

ΠΕΡΙΓΡΑΜΜΑ ΜΑΘΗΜΑΤΟΣ

(1) ΓΕΝΙΚΑ

ΣΧΟΛΗ	Μηχανικών		
ΤΜΗΜΑ	Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών		
ΕΠΙΠΕΔΟ ΣΠΟΥΔΩΝ	Προπτυχιακό		
ΚΩΔΙΚΟΣ ΜΑΘΗΜΑΤΟΣ	5.002	ΕΞΑΜΗΝΟ ΣΠΟΥΔΩΝ	5 ^ο
ΤΙΤΛΟΣ ΜΑΘΗΜΑΤΟΣ	Τεχνολογία Λογισμικού		
ΑΥΤΟΤΕΛΕΙΣ ΔΙΔΑΚΤΙΚΕΣ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ	ΕΒΔΟΜΑΔΙΑΙΕΣ ΩΡΕΣ ΔΙΔΑΣΚΑΛΙΑΣ	ΠΙΣΤΩΤΙΚΕΣ ΜΟΝΑΔΕΣ	
Διαλέξεις και Ασκήσεις Πράξης	4	4	
Εργαστηριακές Ασκήσεις	1	2	
ΣΥΝΟΛΟ	5	6	
ΤΥΠΟΣ ΜΑΘΗΜΑΤΟΣ	Ειδικού υποβάθρου / Κορμού		
ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ ΜΑΘΗΜΑΤΑ			
ΓΛΩΣΣΑ ΔΙΔΑΣΚΑΛΙΑΣ και ΕΞΕΤΑΣΕΩΝ	ΕΛΛΗΝΙΚΗ		
ΤΟ ΜΑΘΗΜΑ ΠΡΟΣΦΕΡΕΤΑΙ ΣΕ ΦΟΙΤΗΤΕΣ ERASMUS	Ναι		
ΗΛΕΚΤΡΟΝΙΚΗ ΣΕΛΙΔΑ ΜΑΘΗΜΑΤΟΣ (URL)	https://eclass.hmu.gr/courses/ECE177/		

(2) ΜΑΘΗΣΙΑΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

Μαθησιακά Αποτελέσματα
<p>Στην σημερινή κοινωνία της πληροφορίας το κόστος ανάπτυξης λογισμικού ανεβαίνει ενώ το κόστος του hardware μειώνεται. Οι χρόνοι ανάπτυξης εφαρμογών αυξάνονται και το κόστος συντήρησης των εφαρμογών μεγαλώνει. Τα λάθη των λογισμικών αυξάνονται ενώ τα λάθη του hardware έχουν μειωθεί στο ελάχιστο. Για να μπορέσουμε να λύσουμε τα παραπάνω προβλήματα τα οποία γίνονται όλο και ποιο έντονα θα πρέπει να ακολουθούμε δομημένους τρόπους ανάπτυξης λογισμικού. Η Τεχνολογία Λογισμικού ή αλλιώς “Software Engineering” συμπεριλαμβάνει όλο το φάσμα εργασιών οι οποίες πρέπει να γίνουν για την σωστή υλοποίηση λογισμικού. Οι εργασίες αυτές συνοψίζονται στην Ανάλυση, Σχεδίαση, Υλοποίηση, Επικύρωση & Επαλήθευση, Διαχείριση έργων και Εξέλιξη & Συντήρηση λογισμικού. Το μάθημα της Τεχνολογίας Λογισμικού έχει σαν σκοπό να ασχοληθεί με θέματα που αφορούν όλα τα στάδια του Software Engineering, Ανάλυση-Σχεδίαση-Υλοποίηση-Επικύρωση & Επαλήθευση, σε εισαγωγικό επίπεδο. Ειδικότερα έχει σαν σκοπό να εξοικειώσει το σπουδαστή με <u>βασικές</u> έννοιες, εργαλεία, διαγράμματα, μοντέλα, μεθόδους και μεθοδολογίες ανάλυσης, σχεδίασης και ελέγχου λογισμικού και έτσι να τον εφοδιάσει με <u>βασικές</u> ικανότητες Software Engineering.</p> <p>Με την επιτυχή ολοκλήρωση του μαθήματος ο φοιτητής / τρια θα είναι σε θέση να:</p> <ul style="list-style-type: none"> ▪ Γνωρίζει βασικές έννοιες ανάλυσης, σχεδίασης και ελέγχου λογισμικών συστημάτων. ▪ Αναλύει, σχεδιάζει και να ελέγχει απλά λογισμικά συστήματα. ▪ Εφαρμόζει μεθόδους και εργαλεία υλοποίησης λογισμικού ▪ Συγκρίνει, αξιολογεί και να εξασφαλίζει την ποιότητα στα λογισμικά συστήματα. ▪ Εφαρμόζει τουλάχιστον ένα CASE tool της UML.

