

Περιεχόμενα

Πρόλογος.....	9
1. Αξιολόγηση της Διαδικτυακής Παρουσίας της Κατασκευαστικής Βιομηχανίας στην Περιοχή της Θεσσαλονίκης <i>Χ. Τριανταφυλλίδης, Ι. Παπαθανασίου, Γ. Ταρνανίδης</i>	11
2. Η Διαδικασία Δικτυακής Ανάλυσης για την Επιλογή της Βέλτιστης Προσφοράς στην Κατασκευαστική Βιομηχανία..... <i>Β. Λεώπουλος, Κ. Κηρυττόπουλος, Δ. Βουλγαρίδου, Β. Διαμάντας</i>	23
3. Μια μη Παραμετρική Μέθοδος για το Πολυκριτήριο Πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα <i>Μ. Κρητικός, Γ. Ιωάννου</i>	41
4. Προβολή και Κατάταξη Τοπικών Αναπτυξιακών Δράσεων στο Διαδίκτυο <i>Ζ. Ανδρεοπούλου, Θ. Κουτρομανίδης, Γ. Αραμπατζής, Β. Μάνος</i>	55
5. Αντιπρότυπα, Δίκτυα Bayes και Πολυκριτήρια Ανάλυση Αποφάσεων στη Διαχείριση Έργων Λογισμικού <i>Δ. Σέττας, Ι. Σταμέλος, Β. Γερογιάννης</i>	67
6. Εκτίμηση των Συναρτήσεων Αξιών στο Σύστημα MUSA με Χρήση Υπερκαθορισμένου Συστήματος .. <i>Χ. Κοίλιας</i>	77
7. Μια Πολυκριτήρια Προσέγγιση για τη Μελέτη Ικανοποίησης Επιχειρησιακών Προγραμμάτων με τη Μέθοδο MUSA..... <i>Γ. Σαμαράς, Π. Υψηλάντης, Ν. Μπλάνας, Ι. Σπύρου</i>	89
8. Μοντελοποίηση των Προτιμήσεων σε Χαρακτηριστικά μη Μονότονης Προτίμησης με τη Χρήση των Οικογενειών Συναρτήσεων Μερικής Προτίμησης του Συστήματος MIIDAS..... <i>Α. Σπυριδάκος, Δ. Γιαννακόπουλος</i>	117
9. Ένας Υβριδικός Αλγόριθμος Περιορισμένης Αναζήτησης και Διαδικασίας Άπληστης Τυχοποιημένης Προσαρμοστικής Αναζήτησης για το Πρόβλημα Ομαδοποίησης <i>Γ. Μαρινάκης, Μ. Μαρινάκη, Μ. Δούμπος, Ν. Ματσατσίνης, Κ. Ζοπουνίδης</i>	133
10. Σχεδιασμός Επιμορφωτικής Παρέμβασης για Εκπαιδευτικούς στη Βάση Πολυκριτήριας Ανάλυσης: Η Περίπτωση της Περιβαλλοντικής Εκπαίδευσης <i>Χ. Σταχτιάς</i>	145
11. Υβριδική Μέθοδος Υποστήριξης Αποφάσεων για την Εκτίμηση Μέσης Ημερήσιας Θερμοκρασίας..... <i>Σ. Κωτσιαντής, Α. Κωστούλας, Σ. Λυκούδης, Α. Αργυρίου, Κ. Μενάγιας</i>	157
12. Εφαρμογή Ενός Νέου Εργαλείου Λήψης Αποφάσεων στην Επιλογή Νέου ΧΥΤΑ στη Δυτική Θεσσαλία..... <i>Β. Βασιλόγλου</i>	167

13. Η Ολοκληρωμένη Διαχείριση Οικοσυστημάτων ως Πολυκριτηριακό Πρόβλημα Λήψης Αποφάσεων	179
<i>Γ. Παυλικάκης, Β. Τσιχριντζής</i>	
14. Μεθοδολογία Υποστήριξης Αποφάσεων για την Προώθηση της Ανανεώσιμης Ενέργειας με Χρήση Γλωσσικών Μεταβλητών	193
<i>Χ. Δούκας, Χ. Παππάς, Ι. Ψαρράς</i>	
15. Διερεύνηση του Επιπέδου Καινοτομίας των Επιχειρήσεων Μέσω των Λογιστικών Καταστάσεων και της Πολυκριτήριας Ανάλυσης	205
<i>Π. Καλαντώνης, Κ. Ζοπουνίδης, Χ. Γαγάνης, Μ. Ροδοσθένους</i>	
16. Εφαρμογή Χωρικών Συστημάτων Υποστήριξης Λήψης Απόφασης σε Περιβαλλοντικές Εφαρμογές: Πιλοτική Εφαρμογή στην Εκτίμηση Κινδύνου Διάβρωσης	215
<i>Γ. Συλλαίος, Ν. Συλλαίος, Ι. Γήτας, Φ. Κατσόγιαννος</i>	
17. Γεωγραφικά Συστήματα Πληροφοριών και Πολυκριτήριες Μέθοδοι Ανάλυσης Αποφάσεων: Μια Επισκόπηση	229
<i>Κ. Αναγνωστόπουλος, Α. Βαβάτσικος</i>	
Κατάλογος Συγγραφέων	245

Αντιπρότυπα, Δίκτυα Bayes και Πολυκριτήρια Ανάλυση Αποφάσεων στη Διαχείριση Έργων Λογισμικού

