



**ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Μοντέλα ανάπτυξης λογισμικού και διαχείριση ρίσκου**

**Παναγιώτα Σ. Κατσώνη**

**Ουρανία Θ. Σιαμπάνη**

**Επιβλέποντες: Δημήτρης Βαρουτάς, Επίκουρος Καθηγητής ΕΚΠΑ**

**Νικόλαος Κανέλλος, Υποψήφιος Διδάκτορας**

**ΑΘΗΝΑ**

**ΜΑΡΤΙΟΣ 2016**

## **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

Μοντέλα ανάπτυξης λογισμικού και διαχείριση ρίσκου

**ΠΑΝΑΓΙΩΤΑ ΚΑΤΣΩΝΗ**

**A.M.: 1115200900110**

**ΟΥΡΑΝΙΑ ΣΙΑΜΠΑΝΗ**

**A.M.: 1115200900253**

**Επιβλέποντες:** **Δημήτρης Βαρουτάς**, Επίκουρος Καθηγητής ΕΚΠΑ  
**Νικόλαος Κανέλλος**, Υποψήφιος Διδάκτορας

## ΠΕΡΙΛΗΨΗ

Στις οικονομίες των αναπτυσσόμενων και υπό-ανάπτυξη χωρών ξεχωρίζει για το δυναμισμό του και τη σημασία του ο τομέας της Πληροφορικής και Επικοινωνιών (ICT). Στον τομέα αυτό περιλαμβάνονται οι οικονομικές δραστηριότητες που αφορούν την πληροφορική και τις ηλεκτρονικές επικοινωνίες. Από τα παραπάνω ξεχωρίζει το λογισμικό που αποτελεί σημαντικό ποσοστό της όλης οικονομικής δραστηριότητας.

Η συγκεκριμένη εργασία έχει ως στόχο να εξετάσει τη συμπεριφορά ορισμένων μοντέλων ανάπτυξης λογισμικού. Τα μοντέλα που αναφέρονται αφορούν τη διαχείριση έργων μεσαίου μεγέθους και είναι τα: V-Model, Spiral, Extreme Programming και Scrum. Για κάθε μοντέλο από τα παραπάνω θα αναλυθούν τα χαρακτηριστικά τους, θα παρουσιαστούν τα θετικά τους και τα αρνητικά τους, καθώς και οι συνθήκες στις οποίες ενδείκνυται να χρησιμοποιείται το καθένα.

Σε δεύτερο στάδιο, θα αναλυθεί ο κύκλος ζωής ενός project χρησιμοποιώντας τα 4 μοντέλα. Στόχος της μελέτης είναι ο εντοπισμός των θετικών και αρνητικών του κάθε μοντέλου καθώς και το πως ανταποκρίνεται το κάθε μοντέλο σε τυχαία σφάλματα. Αυτό πραγματοποιείται μέσα από την ανάλυση ευαισθησίας και ρίσκου διαφορετικών σεναρίων υλοποίησης του έργου, τα οποία έχουν ως στόχο την ανάδειξη των διαφορετικών στοιχείων κάθε μοντέλου. Τέλος, παρουσιάζονται τα στατιστικά και τα συμπεράσματα που προέκυψαν από τη μεθοδολογία που ακολουθήθηκε.

**ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ:** Μοντέλα Ανάπτυξης Λογισμικού, Τεχνοοικονομική Αξιολόγηση

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:** V-Model, Spiral, Extreme Programming, Scrum, Monte Carlo Ανάλυση, Ανάλυση Ευαισθησίας

## **ABSTRACT**

In the economies of developed and underdeveloped countries, the industry of Information and Communication (ICT) stands out for its dynamism and importance. This industry includes the economic activities of information technology and electronic communications. From the above, it can be concluded that software stands out for its significant part in the overall economic activity.

The scope of this thesis is to examine the behavior of some software development models. The chosen models concern the management of medium sized projects and are the following: V-Model, Spiral, Extreme Programming and Scrum. For each model their characteristics, strengths and weaknesses will be analyzed. Also, the circumstances in which it is appropriate to use each model will be presented.

Subsequently, the life cycle of a project will be analyzed using the above four models. The purpose of this document is to identify the pros and cons of each model and derive how each one reacts to random errors. The above is accomplished through a sensitivity and risk analysis of different cases of implementation which aim to highlight the different characteristics of each model. Finally, the statistics and conclusions that arose from the methodology used are presented.

**SUBJECT AREA:** Software Development Models, Techno-economical Evaluation

**KEY WORDS:** V-Model, Spiral, Extreme Programming, Scrum, Monte Carlo Analysis, Sensitivity Analysis

## ΕΥΧΑΡΙΣΤΙΕΣ

Με την ολοκλήρωση της συγκεκριμένης πτυχιακής εργασίας θα θέλαμε να ευχαριστήσουμε όλους τους ανθρώπους οι οποίοι βοήθησαν στην περάτωση της εργασίας μας.

Κατά κύριο λόγο, οφείλουμε να ευχαριστήσουμε τους επιβλέποντες. Αρχικά, τον κύριο Δημήτριο Βαρουτά, Επίκουρο Καθηγητή του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών, για τη βοήθεια στην επιλογή του θέματος της εργασίας αλλά και για την εμπιστοσύνη που μας έδειξε στην ανάθεσή της. Επιπλέον, το Νικόλαο Κανέλλο για την πολύτιμη βοήθεια και υποστήριξη που μας παρείχε σε όλη τη διάρκεια της εργασίας τόσο τεχνικά όσο και ψυχολογικά.

Τέλος, ευχαριστούμε τις οικογένειές μας και όλους εκείνους που μας συμπαράσταθηκαν σε αυτήν την προσπάθεια παρέχοντας απεριόριστη κατανόηση και υποστήριξη.

## ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ.....	13
1. ΕΙΣΑΓΩΓΗ .....	14
1.1 Κόστος Κύκλου Ζωής .....	14
1.2 Μοντέλα Ανάπτυξης Λογισμικού.....	14
1.3 Διαχείριση Ρίσκου.....	15
1.4 Κίνητρο και σκοπός πραγματοποίησης της εργασίας.....	15
1.5 Οργάνωση της εργασίας.....	16
2. ΜΟΝΤΕΛΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ .....	17
2.1 Κριτήρια Αξιολόγησης Μοντέλων .....	18
2.1.1 Απλότητα και σταθερότητα των απαιτήσεων .....	18
2.1.2 Κίνδυνοι έργου .....	19
2.1.3 Συμμετοχή χρηστών.....	19
2.1.4 Κατανόηση της αρχιτεκτονικής.....	19
2.1.5 Αξιολόγηση προόδου του έργου .....	20
2.2 V-Model.....	20
2.3 Spiral Model .....	23
2.4 Extreme Programming.....	27
2.5 Scrum .....	32
3. ΔΙΑΧΕΙΡΙΣΗ ΡΙΣΚΟΥ .....	37
3.1 Θεωρία Διαχείρισης Ρίσκου.....	37
3.2 Εμφάνιση κινδύνου σε έργα ανάπτυξης λογισμικού .....	39
3.3 Ανάλυση Monte Carlo.....	42
3.4 Sensitivity Analysis .....	43
4. ΜΕΘΟΔΟΛΟΓΙΑ ΚΑΙ ΜΟΝΤΕΛΟΠΟΙΗΣΗ .....	44
4.1 Case Study.....	44
4.2 Παραδοχές και τυχαίο σφάλμα .....	45

4.3	Τα σενάρια που θα μελετηθούν .....	48
4.4	Συνοπτική επισκόπηση της μεθοδολογίας.....	50
5.	ΑΝΑΛΥΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ .....	51
5.1	Σενάριο μηδενικού σφάλματος .....	51
5.2	Βασικό σενάριο.....	53
5.2.1	Αποτελέσματα Monte Carlo ανάλυσης βασικού σεναρίου .....	54
5.2.2	Σχολιασμός αποτελεσμάτων βασικού σεναρίου.....	57
5.2.2.1	Συνολικό κόστος.....	57
5.2.2.2	Συνολική διάρκεια.....	59
5.2.2.3	Κόστος development .....	61
5.2.2.4	Διάρκεια development .....	63
5.2.2.5	Κόστος των υπόλοιπων φάσεων.....	64
5.2.2.6	Διάρκεια των υπόλοιπων φάσεων.....	66
5.3	Σενάριο με αλλαγή στην πιθανότητα σφάλματος.....	68
5.4	Σενάριο με αλλαγή στον αριθμό γραμμών κώδικα.....	74
5.5	Σενάριο με αλλαγή στο μέγεθος της ομάδας ανάπτυξης .....	81
5.6	Αποτελέσματα sensitivity ανάλυσης .....	88
5.6.1	Σενάριο με σταθερή ομάδα .....	89
5.6.2	Σενάριο με αλλαγή στο μέγεθος της ομάδας.....	96
6.	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	106
7.	ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΕΚΤΑΣΕΙΣ.....	109
	ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ .....	110
	ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΑ.....	112
	ΑΝΑΦΟΡΕΣ .....	113

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1: Bow tie analysis για έργο ανάπτυξης λογισμικού .....	40
Σχήμα 2: Διάγραμμα συνολικού κόστους για το V-Model .....	54
Σχήμα 3: Διάγραμμα συνολικής διάρκειας σε ημέρες για το V-Model .....	54
Σχήμα 4: Διάγραμμα συνολικού κόστους για το Spiral.....	55
Σχήμα 5: Διάγραμμα συνολικής διάρκειας σε ημέρες για το Spiral .....	55
Σχήμα 6: Διάγραμμα συνολικού κόστους για το XP .....	56
Σχήμα 7: Διάγραμμα συνολικής διάρκειας σε ημέρες για το XP.....	56
Σχήμα 8: Διάγραμμα συνολικού κόστους για το Scrum .....	57
Σχήμα 9: Διάγραμμα συνολικής διάρκειας σε ημέρες για το Scrum .....	57
Σχήμα 10: Mean τιμές κόστους με αλλαγή στην πιθανότητα σφάλματος.....	69
Σχήμα 11: Mean τιμές διάρκειας με αλλαγή στην πιθανότητα σφάλματος.....	70
Σχήμα 12: Maximum τιμές κόστους με αλλαγή στην πιθανότητα σφάλματος .....	71
Σχήμα 13: Maximum τιμές διάρκειας (σε ημέρες) με αλλαγή στην πιθανότητα σφάλματος .....	72
Σχήμα 14: Τιμές κόστους για ποσοστό βεβαιότητας 60% με αλλαγή στην πιθανότητα σφάλματος.....	73
Σχήμα 15: Τιμές διάρκειας (σε ημέρες) για ποσοστό βεβαιότητας 60% με αλλαγή στην πιθανότητα σφάλματος .....	74
Σχήμα 16: Mean τιμές κόστους με αλλαγή στις γραμμές κώδικα .....	76
Σχήμα 17: Mean τιμές διάρκεια με αλλαγή στις γραμμές κώδικα .....	77
Σχήμα 18: Maximum τιμές κόστους με αλλαγή στις γραμμές κώδικα .....	78
Σχήμα 19: Maximum τιμές διάρκειας (σε ημέρες) με αλλαγή στις γραμμές κώδικα.....	79
Σχήμα 20: Τιμές κόστους για ποσοστό βεβαιότητας 60% με αλλαγή στις γραμμές κώδικα .....	80
Σχήμα 21: Τιμές διάρκειας (σε ημέρες) για ποσοστό βεβαιότητας 60% με αλλαγή στις γραμμές κώδικα.....	81
Σχήμα 22: Mean τιμές κόστους με αλλαγή στον αριθμό των Seniors και των Juniors ...	83



Σχήμα 23: Mean τιμές διάρκειας (σε ημέρες) με αλλαγή στον αριθμό των Seniors και των Juniors .....	84
Σχήμα 24: Maximum τιμές κόστους με αλλαγή στον αριθμό των Seniors και των Juniors .....	85
Σχήμα 25: Maximum τιμές διάρκειας (σε ημέρες) με αλλαγή στον αριθμό των Seniors και των Juniors .....	86
Σχήμα 26: Τιμές κόστους για ποσοστό βεβαιότητας 60% με αλλαγή στον αριθμό των Seniors και των Juniors .....	87
Σχήμα 27: Τιμές διάρκειας (σε ημέρες) για ποσοστό βεβαιότητας 60% με αλλαγή στον αριθμό των Seniors και των Juniors .....	88
Σχήμα 28: Sensitivity Chart για το συνολικό κόστος του V-Model .....	93
Σχήμα 29: Sensitivity Chart για το συνολικό κόστος του Spiral.....	93
Σχήμα 30: Sensitivity Chart για το συνολικό κόστος της XP .....	94
Σχήμα 31: Sensitivity Chart για το συνολικό κόστος της Scrum.....	94
Σχήμα 32: Sensitivity Chart για τη συνολική διάρκεια του V-Model .....	95
Σχήμα 33: Sensitivity Chart τη συνολική διάρκεια του Spiral .....	95
Σχήμα 34: Sensitivity Chart τη συνολική διάρκεια της XP .....	96
Σχήμα 35: Sensitivity Chart τη συνολική διάρκεια της Scrum .....	96
Σχήμα 36: Sensitivity Chart για το συνολικό κόστος του V-Model .....	101
Σχήμα 37: Sensitivity Chart για το συνολικό κόστος του Spiral.....	102
Σχήμα 38: Sensitivity Chart για το συνολικό κόστος της XP .....	102
Σχήμα 39: Sensitivity Chart για το συνολικό κόστος της Scrum.....	103
Σχήμα 40: Sensitivity Chart για τη συνολική διάρκεια του V-Model .....	103
Σχήμα 41: Sensitivity Chart τη συνολική διάρκεια του Spiral .....	104
Σχήμα 42: Sensitivity Chart τη συνολική διάρκεια της XP.....	104
Σχήμα 43: Sensitivity Chart τη συνολική διάρκεια της Scrum .....	104

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Στάδια μοντέλων ανάπτυξης λογισμικού.....	17
Εικόνα 2: Αλληλουχία φάσεων κατά το μοντέλο V-Model.....	21
Εικόνα 3: Σπειροειδές Μοντέλο Boehm (Spiral Model).....	24
Εικόνα 4: Επαναληπτική Διαδικασία Scrum .....	32
Εικόνα 5: Συσκέψεις στη Scrum .....	34
Εικόνα 6: Η διεργασία Διαχείρισης Κινδύνου .....	38

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Χρέωση ανθρώπινων πόρων ανά ώρα.....	46
Πίνακας 2: Ημερήσια απόδοση ανθρώπινων πόρων σε γραμμές κώδικα βάσει εμπειρίας .....	47
Πίνακας 3: Ομάδες ανάπτυξης και αντίστοιχη πιθανότητα σφάλματος.....	49
Πίνακας 4: Κόστος development σε περίπτωση μηδενικού σφάλματος.....	51
Πίνακας 5: Διάρκεια development σε περίπτωση μηδενικού σφάλματος.....	52
Πίνακας 6: Συνολικό κόστος σε περίπτωση μηδενικού σφάλματος.....	52
Πίνακας 7: Συνολική διάρκεια σε περίπτωση μηδενικού σφάλματος.....	53
Πίνακας 8: Δεδομένα Base Case.....	53
Πίνακας 9: Στατιστικά στοιχεία για το συνολικό κόστος.....	58
Πίνακας 10: Συνολικό κόστος σε αντιστοίχιση με το ποσοστό βεβαιότητας.....	59
Πίνακας 11: Στατιστικά στοιχεία για τη συνολική διάρκεια.....	60
Πίνακας 12: Συνολική διάρκεια σε αντιστοίχιση με το ποσοστό βεβαιότητας.....	61
Πίνακας 13: Στατιστικά στοιχεία για το κόστος development.....	62
Πίνακας 14: Κόστος development σε αντιστοίχιση με το ποσοστό βεβαιότητας.....	63
Πίνακας 15: Στατιστικά στοιχεία για τη διάρκεια development.....	63
Πίνακας 16: Διάρκεια development σε αντιστοίχιση με το ποσοστό βεβαιότητας.....	64
Πίνακας 17: Στατιστικά στοιχεία για το κόστος υπόλοιπων σταδίων.....	65
Πίνακας 18: Κόστος υπόλοιπων σταδίων σε αντιστοίχιση με το ποσοστό βεβαιότητας.....	65
Πίνακας 19: Στατιστικά στοιχεία για τη διάρκεια υπόλοιπων σταδίων.....	66
Πίνακας 20: Διάρκεια υπόλοιπων σταδίων σε αντιστοίχιση με το ποσοστό βεβαιότητας .....	67
Πίνακας 21: Mean τιμές κόστους με αλλαγή στην πιθανότητα σφάλματος.....	68
Πίνακας 22: Mean τιμές διάρκειας με αλλαγή στην πιθανότητα σφάλματος.....	69
Πίνακας 23: Maximum τιμές κόστους με αλλαγή στην πιθανότητα σφάλματος.....	70
Πίνακας 24: Maximum τιμές διάρκειας (σε ημέρες) με αλλαγή στην πιθανότητα σφάλματος.....	71

Πίνακας 25: Τιμές κόστους για ποσοστό βεβαιότητας 60% με αλλαγή στην πιθανότητα σφάλματος.....	72
Πίνακας 26: Τιμές διάρκειας (σε ημέρες) για ποσοστό βεβαιότητας 60% με αλλαγή στην πιθανότητα σφάλματος.....	73
Πίνακας 27: Mean τιμές κόστους με αλλαγή στις γραμμές κώδικα .....	75
Πίνακας 28: Mean τιμές διάρκεια με αλλαγή στις γραμμές κώδικα .....	76
Πίνακας 29: Maximum τιμές κόστους με αλλαγή στις γραμμές κώδικα .....	77
Πίνακας 30: Maximum τιμές διάρκειας (σε ημέρες) με αλλαγή στις γραμμές κώδικα .....	78
Πίνακας 31: Τιμές κόστους για ποσοστό βεβαιότητας 60% με αλλαγή στις γραμμές κώδικα .....	79
Πίνακας 32: Τιμές διάρκειας (σε ημέρες) για ποσοστό βεβαιότητας 60% με αλλαγή στις γραμμές κώδικα.....	80
Πίνακας 33: Mean τιμές κόστους με αλλαγή στον αριθμό των Seniors και των Juniors.	82
Πίνακας 34: Mean τιμές διάρκειας (σε ημέρες) με αλλαγή στον αριθμό των Seniors και των Juniors.....	83
Πίνακας 35: Maximum τιμές κόστους με αλλαγή στον αριθμό των Seniors και των Juniors.....	84
Πίνακας 36: Maximum τιμές διάρκειας (σε ημέρες) με αλλαγή στον αριθμό των Seniors και των Juniors .....	85
Πίνακας 37: Τιμές κόστους για ποσοστό βεβαιότητας 60% με αλλαγή στον αριθμό των Seniors και των Juniors .....	86
Πίνακας 38: Τιμές διάρκειας (σε ημέρες) για ποσοστό βεβαιότητας 60% με αλλαγή στον αριθμό των Seniors και των Juniors .....	87
Πίνακας 39: Επιρροή της πιθανότητας σφάλματος στο τελικό κόστος.....	89
Πίνακας 40: Επιρροή των γραμμών κώδικα στο τελικό κόστος .....	90
Πίνακας 41: Επιρροή γραμμών κώδικα senior developer στο τελικό κόστος.....	90
Πίνακας 42: Επιρροή γραμμών κώδικα junior developer στο τελικό κόστος.....	91
Πίνακας 43: Επιρροή της πιθανότητας σφάλματος στη τελική διάρκεια.....	91
Πίνακας 44: Επιρροή των γραμμών κώδικα στη τελική διάρκεια .....	91

Πίνακας 45: Επιρροή γραμμών κώδικα senior developer στη τελική διάρκεια.....	92
Πίνακας 46: Επιρροή γραμμών κώδικα senior developer στη τελική διάρκεια.....	92
Πίνακας 47: Επιρροή των γραμμών κώδικα στο τελικό κόστος .....	97
Πίνακας 48: Επιρροή γραμμών κώδικα senior developer στο τελικό κόστος.....	97
Πίνακας 49: Επιρροή γραμμών κώδικα junior developer στο τελικό κόστος.....	98
Πίνακας 50: Επιρροή του αριθμού των senior developers στο τελικό κόστος.....	98
Πίνακας 51: Επιρροή του αριθμού των junior developers στο τελικό κόστος.....	99
Πίνακας 52: Επιρροή των γραμμών κώδικα στη τελική διάρκεια .....	99
Πίνακας 53: Επιρροή γραμμών κώδικα senior developer στη τελική διάρκεια.....	100
Πίνακας 54: Επιρροή γραμμών κώδικα junior developer στη τελική διάρκεια.....	100
Πίνακας 55: Επιρροή του αριθμού των senior developers στη τελική διάρκεια.....	100
Πίνακας 56: Επιρροή του αριθμού των junior developers στη τελική διάρκεια.....	101

## ΠΡΟΛΟΓΟΣ

Η παρούσα πτυχιακή εργασία έγινε στα πλαίσια των προπτυχιακών σπουδών του τμήματος Πληροφορικής και Τηλεπικοινωνιών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών.