- Εφαρμόζει στην πράξη την μεθοδολογία Unified Software Development Process (USDP).

Γενικές Ικανότητες

- Προσαρμογή σε νέες καταστάσεις
- Λήψη αποφάσεων
- Αυτόνομη Εργασία
- Ομαδική Εργασία
- Σχεδιασμός και Διαχείριση Έργων

(3) ΠΕΡΙΕΧΟΜΕΝΟ ΜΑΘΗΜΑΤΟΣ

Ενότητες Θεωρητικών Διαλέξεων

- **Εισαγωγή** στο software engineering και επεξήγηση της σπουδαιότητας του. Ορισμός των βασικότερων όρων του software engineering Εισαγωγή σε θέματα δεοντολογίας και επαγγελματισμού και σε τι αφορούν το software engineers
- **Systems Engineering:** Πως το software engineering επηρεάζεται από θέματα του ευρύτερου τομέα του system engineering Γιατί το περιβάλλον συστήματος θα πρέπει να εμπλακεί κατά τις διαδικασίες σχεδίασης του συστήματος.
- **Διαδικασίες Λογισμικού:** Παρουσίαση και περιγραφή των μοντέλων διαδικασιών λογισμικού. Περιγραφή πως μπορούν να χρησιμοποιηθούν τα μοντέλα διαδικασιών στην καταγραφή απαιτήσεων, στο σχεδιασμό, στην ανάπτυξη, στο testing και στην εξέλιξη του συστήματος. Παρουσίαση της τεχνολογίας CASE και περιγραφή πως αυτή μπορεί να βοηθήσει στα διάφορα στάδια του software engineering.
- **Απαιτήσεις Λογισμικού:** Παρουσίαση των εννοιών «Απαιτήσεις Χρήστη» και «Απαιτήσεις Συστήματος». Περιγραφή των λειτουργικών και μη λειτουργικών απαιτήσεων. Εξήγηση του πως δομείται το έγγραφο περιγραφής των απαιτήσεων λογισμικού
- **Requirements Engineering Processes:** Περιγραφή των βασικών σταδίων του requirements engineering. Παρουσίαση των τεχνικών για την ανάλυση των απαιτήσεων. Περιγραφή της διαδικασίας επικύρωσης των απαιτήσεων. Περιγραφή της διαδικασίας διαχείρισης των απαιτήσεων
- **Μοντέλα Συστήματος:** Γιατί το περιεχόμενο του συστήματος πρέπει να σχεδιαστεί με μοντέλα κατά την διαδικασία του requirements engineering. Περιγραφή του behavioural modelling, data modelling και object modeling. Παρουσίαση κάποιων σημειογραφιών που χρησιμοποιούνται στο Unified Modelling Language (UML). Παρουσίαση των τρόπων που τα CASE workbenches υποστηρίζουν το system modelling
- **Software Prototyping:** Περιγραφή της χρήσης των prototypes σε διάφορους τύπους έργων. Περιγραφή των εξελικτικών prototypes και των throw-away prototyping. Περιγραφή από 3 τεχνικές γρήγορου prototyping, ανάπτυξη με την χρήση γλώσσας προγραμματισμού ανώτερου επιπέδου και επαναλαμβανόμενη χρήση υπαρχόντων κομματιών κώδικα. Εξήγηση της ανάγκης του user interface prototyping
- **Formal Specification:** Εξήγηση των τεχνικών του formal specification και πως αυτές βοηθούν στην ανεύρεση προβλημάτων στις απαιτήσεις συστήματος. Περιγραφή της χρήσης τεχνικών άλγεβρας στον προσδιορισμό του interface. Περιγραφή της χρήσης τεχνικής βασισμένης σε μοντέλα για τον προσδιορισμό της συμπεριφοράς του συστήματος
- **Αρχιτεκτονικός Σχεδιασμός:** Παρουσίαση του αρχιτεκτονικού σχεδιασμού και της σπουδαιότητας του. Εξήγηση του γιατί χρειάζονται πολλαπλά μοντέλα για να τεκμηριωθεί η αρχιτεκτονική του λογισμικού. Περιγραφή των διαφορετικών τύπων μοντέλων αρχιτεκτονικής. Σύγκριση των αρχιτεκτονικών λογισμικού
- **Αρχιτεκτονική Κατανεμημένων Συστημάτων:** Εξήγηση των πλεονεκτημάτων και των μειονεκτημάτων της αρχιτεκτονικής κατανεμημένων συστημάτων. Περιγραφή των

