

Επίλυση Προβλημάτων (Problem Solving) – Εκπαιδευτικό Υλικό

Τι είναι η «Επίλυση Προβλημάτων» ως Δεξιότητα

Η **επίλυση προβλημάτων** είναι μια από τις βασικές ήπιες δεξιότητες (soft skills) που αναφέρεται στην ικανότητα εντοπισμού, ανάλυσης και αντιμετώπισης σύνθετων ζητημάτων με συστηματικό και αποτελεσματικό τρόπο. Στην πράξη, πρόκειται για μια «γενική» ικανότητα που εφαρμόζεται σε πολλά πλαίσια, μαζί με άλλες δεξιότητες όπως η επικοινωνία, η ομαδική εργασία και η διά βίου μάθηση. Η επίλυση προβλημάτων απαιτεί συνδυασμό αναλυτικής σκέψης, δημιουργικότητας και λήψης αποφάσεων, ώστε να βρίσκουμε **λύσεις** σε προκλήσεις που δεν έχουν πάντα μία προφανή ή μοναδική απάντηση. Σύμφωνα με βιβλιογραφία, οι τεχνικές γνώσεις (hard skills) συμβάλλουν μόνο κατά ~15% στην επαγγελματική επιτυχία, ενώ οι ήπιες δεξιότητες – συμπεριλαμβανομένης της επίλυσης προβλημάτων – αποτελούν το υπόλοιπο ~85%. Οι εργοδότες εκτιμούν ιδιαίτερα υποψηφίους με ανεπτυγμένες τέτοιες δεξιότητες, όπως ευελιξία, αυτοδυναμία, αξιοπιστία, ομαδικότητα και ικανότητα να αντιμετωπίζουν προβλήματα με θετική στάση. Με απλά λόγια, η δεξιότητα της επίλυσης προβλημάτων αντικατοπτρίζει την ικανότητα ενός ατόμου να **συνθέτει λύσεις** όταν έρχεται αντιμέτωπος με ένα νέο ζήτημα, αξιοποιώντας γνώσεις, κρίση και συνεργασία.

Γιατί είναι κρίσιμη για Μηχανικούς Πληροφορικής & Ηλεκτρονικών Συστημάτων

Οι μηχανικοί Πληροφορικής και Ηλεκτρονικών Συστημάτων καλούνται κατεξοχήν να επιλύουν πολύπλοκα προβλήματα στο πλαίσιο της εργασίας τους. Παραδοσιακά, οι μηχανικοί αντιμετώπιζαν μεγάλης κλίμακας προβλήματα με καθαρά επιστημονική και μαθηματική προσέγγιση – και αυτό εξακολουθεί να ισχύει – όμως ο ρόλος του σύγχρονου μηχανικού έχει εξελιχθεί ώστε να περιλαμβάνει και **ανθρώπινες** και **οικονομικο-κοινωνικές** παραμέτρους. Στον τομέα της πληροφορικής, για παράδειγμα, ένας μηχανικός δεν αρκεί να γνωρίζει μόνο προγραμματισμό· πρέπει να μπορεί να κατανοήσει τις ανάγκες μιας επιχείρησης, να ηγηθεί ομάδων και να **επιλύει προβλήματα** λαμβάνοντας υπόψη τον τελικό χρήστη και το περιβάλλον χρήσης.

Η σημασία αυτής της δεξιότητας αντανakλάται στο γεγονός ότι πολλοί απόφοιτοι διαθέτουν μεν καλές τεχνικές γνώσεις, αλλά υστερούν σε ζωτικές δεξιότητες όπως η επικοινωνία, η συγγραφή τεχνικών κειμένων, η συνεργατικότητα ή η κριτική και δημιουργική σκέψη. Συχνά παρατηρείται έλλειμμα σε ικανότητες όπως η **επίλυση προβλημάτων** και η αναλυτική σκέψη, κάτι που προκαλεί **δυσαναλογία** μεταξύ των αναγκών της βιομηχανίας και των ικανοτήτων που επιδεικνύουν οι νέοι απόφοιτοι. Οι εργοδότες όχι μόνο αναζητούν αυτές τις δεξιότητες (σε συνδυασμό με θετική στάση και διάθεση μάθησης), αλλά θεωρούν ότι χωρίς αυτές οι νέοι μηχανικοί δυσκολεύονται να προσαρμοστούν, να δεχτούν ανατροφοδότηση και να εξελιχθούν επαγγελματικά.

Για τους παραπάνω λόγους, η **ανάπτυξη της ικανότητας επίλυσης προβλημάτων** κατά τη διάρκεια των σπουδών είναι κρίσιμη. Τα σύγχρονα προγράμματα σπουδών Μηχανικών Πληροφορικής & Η/Σ αναγνωρίζουν αυτήν την ανάγκη: στις κατευθυντήριες οδηγίες τους, δίπλα στις τεχνικές γνώσεις, περιλαμβάνουν ρητά δεξιότητες όπως η γραπτή και προφορική επικοινωνία, η ομαδική συνεργασία, η κριτική σκέψη, καθώς και η **ικανότητα επίλυσης προβλημάτων** και διά βίου μάθησης. Με άλλα λόγια, ένας απόφοιτος του τμήματος μας δεν αρκεί να ξέρει να προγραμματίζει ή να σχεδιάζει ηλεκτρονικά κυκλώματα – πρέπει να είναι σε θέση να **αναγνωρίζει προβλήματα**, να προτείνει **βιώσιμες λύσεις** και να εφαρμόζει λύσεις σε πραγματικές συνθήκες. Αυτό είναι ουσιώδες για την επιτυχία του τόσο σε τεχνικό επίπεδο (λύνει τεχνικές βλάβες, βελτιστοποιεί συστήματα) όσο και σε επίπεδο εργασιακής προσαρμοστικότητας και καινοτομίας. Δεν είναι τυχαίο που πρωτοβουλίες εκπαιδευτικής μεταρρύθμισης στην Ευρώπη δίνουν έμφαση στην παραγωγή μηχανικών που είναι *επιλύτες προβλημάτων*, ικανοί να εργαστούν σε διεπιστημονικές ομάδες και να προσαρμοστούν σε νέες τεχνολογίες.