Δημήτριος Σέττας⁽¹⁾, Ιωάννης Σταμέλος⁽¹⁾, Βασίλης Γερογιάννης⁽²⁾

⁽¹⁾ Τμήμα Πληροφορικής, Αριστοτέλειο Παν/μιο Θεσσαλονίκης

⁽²⁾ Τμήμα Διοίκησης και Διαχείρισης Έργων, ΤΕΙ Λάρισας

Περίληψη

Τα αντιπρότυπα (antipatterns) στη διαχείριση έργων λογισμικού (software project management) είναι μηχανισμοί περιγραφής κοινότυπων λύσεων σε προβλήματα που εμφανίζονται κατά την υλοποίηση του έργου. Η προσέγγιση των Δικτύων Bayes (Bayesian Belief Networks - BBN) έχει πρόσφατα προταθεί για τη μαθηματική αναπαράσταση των αντιπροτύπων. Ένα μοντέλο BBN υποστηρίζει τη διαδικασία λήψης αποφάσεων, αλλά σε πολλές περιπτώσεις αγνοεί τα πολλαπλά και συνήθως αντικρουόμενα κριτήρια που συνδέονται με τις επιλογές του software project manager. Στο πλαίσιο αυτό η εφαρμογή της Πολυκριτήριας Ανάλυσης Αποφάσεων (Multi Criteria Decision Analysis / MCDA) μπορεί να βελτιώσει την αποτελεσματικότητα των αποφάσεων της διαχείρισης ενός έργου. Στο παρόν άρθρο η εφαρμογή αυτή παραδειγματοποιείται μέσω ενός μοντέλου BBN για ένα συγκεκριμένο αντιπρότυπο που αφορά στο σχηματισμό ζευγών προγραμματιστών (pair programming) κατά τη φάση της κατασκευής του κώδικα ενός λογισμικού. Το μοντέλο συμπληρώνεται από την προσέγγιση της MCDA. Το τελικό μοντέλο υποστηρίζει την εκτίμηση της επιρροής των προσωπικοτήτων και των ιδιοσυγκρασιών των προγραμματιστών που συμμετέχουν σε κάθε ζεύγος και χρησιμεύει στην αξιολόγηση του επιπέδου της συνεργασίας μεταξύ τους και της αποτελεσματικότητας των ενεργειών τους.

Λέξεις-κλειδιά: Bayesian Belief Networks, Πολυκριτήρια Ανάλυση Αποφάσεων, Αντιπρότυπα

5.1 Εισαγωγή

Οι διαδικασίες της διαχείρισης έργων (project management) έχουν σημαντικό ρόλο στα έργα ανάπτυξης λογισμικού (software development projects). Είναι απαραίτητες για το συντονισμό των αναλυτών, των σχεδιαστών του λογισμικού, των προγραμματιστών και όλων όσων συμμετέχουν στην υλοποίηση ενός έργου, με απώτερο σκοπό τη μείωση της πιθανότητας αποτυχιών (project failures) [13]. Η μη αποτελεσματική διαχείριση των μεταβλητών που εμπλέκονται σε ένα έργο λογισμικού - δηλαδή των συμμετεχόντων (project stakeholders), της τεχνολογίας (project technology) και των διαδικασιών (project processes) [6, 14] - είναι μια σημαντική αιτία αποτυχιών. Οι μεταβλητές αυτές χαρακτηρίζονται από αβεβαιότητα και ασάφεια, συντελεστές που οι υπάρχουσες τεχνικές της τεχνολογίας λογισμικού (software engineering) συχνά αποδεικνύεται ότι αδυνατούν να τους εξαλείψουν [9].

Στο πλαίσιο αυτό οι software project managers μπορούν να λάβουν υπόψη τεχνικές (πρότυπα – patterns) που χρησιμοποίησαν στο παρελθόν σε «πετυχημένα» έργα, πριν καταλήξουν σε αντίστοιχες αποφάσεις για τρέχοντα έργα. Αναλόγως, οι project managers θα πρέπει να αποφεύγουν πρακτικές (αντιπρότυπα – antipatterns) που σε παρελθόντα έργα οδήγησαν σε αποτυχίες [4, 14]. Τα αντιπρότυπα στη διαχείριση έργων λογισμικού περιγράφουν λύσεις σε προβλήματα που συνδέονται με «δυσλειτουργικές» πρακτικές που αποτρέπουν την επιτυχία ενός

έργου [5, 6, 16]. Πρόκειται για μηχανισμούς που έχουν ως στόχο να διαχειριστούν συστηματικά τις πιθανές αποτυχίες του έργου, επεξηγώντας τους λόγους, τα συμπτώματα, τις επιπτώσεις των προβλημάτων και παρέχοντας πετυχημένες λύσεις που μπορεί να επαναχρησιμοποιηθούν (reused) και σε άλλα έργα. Ένα αντιπρότυπο θεωρείται ως το αποτέλεσμα λανθασμένων επιλογών του project manager εξαιτίας της έλλειψης γνώσης ή εμπειρίας κατά την επίλυση ενός δεδομένου προβλήματος. Κάθε αντιπρότυπο ουσιαστικά περιγράφει δύο δυνατές λύσεις. Η πρώτη είναι μια λύση/επιλογή με αρνητικές επιπτώσεις, ενώ η δεύτερη μια «επανακατασκευασμένη» λύση (refactored solution), που περιγράφει τον τρόπο με τον οποίο η λανθασμένη επιλογή μπορεί να αλλάξει και να οδηγήσει σε θετικά αποτελέσματα [5].

Στο παρόν άρθρο για την αναπαράσταση των αντιπροτύπων βασιζόμαστε σε μοντέλα μηχανικής μάθησης (machine learning models). Συγκεκριμένα, υιοθετούμε τα δίκτυα Bayes - BBN (Bayesian Belief Networks) [11] για να περιγράψουμε ένα συγκεκριμένο αντιπρότυπο που καλείται “shaken but not stirred” [18]. Το αντιπρότυπο αυτό προέρχεται από την πρακτική του σχηματισμού ζευγών προγραμματιστών (pair programming) [20], μια πρακτική που προτείνεται από μεθοδολογίες του ακραίου προγραμματισμού (Extreme Programming - XP) [3], που αποτελεί από τις πιο δημοφιλείς “ευέλικτες” μεθοδολογίες (agile methodologies) της ανάπτυξης λογισμικού. Το pair programming εξετάζει βέλτιστες μεθόδους για τη δημιουργία ομάδων προγραμματιστών των δύο ατόμων που παράγουν μαζί διαφορετικά τμήματα του τελικού κώδικα. Το μοντέλο BBN για το αντιπρότυπο αξιολογεί την επιρροή της προσωπικότητας και της ιδιοσυγκρασίας των προγραμματιστών (programmers) σε ζητήματα επικοινωνίας και συνεργασίας μεταξύ τους. Ένα μοντέλο BBN υποστηρίζει τη λήψη αποφάσεων αλλά σε πολλές περιπτώσεις αγνοεί τα πολλαπλά και συνήθως αντικρουόμενα κριτήρια που συνδέονται με αποφάσεις του software project manager. Για το σκοπό αυτό το μοντέλο BBN του αντιπροτύπου συμπληρώνεται από την προσέγγιση της Πολυκριτήριας Ανάλυσης Αποφάσεων (Multi-Criteria Decision Analysis - MCDA) [12]. Το τελικό μοντέλο λαμβάνει υπόψη, εκτός από ζητήματα επικοινωνίας/συνεργασίας μεταξύ των προγραμματιστών, κριτήρια όπως το χρόνο της ανάπτυξης, την ορθότητα του σχεδιασμού και την ποιότητα του τελικού λογισμικού.