Η εργασία πραγματεύεται τη μελέτη των μοντέλων ανάπτυξης λογισμικού V-Model, Spiral, XP και Scrum και τον έλεγχο της συμπεριφοράς τους σε περιβάλλοντα όπου είναι πιθανή η εμφάνιση σφάλματος.

Η εργασία αξιολογεί τη συμπεριφορά των παραπάνω μεθόδων πραγματοποιώντας Monte Carlo και Sensitivity ανάλυση. Για την εκτέλεση της Monte Carlo και Sensitivity ανάλυσης οι παραπάνω μέθοδοι μοντελοποιήθηκαν μέσω εντολών excel και οι δοκιμές εκτελέστηκαν μέσω του προγράμματος Crystal Ball της Oracle.

Στη συνέχεια, έχοντας ολοκληρώσει την μοντελοποίηση των παραπάνω μεθόδων έγιναν έλεγχοι και δοκιμές για πέντε διαφορετικά σενάρια με σκοπό να εξαχθούν συμπεράσματα για τη συμπεριφορά του κάθε μοντέλου.

Τέλος, τα αποτελέσματα και τα συμπεράσματα παρουσιάζονται με τη βοήθεια πινάκων και σχημάτων και γίνεται εκτενής συζήτηση για τις διαφορές που προέκυψαν στο κάθε μοντέλο βάσει των αποτελεσμάτων της Monte Carlo και της Sensitivity ανάλυσης.

## 1. ΕΙΣΑΓΩΓΗ

### 1.1 Κόστος Κύκλου Ζωής

Η κοστολόγηση του κύκλου ζωής (Life Cycle Costing) είναι μια μεθοδολογία για τον υπολογισμό του συνολικού κόστους ενός συστήματος από την έναρξή του έως την παραγωγική διάθεσή του. Το σύστημα ποικίλει από κλάδο σε κλάδο και θα μπορούσε για παράδειγμα να είναι ένα κτίριο, ένα πλοίο, ένα οπλικό σύστημα ή ένα έργο ανάπτυξης λογισμικού.

Παρόλα αυτά, όποιο κι αν είναι το σύστημα προς υλοποίηση, η τεχνική ανάλυσης του κόστους του κύκλου ζωής του έργου θα είναι η ίδια. Πιο συγκεκριμένα, καθορίζονται τα κυριότερα στοιχεία κόστους στη διάρκεια ζωής του έργου. Στη συνέχεια, τα στοιχεία αυτά υποδιαιρούνται περαιτέρω μέχρι το κόστος του κάθε στοιχείου να μπορεί να οριστεί ως μια μαθηματική εξίσωση. Ένα απλό παράδειγμα εξίσωσης είναι ότι ο αριθμός των εργατωρών μπορεί να πολλαπλασιαστεί με ένα συντελεστή κόστους. Τέλος, τα επιμέρους στοιχεία του κόστους προστίθενται ώστε να δώσουν το συνολικό κόστος για κάθε στοιχείο και ένα γενικό σύνολο για το σύστημα σε όλη τη διάρκεια ζωής του [1].

### 1.2 Μοντέλα Ανάπτυξης Λογισμικού

Κατά την υλοποίηση έργων που αφορούν την ανάπτυξη λογισμικού χρησιμοποιείται πολύ συχνά ο όρος System Development Life Cycle (SDLC). Ο όρος αυτός περιλαμβάνει μια σειρά από διαδικασίες που λαμβάνουν χώρα στη διάρκεια του έργου. Οι διαδικασίες αυτές αφορούν επιγραμματικά την οργάνωση, την υλοποίηση και τον έλεγχο του προϊόντος. Ο στόχος του SDLC είναι η παραγωγή προϊόντων υψηλής ποιότητας τα οποία είναι συμβατά με τις απαιτήσεις του πελάτη και τα οποία παραδίδονται σε σωστό χρονοδιάγραμμα και κόστος. Με δεδομένο ότι τα τεχνολογικά συστήματα είναι ιδιαίτερα περίπλοκα, έχει αναπτυχθεί μεγάλος αριθμός μοντέλων τα οποία ορίζουν διάφορες ενέργειες και φάσεις με σκοπό τη βέλτιστη διαχείριση του έργου [2]. Η συγκεκριμένη εργασία θα επικεντρωθεί στη μελέτη των ακόλουθων μοντέλων: V-Model, Spiral, XP και Scrum.

Αναλυτικότερα, το V-Model και το Spiral ανήκουν στην κατηγορία των traditional μοντέλων. Τα μοντέλα αυτά χαρακτηρίζονται ως traditional μέθοδοι λόγω του γεγονότος ότι αποτελούνται από μια ακολουθιακή σειρά βημάτων. Το έργο ξεκινάει με τον

καθορισμό και την τεκμηρίωση του συνόλου των απαιτήσεων, στη συνέχεια ακολουθεί η ανάλυση κι η αρχιτεκτονική σχεδίαση και τέλος η υλοποίηση και τα στάδια ελέγχου [3].

Οι traditional μέθοδοι απαιτούν ακόμη την πλήρη καταγραφή όλων των απαιτήσεων κι ελέγχων. Λόγω των παραπάνω, δημιουργήθηκε η ανάγκη ανάπτυξης πιο ευέλικτων μοντέλων τα οποία δε δίνουν βάση στην καταγραφή και στη σειριακή ολοκλήρωση των σταδίων αλλά στη συνεχή ανατροφοδότηση [4]. Τα μοντέλα αυτά ονομάζονται agile κι έχουν ως βασικό χαρακτηριστικό την έμφαση στην προσαρμοστικότητα, τη συχνή επικοινωνία με τον πελάτη, την ενίσχυση της ομαδικότητας και τη δημιουργία λειτουργικού λογισμικού χωρίς αναλυτική τεκμηρίωση. Τα μοντέλα XP και Scrum ανήκουν στην κατηγορία των agile μοντέλων [5].

### **1.3 Διαχείριση Ρίσκου**

Η διαχείριση ρίσκου περιλαμβάνει την αναγνώριση, την αξιολόγηση και την προτεραιοποίηση των κινδύνων που ενδέχεται να προκύψουν κατά την ανάπτυξη ενός έργου. Επιπλέον, περιλαμβάνει την ανάπτυξη ενεργειών οι οποίες μειώνουν, παρακολουθούν και ελέγχουν την επίδραση των κινδύνων στην ανάπτυξη του έργου. Ο σκοπός της διαχείρισης κινδύνου είναι η διασφάλιση των στόχων της εκάστοτε επιχείρησης [6]. Η συγκεκριμένη μελέτη θα επικεντρωθεί στο πως διαχειρίζονται τα παραπάνω μοντέλα την εμφάνιση τυχαίων σφαλμάτων πραγματοποιώντας Monte Carlo ανάλυση.

### **1.4 Κίνητρο και σκοπός πραγματοποίησης της εργασίας**

Η διαχείριση του κινδύνου στην ανάπτυξη λογισμικού και η μοντελοποίηση των διάφορων σταδίων της παίζουν σημαντικό ρόλο τόσο σε ακαδημαϊκά και επιστημονικά θέματα όσο και σε θέματα επαγγελματικών κοινοτήτων. Επιπλέον, το γεγονός ότι τα περιβάλλοντα ανάπτυξης λογισμικού είναι ιδιαίτερα ευμετάβλητα αυξάνει την πολυπλοκότητα της μοντελοποίησής τους και του προσδιορισμού των πιθανών αποτελεσμάτων που προκύπτουν. Με δεδομένο τα παραπάνω, θελήσαμε να πραγματοποιήσουμε την ακόλουθη μελέτη ώστε να καταλήξουμε στο πως συμπεριφέρεται το κάθε μοντέλο ανάπτυξης λογισμικού σε περιβάλλοντα όπου είναι πιθανή η εμφάνιση σφάλματος με στόχο τη βελτίωση της διαδικασίας ανάπτυξης και την καλύτερη κοστολόγηση του έργου.



## 1.5 Οργάνωση της εργασίας

Στο δεύτερο κεφάλαιο της εργασίας παρουσιάζονται τα μοντέλα ανάπτυξης λογισμικού που θα χρησιμοποιηθούν στην μελέτη. Πιο συγκεκριμένα, το κεφάλαιο επικεντρώνεται στη θεωρία πάνω στην οποία είναι βασισμένο το κάθε μοντέλο και στα στάδια που περιλαμβάνει το καθένα.

Στο τρίτο κεφάλαιο, παρατίθεται η βασική θεωρία της διαχείρισης κινδύνου καθώς και ανάλυση για το πως εμφανίζεται ο κίνδυνος σε έργα ανάπτυξης λογισμικού. Επιπλέον, περιλαμβάνεται περιγραφή της Monte Carlo ανάλυσης και της Sensitivity ανάλυσης.

Στο τέταρτο κεφάλαιο, αναφέρεται το case study που πραγματεύεται η εργασία. Επιπλέον, παρουσιάζεται η μοντελοποίηση και η μεθοδολογία που ακολουθήθηκε για την εκτέλεση των διάφορων σεναρίων.

Στο πέμπτο κεφάλαιο της εργασίας παρουσιάζονται και σχολιάζονται τα αποτελέσματα που προκύπτουν για το κάθε σενάριο που εκτελέστηκε.

Στο έκτο κεφάλαιο παρατίθενται συνολικά τα συμπεράσματα που έχουν εξαχθεί από το σύνολο των σεναρίων.

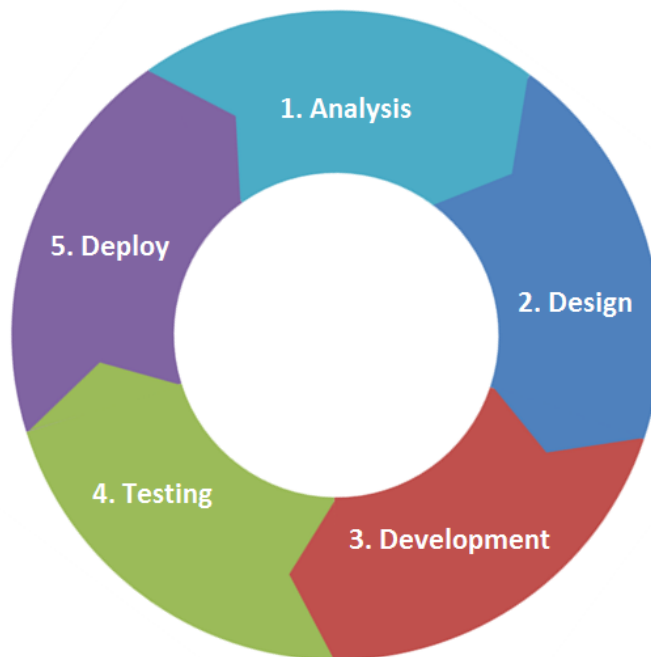
Τέλος, στο έβδομο κεφάλαιο γίνεται αναφορά σε τρόπους με τους οποίους μπορεί να προεκταθεί η συγκεκριμένη μελέτη.

## 2. ΜΟΝΤΕΛΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ

Τα μοντέλα ανάπτυξης λογισμικού είναι ένα σύνολο από διαδικασίες και μεθοδολογίες οι οποίες επιλέγονται για την ανάπτυξη ενός έργου με βάση τους στόχους του. Υπάρχει μεγάλος αριθμός μοντέλων τα οποία έχουν αναπτυχθεί με στόχο την πραγματοποίηση διαφορετικών σκοπών. Τα μοντέλα ορίζουν τα διάφορα στάδια της διαδικασίας καθώς και τη σειρά με την οποία ολοκληρώνονται. Κάποια από τα βασικότερα μοντέλα ανάπτυξης λογισμικού είναι τα ακόλουθα:

1. Waterfall model
2. V-Model
3. Spiral model
4. Big Bang model
5. Incremental model
6. RAD model
7. XP model
8. Scrum model
9. Iterative model
10. RUP

Τα κυριότερα στάδια των μοντέλων ανάπτυξης λογισμικού παρουσιάζονται στο παρακάτω σχήμα.



Εικόνα 1: Στάδια μοντέλων ανάπτυξης λογισμικού

Η συγκεκριμένη εργασία επικεντρώνεται στη μελέτη μεσαίων έργων ανάπτυξης λογισμικού επομένως τα μοντέλα που θα αναλυθούν σε βάθος αφορούν την ανάπτυξη έργων μεσαίου μεγέθους. Αναλυτικότερα, στις υποενότητες που ακολουθούν, παρουσιάζονται τα βασικά κριτήρια αξιολόγησης των μοντέλων καθώς και η θεωρία των V-Model, Spiral, XP και Scrum τα οποία θα χρησιμοποιηθούν κατά την πραγματοποίηση της μελέτης.

## 2.1 Κριτήρια Αξιολόγησης Μοντέλων

Στην υποενότητα αυτή θα παρουσιαστούν τα κριτήρια βάσει των οποίων πραγματοποιείται η αξιολόγηση των μοντέλων ανάπτυξης λογισμικού. Είναι γεγονός ότι δεν υπάρχει κάποιο μοντέλο που να είναι αντικειμενικά καλύτερο από κάποιο άλλο ή που η επιλογή του να εξασφαλίζει την επιτυχία του έργου, ωστόσο τα κριτήρια που θα αναφερθούν στη συνέχεια είναι ιδιαίτερα χρήσιμα. Η χρησιμότητα τους φαίνεται ιδιαίτερα κατά τη διαδικασία επιλογής μοντέλου ανάπτυξης όπου το μοντέλο πρέπει να αξιολογηθεί βάσει των απαιτήσεων του έργου προς υλοποίηση. Τα κριτήρια είναι:

- Απλότητα και σταθερότητα των απαιτήσεων
- Κίνδυνοι έργου
- Συμμετοχή χρηστών
- Κατανόηση της αρχιτεκτονικής
- Αξιολόγηση προόδου του έργου

Για καθένα από τα παραπάνω κριτήρια ακολουθεί μια σύντομη περιγραφή του τι περιλαμβάνουν.

### 2.1.1 Απλότητα και σταθερότητα των απαιτήσεων

Το κριτήριο αυτό αφορά το κατά πόσο οι απαιτήσεις του έργου προς υλοποίηση είναι σχετικά σταθερές ή μεταβλητές. Υπάρχουν διάφορα συστήματα των οποίων οι απαιτήσεις είναι αναμενόμενες, κατανοητές και σπάνια μεταβαλλόμενες σε δευτερεύοντα ζητήματα. Ωστόσο, υπάρχει μεγάλη πιθανότητα να μην είναι εφικτό να περιγραφούν οι απαιτήσεις του λογισμικού με τρόπο τέτοιο που να υπάρχει σταθερότητα κατά την ανάπτυξή του. Ένας λόγος για να συμβεί αυτό, είναι να παίρνουν μέρος στον προσδιορισμό των απαιτήσεων ο πελάτης κι οι χρήστες. Συμπερασματικά, ο προσδιορισμός και η καταγραφή των απαιτήσεων εξαρτάται από την επικοινωνία των

χρηστών και των πελατών με την ομάδα ανάπτυξης. Η ομάδα ανάπτυξης έχει τις τεχνικές γνώσεις να υλοποιήσει το ζητούμενο έργο ωστόσο δεν έχει βαθιά κατανόηση του τι ζητάει ο πελάτης. Αντίστοιχα, οι πελάτες κι οι χρήστες έχουν γνώση του προβλήματος ωστόσο δεν έχουν τις τεχνικές γνώσεις. Επομένως, η διαδικασία καταγραφής και προσδιορισμού των απαιτήσεων εξαρτάται σε μεγάλο βαθμό από την επικοινωνία όλων των μελών γεγονός που συχνά μπορεί να οδηγήσει σε ύπαρξη ελλείψεων και παρερμηνειών. Έτσι λοιπόν, ο τρόπος με τον οποίο θα επιτευχθεί ο προσδιορισμός των απαιτήσεων μπορεί να διαφέρει από έργο σε έργο.

### **2.1.2 Κίνδυνοι έργου**

Οι βασικότεροι κίνδυνοι κατά την υλοποίηση ενός έργου είναι η χαμηλή ποιότητα του τελικού προϊόντος, η υπέρβαση του χρονοπρογράμματος και η υπέρβαση του κόστους. Κάποιοι από τους βασικότερους παράγοντες κινδύνου είναι οι μεταβαλλόμενες απαιτήσεις, η έλλειψη τεχνογνωσίας λόγω χρήσης κάποιας καινούργιας και μη δοκιμασμένης τεχνολογίας, η μη ύπαρξη εμπειρίας σε αντίστοιχα έργα ή ακόμη η υπερβολική αισιοδοξία κατά το σχεδιασμό του έργου. Το πρώτο βήμα για την πρόληψη του κινδύνου είναι να μην υπάρχει προσποίηση ότι δεν υπάρχει κάποιος κίνδυνος και να μην αναβάλλεται η εξάλειψή του. Η επιλογή του μοντέλου ανάπτυξης επηρεάζεται σε μεγάλο βαθμό από τη διαχείριση των κινδύνων του έργου καθώς υπάρχουν μοντέλα που προβλέπουν ενέργειες για την εξάλειψη των κινδύνων.

### **2.1.3 Συμμετοχή χρηστών**

Η συμμετοχή των χρηστών και του πελάτη είναι ιδιαίτερα σημαντική για την επιτυχία ενός έργου λογισμικού. Οι χρήστες είναι αυτοί που θα διατυπώσουν τις απαιτήσεις και θα δεχθούν ή θα απορρίψουν την υλοποίηση. Η σωστότερη προσέγγιση είναι η ενθάρρυνση των χρηστών ώστε να συμμετέχουν στην πορεία ανάπτυξης του έργου βάσει των προδιαγραφών που έχουν δώσει.

### **2.1.4 Κατανόηση της αρχιτεκτονικής**

Κατά την ανάπτυξη ενός έργου λογισμικού, είναι ιδιαίτερα σημαντικό να έχει κατανοήσει η ομάδα ανάπτυξης την αρχιτεκτονική του έργου. Η αρχιτεκτονική του έργου βασίζεται κυρίως στις απαιτήσεις και στις προδιαγραφές. Το στάδιο αυτό είναι αρκετά

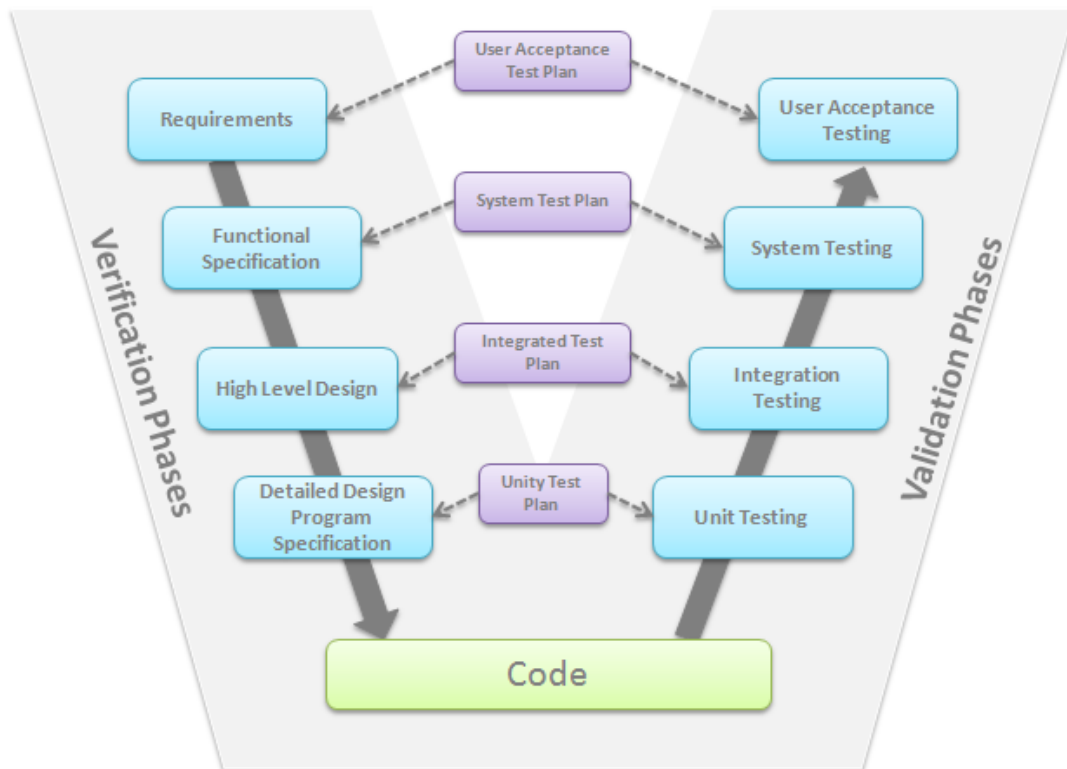
κρίσιμο κι η επιτυχία του έργου έχει μεγάλο βαθμό εξάρτησης από αυτό. Πολλές φορές, η αρχιτεκτονική του προϊόντος προς υλοποίηση μοιάζει με άλλα προϊόντα που ίσως είχε αναπτύξει στο παρελθόν η ομάδα ωστόσο υπάρχουν και φορές που το λογισμικό προς ανάπτυξη έχει χαρακτηριστικά που καθιστούν την αρχιτεκτονική του σχεδίαση μια δύσκολη διαδικασία.