Μεθοδολογίες και Βήματα Επίλυσης Προβλημάτων

Υπάρχουν διάφορες μεθοδολογίες για την επίλυση προβλημάτων, όμως κοινός τους παρονομαστής είναι μια συστηματική προσέγγιση. Ένας τυπικός **κύκλος επίλυσης προβλήματος** μπορεί να περιγραφεί με τα ακόλουθα βήματα:

1. **Καθορισμός του Προβλήματος:** Προσδιορίζουμε με σαφήνεια το πρόβλημα που πρέπει να λυθεί και κατανοούμε τη σημασία του. Σε αυτό το στάδιο διατυπώνουμε τι ακριβώς ζητείται να επιλυθεί και γιατί έχει σημασία. Για παράδειγμα, όταν διαβάζουμε ένα τεχνικό πρόβλημα ή μια εκφώνηση έργου, αναρωτιόμαστε: *«Ποιο είναι το ακριβές πρόβλημα που αντιμετωπίζουμε και γιατί είναι σημαντικό;»*. Η σαφής διατύπωση των στόχων και περιορισμών θέτει τη βάση για όλη τη συνέχεια.

2. **Ανάλυση και Συλλογή Πληροφοριών:** Συγκεντρώνουμε δεδομένα, πληροφορίες και γνώσεις σχετικές με το πρόβλημα. Αυτό μπορεί να περιλαμβάνει βιβλιογραφική έρευνα, μελέτη προηγούμενων παρόμοιων περιπτώσεων ή ανάλυση των απαιτήσεων. Σε ένα έργο μηχανικών, για παράδειγμα, οι φοιτητές μπορεί να χρειαστεί να εκτελέσουν *έρευνα αγοράς ή μελέτη υπάρχουσας βιβλιογραφίας* προκειμένου να κατανοήσουν το πλαίσιο του προβλήματος. Η φάση αυτή βοηθά να χαρτογραφηθούν τα αίτια του προβλήματος και τα διαθέσιμα εργαλεία ή τεχνικές που θα μπορούσαν να αξιοποιηθούν.

3. **Δημιουργία Εναλλακτικών Λύσεων (Brainstorming):** Προτείνονται πιθανές λύσεις ή τρόποι αντιμετώπισης. Εδώ η δημιουργική σκέψη είναι ζωτικής σημασίας – συγκεντρώνουμε ιδέες μέσα από ατομική σκέψη ή ομαδική συζήτηση (*brainstorming*) χωρίς ακόμη να τις απορρίψουμε πρόωρα. Στο πλαίσιο ομαδικών έργων, οι φοιτητές συζητούν τις πληροφορίες που συνέλεξαν, μοιράζονται ιδέες και πιθανούς τρόπους λύσης σε συνεδρίες καταγισμού ιδεών. Για παράδειγμα, σε ένα project ανάπτυξης προϊόντος οι φοιτητές ενδέχεται να σκεφτούν 3-4 διαφορετικές προσεγγίσεις για την επίτευξη του

στόχου. Στόχος σε αυτό το βήμα είναι να υπάρξει μια λίστα εναλλακτικών λύσεων, εξετάζοντας ποικίλες οπτικές γωνίες.

4. **Αξιολόγηση & Επιλογή Λύσης:** Κάθε προτεινόμενη λύση αξιολογείται βάσει συγκεκριμένων κριτηρίων – π.χ. αποτελεσματικότητα, κόστος, χρόνου υλοποίησης, ευκολία εφαρμογής, βιωσιμότητα ή άλλων περιορισμών του προβλήματος. Συγκρίνοντας τις εναλλακτικές, επιλέγουμε εκείνη που ανταποκρίνεται καλύτερα στις απαιτήσεις. Σε ένα τεχνικό project, αυτό το στάδιο μπορεί να σημαίνει ότι οι φοιτητές εφαρμόζουν *κριτήρια αξιολόγησης* (π.χ. **συμπαγές μέγεθος, χαμηλό κόστος, απόδοση, φιλικότητα προς τον χρήστη**) για να ξεχωρίσουν την καλύτερη ιδέα. Για παράδειγμα, μπορεί να καταλήξουν στο **βέλτιστο σχέδιο** αφού συγκρίνουν τις λύσεις ως προς την αποτελεσματικότητά τους. Είναι σημαντικό σε αυτό το βήμα να **τεκμηριώνεται** η επιλογή – δηλαδή να υπάρχει λογική εξήγηση γιατί μια λύση υπερέχει των άλλων.