διαφορετικών τρόπων ανάπτυξης συστημάτων client-server. Εξήγηση των διαφορών μεταξύ client-server αρχιτεκτονικής και αρχιτεκτονικής κατακεντρωμένων συστημάτων.

- **Σχεδιασμός Λογισμικού Object-Oriented:** Πως μπορεί το σχέδιο λογισμικού να αναπαρασταθεί σαν ανεξάρτητα αντικείμενα που αλληλεπιδρούν μεταξύ τους και διαχειρίζονται την κατάσταση που βρίσκονται και τις λειτουργίες τους. Περιγραφή των σταδίων στην διαδικασία του σχεδιασμού object-oriented. Παρουσίαση των μοντέλων τα οποία περιγράφουν τον σχεδιασμό object-oriented. Επίδειξη της δυνατότητας της UML για την παρουσίαση αυτών των μοντέλων
- **Σχεδιασμός Λογισμικού Real-time:** Εξήγηση της έννοιας του συστήματος real-time και τον λόγο για τον οποίο τα συστήματα αυτά αναπτύσσονται με ταυτόχρονες διαδικασίες. Περιγραφή της διαδικασίας σχεδιασμού για συστήματα real-time. Περιγραφή των αρχιτεκτονικών για συστήματα παρακολούθησης, ελέγχου και απόκτησης δεδομένων
- **Σχεδιασμός Λογισμικού με Reuse:** Τα οφέλη του επανάληψης χρήσης λογισμικού και κάποια προβλήματα. Περιγραφή διαφορετικών τύπων reusable μονάδων και διαδικασιών. Παρουσίαση οικογενειών εφαρμογών από τις οποίες μπορούν να χρησιμοποιηθούν κομμάτια κώδικα.
- **Επικύρωση και Επαλήθευση:** Οι έννοιες της επικύρωσης και επαλήθευσης λογισμικού και οι διαφορές τους. Περιγραφή της διαδικασίας ελέγχου του προγράμματος Εξήγηση της στατικής ανάλυσης σαν τεχνική επαλήθευσης
- **Software Testing:** Οι τεχνικές ελέγχου που χρησιμοποιούνται για να βρεθούν τα σφάλματα του προγράμματος. Παρουσίαση των οδηγιών για τον έλεγχο του interface. Προσεγγίσεις για έλεγχο object-oriented. Τα εργαλεία μπορούν να υποστηρίξουν το testing.

Εργαστηριακές Ασκήσεις

Στο εργαστηριακό μέρος του μαθήματος οι φοιτητές έχουν τη δυνατότητα της πρακτικής εφαρμογής των εννοιών της θεωρίας με τη χρήση ασκήσεων που καλύπτουν εκτενώς την ύλη και καλλιεργούν ορθές προγραμματιστικές δεξιότητες για την ανάπτυξη συστημάτων λογισμικού.