### **2.1.5 Αξιολόγηση προόδου του έργου**

Για την ομαλή πορεία κι ανάπτυξη του έργου, πρέπει η πορεία να ελέγχεται ανά τακτά χρονικά διαστήματα τόσο από τη διοίκηση του έργου όσο κι από τον ίδιο τον πελάτη. Ο ευκολότερος κι αποδοτικότερος τρόπος για να ελέγχεται ένα έργο είναι ο ορισμός μικρότερων στόχων μέσα στο έργο. Με αυτόν τον τρόπο, τόσο ο ίδιος ο πελάτης έχει τη δυνατότητα να ελέγχει την καλή πορεία του έργου, όσο κι η ομάδα ανάπτυξης να έχει συγκεκριμένους στόχους και να μπορεί να προέλθει σε διορθωτικές κινήσεις όταν αυτό είναι απαραίτητο.

## **2.2 V-Model**

Το V-Model αποτελείται από δύο βασικούς κύκλους, αυτόν της επαλήθευσης (verification) κι αυτόν της εγκυρότητας (validation) [7]. Στο συγκεκριμένο μοντέλο κάθε φάση από τον κύκλο του verification αντιστοιχεί σε μια φάση από τον κύκλο του validation και οι δύο κύκλοι ενώνονται με το στάδιο του κώδικα (coding). Αυτό σημαίνει ότι οποιαδήποτε διαδικασία λαμβάνει χώρα κατά την υλοποίηση (development), έχει μια αντίστοιχη διαδικασία κατά τον έλεγχο (testing). Το V-Model είναι ιδιαίτερα αυστηρό ως προς την αλληλουχία των σταδίων του καθώς κάθε στάδιο ξεκινάει αφού ολοκληρωθεί το προηγούμενο.



Εικόνα 2: Αλληλουχία φάσεων κατά το μοντέλο V-Model

## Κύκλος Verification

Ο κύκλος του verification είναι ουσιαστικά μια τεχνική στατικής ανάλυσης και σε αυτό το στάδιο, το testing γίνεται χωρίς να έχει υλοποιηθεί το έργο προς ανάπτυξη. Κάποιες από τις τεχνικές που χρησιμοποιούνται για αυτού του είδους το testing αφορούν κατά βάση την αξιολόγηση, τη διερεύνηση και την επεξήγηση. Οι λειτουργίες που αποτελούν τον κύκλο του verification έχουν άμεση σχέση με το development που θα πραγματοποιηθεί κατά το στάδιο του coding κι αυτό που έχει ιδιαίτερο ενδιαφέρον είναι πως παρόλο που το σύνηθες είναι να εμπλέκεται το testing σε στάδια μεταγενέστερα του development, στον κύκλο του verification το testing ξεκινάει από την πρώτη φάση.

## Coding

Το στάδιο του coding είναι αυτό που ενώνει τους δύο κύκλους, δηλαδή τον κύκλο του verification με τον κύκλο του validation. Σε αυτό το στάδιο του μοντέλου γίνεται ουσιαστικά η υλοποίηση όλου του έργου και η ομάδα ανάπτυξης έχει ενεργό ρόλο καθώς από αυτό το στάδιο εξαρτάται σε μεγάλο βαθμό η ποιότητα του παραγόμενου προϊόντος. Στο συγκεκριμένο στάδιο, εκτός από το development, γίνεται αξιολόγηση

των αποτελεσμάτων προηγούμενων σταδίων καθώς επίσης και μια προεργασία για τα επόμενα στάδια που ακολουθούν κατά τον κύκλο του validation.

## Κύκλος Validation

Ο κύκλος του validation είναι ουσιαστικά μια τεχνική δυναμικής ανάλυσης όπου αφού έχει ολοκληρωθεί η υλοποίηση του κώδικα γίνεται το testing. Το testing πραγματοποιείται με την εκτέλεση του κώδικα, δηλαδή του παραγόμενου προϊόντος και περιλαμβάνει τον λειτουργικό έλεγχο αυτού.

Πιο αναλυτικά, σε κάθε στάδιο του development, δημιουργείται κι ένα πλάνο για το κάθε είδος testing:

- Στο στάδιο της ανάλυσης απαιτήσεων δημιουργείται το πλάνο για το User Acceptance Test (UAT). Κατά τη διάρκεια του συγκεκριμένου testing καταγράφεται η ανταπόκριση των χρηστών στο παραγόμενο προϊόν.
- Στο στάδιο του Functional Specifications δημιουργείται το πλάνο για το System Testing. Το πλάνο του test πρέπει να είναι σύμφωνο με την λειτουργικότητα που έχει οριστεί και τις απαιτήσεις. Κατά τη διάρκεια του συγκεκριμένου test, ελέγχεται η συμπεριφορά του συστήματος συνολικά κι είναι το τελευταίο test που πραγματοποιείται από εξειδικευμένους testers κι όχι από απλούς χρήστες.
- Στο στάδιο του High Level Design (HLD), η ομάδα ανάπτυξης επικεντρώνεται στην αρχιτεκτονική του προϊόντος και για αυτό το λόγο δημιουργείται το πλάνο για το System Integration Test (SIT). Κατά τη διάρκεια του SIT ελέγχεται η επικοινωνία μεταξύ των διάφορων μερών του συστήματος κι ο έλεγχος αυτός πραγματοποιείται από εξειδικευμένους integration testers.
- Στο στάδιο του Detailed Design δημιουργείται το unit testing. Το test αυτό αφορά τους developers, οι οποίοι πραγματοποιούν αυτόν τον έλεγχο αφού ολοκληρωθεί το coding ώστε να επιβεβαιώσουν ότι το προϊόν συμφωνεί με τις απαιτήσεις.

Βάσει των παραπάνω προκύπτει ότι το V-Model έχει κάποια θετικά και κάποια αρνητικά στοιχεία τα οποία στηρίζονται στα κριτήρια που έχουν αναφερθεί στο κεφάλαιο 2.1 Κριτήρια Αξιολόγησης Μοντέλων. Σε αυτό το σημείο θα αξιολογηθεί και θα παρουσιαστεί σε ποιες περιπτώσεις ενδείκνυται να χρησιμοποιείται το V-Model.

1. Το βασικότερο πλεονέκτημα του μοντέλου είναι η απλότητα που παρουσιάζει στη χρήση και στην εφαρμογή του. Αυτό το χαρακτηριστικό βοηθάει την υλοποίηση

του έργου προς ανάπτυξη καθώς το μοντέλο δεν περιέχει περίπλοκες διαδικασίες.

2. Όλες οι διαδικασίες στο V-Model είναι πολύ καλά οργανωμένες και κάθε στάδιο ολοκληρώνεται πριν από την αρχή του επόμενου. Η καλή οργάνωση του μοντέλου έχει ως αποτέλεσμα το κάθε στάδιο να έχει ένα συγκεκριμένο στόχο, ένα συγκεκριμένο παραδοτέο στην επομένη φάση και μια διαδικασία αξιολόγησης.
3. Το μοντέλο μπορεί να χρησιμοποιηθεί με μεγάλη επιτυχία σε μικρά έως μεσαία projects όπου οι απαιτήσεις είναι σε πολύ μεγάλο βαθμό κατανοητές και σταθερές. Σε αντίθεση, το μοντέλο δεν είναι κατάλληλο ούτε για αντικειμενοστραφή projects μεγάλης διάρκειας και έκτασης αλλά ούτε για projects όπου οι απαιτήσεις είναι έντονα μεταβαλλόμενες.
4. Ένα ακόμα θετικό του μοντέλου είναι ότι η διαδικασία σύνταξης των test cases ξεκινάει από το πρώτο στάδιο επομένως οι ασάφειες μπορούν να εντοπιστούν από την αρχή. Η δημιουργία των πλάνων για testing γίνεται πριν την υλοποίηση του κώδικα επομένως υπάρχει μεγάλη πιθανότητα επιτυχίας.
5. Ένα μεγάλο μειονέκτημα του V-Model είναι η έλλειψη μηχανισμών για την ανάλυση ρίσκου και για τη διόρθωση λαθών σε προηγούμενα στάδια. Η έλλειψη μηχανισμού για διαχείριση λαθών και η δυσκολία να επιστρέψει η ομάδα σε προγενέστερα στάδια για διόρθωση αυτών αυξάνει το ρίσκο πράγμα που εντείνεται και λόγω της έλλειψης μηχανισμού για τη διαχείριση του ρίσκου και της αβεβαιότητας.

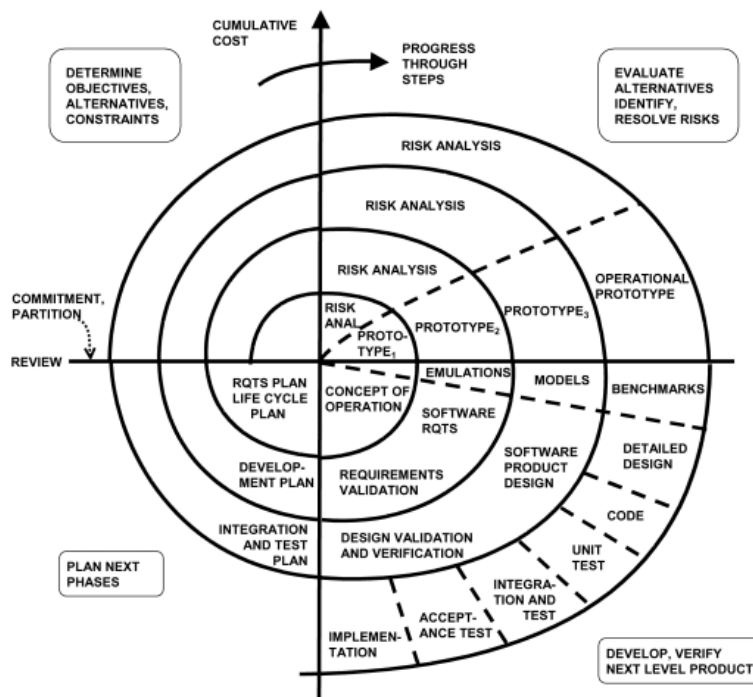
Συμπερασματικά, η χρήση του μοντέλου προτείνεται για μικρά έως μεσαία έργα όπου οι απαιτήσεις είναι προκαθορισμένες και δεν υπάρχουν ασάφειες. Επιπλέον, εκτός από τις απαιτήσεις, πρέπει να είναι σταθερό τόσο το περιβάλλον όσο και η τεχνολογία που χρησιμοποιείται καθώς το μοντέλο δεν είναι ευέλικτο σε αλλαγές [8].

### 2.3 Spiral Model

Το σπειροειδές μοντέλο εισάγει την ιδέα της μη γραμμικής διαδοχής των δραστηριοτήτων της ανάπτυξης [9]. Σχεδιάστηκε ώστε να συμπεριλαμβάνει τα καλύτερα χαρακτηριστικά του μοντέλου του Καταρράκτη (Waterfall Model) και των μοντέλων Πρωτοτύπων (Prototyping Models). Ο όρος σπειροειδές χρησιμοποιείται για να περιγράψει την διαδικασία που ακολουθείται κατά την υλοποίηση του συστήματος.



Συνεχώς εφαρμόζονται αλλαγές πάνω σε μία αρχική έκδοση του συστήματος με βάση τις ανάγκες του πελάτη, επομένως το συνολικό έργο μπορεί να θεωρηθεί και ως σύνολο πολλών μικρών έργων [10]. Η ανάπτυξη του λογισμικού πραγματοποιείται σε κύκλους κατά τους οποίους γίνονται ταυτόχρονα πολλές δραστηριότητες ανάπτυξης. Το τι θα πραγματοποιηθεί σε κάποιο κύκλο του σπειροειδούς μοντέλου καθορίζεται από τους κινδύνους που αντιμετωπίζει το έργο σε κάθε φάση του. Αν, για παράδειγμα, οι ανάγκες του πελάτη είναι εντελώς άγνωστες, τότε δίνεται έμφαση στην αρχική καταγραφή των απαιτήσεων. Αν υπάρχουν ιδιαίτερες απαιτήσεις απόδοσης σε κάποιες λειτουργίες του λογισμικού, τότε μπορεί να αναπτυχθεί ένα πρωτότυπο (prototype) για να εξεταστεί το κατά πόσο μπορούν να ικανοποιηθούν τέτοιου είδους απαιτήσεις. Κάθε κύκλος στο σπειροειδές μοντέλο επιχειρεί να προσδιορίσει όσο το δυνατό καλύτερα το τελικό προϊόν που περιμένει ο πελάτης, μειώνοντας ταυτόχρονα τους κινδύνους του έργου. Ιδιαίτερα σημαντικό πλεονέκτημα του σπειροειδούς μοντέλου είναι το γεγονός ότι μπορεί να συνδυαστεί με άλλα μοντέλα, οπότε είναι αρκετά ευέλικτο ανάλογα με τις ανάγκες του πελάτη [11].



Εικόνα 3: Σπειροειδές Μοντέλο Boehm (Spiral Model)

Η οργάνωση των δραστηριοτήτων γίνεται σε κύκλους ανάπτυξης και σε κάθε κύκλο εκτελούνται τέσσερα βασικά βήματα. Τα βήματα αυτά είναι:

- Καθορισμός στόχων, εναλλακτικών και περιορισμών: περιλαμβάνεται η εξέταση (και επανεξέταση) των στόχων όλων των συμμετεχόντων στο έργο.
- Εντοπισμός και εξάλειψη κινδύνων μαζί με αξιολόγηση εναλλακτικών: επιλέγονται οι κίνδυνοι που θεωρούνται πιο σημαντικοί και επιχειρείται η εξάλειψή τους. Η εξάλειψη κάποιων κινδύνων μπορεί να απαιτεί και την ανάπτυξη πρωτοτύπων. Τα πρωτότυπα μπορεί να αφορούν τις διεπαφές χρήστη και να παρουσιάζονται στους τελικούς χρήστες ενώ άλλα πρωτότυπα μπορεί να αφορούν την αρχιτεκτονική του λογισμικού και να χρησιμοποιούνται για τη μέτρηση της απόδοσης καθώς και για να διαπιστωθεί η εφικτότητα της αρχιτεκτονικής που πρόκειται να επιλεγεί.
- Εκτέλεση δραστηριοτήτων ανάπτυξης: για παράδειγμα, ο προσδιορισμός απαιτήσεων, η σχεδίαση, η κωδικοποίηση κ.τ.λ.. Οι δραστηριότητες ανάπτυξης καθορίζονται σε μεγάλο βαθμό από τα προηγούμενα βήματα δηλαδή, την ικανοποίηση των στόχων των συμμετεχόντων και την εξάλειψη των κινδύνων. Σε αυτό το βήμα μπορεί να γίνεται επανεξέταση και ανάλυση των απαιτήσεων, επανεξέταση της αρχιτεκτονικής του λογισμικού προκειμένου να μειωθούν οι κίνδυνοι που σχετίζονται με αυτή, λεπτομερής σχεδίαση και πιθανόν κωδικοποίηση.
- Κατάρτιση του πλάνου του επόμενου κύκλου: προϋποθέτει τη δέσμευση όλων των συμμετεχόντων του έργου σε αυτό.

Στο σπειροειδές μοντέλο οι πρώτοι κύκλοι είναι οι φθηνότεροι σε ανθρωποπροσπάθεια, χρόνο και κόστος. Κάθε κύκλος αυξάνει αθροιστικά το κόστος ανάπτυξης, μειώνοντας ταυτόχρονα τους κινδύνους του έργου. Το πόσοι θα είναι οι κύκλοι του έργου εξαρτάται από το πόσοι σημαντικοί κίνδυνοι απομένουν να αντιμετωπιστούν. Σε κάθε κύκλο του σπειροειδούς μοντέλου πραγματοποιούνται πολλές δραστηριότητες ανάπτυξης όπως προσδιορισμός απαιτήσεων, σχεδίαση και προγραμματισμός.

Η αξιολόγηση του σπειροειδούς μοντέλου σύμφωνα με τα κριτήρια αξιολόγησης που έχουν παρουσιαστεί στην ενότητα 2.1 Κριτήρια Αξιολόγησης Μοντέλων, καταλήγει στα παρακάτω βασικά συμπεράσματα:

1. Το σπειροειδές μοντέλο μπορεί να αντιμετωπίσει μεταβαλλόμενες απαιτήσεις. Εάν η σταθερότητα των απαιτήσεων διαγνωστεί ως κίνδυνος του έργου, τότε ο

προσδιορισμός απαιτήσεων προτεραιοποιείται. Μπορεί να επιλεχθεί ακόμα και η ανάπτυξη και παρουσίαση πρωτοτύπων, αν υπάρχουν πολλές αμφιβολίες για τη διεπαφή χρήστη.

2. Το σπειροειδές μοντέλο τοποθετεί στο κέντρο της ανάπτυξης τους κινδύνους του έργου. Σε κάθε κύκλο εντοπίζονται οι κίνδυνοι του έργου και όλες οι υπόλοιπες δραστηριότητες κινούνται με σκοπό την εξάλειψή τους. Επομένως η μείωση της επικινδυνότητας είναι αυτή που καθοδηγεί το σύνολο του έργου της ανάπτυξης.
3. Το σπειροειδές μοντέλο βελτιώνει τη συμμετοχή των χρηστών σε σχέση με το V-Model. Η ανάπτυξη των πρωτοτύπων, που παρουσιάζονται σε τελικούς χρήστες, είναι ένα από τα βασικά στοιχεία του μοντέλου. Επιπλέον, με την ολοκλήρωση κάθε κύκλου όλοι οι συμμετέχοντες, όπως χρήστης, πελάτης και ομάδα ανάπτυξης, δεσμεύονται στο πλάνο του επόμενου κύκλου.
4. Η κατανόηση της αρχιτεκτονικής του λογισμικού αναγνωρίζεται ως άλλη μία εστία επικινδυνότητας. Εφόσον η αρχιτεκτονική διαγνωστεί ως κίνδυνος, τότε το σπειροειδές μοντέλο επιτρέπει την εστίαση της προσοχής στην αρχιτεκτονική του λογισμικού. Η κατανόηση της αρχιτεκτονικής υποβοηθείται κι από την ανάπτυξη πρωτοτύπων.
5. Η αξιολόγηση της πορείας του έργου βελτιώνεται σε σχέση με το V-Model. Κάθε κύκλος παράγει αποτελέσματα όπως απαιτήσεις, σχέδιο λογισμικού και κώδικα, τα οποία αξιολογούνται. Η ύπαρξη των κύκλων ανάπτυξης δίνει επίσης και βραχυχρόνιους στόχους στην ομάδα ανάπτυξης για τον καλύτερο έλεγχο του έργου.