5. **Υλοποίηση & Έλεγχος της Λύσης:** Η επιλεγμένη λύση τίθεται σε εφαρμογή. Αυτό μπορεί να σημαίνει σχεδιασμό και κατασκευή ενός προϊόντος, συγγραφή κώδικα, υλοποίηση ενός αλγορίθμου ή οποιαδήποτε άλλη ενέργεια υλοποίησης. Οι μηχανικοί σε αυτό το στάδιο **μετατρέπουν τη θεωρία σε πράξη**. Παράλληλα, διενεργείται έλεγχος και δοκιμές για να διασφαλιστεί ότι η λύση όντως επιλύει το πρόβλημα και ικανοποιεί τις προδιαγραφές. Για παράδειγμα, οι φοιτητές σε ένα έργο θα κατασκευάσουν το πρωτότυπο σύμφωνα με το σχέδιο και θα το δοκιμάσουν επανειλημμένα, έως ότου βεβαιωθούν ότι καλύπτει την ανάγκη του “πελάτη” (ή του τελικού χρήστη). Αν τα αποτελέσματα δεν είναι τα αναμενόμενα, πιθανώς χρειάζεται **αναστροφή** πίσω σε προηγούμενα βήματα: επανεξέταση του σχεδιασμού, επιλογή άλλης εναλλακτικής, ή βελτιώσεις. Η διαδικασία είναι συχνά επαναληπτική (iterative), κάτι που σημαίνει ότι μέσω δοκιμών και λαθών η λύση **βελτιστοποιείται**.

6. **Αποτίμηση και Τεκμηρίωση:** (Προαιρετικό αλλά σημαντικό στάδιο) Μετά την υλοποίηση, είναι καλό να γίνεται μια συνολική αποτίμηση: πέτυχε η λύση να αντιμετωπίσει το πρόβλημα; Τι θα μπορούσε να γίνει καλύτερα; Η ομάδα ή ο φοιτητής εξάγουν συμπεράσματα και *μαθήματα* από τη διαδικασία. Επίσης, ιδιαίτερα σε ακαδημαϊκό περιβάλλον, η λύση και η διαδικασία τεκμηριώνονται σε ένα **τεχνικό κείμενο** ή αναφορά, ώστε να κοινοποιηθούν τα αποτελέσματα. Η σαφής γραπτή αποτύπωση της προβληματικής, της μεθοδολογίας, της λύσης και των συμπερασμάτων είναι μέρος της ολοκληρωμένης επίλυσης προβλήματος – συνδέοντας έτσι τη δεξιότητα αυτή με την **τεχνική συγγραφή** (περισσότερα γι’ αυτό παρακάτω).

Σημειώνεται ότι διαφορετικές προσεγγίσεις (όπως το **Design Thinking**, ο κύκλος Conceive – Design – Implement – Operate (**CDIO**), η μέθοδος **Plan-Do-Check-Act**) έχουν τα δικά τους βήματα, αλλά τα περισσότερα μοιράζονται την παραπάνω λογική: πρώτα **κατανοούμε το πρόβλημα**, ύστερα **σκεπτόμαστε λύσεις**, και τέλος **εφαρμόζουμε και αξιολογούμε** τη λύση. Για παράδειγμα, το εκπαιδευτικό πλαίσιο CDIO ενθαρρύνει ακριβώς αυτή τη λογική στον σχεδιασμό έργων: αντί οι φοιτητές να εστιάζουν μόνο σε θεωρητική ανάλυση, συμμετέχουν ενεργά σε έργα στα οποία **συλλαμβάνουν** μια ιδέα, την **σχεδιάζουν**, την **υλοποιούν** και την

λειτουργούν (δοκιμάζουν) οι ίδιοι . Μάλιστα, στο CDIO η έμφαση δίνεται στην επίλυση **ανοικτών προβλημάτων** πραγματικού τύπου – δηλαδή προβλημάτων που δεν έχουν μία σωστή απάντηση – σε αντίθεση με τυποποιημένες “ασκήσεις” με μοναδική λύση. Αυτό βοηθά τους φοιτητές να αναπτύξουν δημιουργικότητα και να μάθουν να λαμβάνουν αποφάσεις σε συνθήκες αβεβαιότητας, κάτι που είναι ανεκτίμητο στην επαγγελματική πρακτική.

Εφαρμογή της Επίλυσης Προβλημάτων σε Projects, Εργαστήρια και Τεχνικά Κείμενα

Η δεξιότητα της επίλυσης προβλημάτων **διαπερνά οριζόντια** πολλές δραστηριότητες της πανεπιστημιακής ζωής ενός φοιτητή μηχανικού. Παρακάτω θα δούμε πώς αξιοποιείται σε διαφορετικά πλαίσια: στα ομαδικά projects (συμπεριλαμβανομένων μεθοδολογιών PBL και CDIO), στις εργαστηριακές ασκήσεις και στην εκπόνηση τεχνικών κειμένων.

- **Σε Ομαδικά Projects και PBL:** Η μάθηση μέσω επίλυσης προβλημάτων (**Problem-Based Learning - PBL**) είναι μια καθιερωμένη ενεργητική μαθησιακή προσέγγιση στην οποία **η επίλυση ενός προβλήματος είναι ο δρόμος για τη μάθηση**. Οι φοιτητές στο PBL τοποθετούνται σε ρόλο “μικρών επαγγελματιών” που πρέπει να αντιμετωπίσουν ένα ρεαλιστικό πρόβλημα του αντικειμένου τους. Συνήθως οργανώνονται σε ομάδες και τους δίνεται ένα σενάριο/πρόβλημα (π.χ. «Σχεδιάστε ένα σύστημα διαχείρισης αποθήκης για μια μικρή επιχείρηση» ή «Αναλύστε γιατί ένα δίκτυο υπολογιστών παρουσιάζει καθυστερήσεις και προτείνετε λύσεις»). Οι ομάδες συζητούν τι γνωρίζουν ήδη και τι χρειάζεται να μάθουν (γνωστό ως διαδικασία “*Know/Need to know*”), διεξάγουν έρευνα και στη συνέχεια προτείνουν λύσεις. Καθ’ όλη τη διάρκεια, ο ρόλος του διδάσκοντα είναι καθοδηγητικός – παρεμβαίνει ως **μέντορας** όποτε χρειάζεται, παρέχοντας σχόλια ή καθοδήγηση.