Το άρθρο χωρίζεται σε 7 υποενότητες. Στις υποενότητες 5.2 και 5.3 παρουσιάζεται αντίστοιχα μια σύντομη επισκόπηση στην περιοχή των αντιπροτύπων και στις μεθόδους ανάλυσης/αναπαράστασής τους. Στην υποενότητα 5.4 περιγράφεται το προτεινόμενο αντιπρότυπο και στην υποενότητα 5.5 δίνεται η αναπαράστασή του με δίκτυα BBN. Στην υποενότητα 5.6 παρουσιάζεται η προσέγγιση που ενοποιεί τα δίκτυα BBN με μεθόδους MCDA. Τέλος, στην υποενότητα 5.7 συνοψίζονται τα συμπεράσματα της εργασίας.

5.2 Αντιπρότυπα στη διαχείριση έργων λογισμικού

Σε αντίθεση με τις συμβατικές μεθοδολογίες της διαχείρισης έργων λογισμικού, οι λεγόμενες ευέλικτες μεθοδολογίες (agile methodologies) δίνουν έμφαση σε συνεργατικές διαδικασίες μεταξύ των μελών μιας ομάδας κατά τη διάρκεια του κύκλου ζωής του έργου (software development life-cycle) [17]. Μία συμβατική μεθοδολογία - π.χ. μια προσέγγιση που ακολουθεί το μοντέλο του κύκλου ζωής του καταρράκτη (waterfall life cycle) - ορίζει μία αυστηρή ακολουθία βημάτων που πρέπει να ακολουθηθούν κατά τη διάρκεια του κύκλου ζωής (ανάλυση, σχεδίαση, υλοποίηση, έλεγχος). Αντιθέτως, μία ευέλικτη μεθοδολογία θεωρεί ένα έργο λογισμικού ως μία σειρά μικρών σε διάρκεια βημάτων που εκτελούνται με ένα προσαρμόσιμο κατά περίπτωση τρόπο.

Ο ακραίος προγραμματισμός (Extreme Programming - XP) [3] θεωρείται μια από τις πιο δημοφιλείς ευέλικτες μεθοδολογίες της ανάπτυξης λογισμικού που δίνει έμφαση στην ομαδική εργασία και στην άμεση απόκριση σε αλλαγές των απαιτήσεων των χρηστών (user requirements), που εκφράζονται έστω και αργά κατά τον κύκλο ζωής. Οι project managers, οι τελικοί χρήστες και οι προγραμματιστές σε ένα έργο XP αποτελούν μέλη μιας συνεκτικής ομάδας έργου, που έχει ως στόχο να παραδώσει ένα λογισμικό υψηλής ποιότητας που ικανοποιεί τις απαιτήσεις των χρηστών. Οι διαδικασίες της διαχείρισης του έργου έχουν λοιπόν ως αντικειμενικό σκοπό τη βελτίωση του επιπέδου επικοινωνίας και συνεργασίας μεταξύ των μελών της ομάδας έργου.

Η εφαρμογή προτύπων και αντιπροτύπων έχει αποδειχθεί εξαιρετικά αποδοτική σε ένα έργο XP [15]. Η ιδέα της δημιουργίας προτύπων ξεκίνησε ουσιαστικά το 1977 από τον αρχιτέκτονα Christopher Alexander, ο οποίος πρότεινε μια γλώσσα προτύπων για την κατασκευή κτιρίων [1]. Το 1992 σχεδιάζονται τα πρώτα αντικειμενοστρεφή πρότυπα (object-oriented patterns) για έργα λογισμικού [9]. Τέσσερα χρόνια αργότερα, ο Michael Akroyd στο συνέδριο «Object World West» πρότεινε για πρώτη φορά την έννοια των αντιπροτύπων [2]. Εκτοτε η σχετική έρευνα στα θέματα αυτά αποτελεί μια από τις πιο ενεργές περιοχές της τεχνολογίας λογισμικού [5, 16]. Στη σχετική βιβλιογραφία, για παράδειγμα, προτείνονται κατάλογοι αντιπροτύπων (antipatterns catalogues) όπου προδιαγράφονται συγκεκριμένα αντιπρότυπα της διαχείρισης έργων λογισμικού [5]. Ένας κατάλογος αντιπροτύπων αποτελεί χρήσιμο εργαλείο για τους project managers καθώς ανατρέχοντας σε αυτόν μπορούν να λύσουν προβλήματα που συναντούν σε ένα συγκεκριμένο έργο, εφαρμόζοντας επιτυχείς («επανακατασκευασμένες») λύσεις που έχουν δοκιμαστεί σε άλλα έργα. Για παράδειγμα, μια συνήθης πρακτική του XP είναι αυτή του σχηματισμού ζευγών προγραμματιστών [20], σύμφωνα με την οποία κάθε γραμμή του τελικού κώδικα πρέπει να δημιουργηθεί από ένα ζεύγος προγραμματιστών, που μοιράζονται τον ίδιο σταθμό εργασίας. Στόχος της πρακτικής του pair programming είναι η μεταφορά γνώσης, εμπειριών και προγραμματιστικών δεξιοτήτων μεταξύ των δύο προγραμματιστών που συμμετέχουν στην ίδια ομάδα. Ωστόσο υπάρχουν και ανθρώπινοι παράγοντες που πρέπει να αναγνωριστούν για να ανιχνευθούν προβλήματα που προκύπτουν από τη συνεργασία δύο προγραμματιστών που εργάζονται από κοινού. Για την αντιμετώπιση του προβλήματος μια ενεργή κοινότητα ερευνητών προτείνει συγκεκριμένα αντιπρότυπα του pair programming σε έργα που ακολουθούν πρακτικές του XP [15].