Μία δυσκολία του σπειροειδούς μοντέλου είναι η έλλειψη βασικών ορόσημων για το μακροχρόνιο χρονοπρογραμματισμό και τη μακροσκοπική αξιολόγηση της πορείας του έργου. Για το λόγο αυτό ορίζονται συμπληρωματικά ορισμένα βασικά ορόσημα που στοχοθετούν την ανάπτυξη και ορίζουν την δέσμευση της ομάδας προς αυτά. Κάθε ορόσημο προϋποθέτει την επίτευξη κάποιων στόχων που σχετίζονται με την ωριμότητα του έργου. Εάν οι στόχοι που ορίζει κάθε ορόσημο δεν ικανοποιούνται, τότε το έργο δεν προχωρά, αλλά επαναλαμβάνεται η εργασία για την επίτευξη των στόχων αυτών. Ο Boehm προτείνει την συμπλήρωση του σπειροειδούς μοντέλου με τρία βασικά ορόσημα τα οποία είναι [11]:

1. Η διαπίστωση της εφικτότητας του υπό ανάπτυξη συστήματος. Αυτό σημαίνει ότι είναι κατανοητό σε αρκετά καλό βαθμό το πρόβλημα που πρέπει να λυθεί, έχουν

καταγραφεί οι βασικές ανάγκες του πελάτη και υπάρχουν αρκετά καλές εκτιμήσεις για το χρόνο και το κόστος της ανάπτυξης.

2. Ο εντοπισμός, η εξάλειψη των βασικών κινδύνων του έργου και ο ασφαλής προσδιορισμός της αρχιτεκτονικής του λογισμικού.
3. Η αρχική παράδοση του λογισμικού στον πελάτη και τους χρήστες.

Το σπειροειδές μοντέλο απαιτεί πολύ καλές ικανότητες διοίκησης του έργου. Ο project manager θα πρέπει να οργανώσει τους κύκλους του σπειροειδούς μοντέλου με τρόπο που να γίνει εφικτή η ολοκλήρωση του έργου εντός του προκαθορισμένου χρονοδιαγράμματος και κόστους, όπως και να έχει ικανότητες διάγνωσης και ανάλυσης των κινδύνων του έργου.

Η πολυπλοκότητα που εισάγει το σπειροειδές μοντέλο στη διοίκηση του έργου αντισταθμίζεται όμως, από τη συνεχή μείωση των κινδύνων. Επομένως, το σπειροειδές μοντέλο είναι κατάλληλο για την ανάπτυξη μεσαίων προς μεγάλων και πολύπλοκων συστημάτων, για τα οποία δεν υπάρχει αρκετή τεχνογνωσία και υπάρχουν πολλές εστίες σοβαρών κινδύνων.

## 2.4 Extreme Programming

Ο Extreme Programming (XP) [12] είναι μία από τις πιο διαδεδομένες και ευέλικτες διαδικασίες ανάπτυξης. Η ανάπτυξη γίνεται επαναληπτικά και οι επαναλήψεις έχουν χρονική διάρκεια μίας εβδομάδας. Ο XP δίνει μεγαλύτερη έμφαση στην κατασκευή του λογισμικού και προτείνει πρακτικές οι οποίες βοηθούν στο να υλοποιηθεί το προϊόν γρήγορα και να παραδοθεί η λειτουργικότητα στον πελάτη. Το μοντέλο αυτό είναι κατάλληλο για μικρές ομάδες έως 10 ατόμων και για μικρά ή μεσαία έργα.

Το κύριο χαρακτηριστικό του XP είναι η ευελιξία και η γρήγορη προσαρμογή στις αλλαγές. Η μεθοδολογία του XP προτείνει την ελαχιστοποίηση των ενδιάμεσων παραγόμενων προϊόντων εκτός αυτών του κώδικα και των ελέγχων. Το μοντέλο βασίζεται σε μεγάλο βαθμό στην επικοινωνία μεταξύ της ομάδας ανάπτυξης και του πελάτη καθώς επίσης και στην επικοινωνία στο εσωτερικό της ομάδας. Για να επιτευχθούν τα παραπάνω, ο XP θέτει κάποιες αξίες κατά την ανάπτυξη του λογισμικού. Οι αξίες αυτές είναι οι ακόλουθες:

1. Επικοινωνία: Η κακή επικοινωνία ή η έλλειψη αυτής θεωρείται ο σημαντικότερος παράγοντας σφαλμάτων. Η φιλοσοφία του XP θεωρεί πως η προσπάθεια για

καλή επικοινωνία μεταξύ των μελών της ομάδας ανάπτυξης και μεταξύ του πελάτη και της ομάδας ανάπτυξης είναι ένα από τα βασικότερα χαρακτηριστικά για την ομαλή πορεία του έργου.

2. Απλότητα: Η φιλοσοφία του XP θεωρεί την απλότητα σε όλες τις εκφάνσεις του έργου ιδιαίτερα σημαντικό στοιχείο. Είναι σημαντικό να σημειωθεί πως η έννοια της απλότητας δε συμπίπτει με την έννοια της απλοϊκότητας. Για να επιτευχθεί η επιθυμητή απλότητα σε όλες τις δραστηριότητες απαιτούνται γνώσεις, εμπειρία και καλή επικοινωνία μεταξύ των εμπλεκόμενων. Το μοντέλο του XP συστήνει να χρησιμοποιούνται απλά εργαλεία και να αποφεύγεται η πρόωρη αναζήτηση βέλτιστων λύσεων.
3. Ανατροφοδότηση: Η συγκεκριμένη αξία του XP λειτουργεί σαν συμπλήρωμα της καλής επικοινωνίας. Πιο συγκεκριμένα, για οποιαδήποτε δραστηριότητα λαμβάνει χώρα, θα πρέπει να υπάρχει η αντίστοιχη ανατροφοδότηση. Για παράδειγμα, σε περίπτωση που ο πελάτης ζητήσει την υλοποίηση μιας λειτουργίας, θα πρέπει η ομάδα ανάπτυξης να του παρέχει ανατροφοδότηση με την εκτίμηση για το χρόνο υλοποίησης της ζητούμενης λειτουργίας.
4. Θάρρος: Η συγκεκριμένη αξία έχει άμεση σχέση με τη στάση απέναντι στην αλλαγή. Σαν αξία από μόνη της δεν αρκεί ωστόσο συνδυαστικά με όλες τις υπόλοιπες βοηθά σε μεγάλο βαθμό στην επίτευξη των επιθυμητών αποτελεσμάτων.
5. Σεβασμός: Ο σεβασμός είναι ανθρώπινη αξία που ενισχύει το ηθικό ομάδων. Πιο συγκεκριμένα, αφορά το σεβασμό που πρέπει να υπάρχει μεταξύ των μελών της ομάδας ανάπτυξης και μεταξύ του πελάτη και της ομάδας.

Ένα ακόμη χαρακτηριστικό του XP είναι πως δεν προτείνει φάσεις, βήματα και αναλυτικά περιεχόμενα για τα προϊόντα που παράγονται. Ο XP δίνει κατά βάση έμφαση στην εφαρμογή κάποιων προτεινόμενων πρακτικών. Οι πρωτεύουσες πρακτικές (primary practices) που προτείνονται από τον XP είναι 13 και οι επακόλουθες πρακτικές (corollary practices) που προτείνονται είναι 10. Στη συνέχεια θα αναλυθούν οι 13 primary πρακτικές και οι 10 corollary.

### **Primary Practices [13]**

#### 1. Sit together

Η ομάδα μοιράζεται έναν ενιαίο χώρο εργασίας. Η ύπαρξη κοινού χώρου εργασίας ενθαρρύνει τις διαπροσωπικές σχέσεις και την μετάδοση πληροφορίας.

2. Whole team

Όλες οι ευέλικτες διαδικασίες έχουν σαν βασικό χαρακτηριστικό το ότι προάγουν την επικοινωνία. Για σωστή επικοινωνία θα πρέπει η ομάδα να αποτελείται από μέλη που κατέχουν τις δεξιότητες οι οποίες θα οδηγήσουν στην επιτυχία του έργου.

3. Informative workspace

Ο χώρος εργασίας πρέπει να περιέχει ενημερωτικό υλικό, όπως για παράδειγμα πίνακες με την πορεία του έργου.

4. Energized work

Η ομάδα ανάπτυξης πρέπει να είναι ανανεωμένη και να μην υπάρχει μια συνεχής αίσθηση κούρασης. Ο XP συστήνει τον περιορισμό των υπερωριών ώστε τα μέλη της ομάδας να είναι ικανοποιημένα και να έχουν προσωπικό χρόνο.

5. Pair programming

Ο XP συστήνει την ανάπτυξη κώδικα από δύο άτομα στον ίδιο υπολογιστή [14].

6. Stories

Κατά αυτόν τον τρόπο περιγράφονται οι λειτουργίες του συστήματος. Πρόκειται για μικρές περιγραφές που αφορούν λειτουργίες που είναι προφανείς στους πελάτες.

7. Weekly cycle

Η ανάπτυξη του έργου πραγματοποιείται σταδιακά σε εβδομάδες. Στην αρχή κάθε εβδομάδας πραγματοποιείται ένα meeting με τον πελάτη. Στο συγκεκριμένο meeting, ο πελάτης αποφασίζει ποια stories θα αναπτυχθούν μέσα στην εβδομάδα.

8. Quarterly cycle

Πρόκειται για meetings που πραγματοποιούνται κάθε 3 μήνες ώστε να συζητηθεί η πορεία του έργου.

9. Slack

Ο XP προτείνει την προσθήκη εργασιών στο εβδομαδιαίο πλάνο οι οποίες μπορούν να αναβληθούν. Με αυτόν τον τρόπο, σε περίπτωση που προκύψει κάποιο σφάλμα ή κάποια καθυστέρηση κατά τη διάρκεια του κύκλου, θα υπάρχουν εργασίες που θα μπορούν να αναβληθούν χωρίς να επηρεάζεται σε μεγάλο βαθμό ο πελάτης.

10. Ten-minute build

Το build και το testing του έργου πρέπει να ολοκληρώνονται σε 10 λεπτά ώστε να είναι εύκολη η λήψη feedback.

#### 11. Continuous integration

Τα μέλη της ομάδας ανάπτυξης πρέπει να πραγματοποιούν το integration του συστήματος κάθε δύο ώρες ώστε να μειωθούν τα προβλήματα που παρουσιάζονται κατά το integration.

#### 12. Test-first programming

Πριν την ανανέωση και την προσθήκη νέου κώδικα είναι απαραίτητος ο έλεγχος του κώδικα που έχει ήδη υλοποιηθεί.

#### 13. Incremental design

Η ανάπτυξη του design γίνεται σταδιακά κατά τη διάρκεια του development.

### **Corollary Practices [13]**

#### 1. Real customer involvement

Η συγκεκριμένη πρακτική απαιτεί την ανατροφοδότηση από τον πελάτη τουλάχιστον μια φορά την εβδομάδα. Ο στόχος της είναι η αποφυγή ή η μείωση του development σε περιπτώσεις όπου δεν είναι απόλυτα σίγουρο το τι θέλει ο πελάτης.

#### 2. Incremental deployment

Με τη συγκεκριμένη πρακτική, η αντικατάσταση υπάρχοντων συστημάτων ή προϊόντων γίνεται σταδιακά. Τα μεγάλα deployments έχουν μεγάλο ρίσκο και συνήθως δημιουργούν προβλήματα. Σε αντίθεση με την ολική αντικατάσταση, το XP μοντέλο συστήνει τις σταδιακές αλλαγές και την παράλληλη χρήση κάποιων συστημάτων για αυξημένη ασφάλεια. Η παραπάνω διαδικασία έχει σε κάποιες περιπτώσεις διπλάσιο effort, ωστόσο παρέχει μεγαλύτερη ασφάλεια τόσο στην ομάδα ανάπτυξης όσο και στον πελάτη.

#### 3. Team continuity

Η καλή επικοινωνία μεταξύ της ομάδας βοηθάει στο να είναι πιο αποδοτική. Το μοντέλο XP θεωρεί θετικό χαρακτηριστικό την εμπλοκή όλων των μελών σε κάποια εργασία καθώς κατά αυτόν τον τρόπο μεταδίδεται η γνώση και αυξάνεται η εμπειρία στο εσωτερικό της ομάδας και της εταιρίας.

#### 4. Shrinking teams

Οι αποδοτικές ομάδες βελτιώνονται με το πέρασμα του χρόνου και τελικά καταφέρνουν να παράγουν περισσότερο όγκο εργασίας σε λιγότερο χρόνο. Η

συγκεκριμένη πρακτική προτείνει τη συνεχή και σταθερή πορεία του φόρτου εργασίας και τη μείωση των μελών της ομάδας μόλις φτάσει στο επιθυμητό επίπεδο.

#### 5. Root-cause analysis

Όταν γίνεται κάποιο λάθος κατά τη διαδικασία του development, η βέλτιστη λύση δεν είναι η προσωρινή λύση του. Θα πρέπει να βρεθεί μια μόνιμη λύση καθώς επίσης θα πρέπει η ομάδα να κατανοήσει το λόγο για τον οποίο προκλήθηκε το πρόβλημα ώστε να μην επαναληφθεί.

#### 6. Shared code

Οποιοδήποτε μέλος της ομάδας ανάπτυξης θα πρέπει να μπορεί να έχει πρόσβαση σε οποιοδήποτε μέρος του συστήματος και σε οποιοδήποτε προϊόν της ομάδας. Παρ' όλα αυτά για να γίνει αυτό, θα πρέπει η ομάδα να είναι πολύ καλά οργανωμένη κι όλα τα μέλη της να έχουν αίσθηση της υπευθυνότητας.

#### 7. Code and tests

Ο κώδικας και τα tests είναι οι μόνες μόνιμες και ακριβείς περιγραφές του συστήματος. Το μοντέλο XP συστήνει τα δύο παραπάνω να χειρίζονται με ιδιαίτερη προσοχή και να είναι πάντα ενημερωμένα με την τελευταία κατάσταση του παραγόμενου προϊόντος.

#### 8. Single code base

Η ομάδα θα πρέπει να δουλεύει με βάση ένα κώδικα καθώς η ύπαρξη πολλών διαφορετικών εκδόσεων κώδικα μπορεί να καταλήξει μελλοντικά σε προβλήματα ή ακόμα και σε διπλό effort.

#### 9. Daily deployment

Οι διαφορές μεταξύ του κώδικα της παραγωγής και του κώδικα για το development θα πρέπει να είναι μικρές. Με αυτό τον τρόπο, αποφεύγεται το αυξημένο ρίσκο που προκύπτει από μεγάλο όγκο αλλαγών και βοηθάει στο να έχει η ομάδα γρηγορότερη ανατροφοδότηση. Για να επιτευχθεί το παραπάνω, θα πρέπει να γίνονται συχνά deployments στην παραγωγή.

#### 10. Negotiated scope contract

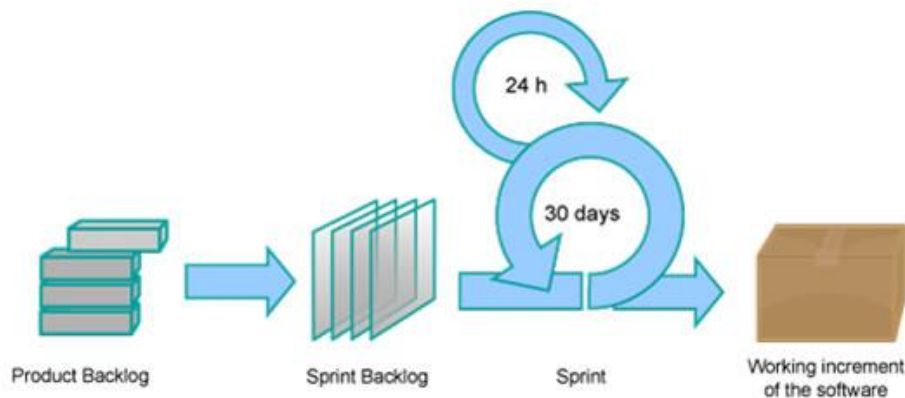
Το μοντέλο XP συστήνει να αποφεύγονται τα μεγάλης διάρκειας συμβόλαια με τον πελάτη. Προτείνει την πραγματοποίηση μικρών releases και την υπογραφή πολλών ξεχωριστών μικρών συμβολαίων. Με αυτό τον τρόπο υπάρχει μεγαλύτερη ευελιξία τόσο στην ανάπτυξη του κώδικα όσο και στην αντιμετώπιση των αλλαγών.



## 2.5 Scrum

Η μέθοδος Scrum πραγματοποιεί τη μοντελοποίηση του έργου και την εύκολη διοίκησή του χωρίς να προσφέρει λεπτομερείς προτάσεις για άλλες δραστηριότητες όπως είναι οι απαιτήσεις και η διοίκηση. Οι βασικότερες λειτουργίες της Scrum είναι οι ακόλουθες:

- Επαναλήψεις 30 ημερών που ονομάζονται sprints.
- Αυτο-οργανούμενες ομάδες.
- Ημερήσιες συσκέψεις της ομάδας για την πρόοδο του έργου.
- Επίδειξη της λειτουργικότητας του λογισμικού στους ενδιαφερόμενους του έργου στο τέλος μιας επανάληψης.
- Προσαρμοστικός σχεδιασμός του έργου στην αρχή κάθε επανάληψης.



Εικόνα 4: Επαναληπτική Διαδικασία Scrum

Πιο συγκεκριμένα, η Scrum είναι μια επαναληπτική μέθοδος η οποία αποτελείται από επαναλήψεις που έχουν χρονική διάρκεια 30 ημερών (sprints). Η μέθοδος απευθύνεται σε ομάδες ανάπτυξης έως 7 ατόμων και το βασικό της στοιχείο είναι πως οι ομάδες αυτές είναι αυτο-οργανούμενες. Επιπλέον, στο εσωτερικό της ομάδας δεν υπάρχει κάποιος project manager ο οποίος λαμβάνει τις αποφάσεις κι απλά τις μεταφέρει αλλά αντίθετα η λήψη αποφάσεων γίνεται από το σύνολο της ομάδας επομένως κι η ευθύνη υλοποίησης των αποφάσεων ανήκει στην ομάδα [15].

Στη συνέχεια, θα αναλυθούν οι βασικοί ρόλοι του έργου ανάπτυξης, τα προϊόντα που συντηρούνται κι ο τρόπος με τον οποίο οργανώνονται οι επαναλήψεις.

## Βασικοί Ρόλοι

Οι βασικοί ρόλοι σε ένα έργο ανάπτυξης είναι ο ιδιοκτήτης του προϊόντος (Product Owner), ο Scrum Master και η ομάδα ανάπτυξης (Team). Ο product owner είναι το πρόσωπο ή η ομάδα που εκπροσωπεί τον πελάτη και αναλαμβάνει να περιγράψει και να θέσει τις προτεραιότητες για τις απαιτήσεις. Ο Scrum Master έχει μια αντίστοιχη θέση με τον project manager ωστόσο ο ρόλος του έχει κατά βάση ως στόχο την επικοινωνία μεταξύ της ομάδας ανάπτυξης και των πελατών. Με τη βελτίωση της παραπάνω επικοινωνίας απομακρύνονται πιθανά εμπόδια κι ασάφειες κατά την ανάπτυξη του έργου. Όσον αφορά την ομάδα, ο ρόλος της, εκτός από το να υλοποιήσει το ζητούμενο προϊόν, περιλαμβάνει και το να είναι υπεύθυνη για τη λήψη αποφάσεων. Πιο αναλυτικά, οι αρμοδιότητες των παραπάνω ρόλων περιλαμβάνουν τα εξής:

### 1. Product Owner:

- a. Αντιπροσωπεύει τα συμφέροντα των πελατών.
- b. Έχει λόγο για την εξέλιξη του project.
- c. Είναι αρμόδιος για τη χρηματοδότηση του έργου.
- d. Είναι υπεύθυνος για τη δημιουργία των αρχικών απαιτήσεων και των στόχων απόδοσης της επένδυσης.
- e. Είναι υπεύθυνος για τη διάδοση του προϊόντος στην αγορά.

### 2. Scrum Master:

- a. Είναι υπεύθυνος για τη διαδικασία Scrum.
- b. Διδάσκει τη μέθοδο Scrum σε όλους όσοι συμμετέχουν στο project.
- c. Προστατεύει την ομάδα από εμπόδια.
- d. Βοηθά θέτοντας ερωτήματα και παρέχοντας συμβουλές.
- e. Λειτουργεί με βάση την κουλτούρα του οργανισμού.

### 3. Team:

- a. Είναι αρμόδια για την ανάπτυξη της λειτουργικότητας.
- b. Έχει πολλαπλές λειτουργίες, είναι αυτο-διοικούμενη κι αυτο-οργανούμενη.
- c. Τα μέλη της ομάδας είναι συλλογικά υπεύθυνα για την επιτυχία κάθε επανάληψης όπως επίσης και του project συνολικά.