Μέσα από τέτοιες δραστηριότητες, οι φοιτητές όχι μόνο εμπεδώνουν καλύτερα τη θεωρία (αφού τη βλέπουν να εφαρμόζεται σε πραγματικό πρόβλημα), αλλά καλλιεργούν σημαντικές δεξιότητες: μαθαίνουν να συνεργάζονται, να διαχειρίζονται χρόνο και πόρους, να αντιμετωπίζουν διαφωνίες στην ομάδα και φυσικά να βελτιώνουν τις ικανότητές τους στην επίλυση προβλημάτων. Δεν είναι ασυνήθιστο κατά τη διάρκεια ενός PBL project να προκύψουν διαφωνίες ή πολλαπλές ανταγωνιστικές ιδέες – οι φοιτητές πρέπει τότε να αξιολογήσουν αντικειμενικά τις καταστάσεις και να βρουν **συναινετικές λύσεις**, γεγονός που ενισχύει περαιτέρω τις διαπροσωπικές και ηγετικές τους δεξιότητες. Μελέτες έχουν δείξει ότι προγράμματα που ενσωματώνουν PBL μπορούν να βελτιώσουν μια σειρά από soft skills στους φοιτητές, συμπεριλαμβανομένης της δημιουργικότητας, της επικοινωνίας και της επίλυσης προβλημάτων. Στο Τμήμα μας, τέτοια projects μπορεί να λάβουν χώρα σε μαθήματα σχεδίασης συστημάτων, ανάπτυξης λογισμικού ή και διεπιστημονικές εργασίες, όπου οι φοιτητές από την αρχή μέχρι το τέλος «*μαθαίνουν λύνοντας*».

- **Στο Πλαίσιο CDIO και Project-Based Learning:** Μια συγγενική προσέγγιση είναι η μάθηση μέσω έργων (Project-Based Learning – PBL, όχι να συγχέεται με το Problem-Based Learning) και η υιοθέτηση του πλαισίου **CDIO**. Στο πλαίσιο CDIO (Conceive-Design-

Implement-Operate), οι φοιτητές αναλαμβάνουν ολοκληρωμένα έργα που προσομοιώνουν τον πλήρη κύκλο ανάπτυξης ενός προϊόντος ή συστήματος. Για παράδειγμα, σε ένα project τρίτου έτους μηχανικών (όπως αναφέρεται σε πρόσφατη εφαρμογή του CDIO), οι φοιτητές χωρίζονται σε ομάδες και καλούνται **να συλλάβουν μια ιδέα προϊόντος που καλύπτει μια ανάγκη της κοινωνίας/αγοράς, να σχεδιάσουν τη λύση, να την υλοποιήσουν και να την λειτουργήσουν/δοκιμάσουν** στην πράξη. Μια τέτοια ολιστική εμπειρία δίνει στους φοιτητές την ευκαιρία να εξασκηθούν σε πραγματικές συνθήκες: ξεκινούν εντοπίζοντας μια πραγματική ανάγκη ή πρόβλημα (π.χ. «**Πώς μπορεί να αυτοματοποιηθεί μια χρονοβόρα χειρωνακτική διαδικασία;**»), στη συνέχεια **γεννούν ιδέες λύσεων**, κάνουν σχέδια και πρωτότυπα, και εντέλει **κατασκευάζουν** την λύση και την **δοκιμάζουν** για να δουν αν όντως επιλύει το αρχικό πρόβλημα.