5.3 Αναπαράσταση και ανάλυση των αντιπροτύπων

Ενδιαφέρον παρουσιάζει η αναπαράσταση αντιπροτύπων με μοντέλα μηχανικής μάθησης (machine learning models) με στόχο να παρατηρηθούν αποτελέσματα αποφάσεων που οφείλονται σε αβεβαιότητα, αναφορικά με τις μεταβλητές ενός έργου λογισμικού (συμμετέχοντες, τεχνολογία, διαδικασίες), και τελικά να υποστηριχθεί η απόφαση χρησιμοποίησης της επανακατασκευασμένης λύσης ενός αντιπροτύπου. Τα δίκτυα Bayes (BBN) έχουν προταθεί ως ένας κατάλληλος φορμαλισμός για τη μοντελοποίηση των αντιπροτύπων [18]. Η οπτική αναπαράσταση ενός δικτύου BBN δίνεται με ένα κατευθυνόμενο γράφημα (directed graph) όπου οι κόμβοι αναπαριστούν αβέβαιες μεταβλητές και τα βέλη (οι κατευθυνόμενες ακμές) λογικές σχέσεις (αίτια/αποτελέσματα) μεταξύ των μεταβλητών. Κάθε κόμβος συνδέεται με ένα σύνολο (υπό συνθήκη) πιθανοτήτων που ποσοτικοποιούν τη λογική σχέση κάθε κόμβου με τον αντίστοιχο κόμβο από τον οποίο είναι λογικά εξαρτώμενος.

Η αναπαράσταση των αντιπροτύπων με δίκτυα BBN υποστηρίζει την ανάλυση των επιπτώσεων της αβεβαιότητας των διαφόρων μεταβλητών ενός έργου σε συνδυασμό με συγκεκριμένες μη λειτουργικές απαιτήσεις (non functional requirements), όπως είναι, για παράδειγμα, η ποιότητα του τελικά παραγόμενου λογισμικού. Το βασικό πλεονέκτημα αυτής της προσέγγισης είναι η αξιοποίηση της αναλυτικής ικανότητας (analytical power) των δικτύων BBN καθώς ο φορμαλισμός τους επιτρέπει την ανάλυση αντιπροτύπων με χρήση μαθηματικών μοντέλων, συνεπώς και τη διαχείριση των μεταβλητών του έργου και της αβεβαιότητας που τις χαρακτηρίζει (όταν δηλαδή τα δεδομένα για τις μεταβλητές του έργου είναι ελλιπή). Ωστόσο η προσέγγιση αυτή λαμβάνει υπόψη μόνο ένα υποσύνολο των πληροφοριών που ο project manager χρειάζεται, πριν αποφασίσει να χρησιμοποίησει την επανακατασκευασμένη λύση ενός αντιπροτύπου. Ο project manager μπορεί να ενδιαφέρεται και για άλλα χαρακτηριστικά, όπως το κόστος της ανάπτυξης και τη λειτουργικότητα (functionality) του τελικού λογισμικού, τα οποία δεν περιγράφονται από το μοντέλο BBN [11].

Για τη λύση του προβλήματος χρήσιμη μπορεί να αποδειχτεί η εφαρμογή μεθοδολογιών της Πολυκριτήριας Ανάλυσης Αποφάσεων (MCDA). Στόχος της εφαρμογής της MCDA είναι υποστήριξη της διαδικασίας λήψης απόφασης σε προβλήματα που συνδέονται με ένα σύνολο εναλλακτικών επιλογών, καθεμία από τις οποίες αξιολογείται σύμφωνα με πολλαπλά και συχνά αλληλοσυγκρουόμενα κριτήρια. Στη σχετική βιβλιογραφία [12] οι περισσότερες από τις μεθόδους MCDA επικεντρώνονται σε αποφάσεις που λαμβάνονται έχοντας σχετική βε-

βαιότητα για τις εμπλεκόμενες μεταβλητές. Ωστόσο υπάρχουν και μέθοδοι που αναφέρονται σε προβλήματα που εμπιρεύουν την αβεβαιότητα ως εγγενή παράγοντα [10, 21] και προτείνουν αρχικά τον καθορισμό ενός συνόλου κριτηρίων και περιορισμών. Στη συνέχεια, ο φορμαλισμός των μοντέλων BBN χρησιμοποιείται για να αναπαρστήσει και να βαθμολογήσει τα αρχικά κριτήρια.

5.4 Το αντιπρότυπο “Shaken but not stirred”

Ας υποθέσουμε ότι ακολουθείται η πρακτική του pair programming κατά το στάδιο της συγγραφής του κώδικα ενός λογισμικού. Ο software project manager στο στάδιο αυτό επιθυμεί να αποφασίσει και να ανακατατάξει τα σχηματισθέντα ζεύγη προγραμματιστών κατά τον καλύτερο δυνατό τρόπο. Βασικός στόχος είναι να εξασφαλιστεί η ποιότητα του κώδικα, δηλαδή, για παράδειγμα, να εξασφαλιστεί ότι ο τελικός κώδικας θα έχει όσο το δυνατόν λιγότερα λάθη). Για να επιτευχθεί ο στόχος αυτός δεν αρκεί μόνο η αποφυγή λαθών κατά τη σχεδίαση του κώδικα. Ταυτόχρονα θα πρέπει να ληφθεί υπόψη το επίπεδο γνώσης/εμπειρίας σε θέματα προγραμματισμού αλλά και πιθανές “διαφορές/συγκρούσεις” μεταξύ των συμμετεχόντων στα ζεύγη. Θα πρέπει δηλαδή να εξασφαλιστεί ένα υψηλό επίπεδο επικοινωνίας και συνεργασίας μεταξύ τους.

Το συγκεκριμένο πρόβλημα μπορεί να θεωρηθεί ως παράδειγμα προβλήματος λήψης απόφασης. Στον project manager παρουσιάζονται ως επιλογές [19, 22]:

- Να ανακατατάξει τα ζεύγη προγραμματιστών με «ετερογενή» τρόπο, σύμφωνα με το επίπεδο γνώσεων/εμπειριών/ προγραμματιστικών δεξιοτήτων του καθενός, ώστε σε κάθε ζεύγος να συμμετέχει ένας αρχάριος και ένας έμπειρος προγραμματιστής.
- Να σχηματίσει ζεύγη προγραμματιστών, όπου οι συμμετέχοντες χαρακτηρίζονται το ίδιο (περίπου) επίπεδο γνώσεων/εμπειριών/δεξιοτήτων.
- Να ανακατατάξει τα ζεύγη προγραμματιστών, λαμβάνοντας υπόψη την προσωπικότητα και την ιδιοσυγκρασία των συμμετεχόντων.