## Προϊόντα

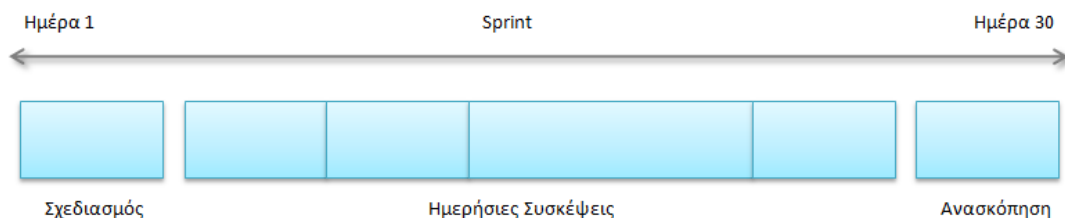
Η μέθοδος Scrum προβλέπει έναν πολύ μικρό αριθμό προϊόντων τα οποία εστιάζονται περισσότερο στη διοίκηση του έργου.

Οι λειτουργικές και μη λειτουργικές απαιτήσεις συγκεντρώνονται στον κατάλογο εκκρεμοτήτων του προϊόντος (Product Backlog). Ο κατάλογος αυτός περιέχει τα λειτουργικά χαρακτηριστικά του προϊόντος (features), τις περιπτώσεις χρήσης και μη λειτουργικές απαιτήσεις. Ο Product Owner συντηρεί τον κατάλογο και δίνει προτεραιότητα στην υλοποίηση των απαιτήσεων που του προσφέρουν τη μεγαλύτερη αξία. Το product backlog εξελίσσεται και αλλάζει βάσει της προόδου του έργου αλλά και βάσει των αλλαγών που προκύπτουν από το περιβάλλον του έργου. Η οργάνωση των καταχωρήσεων στο product backlog γίνεται με τρόπο που να προσδιορίζουν τις εκδόσεις κυκλοφορίας (releases) κάθε λογισμικού. Μια εναλλακτική στο παραπάνω είναι η δημιουργία ενός υποκαταλόγου, δηλαδή ενός καταλόγου εκκρεμοτήτων της κυκλοφορίας (release backlog) ο οποίος περιλαμβάνει καταχωρήσεις οι οποίες σχεδιάζονται για το επόμενο release.

Το δεύτερο προϊόν από το οποίο αποτελείται η μέθοδος Scrum είναι ο κατάλογος εκκρεμοτήτων της επανάληψης (sprint backlog). Το sprint backlog περιλαμβάνει καταχωρήσεις οι οποίες συμπεριλαμβάνονται στην επανάληψη καθώς επίσης κι όλα τα tasks που θα πραγματοποιηθούν στην τρέχουσα επανάληψη. Ο στόχος της ομάδας είναι να μετατρέψει τις καταχωρήσεις του product backlog σε tasks διάρκειας 4 έως 16 ωρών. Τέλος, στο sprint backlog περιλαμβάνονται και οι αναθέσεις των tasks σε μέλη της ομάδας.

## Επαναλήψεις

Η μέθοδος Scrum δίνει μεγάλη έμφαση στις διαπροσωπικές σχέσεις και στην επικοινωνία μεταξύ των μελών της ομάδας. Για αυτό το λόγο, ο συντονισμός κι ο σχεδιασμός του έργου γίνεται μέσω συσκέψεων οι οποίες είναι χρονικά πλαισιωμένες [16].



**Εικόνα 5: Συσκέψεις στη Scrum**

Στη μέθοδο του Scrum, τα είδη συσκέψεων που πραγματοποιούνται είναι τα ακόλουθα:

- Σύσκεψη σχεδιασμού.
- Ημερήσια σύσκεψη.
- Σύσκεψη ανασκόπησης επανάληψης.
- Σύσκεψη αναθεώρησης επανάληψης.

Πιο αναλυτικά, η σύσκεψη σχεδιασμού γίνεται στην αρχή μια επανάληψης κι αφορά το σχεδιασμό. Σε αυτήν παίρνουν μέρος η ομάδα ανάπτυξης, ο Scrum Master κι ο Product Owner. Πριν τη σύσκεψη ο Product Owner πρέπει να έχει προετοιμάσει το product backlog και κατά τη διάρκεια της σύσκεψης πρέπει να δώσει προτεραιότητα και να επιλέξει κάποιες από τις καταχωρήσεις του product backlog ώστε να υλοποιηθούν. Η ομάδα ανάπτυξης κάνει εκτιμήσεις για το χρόνο που χρειάζεται η υλοποίηση ώστε να καταλήξουν όλοι μαζί στο τέλος της σύσκεψης στο ποιες καταχωρήσεις του product backlog θα υλοποιηθούν. Αφού η ομάδα κι ο Product Owner καταλήξουν στις καταχωρήσεις προς υλοποίηση, δημιουργείται το sprint backlog το οποίο εμπλουτίζεται με την ανάθεση των tasks στα μέλη της ομάδας και το οποίο μπορεί να τροποποιείται από τα μέλη της ομάδας καθώς προσδιορίζει το έργο της τρέχουσας επανάληψης. Τέλος, η σύσκεψη σχεδιασμού διαρκεί περίπου 8 ώρες και μόλις ολοκληρωθεί σε αυτήν ο σχεδιασμός της επανάληψης μπορούν να ξεκινήσουν οι δραστηριότητες ανάπτυξης.

Όσον αφορά την ημερήσια σύσκεψη (daily scrum) διαρκεί 15 λεπτά. Οι ημερήσιες συσκέψεις διεξάγονται από τον Scrum Master και την ομάδα ανάπτυξης και κατά τη διάρκεια αυτών όλα τα μέλη είναι όρθια (stand-up meetings). Κατά τη διάρκεια των ημερήσιων stand-up meetings κάθε μέλος πρέπει να απαντήσει στα τρία ακόλουθα ερωτήματα:

1. Τι έκανε για το έργο από το προηγούμενο daily scrum;
2. Τι θα κάνει έως το επόμενο daily scrum;
3. Ποια είναι τα εμπόδια που έχει συναντήσει για την επίτευξη του στόχου της τρέχουσας επανάληψης;

Ο κύριος στόχος των daily scrum είναι να γίνεται καθημερινός έλεγχος της πορείας του έργου.

Αφού ολοκληρωθεί κάθε επανάληψη, πραγματοποιείται η σύσκεψη ανασκόπησης (sprint review) η οποία έχει διάρκεια 4 ωρών. Στη συνάντηση αυτή, η ομάδα ανάπτυξης

παρουσιάζει στον Product Owner και σε άλλους ενδιαφερόμενους τη λειτουργικότητα του προϊόντος που έχει αναπτύξει κατά τη διάρκεια του sprint. Ο Product Owner μπορεί να αποφασίσει αν η έκδοση του προϊόντος που του παρουσιάζεται μπορεί να τεθεί σε παραγωγική λειτουργία.

Τέλος, στις υπολειπόμενες 4 ώρες διεξάγεται η σύσκεψη αναθεώρησης (sprint retrospective) κατά τη διάρκεια της οποίας ο Scrum Master κι η ομάδα ανάπτυξης πραγματοποιούν μια ανασκόπηση των πρακτικών που ακολούθησαν. Εάν το αποτέλεσμα είναι η πιθανή αναθεώρηση αυτών των πρακτικών τότε είναι αρμοδιότητα της ομάδας να αποφασίσει το ποιες πρακτικές τεχνικού χαρακτήρα θα χρησιμοποιηθούν καθώς ο Scrum Master δεν επιβάλλει τον τρόπο με τον οποίο θα υλοποιηθεί η ανάπτυξη του λογισμικού.

Όσον αφορά την αξιολόγηση της Scrum, είναι εμφανώς μια agile μέθοδος, στοιχείο που φαίνεται από τα παρακάτω:

1. Το χρονικό διάστημα της κάθε επανάληψης δεν είναι προκαθορισμένο και καθορίζεται βάσει των αναγκών της τρέχουσας επανάληψης.
2. Κατά τη διάρκεια του χρονικού διαστήματος της επανάληψης, η ομάδα ανάπτυξης αναλαμβάνει ένα κομμάτι της τελικής λειτουργικότητας του προϊόντος και προσπαθεί να κατανοήσει πως θα το φέρει εις πέρας. Με αυτόν τον τρόπο, η ομάδα επικεντρώνεται μόνο σε αυτούς τους στόχους και όχι στα γενικότερα παραδοτέα του προϊόντος.
3. Το γεγονός ότι σε κάθε επανάληψη δημιουργείται ένα προϊόν το οποίο ελέγχεται έχει ως αποτέλεσμα ο αριθμός των αδυναμιών του τελικού προϊόντος να μην είναι ποτέ μεγάλος.
4. Ο τρόπος με τον οποίο είναι δομημένη η ομάδα ανάπτυξης, δηλαδή το ότι είναι αυτο-διοικούμενη και αυτο-οργανούμενη έχει σαν αποτέλεσμα να είναι ιδιαίτερα ευέλικτη και να λειτουργεί βάσει των αναγκών της τρέχουσας υλοποίησης.

### 3. ΔΙΑΧΕΙΡΙΣΗ ΡΙΣΚΟΥ

#### 3.1 Θεωρία Διαχείρισης Ρίσκου

Διαχείριση κινδύνου (Risk Management - RM) ενός έργου ορίζεται ως «το σύνολο των διαδικασιών αναγνώρισης, ανάλυσης, ανταπόκρισης και παρακολούθησης των κινδύνων κατά τη διάρκεια της ζωής ενός έργου, με στόχο την επίτευξη των αρχικών στόχων του» [17].

Η διαχείριση κινδύνου είναι κεντρικός πυρήνας της διαχείρισης στρατηγικής κάθε οργανισμού. Είναι η διεργασία με την οποία οι οργανισμοί προσεγγίζουν μεθοδικά τους κινδύνους που σχετίζονται με τις δραστηριότητές τους με σκοπό την επίτευξη αειφόρου οφέλους σε κάθε δραστηριότητα [18].

Το επίκεντρο της καλής διαχείρισης κινδύνου είναι η αναγνώριση και ο χειρισμός αυτών των κινδύνων. Ταξινομεί την κατανόηση των πιθανών κερδών και απειλών όλων εκείνων των παραγόντων που μπορούν να επηρεάσουν τον οργανισμό. Αυξάνει την πιθανότητα επιτυχίας και μειώνει αμφότερα την πιθανότητα αποτυχίας και την αβεβαιότητα επίτευξης των συνολικών στόχων του οργανισμού.

Η διαχείριση κινδύνου προστατεύει και προσθέτει αξία στον οργανισμό και στους ενδιαφερομένους (stakeholders) υποστηρίζοντας τους στόχους του οργανισμού με:

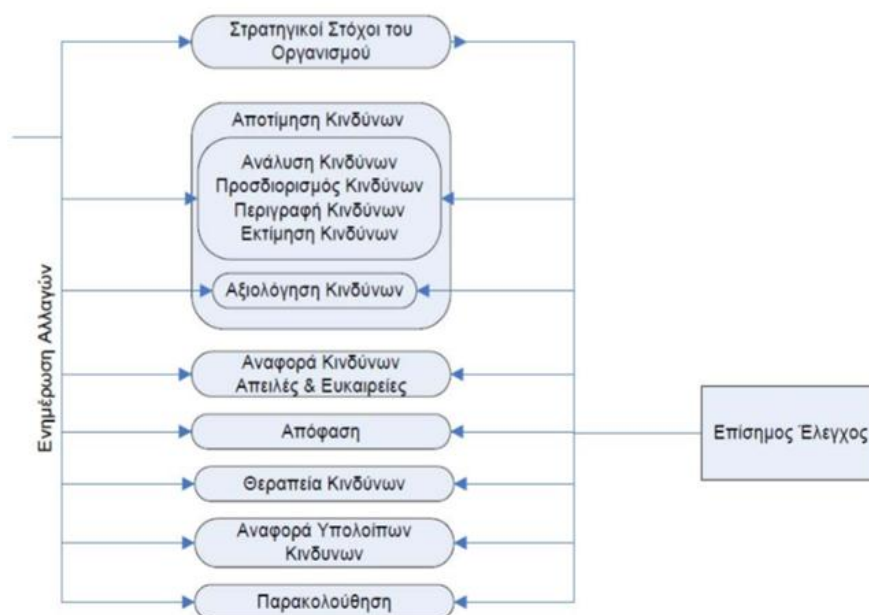
- την παροχή ενός πλαισίου στον οργανισμό το οποίο καθιστά δυνατή την ύπαρξη μελλοντικής δραστηριότητας με σταθερό και ελεγχόμενο τρόπο,
- τη βελτίωση της λήψης αποφάσεων, του προγραμματισμού και του καθορισμού προτεραιότητας μέσω της ευρείας και δομημένης κατανόησης της επιχειρησιακής δραστηριότητας, της αστάθειας και των ευκαιριών / απειλών των έργων,
- την συμβολή σε μία πιο αποδοτική χρήση / κατανομή του κεφαλαίου και των πόρων εντός του οργανισμού,
- τη μείωση της μεταβλητότητας στις μη βασικές επιχειρησιακές περιοχές,
- την προστασία και βελτίωση των παγίων και της εικόνας της εταιρείας,
- την ανάπτυξη και την υποστήριξη των ανθρώπων και της βάσης γνώσης του οργανισμού,
- τη βελτιστοποίηση της λειτουργικής αποδοτικότητας.

Ο χειρισμός κινδύνου είναι η διεργασία της επιλογής και εφαρμογής μέτρων για να τροποποιηθεί ο κίνδυνος. Ο χειρισμός κινδύνου περιλαμβάνει, ως το πιο σημαντικό του στοιχείο, τον έλεγχο και τη μείωση κινδύνου, αλλά εκτείνεται ακόμη, στην αποφυγή κινδύνου, τη μεταφορά κινδύνου, τη χρηματοδότηση κινδύνου, κλπ.

Οποιοδήποτε σύστημα χειρισμού κινδύνου θα έπρεπε να παρέχει, ως το ελάχιστο:

- αποτελεσματική και αποδοτική λειτουργία του οργανισμού,
- αποτελεσματικά εσωτερικά μέτρα ελέγχου,
- συμμόρφωση με νόμους και κανονισμούς.

Η διεργασία ανάλυσης κινδύνου βοηθάει την αποτελεσματική και αποδοτική λειτουργία του οργανισμού με την αναγνώριση εκείνων των κινδύνων οι οποίοι απαιτούν την προσοχή της διοίκησης. Η διοίκηση θα πρέπει να θέσει προτεραιότητα στις ενέργειες ελέγχου κινδύνου με βάση τη δυνατότητά τους να ωφελήσουν τον οργανισμό.



**Εικόνα 6: Η διεργασία Διαχείρισης Κινδύνου**

Οι κίνδυνοι που αντιμετωπίζει ένας οργανισμός και οι λειτουργίες του μπορεί να είναι αποτέλεσμα παραγόντων αμφότερα εξωτερικών και εσωτερικών του οργανισμού. Ως κίνδυνοι θα μπορούσαν να οριστούν οι παράγοντες που επηρεάζουν την λειτουργικότητα του Business Plan και πολλές φορές καθιστούν αδύνατη την υλοποίησή

του. Η πιθανότητα κάποιου ανεπιθύμητου γεγονότος σε κάποια χρονική στιγμή ορίζεται ως μονάδα κινδύνου που έχει σαν αποτέλεσμα αυξημένο κόστος. Οι κίνδυνοι πλέον είναι ποικιλόμορφοι και μπορούν να συνοψιστούν σε τρεις μεγάλες κατηγορίες:

- χρηματοοικονομικοί κίνδυνοι,
- λειτουργικοί κίνδυνοι και
- κίνδυνοι αγοράς.

Οι κίνδυνοι είναι πολυδιάστατοι και αλληλένδετοι. Η αντίληψη λοιπόν του κινδύνου από κάθε διοικητικό στέλεχος είναι διαφορετική. Εξαρτάται από τον τύπο του κινδύνου, το πεδίο του κινδύνου και την επίδρασή του. Κάθε στέλεχος δίνει διαφορετική έμφαση από οργανισμό σε οργανισμό.

Η διαχείριση κινδύνων είναι βασισμένη στον προσδιορισμό, την ανάλυση, τον έλεγχο και την όσο γίνεται πιο αποτελεσματική αντιμετώπιση εκείνων των γεγονότων που έχουν την δυνατότητα να προκαλούν μια ανεπιθύμητη αλλαγή στο Business Plan. Είναι λοιπόν απαραίτητη η ανάπτυξη της κατάλληλης, για κάθε παράγοντα κινδύνου, μεθοδολογίας για την αξιολόγηση και αντιμετώπιση του κινδύνου με την προσφορά κάποιας εναλλακτικής επιλογής η οποία να επιτρέπει και να ενθαρρύνει το Business Plan να αναπτυχθεί παρ' όλες τις διορθωτικές αλλαγές.

### **3.2 Εμφάνιση κινδύνου σε έργα ανάπτυξης λογισμικού**

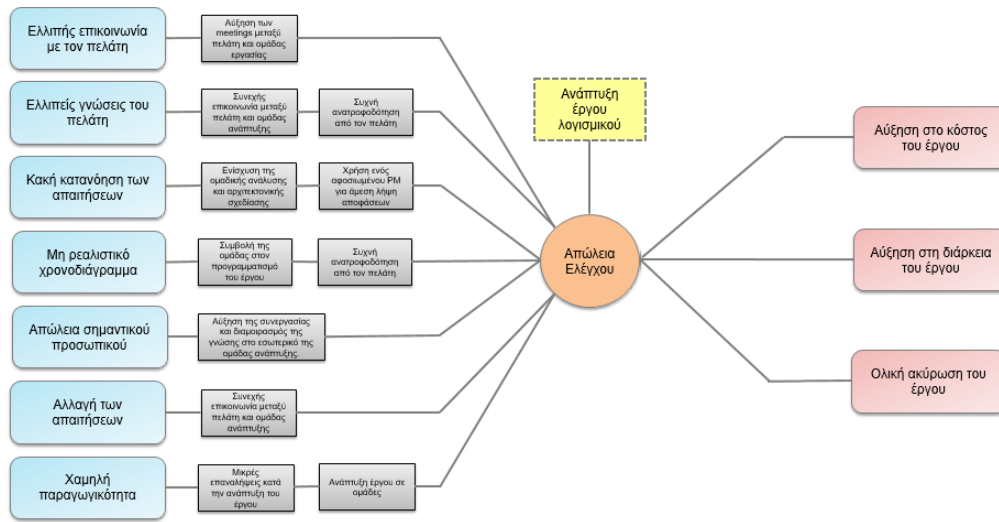
Οι κίνδυνοι μπορεί να προκύψουν από διάφορες πηγές όπως για παράδειγμα από αβεβαιότητα στην αγορά, απειλές από σφάλματα που προκύπτουν στις διάφορες φάσεις του έργου, ατυχήματα ή ακόμη και φυσικές καταστροφές. Σύμφωνα με τον Boehm οι σημαντικότεροι και πιθανότεροι κίνδυνοι στα έργα ανάπτυξης λογισμικού είναι οι ακόλουθοι [19]:

- Έλλειψη προσωπικού
- Μη ρεαλιστικό χρονοδιάγραμμα
- Κακή κατανόηση των απαιτήσεων
- Δύσχρηστη διεπαφή του χρήστη με το λογισμικό
- Λογισμικό υπερβολικά περίπλοκο για τις ανάγκες του πελάτη
- Μη ύπαρξη ελέγχου σε αλλαγές απαιτήσεων
- Προβλήματα σε επαναχρησιμοποιούμενα εξαρτήματα ή σε εξωτερικό λογισμικό
- Προβλήματα σε έργα που εκτελούνται από τρίτους



- Χαμηλός χρόνος απόκρισης
- Έργο πέρα από τις δυνατότητες της τεχνολογίας

Στο Σχήμα 1 πραγματοποιείται μια bow tie analysis με σκοπό την καλύτερη παρουσίαση των πιθανών απειλών. Στο συγκεκριμένο διάγραμμα παρουσιάζονται οι πιθανοί κίνδυνοι που μπορεί να προκύψουν καθώς και κάποια μέτρα αποφυγής των συγκεκριμένων κινδύνων. Επιπλέον, παρουσιάζονται τα πιθανά αποτελέσματα σε περίπτωση μη διαχείρισης των παραπάνω κινδύνων.