Το CDIO, εκτός από την τεχνική γνώση, δίνει μεγάλη έμφαση στην ανάπτυξη δεξιοτήτων όπως η συνεργασία, η ηγεσία, η δημιουργικότητα και η επίλυση προβλημάτων. Οι φοιτητές μαθαίνουν να δουλεύουν σε ομάδες και να ενώνουν διάφορες τεχνικές γνώσεις για έναν κοινό σκοπό. Σε πρόσφατα παραδείγματα, φοιτητικές ομάδες που δούλεψαν υπό το CDIO ανέπτυξαν **καινοτόμα προϊόντα** (μερικά απέκτησαν και διπλώματα ευρεσιτεχνίας) – ενδεικτικά: μια ηλεκτρική οδοντόβουρτσα που καθαρίζει ταυτόχρονα τρεις πλευρές των δοντιών, ένα ρομπότ ακριβείας για γεωργικές εφαρμογές, μια αυτόματη μηχανή παρασκευής παραδοσιακών τροφίμων (π.χ. πίτες/ροτί) κ.ά. . Ένα συγκεκριμένο project ξεχώρισε: ομάδα φοιτητών εντόπισε την ανάγκη για μια συμπαγή και οικονομική μηχανή που θα παράγει **μαζικά ινδικές πίτες (chapatti/roti)** για χώρους εστίασης. Κατά την εξέλιξη του έργου, οι φοιτητές ερεύνησαν την αγορά και τα υπάρχοντα προϊόντα, πρότειναν εναλλακτικές λύσεις (όπως σύστημα πλάστη με υδραυλική πίεση έναντι συστήματος με κυλίνδρους), έθεσαν κριτήρια επιλογής (ταχύτητα παραγωγής, κόστος, μέγεθος μηχανής, ευκολία χρήσης), και κατέληξαν στο βέλτιστο σχέδιο. Στη συνέχεια σχεδίασαν τη μηχανική λύση με λογισμικό CAD, κατασκεύασαν το πρωτότυπο στο εργαστήριο του πανεπιστημίου και το **δοκίμασαν σε πραγματικές συνθήκες**. Μετά από βελτιώσεις βάσει σχολίων ειδικών και τελικών χρηστών, η μηχανή πέτυχε να παράγει τον απαιτούμενο αριθμό πιτών ανά ώρα τηρώντας τα υγειονομικά στάνταρ, και παρουσιάστηκε στους υποψήφιους χρήστες για τελική έγκριση. Αυτό το παράδειγμα καταδεικνύει πώς ένα εκπαιδευτικό project με βάση το CDIO λειτουργεί **ως πραγματικό σενάριο επίλυσης προβλήματος**: από τον εντοπισμό της ανάγκης μέχρι την παράδοση μιας λύσης έτοιμης προς χρήση. Οι φοιτητές όχι μόνο εμβαθύνουν στις τεχνικές δεξιότητες (π.χ. μηχανολογικός σχεδιασμός, προγραμματισμός μικροελεγκτών κτλ.), αλλά παράλληλα εξασκούνται εντατικά στην επίλυση προβλημάτων, συνεργαζόμενοι και **βιώνοντας τον κύκλο αποτυχίας-διόρθωσης-επιτυχίας** που χαρακτηρίζει τα πραγματικά έργα. Το αποτέλεσμα είναι πιο **ολοκληρωμένοι απόφοιτοι**, με αυτοπεποίθηση να αναλαμβάνουν πολύπλοκες προκλήσεις.

- **Σε Εργαστηριακές Ασκήσεις:** Κάθε εργαστηριακή άσκηση μπορεί να αντιμετωπιστεί σαν μια μικρογραφία διαδικασίας επίλυσης προβλήματος. Στα εργαστήρια του τμήματος (ηλεκτρονικά, προγραμματισμού, δικτύων κ.λπ.), οι φοιτητές συχνά αντιμετωπίζουν πρακτικά προβλήματα: από το να συνδέσουν σωστά ένα κύκλωμα ώστε να επιτύχουν μια συγκεκριμένη λειτουργία, μέχρι να **εντοπίσουν και να διορθώσουν ένα σφάλμα (bug) σε προγραμματιστικό κώδικα** που εμποδίζει το λογισμικό να τρέξει σωστά.

Σε τέτοιες περιπτώσεις, ακολουθούν τα βήματα που περιγράφηκαν παραπάνω, έστω και άτυπα. Για παράδειγμα, αν σε μια εργαστηριακή εργασία ο κώδικας ενός φοιτητή δεν **συμπεριφέρεται** όπως αναμένεται, ο φοιτητής πρέπει: (1) να **καθορίσει το πρόβλημα** (π.χ. “το πρόγραμμα κρασάρει όταν ο χρήστης εισάγει μη έγκυρα δεδομένα”), (2) να **αναλύσει** το τι μπορεί να προκαλεί το πρόβλημα (διαβάζοντας τα μηνύματα λάθους, χρησιμοποιώντας *debugger*, συμβουλευόμενος σημειώσεις ή πηγές), (3) να **σκεφτεί πιθανές λύσεις** (διορθώσεις στον κώδικα, αλλαγή στον έλεγχο εισόδων, κ.λπ.), (4) να **δοκιμάσει** τις λύσεις αυτές μία-μία μέχρι να βρει ποια λύνει το πρόβλημα, και τέλος (5) να **επιβεβαιώσει** ότι η λύση δεν δημιουργεί νέες παρενέργειες. Αυτή η διαδικασία – που στους φοιτητές γίνεται όλο και πιο ενστικτώδης με την εμπειρία – είναι ουσιαστικά η εφαρμογή της δεξιότητας επίλυσης προβλημάτων. Πέρα από τα τεχνικά αποτελέσματα (π.χ. ένα λειτουργικό κύκλωμα ή πρόγραμμα), οι φοιτητές μαθαίνουν να είναι **μεθοδικοί, υπομονετικοί και δημιουργικοί** στην προσέγγισή τους. Τα εργαστήρια παρέχουν επίσης ένα ασφαλές περιβάλλον για λάθη: οι φοιτητές βλέπουν ότι η αποτυχία είναι μέρος της διαδικασίας και μαθαίνουν να **αναστοχάζονται (self-reflect)** – δηλαδή να σκέφτονται «τι πήγε λάθος; τι έμαθα; πώς θα το αποφύγω στο μέλλον;». Αυτή η νοοτροπία είναι κεντρική στην δεξιότητα επίλυσης προβλημάτων, διότι τους προετοιμάζει για τη συνεχή **βελτίωση** και **προσαρμοστικότητα** που χρειάζεται ένας μηχανικός στην καριέρα του.