Αναφορικά με την τρίτη επιλογή, ο project manager μπορεί να ακολουθήσει τους κανόνες του αντιπροτύπου “shaken but not stirred” [18], το οποίο και παρουσιάζουμε στη συνέχεια ακολουθώντας τα κατηγορήματα των Laplante & Neill [16] για την περιγραφή αντιπροτύπων.

Όνομα (Antipattern Name): “shaken but not stirred”.

Σύνοψη – Βασική Ιδέα (Central Concept): Η απόφαση για το σχηματισμό ζευγών προγραμματιστών δεν λαμβάνει υπόψη την προσωπικότητα και την ιδιοσυγκρασία των συμμετεχόντων.

Δυσλειτουργία (Dysfunction): Ο project manager αποφασίζει τα ζεύγη προγραμματιστών εμπλέκοντας στην ίδια ομάδα (ζεύγος) έναν έμπειρο και έναν αρχάριο προγραμματιστή.

Τεκμηρίωση (Explanation): Υιοθετούμε την ονομασία “shaken but not stirred” γιατί δύο προγραμματιστές εμπλέκονται μαζί με βάση μόνο τις δεξιότητες τους και όχι τα χαρακτηριστικά της προσωπικότητάς τους. Αυτή η επιλογή μπορεί να επηρεάσει αρνητικά το συνολικό χρόνο ανάπτυξης, το επίπεδο συνεργασίας και επικοινωνίας μεταξύ των προγραμματιστών αλλά και την ποιότητα του τελικού λογισμικού.

Ανίχνευση του Προβλήματος (Identification): Το αντιπρότυπο “shaken but not stirred” παρουσιάζεται όταν συνεχώς διαπιστώνονται λάθη στον τελικά παραγόμενο κώδικα από την ίδια ομάδα προγραμματιστών, όταν οι προγραμματιστές αντιλαμβάνονται με διαφορετικό τρόπο τις προδιαγραφές της σχεδίασης ή όταν τα μέλη της ομάδας επιμένουν να εργάζονται ατομικά και δεν συνεργάζονται μεταξύ τους.

Επίλυση (Self-Repair): Ο project manager πρέπει να αναγνωρίσει το συντομότερο δυνατό προβλήματα επικοινωνίας και συνεργασίας μεταξύ των προγραμματιστών.

Επανακατασκευασμένη Λύση (Refactoring): Ο project manager μπορεί να εφαρμόσει ένα επιστημονικά τεκμηριωμένο test προσωπικότητας/ιδιοσυγκρασίας ατόμων (π.χ. το Keirsey Temperament Sorter test - www.keirsey.com) προκειμένου να ανιχνεύσει τα χαρακτηριστικά της προσωπικότητας των προγραμματιστών και να βελτιώσει το επίπεδο επικοινωνίας / συνεργασίας μεταξύ τους.

5.5 Μοντέλο BBN του αντιπροτύπου

Για να καταλήξουμε σε ένα δίκτυο BBN για την αναπαράσταση του αντιπροτύπου “shaken but not stirred” πραγματοποιήθηκε ένα πείραμα στο οποίο συμμετείχαν 84 φοιτητές που παρακολουθούν το μάθημα της Τεχνολογίας Λογισμικού στο 4ο εξάμηνο του προγράμματος σπουδών του Τμήματος Πληροφορικής του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης. Οι φοιτητές χωρίστηκαν από τον project manager (καθηγητή) σε ομάδες των δύο ατόμων, ανάλογα με την προσωπικότητα τους, με βάση τις απαντήσεις τους σε tests ταξινόμησης προσωπικότητας (ακολουθώντας το test Keirsey). Ο σκοπός του πειράματος ήταν να σχεδιαστεί και να υλοποιηθεί κώδικας δύο απλών προγραμμάτων σε Java.

Τα δεδομένα από το πείραμα συγκεντρώθηκαν σε ένα πίνακα που περιέχει 8 μεταβλητές (Πίνακας 5.1). Μετά την καταγραφή των δεδομένων του πειράματος, πραγματοποιήθηκε με χρήση του περιβάλλοντος BBN PowerSoft Package (Data PreProcessor, PowerConstructor & PowerPredictor) [7]:

- α) προεπεξεργασία των δεδομένων,
- β) κατασκευή του αντίστοιχου δικτύου BBN και
- γ) επίλυση του μοντέλου (www.cs.ualberta.ca/~jcheng/bnsoft.htm).

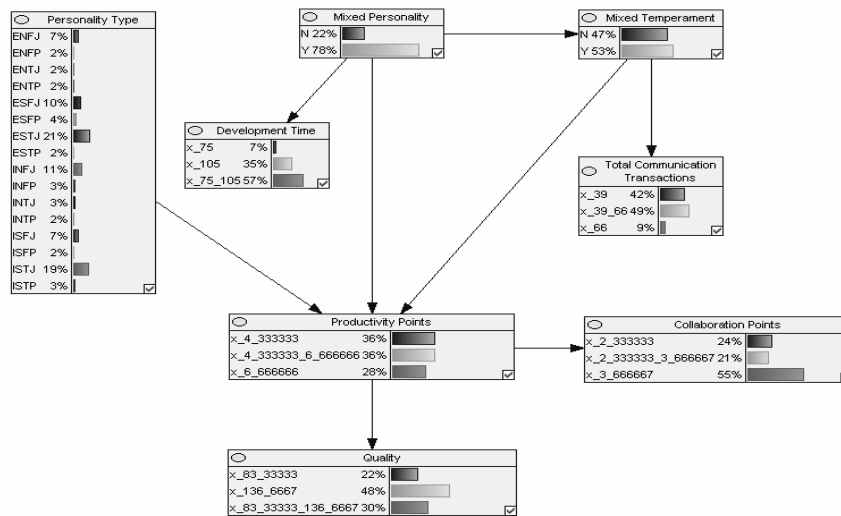
Η προεπεξεργασία των δεδομένων οδήγησε στην κατηγοριοποίηση συνεχών τιμών με βάση ένα καθορισμένο αριθμό διαστημάτων. Το εργαλείο PowerConstructor χρησιμοποιήθηκε για την αναπαράσταση της γνωστικής περιοχής του προβλήματος (domain knowledge). Με βάση τις γνώσεις και την εμπειρία του project manager καθορίστηκαν οι “ρίζες” και τα “φύλλα” του BBN (leaf- root fields). Η εμπειρία του project manager απαιτήθηκε και για τον καθορισμό των λογικών εξαρτήσεων μεταξύ των κόμβων. Αυτή η διαδικασία διευκόλυνε τη διαδικασία κατασκευής του μοντέλου και καθόρισε τις αιτιολογικές (cause-effect) σχέσεις μεταξύ των μεταβλητών.