Σχήμα 1: Bow tie analysis για έργο ανάπτυξης λογισμικού

Στη συγκεκριμένη εργασία, τα μοντέλα που θα μελετηθούν για την ανάπτυξη ενός έργου λογισμικού είναι το V-Model, η Spiral, η XP και η Scrum. Το καθένα από τα συγκεκριμένα μοντέλα διαχειρίζεται με διαφορετικό τρόπο κάποιους από τους παραπάνω κινδύνους.

Αναλυτικότερα, το V-Model ορίζει εξαρχής την ανάπτυξη των test cases που θα ελεγχθούν κατά το στάδιο του testing, στοιχείο που βοηθάει στην μείωση των λαθών κατά την κατανόηση των απαιτήσεων. Όσον αφορά τη Spiral, η ανάπτυξη πρωτοτύπων βοηθάει σε μεγάλο βαθμό την εξάλειψη των κινδύνων. Σε κάθε κύκλο εντοπίζονται οι κίνδυνοι κι ο τρόπος ανάπτυξης του πρωτοτύπου πραγματοποιείται με στόχο την εξάλειψή τους. Επιπλέον, η παρουσίαση του πρωτοτύπου στον πελάτη ενισχύει την καλύτερη επικοινωνία μεταξύ της ομάδας ανάπτυξης και του πελάτη. Τέλος, οι διαδικασίες που λαμβάνουν χώρα στη Spiral απαιτούν πολύ καλές ικανότητες διοίκησης χαρακτηριστικό που βοηθάει στην οργάνωση της ομάδας και στην καλύτερη διαχείριση των απαιτήσεων.

Όσον αφορά τα agile μοντέλα, δηλαδή την XP και τη Scrum, η εξάλειψη των κινδύνων ενισχύεται σε μεγάλο βαθμό από το γεγονός ότι τα μοντέλα προωθούν την καλή επικοινωνία όλων των μελών του έργου.

Πιο συγκεκριμένα, οι πρακτικές που εφαρμόζονται στην XP έχουν ως στόχο την καλή επικοινωνία στο εσωτερικό της ομάδας ανάπτυξης. Για να επιτευχθεί το παραπάνω υπάρχουν διάφορες κύριες αλλά και δευτερεύουσες πρακτικές οι οποίες συστήνουν την ανάπτυξη λογισμικού σε ζευγάρια, την ύπαρξη ενός ενιαίου χώρου για όλη την ομάδα, τον περιορισμό των υπερωριών και την εμπλοκή όλων των μελών της ομάδας στις διάφορες δραστηριότητες. Επιπλέον, υπάρχουν διάφορες πρακτικές οι οποίες βοηθούν στην επικοινωνία μεταξύ της ομάδας ανάπτυξης και του πελάτη. Οι πρακτικές αυτές περιλαμβάνουν την ανάπτυξη συνοπτικών περιγραφών για διάφορες λειτουργίες που παρουσιάζονται στον πελάτη στην αρχή κάθε εβδομάδας με σκοπό να αποφασίσει σε ποιες λειτουργίες θα δώσει προτεραιότητα, τα τριμηνιαία meetings με τον πελάτη με σκοπό τον έλεγχο της πορείας του έργου και τέλος την εβδομαδιαία ανατροφοδότηση από τον πελάτη, χαρακτηριστικό που ενισχύει τον σωστό χρονοπρογραμματισμό. Τέλος, το γεγονός ότι η ανάπτυξη του έργου ολοκληρώνεται με εβδομαδιαίες επαναλήψεις ενισχύει την παραγωγικότητα των εργαζομένων.

Αντίστοιχα, η Scrum επιτυγχάνει τη βελτίωση της επικοινωνίας δίνοντας πρωτεύοντα ρόλο στην ύπαρξη meetings. Αναλυτικότερα, η πραγματοποίηση των ημερήσιων συσκέψεων αλλά και της σύσκεψης αναθεώρησης ενισχύουν την καλή επικοινωνία στο εσωτερικό της ομάδας. Επιπλέον, η λογική της αυτο-οργανούμενης ομάδας βοηθάει στην καλύτερη οργάνωση του έργου και στην εμπλοκή όλων των μελών της ομάδας στον προγραμματισμό των επόμενων κινήσεων. Ένα ακόμη χαρακτηριστικό είναι πως η ανάπτυξη του έργου γίνεται σε επαναλήψεις οι οποίες διαρκούν ένα μήνα, στοιχείο που βελτιώνει την παραγωγικότητα της ομάδας ανάπτυξης. Την καλή οργάνωση της ομάδας και την καλύτερη κατανόηση των απαιτήσεων του πελάτη ενισχύει και η ύπαρξη του Scrum Master ο οποίος έχει ως στόχο την προστασία της ομάδας από εμπόδια. Τέλος, η καλή επικοινωνία με τον πελάτη ενισχύεται από την ύπαρξη των συσκέψεων σχεδιασμού και των συσκέψεων ανασκόπησης στις οποίες παίρνουν μέρος τόσο ο πελάτης όσο και η ομάδα ανάπτυξης.

Συμπερασματικά, για τη διαχείριση της εμφάνισης κινδύνου είναι απαραίτητη η μοντελοποίηση των παραπάνω μεθόδων και η χρήση κάποιας διαδικασίας για τον ποσοτικό προσδιορισμό του κινδύνου. Οι τεχνικές ποσοτικού προσδιορισμού έχουν ως

στόχο την εκτίμηση της πιθανότητας να συμβεί κάποιο σφάλμα και στη συνέχεια τον προσδιορισμό της έκτασης της επίπτωσης του σφάλματος που έχει προκύψει. Με δεδομένο ότι το περιβάλλον της ανάπτυξης λογισμικού επηρεάζεται από πολλούς παράγοντες, η διαδικασία που πρέπει να χρησιμοποιηθεί για τον ποσοτικό προσδιορισμό του κινδύνου απαιτεί τη χρήση τυχαιότητας. Στην παρούσα εργασία, η μέθοδος που θα χρησιμοποιηθεί για την προσομοίωση και την αξιολόγηση των πιθανών αποτελεσμάτων είναι η Monte Carlo. Επιπλέον, για τον εντοπισμό των μεταβλητών που επηρεάζουν σε μεγαλύτερο βαθμό το τελικό αποτέλεσμα θα πραγματοποιηθεί μία Sensitivity ανάλυση.

### 3.3 Ανάλυση Monte Carlo

Η Monte Carlo μέθοδος (ή πείραμα / προσομοίωση Monte Carlo) είναι μια στοχαστική διαδικασία όπου με χρήση τυχαίων αριθμών και τη στατιστική προσπαθεί να λύσει ένα πρόβλημα. Παρουσιάστηκε το 1949 με την δημοσίευση των N. Metropolis και S. Ulam "Η μέθοδος Monte Carlo" στο περιοδικό Journal of the American Statistics Association [20] και αποτελεί την πιο διαδεδομένη μέθοδο στατιστικής δειγματοληψίας.

Λόγω της αποτελεσματικότητάς της και της γενικής της εφαρμογής είναι ιδιαίτερα χρήσιμη στη μελέτη συστημάτων με μεγάλο βαθμό ελευθερίας. Υπάγεται σε μια κατηγορία υπολογιστικών αλγορίθμων που στηρίζονται σε επαναλαμβανόμενες τυχαίες δειγματοληψίες για τον υπολογισμό των αποτελεσμάτων. Γενικότερα, είναι χρήσιμη για τη μοντελοποίηση φαινομένων με σημαντική αβεβαιότητα και είναι ευρέως επιτυχής στην ανάλυση κινδύνου σε σύγκριση με εναλλακτικές μεθόδους ή την ανθρώπινη διαίσθηση. Έχει εφαρμοστεί για την εξερεύνηση και εκμετάλλευση του πετρελαίου, την πραγματική παρατήρηση βλαβών, για τις υπερβάσεις κόστους και χρονοδιαγράμματος κι είναι συνήθως καλύτερη από τις προσομοιώσεις ανθρώπινης διαίσθησης ή εναλλακτικά ευέλικτων μεθόδων.

Επιπρόσθετα, μπορεί να χρησιμοποιηθεί με επιτυχία σε περιπτώσεις όπου το πρόβλημα προς επίλυση κρίνεται μέσω πιθανοτήτων. Πιο συγκεκριμένα, τα αποτελέσματα της παραπάνω ανάλυσης παρέχουν στο διαχειριστή του έργου τα πιθανά αποτελέσματα και τις αντίστοιχες πιθανότητες εμφάνισής τους.

Για να πραγματοποιηθεί η Monte Carlo ανάλυση στα μοντέλα ανάπτυξης λογισμικού της συγκεκριμένης εργασίας, έχει χρησιμοποιηθεί η εφαρμογή Crystal Ball. Το Crystal Ball είναι μια εφαρμογή της Oracle που μπορεί να εκτελεστεί μέσω excel και

πραγματοποιεί προβλέψεις για τα πιθανά αποτελέσματα βάσει των μεταβλητών που επηρεάζουν σε μεγαλύτερο βαθμό το έργο ανάπτυξης.

### 3.4 Sensitivity Analysis

Η ανάλυση ευαισθησίας (Sensitivity Analysis) είναι η μελέτη του πως η αβεβαιότητα στην έξοδο ενός μαθηματικού μοντέλου ή συστήματος μπορεί να καταναμηθεί σε διάφορες πηγές αβεβαιότητας στην είσοδο του συστήματος [21]. Η διαδικασία του εκ νέου υπολογισμού των αποτελεσμάτων με εναλλακτικές παραδοχές (assumptions) για να διαπιστωθεί η επίδραση της μεταβλητής υπό ανάλυση ευαισθησίας μπορεί να είναι χρήσιμη για τους εξής λόγους [22]:

- έλεγχος της ευρωστίας των αποτελεσμάτων ενός μοντέλου ή συστήματος σε συνδυασμό με την παρουσία της αβεβαιότητας,
- αυξημένη κατανόηση των σχέσεων μεταξύ των μεταβλητών εισόδου και εξόδου σε ένα σύστημα ή μοντέλο,
- μείωση της αβεβαιότητας μέσω του εντοπισμού των μεταβλητών εισόδου που προκαλούν σημαντική αβεβαιότητα στην έξοδο και θα πρέπει, συνεπώς, να είναι το επίκεντρο της προσοχής σε περίπτωση που είναι επιθυμητή η αύξηση της ευρωστίας του συστήματος [23],
- έλεγχος σφαλμάτων του μοντέλου μέσω της παρατήρησης των σχέσεων μεταξύ των μεταβλητών εισόδου και εξόδου,
- απλούστευση του μοντέλου μέσω του εντοπισμού μεταβλητών εισόδου που δεν έχουν καμία επίδραση στην έξοδο και
- εντοπισμός των βέλτιστων τιμών εισόδου με σκοπό την βελτιστοποίηση των αποτελεσμάτων στις μεταβλητές εξόδου.

Στην παρούσα εργασία, η Sensitivity ανάλυση θα πραγματοποιηθεί μέσω του προγράμματος Crystal Ball.

## 4. ΜΕΘΟΔΟΛΟΓΙΑ ΚΑΙ ΜΟΝΤΕΛΟΠΟΙΗΣΗ

### 4.1 Case Study

Όπως αναφέρθηκε παραπάνω (υποενότητα 3.2) υπάρχουν πολλοί παράγοντες που μπορούν να επηρεάσουν την ομαλή εξέλιξη ενός έργου. Αυτοί μπορεί να έχουν ως αποτέλεσμα τη σημαντική αύξηση του χρόνου υλοποίησης ή/και του κόστους του έργου, καθώς ακόμα και την εγκατάλειψή του, με ό,τι συνεπάγεται το καθένα τόσο για την ανάδοχο εταιρία, όσο και για την εντολοδόχο [24].

Επιπρόσθετα, τα μοντέλα ανάπτυξης λογισμικού σχεδιάστηκαν για να βοηθήσουν τη διαδικασία της ανάπτυξης, περιορίζοντας ορισμένους από αυτούς τους παράγοντες όπως για παράδειγμα τη χαμηλή παραγωγικότητα της ομάδας ανάπτυξης, το μη ρεαλιστικό χρονοδιάγραμμα, την κακή επικοινωνία στο εσωτερικό της ομάδας κλπ. Εντούτοις, η εμφάνιση σφαλμάτων είναι αναπόφευκτη και μπορεί να οφείλεται σε ελλιπή επικοινωνία με τον πελάτη, σε κακή κατανόηση των απαιτήσεων από την ομάδα ανάπτυξης ή σε αλλαγή των απαιτήσεων από τον πελάτη κλπ. Καθένα από τα μοντέλα ανάπτυξης λογισμικού αντιμετωπίζει με διαφορετικό τρόπο την εμφάνιση αυτών των σφαλμάτων. Για το λόγο αυτό η χρήση διαφορετικών μοντέλων μπορεί να οδηγήσει σε διαφορετικά αποτελέσματα αναφορικά με το κόστος και το χρόνο υλοποίησης του έργου.

Ο σκοπός της εργασίας είναι να διερευνηθεί ο τρόπος με τον οποίο συγκεκριμένα μοντέλα ανάπτυξης λογισμικού αντιμετωπίζουν τα σφάλματα που εμφανίζονται στην ανάπτυξη λογισμικού με σκοπό τη βελτίωση της διαδικασίας ανάπτυξης και την καλύτερη κοστολόγηση του έργου. Πιο συγκεκριμένα, θα εξεταστεί η περίπτωση ανάπτυξης ενός έργου μεσαίου μεγέθους κάνοντας χρήση 4 μοντέλων, 2 agile και 2 traditional. Αρχικά, θα εξεταστεί ο τρόπος με τον οποίο τα χαρακτηριστικά του έργου επηρεάζουν το κόστος και τη διάρκεια του τελευταίου. Ακολούθως, θα γίνει εισαγωγή τυχαίων ομοιόμορφα κατανομημένων σφαλμάτων, προκειμένου να εντοπιστεί η επίδραση των τελευταίων στο κόστος και στη διάρκεια του έργου.

Αναλυτικότερα, με δεδομένο ότι στόχος της μελέτης είναι να υλοποιηθεί το έργο στο χαμηλότερο ή στο ζητούμενο δυνατό κόστος και χρονικό διάστημα, θα πρέπει να επιλεγεί το μοντέλο το οποίο διαχειρίζεται καλύτερα την εμφάνιση τυχαίων σφαλμάτων. Το αποτέλεσμα της εμφάνισης τυχαίου σφάλματος είναι ένα χαρακτηριστικό που δε μπορεί να προβλεφθεί εμπειρικά κι επομένως για να υπολογισθεί θα χρειαστεί να γίνει εισαγωγή του σφάλματος σαν βασική μεταβλητή στα μοντέλα ανάπτυξης λογισμικού. Η

αξιολόγηση της μεταβλητής τους σφάλματος γίνεται με τη χρήση της Monte Carlo ανάλυσης, η οποία μέσω των πολλαπλών επαναλήψεων επιτρέπει την εξαγωγή αξιόπιστων συμπερασμάτων αναφορικά με την συμπεριφορά του κάθε μοντέλου στην τυχαία εμφάνιση σφάλματος. Με αυτό τον τρόπο λοιπόν μπορεί να γίνει επιλογή του καταλληλότερου μοντέλου ανάπτυξης λογισμικού, που θα επιφέρει το επιθυμητό αποτέλεσμα με μικρότερο ρίσκο. Η ανάλυση θα ολοκληρωθεί με την πραγματοποίηση μιας sensitivity analysis επί του συνόλου των ανωτέρω παραγόντων.

## 4.2 Παραδοχές και τυχαίο σφάλμα

Η μοντελοποίηση των μεθόδων έγινε σε excel χρησιμοποιώντας διάφορες λογικές και μαθηματικές συναρτήσεις. Για την προσομοίωση του περιβάλλοντος που περιγράφεται στο case study έχουν χρησιμοποιηθεί τα ακόλουθα δεδομένα. Αρχικά, σαν μέτρο για το μέγεθος του έργου χρησιμοποιούνται οι γραμμές κώδικα. Όσον αφορά τους ανθρώπινους πόρους το excel έχει μοντελοποιηθεί με τέτοιο τρόπο ώστε το εκάστοτε μοντέλο να χρησιμοποιεί σε κάθε φάση τους απαραίτητους πόρους βάσει της θεωρίας του.

Αρχικά στο V-Model, έχουν συμπεριληφθεί όλα τα στάδια που ορίζει η θεωρία του. Τα στάδια αυτά περιλαμβάνουν τη συλλογή των requirements, το functional specification, το high level design, το detailed design, το coding, το unit testing, το integration testing, το system testing και το user acceptance testing. Με δεδομένο ότι η θεωρία της Spiral δεν ορίζει το ποια μέλη πρέπει να παίρνουν μέρος σε κάθε στάδιο, έχουν γίνει οι εξής παραδοχές κατά την επιλογή των resources του κάθε σταδίου. Οι seniors παίρνουν μέρος σε όλα τα στάδια της ανάλυσης και του design. Κατά το στάδιο της υλοποίησης και του unit testing παίρνουν μέρος όλοι οι senior και junior developers. Επιπλέον, στα υπόλοιπα στάδια testing χρησιμοποιείται ένας senior κι ένας junior και τέλος ο PM εμπλέκεται σε στάδια που αφορούν την ανάλυση κι οργάνωση του έργου.

Σε αντιστοιχία με τα παραπάνω, στο Spiral έχουν συμπεριληφθεί όλα τα στάδια που υπάγονται σε κάθε φάση και σε κάθε κύκλο. Αναλυτικότερα, οι φάσεις από τις οποίες αποτελείται το Spiral είναι οι ακόλουθες: ο προσδιορισμός στόχων, εναλλακτικών και περιορισμών, η αξιολόγηση των παραπάνω και ο προσδιορισμός κινδύνων, η υλοποίηση και επιβεβαίωση του προϊόντος για το επόμενο στάδιο και τέλος η οργάνωση των επόμενων φάσεων για τον κύκλο που ακολουθεί. Όσον αφορά τα διαθέσιμα resources στο κάθε στάδιο, η λογική που έχει χρησιμοποιηθεί είναι ακριβώς

ίδια με αυτή του V-Model καθώς και πάλι η θεωρία του Spiral δεν καθορίζει το ποια μέλη της ομάδας πρέπει να είναι ενεργά σε κάθε στάδιο.

Η μοντελοποίηση της XP συμπεριλαμβάνει όλα τα στάδια που ορίζει η θεωρία της. Τα στάδια αυτά είναι τα εβδομαδιαία meetings, το design, η υλοποίηση κώδικα, το testing και τα τριμηνιαία meetings. Στο στάδιο της υλοποίησης κώδικα έχουν χρησιμοποιηθεί και πάλι όλοι οι senior και junior developers. Όσον αφορά το testing, με δεδομένο ότι η θεωρία της XP δεν ορίζει ξεχωριστά το unit testing, το system integration testing και το user acceptance testing έχει οριστεί ως παραδοχή ότι στο γενικό στάδιο του testing χρησιμοποιούνται όλοι οι developers. Τέλος, στο στάδιο του design χρησιμοποιούνται μόνο οι senior developers ενώ σε όλα τα στάδια meeting παίρνει μέρος όλη η ομάδα ανάπτυξης κι ο PM όπως ορίζει η θεωρία του μοντέλου.

Όσον αφορά τη μοντελοποίηση της Scrum περιλαμβάνει τα ακόλουθα στάδια βάσει της θεωρίας της: design meeting, daily meetings, development, test, reconsideration meeting, review meeting και scrum training. Σε αντιστοιχία με την XP, στο στάδιο του development και του testing παίρνουν μέρος όλοι οι developers. Επιπλέον στα meetings τα οποία ορίζονται από τη θεωρία του μοντέλου πρέπει να παίρνει μέρος όλη η ομάδα ανάπτυξης καθώς κι ο Scrum Master. Τέλος, στο scrum training χρησιμοποιούνται και πάλι όλα τα resources, δηλαδή ομάδα ανάπτυξης και Scrum Master.

Συμπερασματικά, η μοντελοποίηση όλων των μοντέλων στηρίχθηκε στην παραδοχή ότι σε στάδια που είναι κοινά σε όλα τα μοντέλα χρησιμοποιούνται τα ίδια resources ενώ σε στάδια που απαιτούν συγκεκριμένη διαχείριση βάσει της θεωρίας του κάθε μοντέλου τα resources διαφοροποιούνται ανάλογα με τις απαιτήσεις.