- **Στην Τεχνική Συγγραφή και σε Τεχνικά Κείμενα:** Ίσως να μην είναι προφανές εκ πρώτης όψευς, αλλά η επίλυση προβλημάτων συνδέεται στενά και με τη **συγγραφή τεχνικών κειμένων** – όπως εκθέσεις, εργαστηριακές αναφορές, πτυχιακές εργασίες και επιστημονικά άρθρα. Κάθε τέτοιο τεχνικό κείμενο δομείται συνήθως γύρω από ένα **κεντρικό πρόβλημα** ή ερώτημα και την προτεινόμενη λύση ή απάντηση σε αυτό. Για παράδειγμα, μια τυπική πτυχιακή εργασία θα ξεκινήσει στην *Εισαγωγή* της περιγράφοντας το πρόβλημα/θέμα που εξετάζει και γιατί αυτό είναι σημαντικό, θα παρουσιάσει στη συνέχεια το *θεωρητικό υπόβαθρο* και τη *μεθοδολογία* (δηλαδή πώς προσέγγισε ο φοιτητής την επίλυση του προβλήματος), έπειτα τα *αποτελέσματα* (λύσεις, ευρήματα) και τέλος θα καταλήξει σε *συμπεράσματα* σχετικά με το πόσο επιτυχημένη ήταν η λύση. Από την πλευρά του αναγνώστη, διαβάζοντας την *περίληψη (abstract)* ή την εισαγωγή ενός άρθρου, οι πρώτες πληροφορίες που αναζητάμε είναι: «*Τι είδους πρόβλημα πραγματεύεται η εργασία και ποια λύση προτείνουν οι συγγραφείς;*». Αυτές ακριβώς οι ερωτήσεις αποτελούν μέρος της κριτικής ανάγνωσης τεχνικών κειμένων και είναι ενδεικτικές της σημασίας του δίπολου **Πρόβλημα–Λύση** σε κάθε τεχνική αναφορά.

Κατά τη συγγραφή ενός τεχνικού κειμένου, ο φοιτητής ουσιαστικά αφηγείται την ιστορία της επίλυσης ενός προβλήματος: παρουσιάζει το **πλαίσιο** και το **πρόβλημα** (π.χ. «πώς μπορούμε να βελτιστοποιήσουμε την απόδοση ενός ασύρματου δικτύου αισθητήρων;»), περιγράφει τη μεθοδολογία που ακολούθησε (*πώς* επιχείρησε να το λύσει – π.χ. μέσω προσομοιώσεων, πειραμάτων, κ.λπ.), παραθέτει τα αποτελέσματα/ευρήματα (**τι λύση βρήκε** ή τι συνέβη) και τέλος αξιολογεί το αποτέλεσμα ως προς το αν και σε ποιο βαθμό έλυσε το αρχικό πρόβλημα. Ένα **καλά δομημένο τεχνικό κείμενο** κάνει εμφανές στον αναγνώστη ποιο πρόβλημα λύθηκε και με ποιον τρόπο – στην πραγματικότητα, ένα από τα *προσδοκώμενα αποτελέσματα* όταν μελετάμε μια εργασία είναι να μπορούμε να απαντήσουμε ξεκάθαρα: «*Ποιο πρόβλημα αντιμετωπίστηκε και ποια είναι η προτεινόμενη λύση;*». Επίσης, ένα τεχνικό κείμενο

αξιολογείται στο κατά πόσο **τεκμηριώνει και υποστηρίζει** ικανοποιητικά τη λύση του προβλήματος. Για παράδειγμα, στην ενότητα της *Συζήτησης/Ανάλυσης* μιας επιστημονικής εργασίας, ο συγγραφέας πρέπει να πείσει ότι η προτεινόμενη λύση όντως λειτουργεί και ότι τα αποτελέσματα επαρκούν για να στηρίξουν τα συμπεράσματά του. Αυτό απαιτεί επίδειξη κριτικής σκέψης και ικανότητα επίλυσης προβλήματος: αν τα δεδομένα δεν υποστηρίζουν πλήρως τη λύση, ο συγγραφέας χρειάζεται να αναγνωρίσει τα όρια και να προτείνει μελλοντικές βελτιώσεις (“future work”), δηλαδή ουσιαστικά **νέα προβλήματα προς επίλυση**.

Συνεπώς, μέσω της τεχνικής γραφής οι φοιτητές εξασκούν την επίλυση προβλημάτων σε μετα-επίπεδο: όχι μόνο λύνουν ένα πρόβλημα πρακτικά, αλλά και *επικοινωνούν* το πρόβλημα και τη λύση του με σαφήνεια. Η διαδικασία συγγραφής απαιτεί να οργανώσουν τη σκέψη τους λογικά – να παρουσιάσουν το **πρόβλημα**, να εξηγήσουν **πώς το προσέγγισαν** (μεθοδολογία), να δείξουν **τι αποτελέσματα** πήραν και να καταλήξουν **αν τελικά λύθηκε** (συμπεράσματα). Αυτός ο δομημένος τρόπος σκέψης δεν είναι παρά μια προέκταση της ίδιας της δεξιότητας επίλυσης προβλημάτων, εφαρμοσμένης στην επικοινωνία. Εξάλλου, μια λύση έχει αξία μόνο όταν μπορεί να **μεταδοθεί και να υιοθετηθεί** από άλλους – και εδώ η συγγραφή παίζει καθοριστικό ρόλο.