Η ενσωμάτωση της “γνώσης” του project manager βελτίωσε τα αποτελέσματα του μοντέλου, σε σύγκριση με ένα μοντέλο που δεν θα περιελάμβανε τη γνώση των ειδικών. Στη συνέχεια αξιοποιήθηκε το εργαλείο Genie 2.0 (genie.sis.pitt.edu) για την περαιτέρω επεξεργασία ώστε να απαλειφθεί η αβεβαιότητα (δηλαδή να δημιουργηθεί evidence) στις τιμές των δεδομένων και να παραχθούν αποτελέσματα. Ανανεώνοντας τις τιμές πιθανοτήτων (belief values), κάθε φορά που δίνονταν νέα στοιχεία (πιθανότητες), το αρχικό μοντέλο (Εικόνα 5.1) ελέγχονταν για να διαπιστωθεί η ορθότητά του από τον project manager. Αυτή η τεχνική μας επέτρεψε να απαλείψουμε την αβεβαιότητα (Set Evidence = Yes) στους κόμβους Mixed Personality και Mixed Temperament και τελικά να απεικονιστεί η επανακατασκευασμένη λύση (Εικόνα 5.2).

Πίνακας 5.1. Μεταβλητές του μοντέλου BN

Mixed Personality: NAI/OXI ένδειξη ετερογενούς προσωπικότητας των προγραμματιστών
Mixed Temperament: NAI/OXI ένδειξη ετερογενούς ιδιοσυγκρασίας προγραμματιστών, σύμφωνα με την κατάταξη του test Keirsey (www.keirsey.com)
Personality Type: 16 τύποι προσωπικοτήτων σύμφωνα με την ένδειξη τύπου Myers-Briggs (www.myersbriggs.org).
Development Time: Ο συνολικός χρόνος που χρειάστηκε το ζεύγος προγραμματιστών για να ολοκληρώσει τα 2 προγράμματα (σε λεπτά).

<p>Total Communication Transactions: Το συνολικό πλήθος των «συναλλαγών» μεταξύ των προγραμματιστών (της επικοινωνίας μεταξύ τους) για την κατανόηση της αρχικής σχεδίασης, τον προγραμματισμό, τον έλεγχο και την επιθεώρηση του τελικού κώδικα και του αρχικού σχεδιασμού.</p>
<p>Collaboration: Ο βαθμός συνεργασίας του κάθε προγραμματιστή με το συνεργάτη του (σε βαθμούς).</p>
<p>Design Correctness: Εκφράζεται σε βαθμούς για κάθε ζεύγος και για τα δύο προγραμματιστικά καθήκοντα. Οι βαθμοί μετρήθηκαν βασισμένοι σε συγκεκριμένες περιπτώσεις ελέγχου με σκοπό την εκτίμηση των αποτελεσμάτων των συμμετοχόντων.</p>
<p>Quality: Εκφράζεται σε συνάφεια με τα αποτελέσματα του ελέγχου αποδοχής (acceptance testing) των 2 προγραμμάτων.</p>



Εικόνα 5.1 Μοντέλο BBN του αντιπρότυπου “Shaken but not stirred”

5.6 Πολυκριτήρια ανάλυση σε αντιπρότυπα

Το αντικείμενο ενός προβλήματος απόφασης [11] ορίζεται κάθε φορά υπό την οπτική γωνία των συμμετεχόντων (stakeholders) που μπορεί να είναι είτε οι ενδιαφερόμενοι (που επηρεάζονται άμεσα από την απόφαση) είτε οι αποφασίζοντες (που συνδυάζουν τους στόχους και τους περιορισμούς των συμμετεχόντων και λαμβάνουν την απόφαση). Το πρόβλημα λοιπόν του σχηματισμού ζευγών προγραμματιστών μπορεί να παρουσιαστεί ως ένα πρόβλημα πολυκριτήριας απόφασης ως ακολούθως:

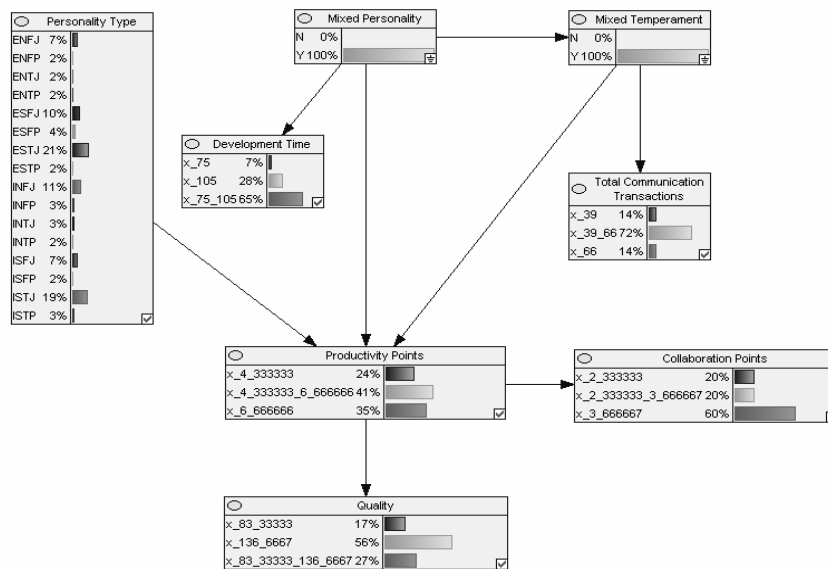
Αντικείμενο της Απόφασης: Η ανακατάταξη των ζευγών προγραμματιστών με τον καλύτερο δυνατό τρόπο.

Συμμετέχοντες: Αποφασίζων: Project Manager. Ενδιαφερόμενοι: Προγραμματιστές.

Πιθανές δράσεις: α) ο project manager ανακατατάσσει τα ζεύγη προγραμματιστών με ετερογενή τρόπο, σύμφωνα με το επίπεδο των ικανοτήτων τους, β) ο project manager ανακατατάσσει τα ζεύγη προγραμματιστών χρησιμοποιώντας προγραμματιστές του ίδιου περίπου επιπέδου και γ) ο project manager εφαρμόζει το αντιπρότυπο “shaken but not stirred”, δηλαδή να ανακατατάσσει τα ζεύγη προγραμματιστών σύμφωνα με την προσωπικότητα τους.