Επιπρόσθετα, στον Πίνακα 1 φαίνεται η χρέωση ανά ώρα για τους ανθρώπινους πόρους και αντίστοιχα στον Πίνακα 2 φαίνεται η ημερήσια απόδοση των εργαζομένων σε γραμμές κώδικα ανάλογα με την εμπειρία τους. Οι παρακάτω πίνακες έχουν στηριχθεί στο case study και τα συγκεκριμένα δεδομένα χρησιμοποιούνται για τον υπολογισμό του κόστους και της διάρκειας του έργου.

**Πίνακας 1: Χρέωση ανθρώπινων πόρων ανά ώρα**

	Χρέωση ανά ώρα (ευρώ)
Junior	50
Senior	60
PM	75

	Χρέωση ανά ώρα (ευρώ)
Scrum Master	75

**Πίνακας 2: Ημερήσια απόδοση ανθρώπινων πόρων σε γραμμές κώδικα βάσει εμπειρίας**

	Γραμμές κώδικα (LOCs)
Junior	200
Senior	250

Επιπλέον, ένας πολύ σημαντικός παράγοντας στη μοντελοποίηση ήταν η εμφάνιση τυχαίου σφάλματος. Η εμφάνιση του σφάλματος περιλαμβάνει δύο στάδια. Το πρώτο στάδιο είναι το εάν θα εμφανιστεί σφάλμα και το δεύτερο στάδιο είναι το πόσο μεγάλο αντίκτυπο θα έχει το σφάλμα στις γραμμές κώδικα που έχουν υλοποιηθεί. Πιο συγκεκριμένα, σε πρώτη φάση ορίζεται στο excel η επιθυμητή πιθανότητα σφάλματος και βάσει αυτής της πιθανότητας σφάλματος υπολογίζεται το εάν θα έχουμε σφάλμα ή όχι. Η εμφάνιση τυχαίου σφάλματος καθώς και το μέγεθος του σφάλματος στις γραμμές κώδικα πραγματοποιούνται με τη χρήση ομοιόμορφα κατανομημένων μεταβλητών. Έχει τεθεί ως παραδοχή ότι το μέγιστο πιθανό σφάλμα δεν μπορεί να ξεπερνά το συνολικό αριθμό γραμμών κώδικα που εκτελείται σε κάθε στάδιο υλοποίησης. Το παραπάνω σφάλμα μπορεί να εντοπιστεί σε όλα τα στάδια testing του κάθε μοντέλου και επηρεάζει την πορεία του έργου ανάλογα με τη θεωρία του εκάστοτε μοντέλου.

Αναλυτικότερα, η θεωρία του V-Model ορίζει ότι σε περίπτωση σφάλματος στο unit testing επαναλαμβάνονται όλα τα στάδια από το detailed design και μετά, σε περίπτωση εμφάνισης σφάλματος στο integration testing επαναλαμβάνονται όλα τα στάδια από το high level design και μετά, σε περίπτωση εμφάνισης λάθους στο system testing επαναλαμβάνονται όλα τα στάδια από το functional specification και μετά και τέλος σε περίπτωση εμφάνισης λάθους στο user acceptance testing πρέπει να επαναληφθούν όλα τα στάδια από την αρχή. Συμπερασματικά, εάν εμφανιστεί σφάλμα στο excel, επαναλαμβάνονται για τις προβληματικές γραμμές κώδικα τα αντίστοιχα στάδια βάσει του σημείου στο οποίο εμφανίστηκε το λάθος. Η διαδικασία ολοκληρώνεται μόλις σταματήσει να εμφανίζεται σφάλμα και ουσιαστικά δεν έχουν απομείνει γραμμές κώδικα προς υλοποίηση.

Αντίστοιχα, στο Spiral ορίζεται ότι σε οποιοδήποτε κύκλο εμφανιστεί σφάλμα θα πρέπει να επαναληφθεί ολόκληρος ο κύκλος εξ αρχής. Αυτό πρακτικά σημαίνει ότι εάν



εμφανιστεί σφάλμα κατά τον έλεγχο του Prototype 1 θα πρέπει να επαναληφθεί ο πρώτος κύκλος για τις γραμμές κώδικα που είχαν σφάλμα. Επιπλέον, η θεωρία του Spiral ορίζει ότι σε περίπτωση που ολοκληρωθεί κάποιος κύκλος επιτυχώς, το μοντέλο δεν επιστρέφει ξανά σε αυτόν. Πρακτικά εάν εμφανιστεί σφάλμα σε μεταγενέστερο κύκλο, για παράδειγμα στον 3<sup>ο</sup> κύκλο, τότε δε θα επαναληφθούν όλοι οι προηγούμενοι κύκλοι της Spiral για τις προβληματικές γραμμές αλλά μόνο ο 3<sup>ος</sup> κύκλος. Η διαδικασία ολοκληρώνεται μόλις ολοκληρωθεί επιτυχώς ο 4<sup>ος</sup> κύκλος.

Όσον αφορά την XP και τη Scrum, η διαχείριση της εμφάνισης σφάλματος γίνεται με τον ίδιο τρόπο και στις δύο. Σε περίπτωση που εμφανιστεί σφάλμα, οι συνολικές λανθασμένες γραμμές προστίθενται στις υπολειπόμενες γραμμές κώδικα προς υλοποίηση. Στην XP επαναλαμβάνεται ο εβδομαδιαίος κύκλος όσες φορές χρειάζεται μέχρι να μην υπάρχουν υπολειπόμενες γραμμές κώδικα κι αντίστοιχα στη Scrum επαναλαμβάνεται η διαδικασία των Sprints.

### 4.3 Τα σενάρια που θα μελετηθούν

Τα στοιχεία που υπολογίζονται σε κάθε μοντέλο είναι το τελικό συνολικό κόστος, η τελική συνολική διάρκεια, το τελικό κόστος του development, η τελική διάρκεια του development, το τελικό κόστος των υπόλοιπων φάσεων εκτός του development και η τελική διάρκεια των υπόλοιπων φάσεων εκτός του development. Τα κόστη και οι διάρκειες υπολογίζονται προσθέτοντας το κόστος της κάθε επανάληψης με αυτό της προηγούμενης.

Στα πλαίσια της μελέτης, εκτελέστηκε ένας αριθμός από σενάρια τα αποτελέσματα των οποίων θα αναλυθούν και θα σχολιασθούν στο επόμενο κεφάλαιο. Η μελέτη επικεντρώθηκε στο πως επηρεάζεται το κάθε μοντέλο από την ύπαρξη τυχαίου σφάλματος, από την αλλαγή της πιθανότητας εμφάνισής του, από την αλλαγή των γραμμών κώδικα προς υλοποίηση κι από την αύξηση των ανθρώπινων πόρων ή την αλλαγή εμπειρίας τους. Οι υπόλοιπες τιμές, δηλαδή αυτές του πίνακα 1 και 2, παραμένουν σταθερές σε όλα τα σενάρια που εκτελέστηκαν.

Το base case σενάριο έχει εκτελεστεί με τα ακόλουθα δεδομένα. Οι συνολικές γραμμές κώδικα είναι 65.000, η πιθανότητα εμφάνισης σφάλματος είναι 0,2 και η ομάδα ανάπτυξης αποτελείται από 2 seniors και 3 juniors. Όσον αφορά τις γραμμές κώδικα, τα επιπλέον σενάρια που εκτελέστηκαν περιλαμβάνουν μεγέθη από 55.000 έως 75.000 γραμμές με βήμα 5.000. Αντίστοιχα, τα σενάρια για την πιθανότητα σφάλματος

περιλαμβάνουν περιπτώσεις από μηδενική πιθανότητα εμφάνισης σφάλματος έως και πιθανότητα εμφάνισης 0,5 με βήμα 0,1. Η τελευταία ομάδα σεναρίων που εκτελέστηκε είναι η μεταβολή του αριθμού και της εμπειρίας των εργαζομένων που παίρνουν μέρος στο στάδιο του development. Πιο συγκεκριμένα, κατά το base case σενάριο, η ομάδα ανάπτυξης αποτελείται από 2 seniors και 3 juniors και η πιθανότητα εμφάνισης σφάλματος είναι 0,2 επομένως ο μαθηματικός τύπος για τον υπολογισμό της πιθανότητας είναι ο ακόλουθος:

$$2x + 3y = 0,2.$$

Για τον υπολογισμό της μεταβολής της πιθανότητας σφάλματος σε περίπτωση αλλαγής της ομάδας υλοποίησης, έχει τεθεί ως παραδοχή ότι ένας senior επηρεάζει την πιθανότητα σφάλματος κατά το μισό από ένα junior. Επομένως η δεύτερη μαθηματική σχέση που προκύπτει είναι η ακόλουθη:

$$x = y/2$$

Λύνοντας το σύστημα των δύο παραπάνω εξισώσεων προκύπτει ότι ένας senior επηρεάζει την πιθανότητα εμφάνισης σφάλματος κατά 0,025 ενώ ένας junior κατά 0,05. Με δεδομένο τα παραπάνω, στον Πίνακα 3 παρουσιάζονται οι ομάδες ανάπτυξης για τις οποίες εκτελέστηκαν τα σενάρια και η πιθανότητα εμφάνισης σφάλματος που αντιστοιχεί σε κάθε ομάδα.

**Πίνακας 3: Ομάδες ανάπτυξης και αντίστοιχη πιθανότητα σφάλματος**

Αριθμός από Seniors	Αριθμός από Juniors	Πιθανότητα σφάλματος
3	2	0,175
2	3	0,2
1	4	0,225
3	4	0,275
2	5	0,3

Κατά την εκτέλεση των παραπάνω σεναρίων θα δοθεί βάση στη μελέτη των ακόλουθων μεταβλητών: mean value, maximum και ποσοστό βεβαιότητας 60%. Επισημαίνεται ότι το ποσοστό βεβαιότητας 60% επιλέχθηκε γιατί θεωρείται ότι καλύπτει το ρίσκο που καλείται να αναλάβει αυτός που αναπτύσσει το λογισμικό. Το ποσοστό αυτό γίνεται να αυξομειωθεί ανάλογα με την αποστροφή προς τον κίνδυνο αυτού που αναπτύσσει το λογισμικό.

#### 4.4 Συνοπτική επισκόπηση της μεθοδολογίας

Με βάση τα παραπάνω, η μεθοδολογία που ακολουθήθηκε παρουσιάζεται συνοπτικά παρακάτω:

1. Monte Carlo ανάλυση για περίπτωση μηδενικού σφάλματος.
2. Monte Carlo ανάλυση στο βασικό σενάριο το οποίο εκτελέστηκε για 65.000 γραμμές κώδικα, πιθανότητα εμφάνισης σφάλματος 0,2 και ομάδα ανάπτυξης με 2 seniors και 3 juniors.
3. Monte Carlo ανάλυση αλλάζοντας την πιθανότητα εμφάνισης σφάλματος.
4. Monte Carlo ανάλυση αλλάζοντας τον αριθμό γραμμών κώδικα.
5. Monte Carlo ανάλυση αλλάζοντας το μέγεθος και την εμπειρία της ομάδας υλοποίησης.
6. Sensitivity ανάλυση σε συγκεκριμένες μεταβλητές του περιβάλλοντος έχοντας σταθερή ομάδα υλοποίησης.
7. Sensitivity ανάλυση σε συγκεκριμένες μεταβλητές του περιβάλλοντος αλλάζοντας το μέγεθος και την εμπειρία της ομάδας υλοποίησης.

Τα αποτελέσματα της παραπάνω μεθοδολογίας παρουσιάζονται στο κεφάλαιο 5.

## 5. ΑΝΑΛΥΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Στη συγκεκριμένη ενότητα παρουσιάζονται συγκεντρωτικά τα αποτελέσματα όλων των σεναρίων που εκτελέστηκαν. Αρχικά, παρουσιάζονται τα αποτελέσματα σε περίπτωση μηδενικού σφάλματος. Στη συνέχεια, πραγματοποιείται εκτενής ανάλυση των αποτελεσμάτων που έχουν προκύψει για το βασικό σενάριο του case study. Ακολουθούν τα αποτελέσματα των σεναρίων που σχετίζονται με την αλλαγή των μεταβλητών του συστήματος και τέλος πραγματοποιείται μια sensitivity ανάλυση για τον εντοπισμό της σχέσης μεταξύ των μεταβλητών του συστήματος και του τελικού αποτελέσματος.

### 5.1 Σενάριο μηδενικού σφάλματος

Στη συγκεκριμένη υποενότητα θα παρουσιαστούν και θα σχολιαστούν τα αποτελέσματα για το κάθε μοντέλο σε περίπτωση που η πιθανότητα εμφάνισης σφάλματος είναι μηδενική.

Στον Πίνακα 4 που ακολουθεί φαίνονται συγκεντρωτικά τα αποτελέσματα που έχουν προκύψει για το κόστος του development σε περίπτωση που δεν εμφανιστεί κανένα σφάλμα. Το συμπέρασμα που εξάγεται από τα παρακάτω είναι πως το κόστος για το development είναι ίδιο σε όλα τα μοντέλα εάν δεν εμφανιστεί σφάλμα κατά τη διαδικασία υλοποίησης, καθώς το μέγεθος (65.000 γραμμές) και οι πόροι (2 Seniors και 3 Juniors) του έργου είναι συγκεκριμένοι και σταθεροί.

**Πίνακας 4: Κόστος development σε περίπτωση μηδενικού σφάλματος**

	Κόστος (ευρώ)
V-Model	127.656,00
Spiral	127.677,60
XP	127.636,37
Scrum	127.637,10

Στον Πίνακα 5 παρουσιάζονται τα αντίστοιχα αποτελέσματα για τη διάρκεια της φάσης του development. Όπως είναι αναμενόμενο, η διάρκεια του development σε περίπτωση μηδενικού σφάλματος είναι ίδια για όλα τα μοντέλα για τους ίδιους λόγους που αναφέρθηκαν παραπάνω.

**Πίνακας 5: Διάρκεια development σε περίπτωση μηδενικού σφάλματος**

	Διάρκεια (ημέρες)
V-Model	59,1
Spiral	59,11
XP	59,09
Scrum	59,09

Το συνολικό συμπέρασμα που προκύπτει από τα παραπάνω είναι πως σε περίπτωση μηδενικού σφάλματος το κόστος και η διάρκεια του development είναι ίδια για όλα τα μοντέλα κι επομένως η τελική επιλογή του μοντέλου βασίζεται στη διάρκεια και το κόστος των προκαθορισμένων φάσεων του κάθε μοντέλου. Στους πίνακες που ακολουθούν φαίνεται το συνολικό κόστος και η συνολική διάρκεια που έχει προκύψει για το κάθε μοντέλο.

Ο Πίνακας 6 παρουσιάζει τα αποτελέσματα για το συνολικό κόστος του έργου όταν η πιθανότητα εμφάνισης σφάλματος είναι μηδενική. Βάσει των αποτελεσμάτων, φαίνεται ότι το χαμηλότερο κόστος το εμφανίζει η Scrum και στη συνέχεια το V-Model. Όσον αφορά την XP, είναι η ακριβότερη σε σύγκριση με τις υπόλοιπες μεθόδους στοιχείο που δείχνει ότι οι εβδομαδιαίες διαδικασίες που έχει αυξάνουν αρκετά το συνολικό κόστος.

**Πίνακας 6: Συνολικό κόστος σε περίπτωση μηδενικού σφάλματος**

	Κόστος (ευρώ)
V-Model	225.816,00
Spiral	229.297,60
XP	255.278,87
Scrum	212.747,10

Στον Πίνακα 7 φαίνονται τα αποτελέσματα για τη συνολική διάρκεια του έργου με χρήση του εκάστοτε μοντέλου σε περίπτωση όπου το σφάλμα είναι μηδενικό. Το συμπέρασμα που προκύπτει από το παρακάτω είναι πως η Scrum έχει τη μικρότερη συνολική χρονική διάρκεια σε σύγκριση με τις υπόλοιπες μεθόδους. Επιπλέον, η XP φαίνεται πως έχει μικρότερη διάρκεια σε σχέση με τις traditional μεθόδους παρόλο που λόγω των εβδομαδιαίων συσκέψεων παρουσιάζει αυξημένο κόστος. Τέλος, όσον αφορά

τις traditional μεθόδους, η αυξημένη διάρκεια προκύπτει από τις επιπλέον φάσεις ανάλυσης και ελέγχου που απαιτεί η δομή τους.

**Πίνακας 7: Συνολική διάρκεια σε περίπτωση μηδενικού σφάλματος**

	Διάρκεια (ημέρες)
V-Model	143,1
Spiral	141,12
XP	119,2
Scrum	95,72

Εν κατακλείδι, η Scrum είναι καλύτερη τόσο από άποψη χρόνου όσο και κόστους. Τόσο το V-Model όσο και το Spiral έχουν μικρή διαφορά κόστους σε σύγκριση με τη Scrum. Αντίστοιχα, η XP είναι ακριβότερη από όλες τις μεθόδους αλλά εμφανίζει μικρή συνολική διάρκεια.

Τέλος, οι τιμές που παρουσιάζονται στους παραπάνω πίνακες είναι οι βέλτιστες (ελάχιστες) τιμές που μπορεί να προκύψουν στην υλοποίηση του έργου υπό εξέταση. Η εισαγωγή τυχαίων σφαλμάτων κατά τη διάρκεια του development αναμένεται να αυξήσει τις παραπάνω τιμές

## 5.2 Βασικό σενάριο

Στην υποενότητα αυτή θα αναλυθούν τα αποτελέσματα και τα συμπεράσματα που έχουν εξαχθεί για το βασικό σενάριο το οποίο εκτελέστηκε με τα δεδομένα που φαίνονται στον ακόλουθο πίνακα.

**Πίνακας 8: Δεδομένα Base Case**

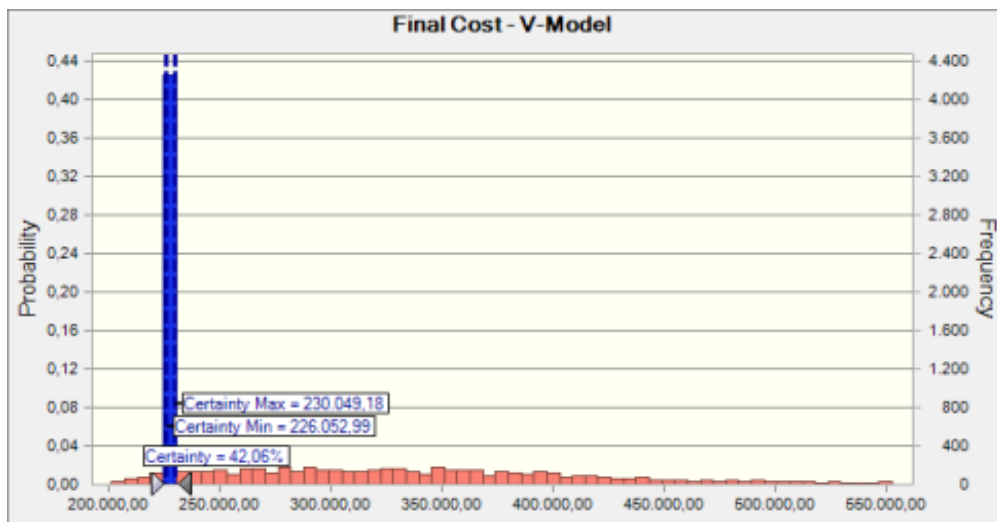
LOCs	65.000
Fault Probability	0,2
Αριθμός από Seniors	2
Αριθμός από Juniors	3

Τα αποτελέσματα που παρατίθενται στην υποενότητα 5.2.1 αφορούν το κόστος και τη διάρκεια του έργου συνολικά. Επισημαίνεται ότι το διάγραμμα του μοντέλου XP

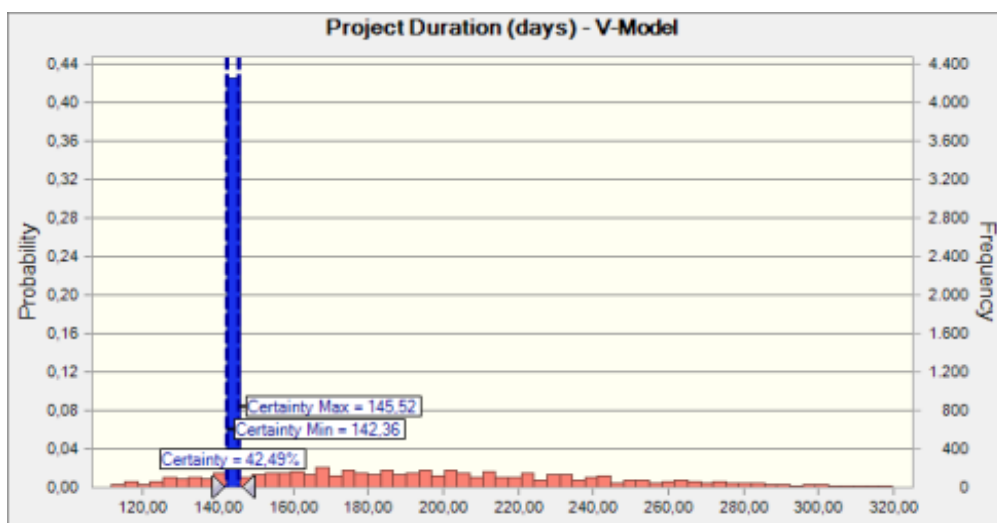
παρουσιάζει κανονική κατανομή ενώ τα υπόλοιπα μοντέλα, V-Model, Spiral και Scrum, παρουσιάζουν κατανομή γύρω από μία nominal τιμή, η οποία είναι η πιο πιθανή τιμή εμφάνισης σε όλες τις επαναλήψεις. Τέλος, στην υποενότητα 5.2.2 σχολιάζονται τα αποτελέσματα της Monte Carlo ανάλυσης, τόσο για το συνολικό κόστος και τη συνολική διάρκεια, όσο και μεμονωμένα για το στάδιο του development και των υπόλοιπων σταδίων του εκάστοτε μοντέλου.