Παραδείγματα Εκπαιδευτικών Σεναρίων & Φοιτητικών Εφαρμογών

Για να εμβαθύνουμε περισσότερο, ας εξετάσουμε μερικά **πρακτικά παραδείγματα** όπου η δεξιότητα της επίλυσης προβλημάτων ενεργοποιείται στο πανεπιστημιακό περιβάλλον:

- *Παράδειγμα 1 – Σενάριο PBL σε Μάθημα Λογισμικού:* Σε ένα μάθημα Μηχανικής Λογισμικού, ο διδάσκων δίνει στις ομάδες φοιτητών ένα πραγματικό σενάριο: μια μικρή επιχείρηση χρειάζεται μια εφαρμογή για τη διαχείριση των ραντεβού πελατών της. Οι φοιτητές πρέπει να **κατανοήσουν το πρόβλημα** (τι απαιτήσεις έχει η επιχείρηση;), να **αναλύσουν τις επιλογές** (διαφορετικοί τρόποι σχεδιασμού της βάσης δεδομένων, του UI, κ.ο.κ.), να προτείνουν διάφορες λύσεις (ίσως μια web εφαρμογή, ή μια mobile app, κλπ.) και να αιτιολογήσουν ποια είναι βέλτιστη. Στη συνέχεια υλοποιούν σε πρωτότυπο την εφαρμογή και την παρουσιάζουν. Κατά τη διαδικασία αυτή, αντιμετωπίζουν ζητήματα όπως bug, αλλαγές απαιτήσεων από τον “πελάτη” (που παίζει ο διδάσκων) και περιορισμούς χρόνου. Μαθαίνουν έτσι να προσαρμόζουν το πλάνο τους – ίσως χρειαστεί να επιλύσουν προβλήματα απόδοσης του κώδικα ή ασυμβατότητας με κάποιο σύστημα. Το μάθημα ολοκληρώνεται με τους φοιτητές να καταγράφουν σε μια αναφορά **τι προβλήματα συνάντησαν και πώς τα έλυσαν**, κάνοντας αυτοκριτική στην προσέγγισή τους. Έρευνες έχουν δείξει ότι τέτοιου είδους σενάρια PBL βελτιώνουν αισθητά την αυτοπεποίθηση των φοιτητών στην επίλυση προβλημάτων και άλλες συναφείς δεξιότητες .

- *Παράδειγμα 2 – Σενάριο CDIO/Project σε Μάθημα Ηλεκτρονικών:* Σε ένα εργαστηριακό μάθημα ηλεκτρονικών συστημάτων, οι φοιτητές καλούνται (σε ομάδες) να **σχεδιάσουν, υλοποιήσουν και δοκιμάσουν** έναν αυτόνομο σταθμό μέτρησης περιβαλλοντικών παραμέτρων (θερμοκρασία, υγρασία, κ.λπ.) που θα τροφοδοτείται με

ηλιακή ενέργεια και θα στέλνει τα δεδομένα ασύρματα σε μια βάση δεδομένων. Πρόκειται για ένα έργο τύπου CDIO όπου από το μηδέν πρέπει να **συλλάβουν την ιδέα** (τί ακριβώς θα κάνει το σύστημα, ποια προβλήματα λύνει – π.χ. παρέχει real-time δεδομένα για αγροτική χρήση), να **σχεδιάσουν το κύκλωμα και τον κώδικα** (επιλογή αισθητήρων, μικροελεγκτή, πρωτοκόλλου επικοινωνίας), να **υλοποιήσουν** το σύστημα (συναρμολόγηση hardware, προγραμματισμός λογισμικού) και να το **δοκιμάσουν σε πραγματικές συνθήκες** (έκθεση στον ήλιο, δοκιμή εμβέλειας σήματος κ.λπ.). Κατά τη διάρκεια, αντιμετωπίζουν πλήθος προβλημάτων: ο αρχικός αισθητήρας μπορεί να αποδειχθεί αναξιόπιστος, το ηλιακό πάνελ να μη δίνει αρκετή ισχύ τον χειμώνα, η ασύρματη μετάδοση να διακόπτεται. Οι φοιτητές πρέπει να επιλύσουν κάθε επιμέρους πρόβλημα, πιθανώς αναθεωρώντας μέρος του σχεδιασμού (π.χ. επιλογή διαφορετικού αισθητήρα ή μεγαλύτερης μπαταρίας). Με την ολοκλήρωση του project, οι φοιτητές συνειδητοποιούν ότι έχουν αναπτύξει όχι μόνο ένα λειτουργικό σύστημα, αλλά και ένα **συνολικό πλάνο επίλυσης προβλημάτων**: έμαθαν πώς να διαχειρίζονται πολύπλοκα έργα σπάζοντάς τα σε μικρότερα προβλήματα και λύνοντάς τα ένα ένα. Αυτή η εμπειρία τους προετοιμάζει άριστα για **πραγματικές συνθήκες εργασίας**, όπου τέτοιου είδους ολοκληρωμένες προκλήσεις είναι ο κανόνας.