Κριτήρια: Ο συνολικός χρόνος ανάπτυξης, η επικοινωνία και η συνεργασία μεταξύ των προγραμματιστών, η ορθότητα του σχεδιασμού και η ποιότητα του τελικού κώδικα.

Περιορισμοί: Ο μειωμένος χρόνος ανάπτυξης, η βελτιωμένη επικοινωνία, η βελτίωση της συνεργασίας και η ορθότητα σχεδιασμού, η βελτιωμένη ποιότητα λογισμικού.



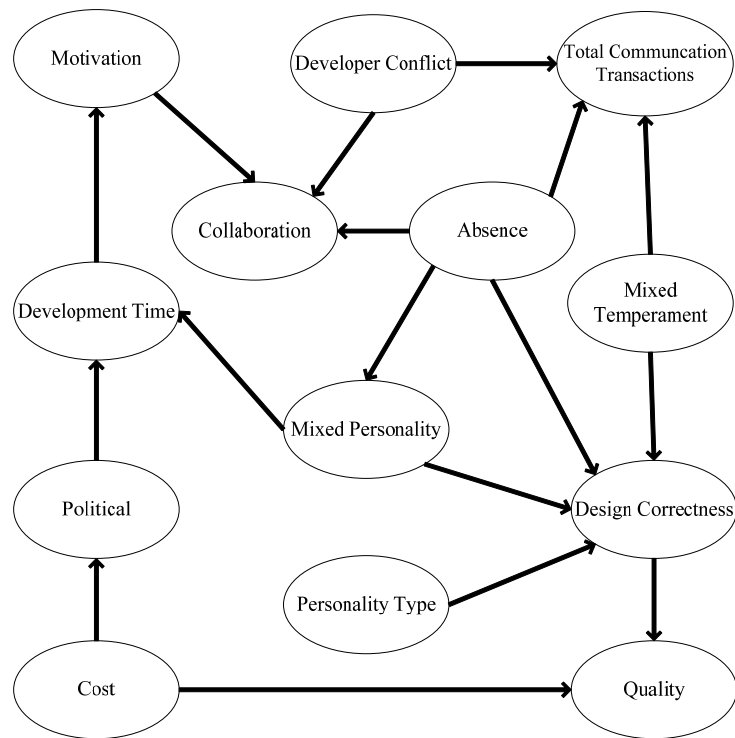
Εικόνα 5.2 Μοντέλο BBN με την επανακατασκευασμένη λύση

Εξωτερικοί παράγοντες: Τα κίνητρα των προγραμματιστών, οι συγκρούσεις, η απουσία ενός προγραμματιστή, το κόστος ανάπτυξης του λογισμικού.

Εσωτερικοί παράγοντες: Το περιβάλλον ανάπτυξης, οι παρεμβάσεις του project manager στη διαδικασία της ανάπτυξης λογισμικού, η προσωπική επικοινωνία με τους προγραμματιστές.

Οι αποφάσεις του project manager σε ένα έργο λογισμικού συνδέονται πολλές φορές με γενικά κριτήρια που δεν είναι εύκολο να καθοριστούν με απόλυτη βεβαιότητα (με συγκεκριμένες τιμές) [11]. Τα BBN μπορούν να αντιμετωπίσουν την έμφυτη αβεβαιότητα των κριτηρίων. Έτσι ένα ασαφές κριτήριο μπορεί να αποσυντίθεται σε κριτήρια που καθορίζονται με περισσότερη σαφήνεια. Για παράδειγμα, η ιδιότητα της εξάρτησης (dependability) του λογισμικού μπορεί να αποσυντεθεί στις ιδιότητες της ασφάλειας (security), της αξιοπιστίας (reliability), της διαθεσιμότητας (availability) και της συντηρησιμότητας (maintainability), ιδιότητες για τις οποίες υπάρχουν συγκεκριμένες μετρικές [10]. Επιπλέον, σε ένα πρόβλημα πολυκριτήριας απόφασης, ορισμένα κριτήρια μπορεί να υπολογίζονται με αβεβαιότητα μετά από μια διαδικασία εξαγωγής συμπεράσματος (inference).

Συγκεκριμένα, εμφανίζονται παράγοντες (εξωτερικοί και εσωτερικοί) που προκαλούν την ανάγκη για εξαγωγή συμπεράσματος με αβεβαιότητα. Οι εξωτερικοί παράγοντες δεν είναι δυνατό να ελεγχθούν, αλλά επηρεάζουν την τιμή ενός κριτηρίου για μία δεδομένη δράση. Όλοι αυτοί οι παράγοντες σε συνδυασμό με τα “αβέβαια” κριτήρια μπορούν να σχηματίσουν κόμβους σε ένα νέο δίκτυο BBN, το οποίο μπορεί να χρησιμοποιηθεί για να προβλεφθούν οι τιμές των αβέβαιων κριτηρίων. Για παράδειγμα, στην Εικόνα 5.3 παρουσιάζεται το τελικό μοντέλο BBN που απεικονίζει αβέβαια κριτήρια και τους εξωτερικούς παράγοντες που επηρεάζουν αυτά τα κριτήρια. Ωστόσο οι εσωτερικοί παράγοντες είναι δυνατό να ελεγχθούν και να επηρεάσουν την τιμή ενός κριτηρίου για μία δεδομένη απόφαση. Ένα παράδειγμα εσωτερικού παράγοντα που θα μπορούσε να συμπεριληφθεί στο μοντέλο BBN (Εικόνα. 5.3) είναι το ποσοστό της “παρέμβασης” του project manager στην ανάπτυξη του λογισμικού. Με αυτόν τον τρόπο θα αυξήσουμε το ποσοστό συμμετοχής του project manager σε περίπτωση που δεχτούμε ως βεβαιότητα ότι έχει μειωθεί το κίνητρο των προγραμματιστών. Έτσι με το τελικό μοντέλο BBN μπορεί να υπολογισθούν οι τιμές των αβέβαιων κριτηρίων, να συνδυαστούν οι τιμές για κάθε απόφαση και τελικά να αξιολογηθούν οι πιθανές αποφάσεις.



