborazione delle informazioni Dipartimento di Ingegneria Elettrica ed Elettronica, Università degli Studi di Cagliari 070 675 5757 michele@diee.unica.it martedì 13-14, mercoledì 14-15 http://agile.diee.unica.it/agroup/it/marchesi.wp
Curriculum scientifico	Michele Marchesi è professore ordinario di Sist. di Elaboraz. delle Informazioni ed è coordinatore del gruppo di ricerca di ingegneria del software. E' autore di circa 200 pubblicazioni internazionali ed è stato ed è coordinatore di vari progetti di ricerca italiani ed europei, per oltre 2.000.000 Euro. Principali pubblicazioni: A. Corana, M. Marchesi, C. Martini, S. Ridella, “Minimizing Multimodal Functions of Continuous Variables with the 'Simulated Annealing' Algorithm” ACM Trans. on Mathematical Software, v. 13, September 1987. T.Lux, M. Marchesi, “Scaling and criticality in a stochastic multi-agent model of a financial market”, Nature, v. 397, 2 February 1999. G. Succi, M. Marchesi, Extreme Programming Examined, Addison-Wesley, 2001. G. Concas, M. Locci, M. Marchesi, S. Pinna, I. Turnu, “Fractal dimension in software networks”, Europhysics Lett., v.76, December 2006. G. Concas, M. Marchesi, S. Pinna, N. Serra, “Power-Laws in a Large Object-Oriented Software System”, IEEE Trans. Sw. Eng, v. 33, October 2007.
Contenuto schematico del corso di insegnamento	Approfondimento, anche pratico, di strumenti e metodologie dell'ingegneria del software per la pianificazione dello sviluppo, realizzazione e valutazione di applicazioni software complesse, in particolare legate allo sviluppo di software ad oggetti.
Obiettivi formativi e risultati attesi (secondo i descrittori di Dublino)	<p>Conoscenza e capacità di comprensione: lo studente conoscerà i fondamenti dell'ingegneria del software come scienza che studia strumenti e metodologie per la pianificazione dello sviluppo, realizzazione e valutazione di applicazioni software complesse..</p> <p>Capacità di applicare la conoscenza e capacità di comprensione: lo studente sarà in grado di comprendere i principi e l'organizzazione dei moderni processi di produzione del software, di effettuare analisi ad oggetti di semplici sistemi e di pianificarne lo sviluppo usando i principi delle metodologie agili.</p> <p>Autonomia di giudizio: lo studente saprà valutare l'adeguatezza dei processi per lo sviluppo del software rispetto a specifici progetti.</p> <p>Abilità comunicative: lo studente sarà in grado di dialogare con specialisti informatici sulla pianificazione dello sviluppo e del testing di sistemi informativi. Sarà inoltre in grado di comunicare e di leggere modelli di analisi ad oggetti utilizzando il linguaggio</p>

	<p>UML.</p> <p>Capacità di apprendere autonomamente: lo studente sarà in grado di apprendere le problematiche e le metodologie relative ai processi di sviluppo del software, nonché i metodi di analisi di sistemi a oggetti, applicando con flessibilità i concetti di base forniti nel corso.</p>
Articolazione del corso	<p>La disciplina dell'Ingegneria del Software (lezioni: 4 ore) Prospettiva storica, impatto economico, temi principali, sfide attuali. Caratteristiche del software come prodotto ingegneristico: corretto, affidabile, efficiente, usabile, robusto, interoperabile, manutenibile, evolvibile, verificabile, riusabile, portabile.</p> <p>Il ciclo di vita del software (lezioni: 6 ore) Le fasi del ciclo di vita del software: raccolta dei requisiti, analisi, progetto, codifica, testing, rilascio, manutenzione. I principali modelli del processo di sviluppo: "waterfall", incrementale-iterativo, a spirale, RUP, agile.</p> <p>Analisi e progetto ad oggetti dei sistemi (lezioni: 10 ore; esercitazioni: 10 ore) L'approccio ad oggetti. Analisi e progettazione di sistemi software. La raccolta dei requisiti: casi d'uso e "user stories". Identificazione degli oggetti e delle operazioni. Il metodo "Class, Responsibility, Collaboration" (CRC). Dall'OOA all'OOD. Principi di buona progettazione software.</p> <p>Linguaggio UML (lezioni: 10 ore; esercitazioni: 4 ore) Il linguaggio di modellazione standard UML. Diagrammi dei casi d'uso, diagrammi delle classi, diagrammi di stato, diagrammi sequenziali e di collaborazione, diagrammi di attività.</p> <p>Le metodologie agili (lezioni: 6 ore) L'Agile Manifesto. La programmazione estrema: motivazioni, valori e pratiche. Sviluppo incrementale-iterativo e "Planning Game". Programmazione a coppie, refactoring, "Test First Programming".</p>
Propedeuticità	Fondamenti di Informatica 1
Anno di corso e semestre	1° anno/ 2° sem.
Testi di riferimento	<p>Lucidi del corso, reperibili su Internet nel sito: agile.diee.unica.it/corsi.</p> <p>M. Fowler, UML Distilled-Terza Edizione, Pearson Education Italia, 2004.</p>
Modalità di erogazione dell'insegnamento	Tradizionale
Modalità di frequenza	Facoltativa
Metodi di valutazione	Prova scritta
Organizzazione della didattica	50 ore, di cui 36 ore di lezione e 14 ore di esercitazione