- *Παράδειγμα 3 – Εργαστηριακή Αναφορά:* Σε ένα εργαστήριο τηλεπικοινωνιών, ζητείται από τους φοιτητές να μετρήσουν την απόδοση ενός φίλτρου και να εντοπίσουν τυχόν αποκλίσεις από τη θεωρία. Μετά τις μετρήσεις, οι φοιτητές παρατηρούν ότι το φίλτρο παρουσιάζει απώλειες μεγαλύτερες από τις αναμενόμενες. Στην **εργαστηριακή αναφορά** που θα παραδώσουν, πρέπει να εξηγήσουν γιατί συμβαίνει αυτό. Εδώ έρχεται η επίλυση προβλήματος: θα χρειαστεί να σκεφτούν πιθανά αίτια (π.χ. ανοχές εξαρτημάτων, ατελής βαθμονόμηση οργάνων, εξωτερικές παρεμβολές), να συγκρίνουν τα ευρήματά τους με βιβλιογραφία (ανάλογες μετρήσεις άλλων) και ενδεχομένως να προτείνουν διορθωτικές ενέργειες (π.χ. καλύτερη θωράκιση του κυκλώματος, διαφορετική βαθμονόμηση). Η **δομή της αναφοράς** τους θα ακολουθήσει τη λογική Problem-Solving: *Εισαγωγή* (περιγραφή του προβλήματος και στόχος πειράματος), *Μέθοδος* (πώς προσέγγισαν τη μέτρηση), *Αποτελέσματα* (τί βρήκαν, ποιο είναι το πρόβλημα – οι υψηλές απώλειες), *Συζήτηση* (ανάλυση γιατί συμβαίνει – επίλυση του “μυστηρίου”), *Συμπεράσματα* (τι μάθαμε, πώς λύθηκε ή θα μπορούσε να λυθεί καλύτερα το ζήτημα). Έτσι, ακόμη και μια απλή εργαστηριακή άσκηση καθίσταται ευκαιρία να καλλιεργηθεί τρόπος σκέψης προσανατολισμένος στη λύση προβλημάτων.

Συνοψίζοντας, η **επίλυση προβλημάτων** δεν είναι απλώς μια θεωρητική έννοια αλλά μια **βιωματική δεξιότητα** που οι φοιτητές του τμήματος Μηχανικών Πληροφορικής & Ηλεκτρονικών Συστημάτων εξασκούν καθημερινά: από το πώς θα διορθώσουν ένα bug, μέχρι το πώς θα σχεδιάσουν ένα ολόκληρο σύστημα ή θα συντάξουν μια τεχνική μελέτη. Είναι η δεξιότητα που τους επιτρέπει να μετατρέπουν τη γνώση σε πράξη και τα προβλήματα σε ευκαιρίες για μάθηση. Σε έναν κόσμο όπου η τεχνολογία και οι προκλήσεις εξελίσσονται συνεχώς, η ικανότητα να **επιλύεις προβλήματα αποτελεσματικά** αποτελεί ίσως το πιο **πολύτιμο εφόδιο** ενός νέου μηχανικού. Μέσα από τα projects, τα εργαστήρια και τις γραπτές εργασίες, οι φοιτητές μαθαίνουν όχι μόνο να λύνουν τα προβλήματα του σήμερα, αλλά και να είναι **έτοιμοι για τα προβλήματα του αύριο** – με αυτοπεποίθηση, δημιουργικότητα και συνεργατικό πνεύμα.

Πηγές:

Felipe L. F. Barros et al., *“Evaluating the Influence of PBL on the Development of Soft Skills in a Computer Engineering Undergraduate Program”*, IEEE, 2018.

Sushma Kulkarni et al., *“Adoption of the Conceive-Design-Implement-Operate Approach to the Third Year Project in a Team-Based Design-Build Environment”*, Procedia Computer Science, 2020.

Meenu Wats & Rakesh K. Wats, *“Developing Soft Skills in Students”*, Int. Journal of Learning, 2009.

Cao et al., *“Application of the CDIO Approach in Developing Student’s Competencies”*, Education Sciences, 2025.

Επίλυση Προβλημάτων – Επίπεδο Αρχάριο (Beginner)

Στόχος: Ανάπτυξη βασικής δομημένης σκέψης, μείωση σύγχυσης, ενίσχυση επιμονής και ικανότητας ανάλυσης.

Επεξήγηση

Οι φοιτητές αυτού του επιπέδου:

- δυσκολεύονται με ανοιχτά προβλήματα,
- μπερδεύονται εύκολα,
- προσπαθούν χωρίς σχέδιο,
- εγκαταλείπουν γρήγορα όταν κολλήσουν.

Ο στόχος είναι να μάθουν **δομή και μεθοδικότητα**.

Συμπεριφορές προς Ανάπτυξη

- Κατάτμηση προβλήματος σε μικρά κομμάτια.
 - Ερωτήσεις κατανόησης.
 - Καταγραφή του τι γνωρίζω / δεν γνωρίζω.
 - Χρήση απλών βημάτων (κατανόηση → ανάλυση → δοκιμή → έλεγχος).
 - Μικρές, σταθερές προσπάθειες.
-

Βασικές Δραστηριότητες

A. Χαρτογράφηση Προβλήματος

Για κάθε άσκηση:

- Πρόβλημα
- Στόχος
- Περιορισμοί
- 3 ερωτήσεις κατανόησης

B. Άσκηση Σταδιακής Προσέγγισης

Κύκλος:

1. Κατανοώ
2. Αναλύω
3. Δοκιμάζω
4. Ελέγχω
5. Διορθώνω

Γ. Μικρές Προκλήσεις Debugging

- διόρθωση μικρών λαθών,
 - εύρεση ενός bug,
 - έλεγχος κυκλώματος.
-

Ρόλοι σε Project

- Διευκρινιστής απαιτήσεων
 - Βοηθός tester
 - Συλλέκτης δεδομένων
 - Βοηθός έρευνας
-

Αναστοχασμός (Reflection)

- Τι με μπέρδεψε;
 - Ποια βήματα ακολούθησα;
 - Τι έμαθα από τα λάθη μου;
-

Δείκτες Προόδου

Ο φοιτητής περνά στο μεσαίο επίπεδο όταν:

- ορίζει ξεκάθαρα προβλήματα,
- κάνει στοχευμένες ερωτήσεις,
- δοκιμάζει περισσότερες από μία λύσεις,
- αποφεύγει την τυχαία προσπάθεια,
- επιμένει συστηματικά.