Εικόνα 5.3 Το τελικό μοντέλο BBN του αντιπροτύπου

5.7 Επίλογος

Στο άρθρο προτάθηκε η εφαρμογή μοντέλων Bayes για τη μοντελοποίηση των αντιπροτύπων της διαχείρισης έργων ανάπτυξης λογισμικού, λαμβάνοντας υπόψη ευρύτερα προβλήματα αποφάσεων που βασίζονται σε πολλαπλά κριτήρια. Χρησιμοποιώντας το αντιπρότυπο του ακραίου προγραμματισμού “shaken but not stirred” δόθηκε ένα παράδειγμα της προσέγγισης. Το τελικό μοντέλο παρέχει περισσότερες πληροφορίες, όπως στοιχεία για το κόστος της ανάπτυξης και τη λειτουργικότητα του τελικού λογισμικού, τις οποίες θα πρέπει να λάβει υπόψη ο project manager πριν να επιλέξει την απόφαση να υιοθετήσει την επανακατασκευασμένη λύση του αντιπροτύπου.

Ο συνδυασμός της πολυκριτήριας ανάλυσης αποφάσεων και των δικτύων Bayes μπορεί να χρησιμοποιηθεί σε περιπτώσεις που η χρήση των δικτύων Bayes για την παρατήρηση των αποτελεσμάτων της αβεβαιότητας σε μία συγκεκριμένη μεταβλητή του έργου δεν είναι επαρκής και, συνεπώς, δεν μπορεί να υποστηριχθεί η απόφαση της επιλογής ή μη της επανακατασκευασμένης λύσης του αντιπροτύπου. Με βάση την πολυκριτήρια ανάλυση υπολογίζεται η τιμή κάθε κριτηρίου, για κάθε δυνατή απόφαση και, ως αποτέλεσμα, υποστηρίζεται ο project manager στη λήψη της πιο “κατάλληλης” απόφασης. Οι ερευνητικές μας δραστηριότητες προσανατολίζονται στην εγκυροποίηση (validation) της προσέγγισης με ένα πιο ολοκληρωμένο σύνολο εμπειρικών πειραμάτων. Τα πρώτα συμπεράσματα οδηγούν στην ανάγκη ανάλυσης σύνθετων μοντέλων BBN που καθιστούν απαραίτητη την υποστήριξη από εργαλεία λογισμικού που πραγματοποιούν πολυκριτήρια ανάλυση.

Βιβλιογραφία

- [1] Akroyd, M. (1996), “Antipatterns Session Notes”, Object World West, San Francisco.
- [2] Alexander, C. (1977), “A Pattern Language: Towns, Buildings, Construction”, Oxford University Press Inc.
- [3] Beck, K. (1999), “Extreme Programming Explained: Embrace Change”, Addison-Wesley.
- [4] Belbin, M. (1996), “Management Teams: Why They Succeed or Fail”, Butterworth-Heinemann.
- [5] Brown, W. J., Malveau, R. C., McCormick H. & Thomas, S. W. (1998), “AntiPatterns: Refactoring Software, Architectures & Projects in Crisis”, Wiley Computer Publishing.
- [6] Brown, W. J., McCormick H. & Thomas, S. W. (2000), “AntiPatterns in Project Management”, Wiley Computer Publishing.
- [7] Cheng, J., Hatzis, C., Hayashi, H., Krogel, M., Morishita, S., Page, D. & Sese, J. (2002), “KDD Cup 2001 Report”, ACM SIGKDD Explorations, vol. 3, issue 2, January 2002.
- [8] Coad Peter, (1992), “Object-Oriented Patterns”, *Communications of the ACM*, vol. 35(9), pp. 152 – 159.
- [9] Fan, C. F. & Yu, Y. C. (2004), “BBN-based Software Project Risk Management”, *The Journal of Systems & Software*, vol. 73 (2), pp. 193-203.
- [10] Fenton, N. E., Marsh, W., Neil, M., Cates, P., Forey, S. & Tailor, M (2004), “Making Resource Decisions for Software Projects”, *Proc. of the 26th International Conference on Software Engineering*, IEEE Comp. Soc., pp. 397-406.
- [11] Fenton, N. E. & Neil, M. (2001), “Making Decisions: Using Bayesian Nets & MCDA”, *Knowledge-Based Systems*, vol. 14(7), pp. 307-325.
- [12] Figueira, J., Greco, S. & Ehrgott, M. (2005), “Multiple Criteria Decision Analysis: State of the Art Surveys”, Springer Science.
- [13] Fox, T. & Wayne, S. J. (2005), “The Effect of Decision Style on the Use of a Project Management Tool: An Empirical Laboratory Study”, *The Database for Advances in Information Systems*, vol. 36(2), pp. 28-42.
- [14] Hughes, R. & Cotterell, M. (1999), “Software Project Management”, McGraw-Hill.
- [15] Kuranuki, Y. & Hiranabe, K. (2004), “Antipractices: AntiPatterns for XP Practices”, *Proc. of the Agile Development Conference*, IEEE Comp. Soc., pp. 83-86.
- [16] Laplante, P. A. & Neill, C. J. (2006), “Antipatterns: Identification, Refactoring & Management”, Taylor & Francis.
- [17] Larman, C. (2004), “Agile & Iterative Development: A Manager’s Guide”, Addison-Wesley.
- [18] Settas, D., Bibi, S., Sfetsos, P., Stamelos, I. & Gerogiannis, V. C. (2006), “Using Bayesian Belief Networks to Model Software Project Management Antipatterns”, *Proc. of the Fourth International Conference on Software Engineering Research, Management & Applications*, IEEE Comp. Soc., pp. 117-124.
- [19] Sfetsos, P., Stamelos, I., Angelis, L. & Deligiannis, I. (2006), “Investigating the Impact of Personality Types on Communication & Collaboration-Viability in Pair Programming - An Empirical Study”, *Proc. of the Extreme Programming & Agile Processes in Software Engineering Conference*, LNCS 4004, Springer, pp. 43-52.
- [20] Stephens, M. & Rosenberg, D. (2003), “Extreme Programming Refactored: The Case Against XP”, Apress, 2003.
- [21] Waththayu, W. & Peng, Y. (2004), “A Bayesian Network Based Framework for Multi-Criteria Decision Making”, *Proc. of Multi-Criteria Decision Making Conference*, Whistler, B. C., pp. 6-11.
- [22] Williams, L. & Kessler, R. (2002), “Pair Programming Illuminated”, Addison-Wesley.