(4) ΔΙΔΑΚΤΙΚΕΣ και ΜΑΘΗΣΙΑΚΕΣ ΜΕΘΟΔΟΙ - ΑΞΙΟΛΟΓΗΣΗ

ΤΡΟΠΟΣ ΠΑΡΑΔΟΣΗΣ	Στην τάξη Πρόσωπο με πρόσωπο	
ΧΡΗΣΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ	Εξειδικευμένο Λογισμικό ανάλυσης και σχεδίασης λογισμικού. Χρήση Τ.Π.Ε. στη διδασκαλία Χρήση Τ.Π.Ε. στην εργαστηριακή εκπαίδευση Χρήση Τ.Π.Ε. στην επικοινωνία με τους φοιτητές μέσω της ηλεκτρονικής πλατφόρμας e-class	
ΟΡΓΑΝΩΣΗ ΔΙΔΑΣΚΑΛΙΑΣ	Δραστηριότητα	Φόρτος Εργασίας Εξαμήνου
	Διαλέξεις	56
	Φροντιστήρια	13
	Μικρές ατομικές εργασίες εξάσκησης	24
	Ομαδική Εργασία σε μελέτη περίπτωσης.	50
	Αυτοτελής Μελέτη	40
	Σύνολο Μαθήματος	180

ΑΞΙΟΛΟΓΗΣΗ ΦΟΙΤΗΤΩΝ	<p>Γλώσσα Αξιολόγησης: Ελληνική</p> <p>Θεωρία: Τελική γραπτή εξέταση στο σύνολο της ύλης (100%). Η εξέταση περιλαμβάνει ερωτήματα θεωρίας (από 3 έως 5) και ασκήσεις πράξεις (από 1 έως 2).</p> <p>Εργαστήριο: Ο τελικός βαθμός συναθροίζεται από συγγραφή εργαστηριακών εργασιών (10%), εκπόνηση project (50%) και τελικής εξέτασης (40%)</p> <p>Τα κριτήρια αξιολόγησης ανακοινώνονται στους φοιτητές κατά την έναρξη του εξαμήνου και βρίσκονται αναρτημένα στην ιστοσελίδα του μαθήματος στο eClass.</p>
----------------------------	--

(5) ΣΥΝΙΣΤΩΜΕΝΗ-ΒΙΒΛΙΟΓΡΑΦΙΑ

- Προτεινόμενη Βιβλιογραφία:

- Bruegge, B., & Dutoit, A. H. (2009). *Object--Oriented Software Engineering. Using UML, Patterns, and Java. Learning, 5(6), 7.*
- Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (2012). *Non-functional requirements in software engineering (Vol. 5). Springer Science & Business Media.*
- Humphrey, W. S. (1995). *A discipline for software engineering. Addison-Wesley Longman Publishing Co., Inc..*
- Jacobson, I. (1993). *Object-oriented software engineering: a use case driven approach. Pearson Education India.*
- Jacobson, I. (1999). *The unified software development process. Pearson Education India.*
- Kotonya, G., & Sommerville, I. (1998). *Requirements engineering: processes and techniques. Wiley Publishing.*
- Pohl, K. (2010). *Requirements engineering: fundamentals, principles, and techniques. Springer Publishing Company, Incorporated.*
- Pree, W., & Gamma, E. (1995). *Design patterns for object-oriented software development (Vol. 183). Reading, MA: Addison-wesley.*
- Pressman, R. S. (2005). *Software engineering: a practitioner's approach. Palgrave Macmillan.*
- Sommerville, I. (2011). *Software engineering 9th Edition. ISBN-10, 137035152.*
- *The Carnegie Mellon Software Engineering Home: <http://www.sei.cmu.edu/sei-home.html>*
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in software engineering. Springer Science & Business Media.*

- Συναφή επιστημονικά περιοδικά:

- *Advances in Engineering Software - Journal – Elsevier*

- *Journal of Software Engineering Research and Development, Springer*
- *IEEE Transactions on Software Engineering*
- *ACM Transactions on Programming Languages and Systems*
- *ACM Transactions on Software Engineering and Methodology*
- *IEEE Software*
- *Information and Software Technology, Elsevier*