### 5.2.1 Αποτελέσματα Monte Carlo ανάλυσης βασικού σεναρίου

Στα διαγράμματα που ακολουθούν, παρουσιάζονται οι κατανομές συνολικού κόστους και συνολικής διάρκειας για το V-Model.

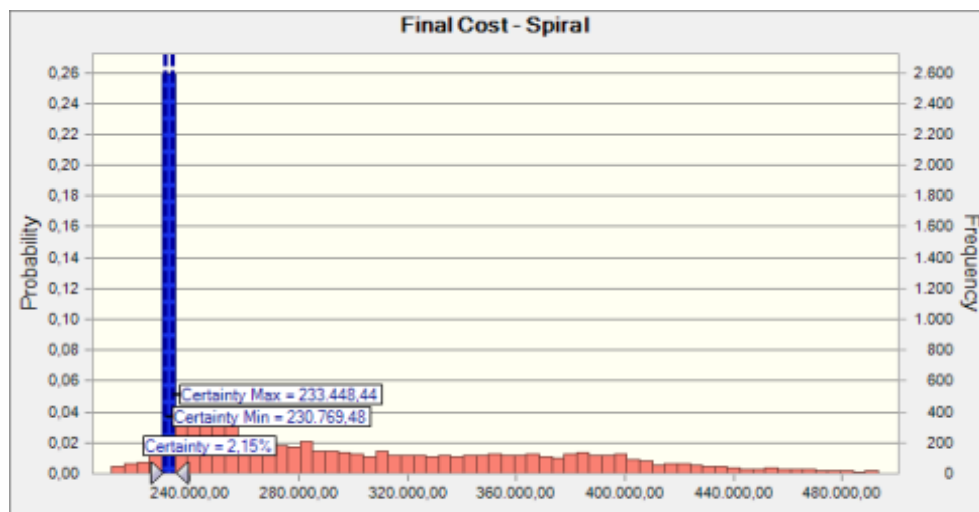


Σχήμα 2: Διάγραμμα συνολικού κόστους για το V-Model

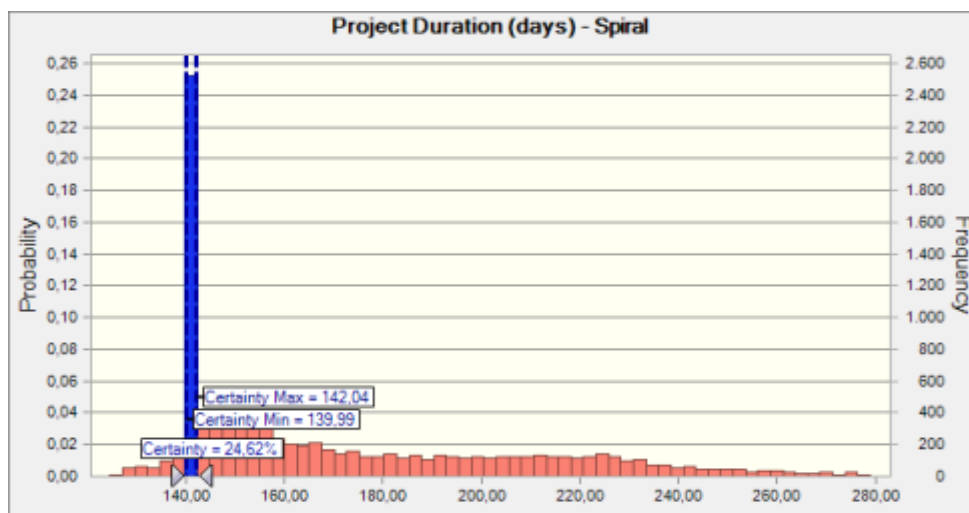


Σχήμα 3: Διάγραμμα συνολικής διάρκειας σε ημέρες για το V-Model

Στη συνέχεια, παρατίθενται τα αντίστοιχα διαγράμματα κόστους και διάρκειας για το Spiral.



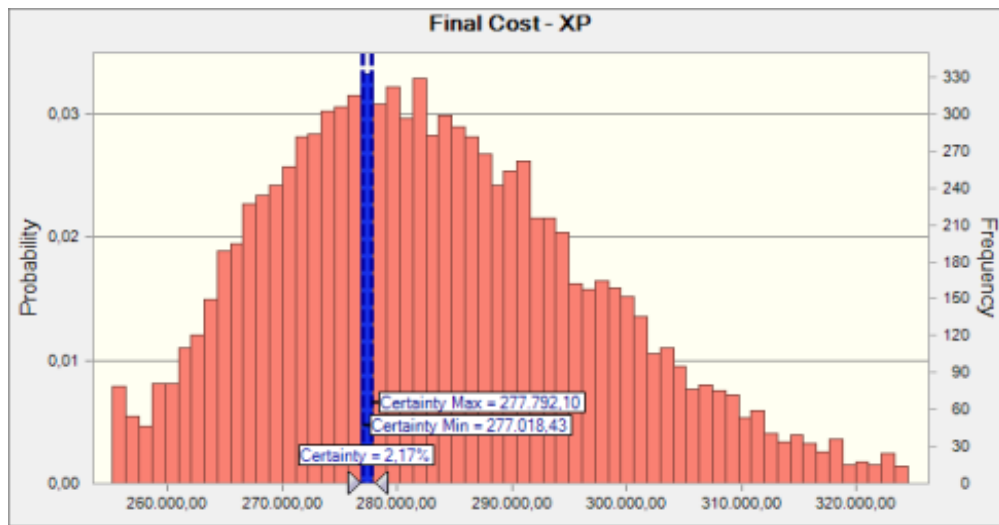
Σχήμα 4: Διάγραμμα συνολικού κόστους για το Spiral



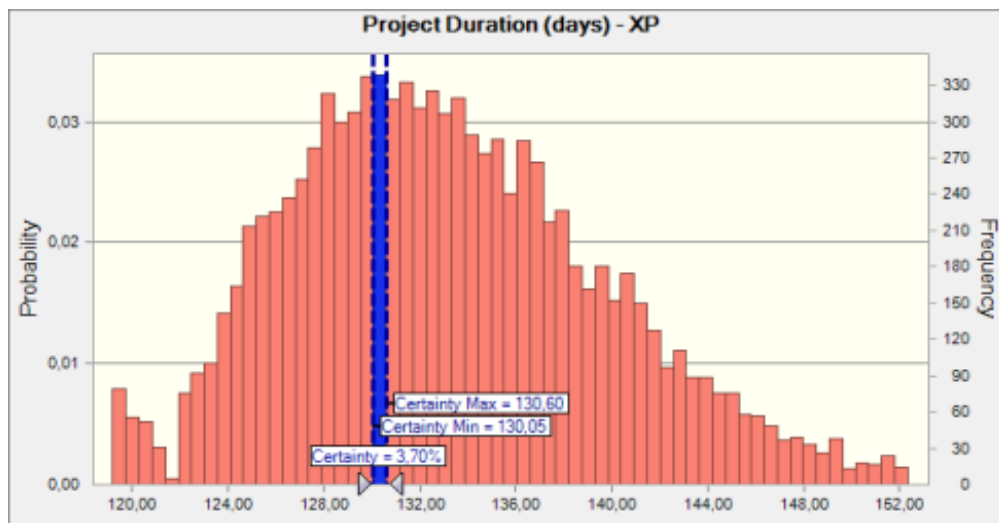
Σχήμα 5: Διάγραμμα συνολικής διάρκειας σε ημέρες για το Spiral



Τα αντίστοιχα διαγράμματα του συνολικού κόστους και διάρκειας για το μοντέλο ΧΡ παρουσιάζονται στα σχήματα 6 και 7.

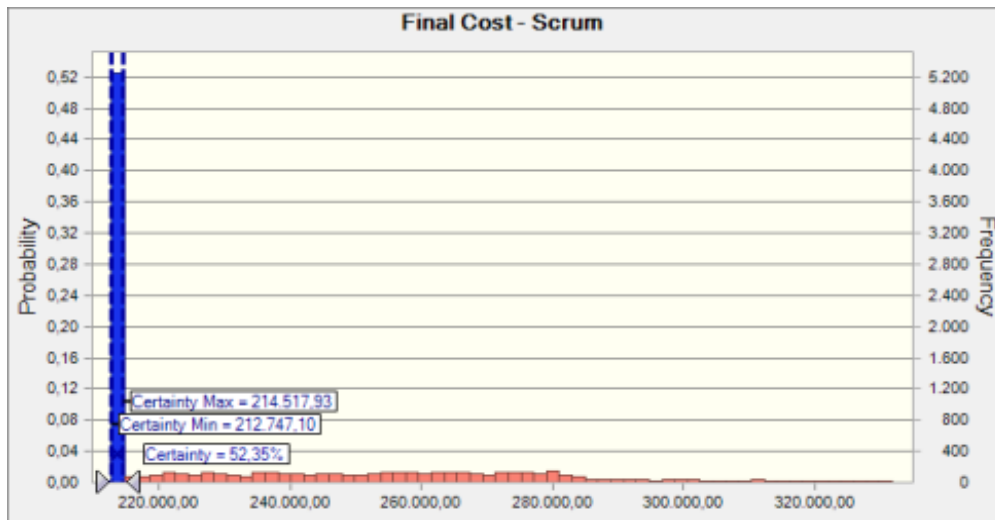


Σχήμα 6: Διάγραμμα συνολικού κόστους για το ΧΡ

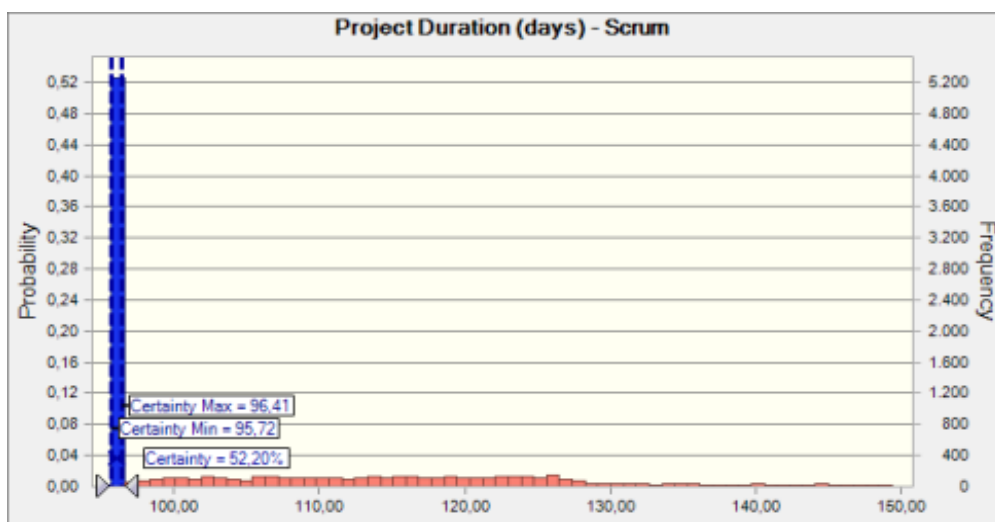


Σχήμα 7: Διάγραμμα συνολικής διάρκειας σε ημέρες για το ΧΡ

Τέλος, ακολουθούν τα διαγράμματα συνολικού κόστους και διάρκειας του μοντέλου Scrum.



Σχήμα 8: Διάγραμμα συνολικού κόστους για το Scrum



Σχήμα 9: Διάγραμμα συνολικής διάρκειας σε ημέρες για το Scrum

## 5.2.2 Σχολιασμός αποτελεσμάτων βασικού σεναρίου

### 5.2.2.1 Συνολικό κόστος

Στον Πίνακα 9 που ακολουθεί παρουσιάζονται συγκεντρωτικά κάποια στατιστικά στοιχεία και κάποιες χαρακτηριστικές τιμές των κατανομών του συνολικού κόστους. Πιο συγκεκριμένα αναλύοντας τα παρακάτω στοιχεία εξάγονται διάφορα συμπεράσματα για το πως διαχειρίζεται το συνολικό κόστος το κάθε μοντέλο με την ύπαρξη σφάλματος.

Αρχικά, από τα στατιστικά προκύπτει ότι τα traditional μοντέλα ανάπτυξης λογισμικού, δηλαδή το V-Model και η Spiral, εμφανίζουν πολύ μεγάλο εύρος τιμών. Η μεγάλη διακύμανση (variance) και τυπική απόκλιση (standard deviation) δείχνουν ότι τα traditional μοντέλα δεν μπορούν να διαχειριστούν σε μεγάλο βαθμό την τυχαία εμφάνιση σφαλμάτων. Το παραπάνω προκύπτει λόγω των προκαθορισμένων διαδικασιών που ορίζει η θεωρία των traditional μοντέλων σύμφωνα με την οποία όλα τα στάδια του ελέγχου πραγματοποιούνται στο τέλος της υλοποίησης με αποτέλεσμα τον αργό εντοπισμό λαθών.

Τα agile μοντέλα που έχουν επιλεγθεί, XP και Scrum, φαίνεται πως διαχειρίζονται καλύτερα την εμφάνιση σφάλματος, το καθένα με διαφορετικό τρόπο. Πιο συγκεκριμένα, η XP παρουσιάζει την μικρότερη διακύμανση και τυπική απόκλιση από όλα τα μοντέλα, χαρακτηριστικό που δείχνει ότι αν προκύψει σφάλμα, το μοντέλο μπορεί να το διαχειριστεί λόγω της εβδομαδιαίας επανάληψης των σταδίων. Όσον αφορά τη Scrum, έχει μεγαλύτερη διακύμανση και τυπική απόκλιση από την XP καθώς οι διαδικασίες επαναλαμβάνονται μηνιαία κι όχι εβδομαδιαία.

**Πίνακας 9: Στατιστικά στοιχεία για το συνολικό κόστος**

Στατιστικά	V-Model	Spiral	XP	Scrum
Mean (ευρώ)	296.342,41	291.490,21	283.565,84	236.527,08
Median (ευρώ)	250.308,00	261.169,40	281.978,06	212.747,10
Mode (ευρώ)	227.760,00	229.870,00	255.278,87	212.747,10
Standard Deviation	91.412,62	72.046,06	14.590,10	34.028,62
Variance	8.356.267.030,49	5.190.635.102,14	212.870.923,80	1.157.947.061,04
Minimum (ευρώ)	225.816,00	229.297,60	255.278,87	212.747,10
Maximum (ευρώ)	888.332,00	852.636,40	361.891,05	495.779,40
Range Width (ευρώ)	662.516,00	623.338,80	106.612,18	283.032,30

Ο Πίνακας 10 παρουσιάζει το ποσοστό εμφάνισης των τιμών του κόστους για το κάθε μοντέλο.

Βάσει των παρακάτω στατιστικών εξάγεται το συμπέρασμα πως τα δύο traditional μοντέλα εμφανίζουν μεγάλες τιμές κόστους σε ποσοστό βεβαιότητας πάνω από 40%. Πιο συγκεκριμένα για το V-Model έως και το 40% παρουσιάζει τιμές που είναι κοντά

στην ελάχιστη τιμή στη συνέχεια όμως παρουσιάζει μεγάλη αύξηση. Αντίστοιχα το Spiral μοντέλο μέχρι το ποσοστό βεβαιότητας 40% έχει μικρό ρυθμό αύξησης του κόστους ωστόσο για μεγαλύτερα ποσοστά βεβαιότητας οι τιμές κόστους αυξάνονται με πολύ μεγαλύτερο ρυθμό. Τέλος, για ποσοστά βεβαιότητας μεγαλύτερα του 90% και στα δύο μοντέλα οι τιμές κόστους αυξάνονται πάρα πολύ.

Όσον αφορά τα agile μοντέλα, όπως φαίνεται από τον πίνακα παρουσιάζουν μικρότερο ρυθμό αύξησης σε σύγκριση με τα traditional και καταλήγουν σε μικρότερες τιμές. Συγκεκριμένα για τη Scrum προκύπτει ότι μέχρι το ποσοστό βεβαιότητας του 50% εμφανίζει την ελάχιστη δυνατή τιμή κόστους. Επιπλέον, μέχρι το ποσοστό βεβαιότητας 90% οι τιμές κόστους έχουν μικρό ρυθμό αύξησης ενώ από 90 μέχρι το 100% οι τιμές σχεδόν διπλασιάζονται. Αντίστοιχα η XP σε όλο το διάστημα της εμφανίζει ομοιόμορφο ρυθμό αύξησης.

Τελικά, βάσει των ποσοστών βεβαιότητας των τελικών τιμών του κόστους και με δεδομένο ότι λαμβάνουμε υπόψη μας τα ποσοστά βεβαιότητας από 60% και πάνω, προκύπτει ότι στο 90% των περιπτώσεων η Scrum έχει το μικρότερο κόστος.

**Πίνακας 10: Συνολικό κόστος σε αντιστοίχιση με το ποσοστό βεβαιότητας**

Ποσοστά βεβαιότητας	V-Model (ευρώ)	Spiral (ευρώ)	XP (ευρώ)	Scrum (ευρώ)
0%	225.816,00	229.297,60	255.278,87	212.747,10
10%	227.760,00	229.870,00	265.971,85	212.747,10
20%	227.760,00	229.870,00	270.925,85	212.747,10
30%	227.760,00	234.227,20	274.827,09	212.747,10
40%	227.760,00	245.724,40	278.392,83	212.747,10
50%	250.301,60	261.158,80	281.973,02	212.747,10
60%	289.627,20	284.496,40	285.789,36	229.564,50
70%	328.381,60	318.952,80	290.149,85	247.902,00
80%	367.940,80	358.786,00	295.238,87	264.789,60
90%	424.892,80	397.313,20	303.032,04	280.848,30
100%	888.332,00	852.636,40	361.891,05	495.779,40

### 5.2.2.2 Συνολική διάρκεια

Στον Πίνακα 11 που ακολουθεί παρουσιάζονται συγκεντρωτικά κάποια στατιστικά στοιχεία και κάποιες χαρακτηριστικές τιμές των κατανομών της συνολικής διάρκειας. Πιο συγκεκριμένα αναλύοντας τα παρακάτω στοιχεία μπορούμε να εξάγουμε διάφορα

συμπεράσματα για το πως η ύπαρξη σφάλματος επηρεάζει την συνολική διάρκεια του κάθε μοντέλου.

Αρχικά, από τα στατιστικά προκύπτει ότι τα traditional μοντέλα ανάπτυξης λογισμικού, δηλαδή το V-Model και η Spiral, εμφανίζουν τις μεγαλύτερες τιμές στη συνολική διάρκεια καθώς και πολύ μεγάλο εύρος τιμών. Επιπλέον, η έντονη διακύμανση (variance) και τυπική απόκλιση (standard deviation) δείχνουν ότι τα traditional μοντέλα δεν μπορούν να διαχειριστούν σε μεγάλο βαθμό την τυχαία εμφάνιση σφαλμάτων.

Αντίθετα, τα agile μοντέλα, XP και Scrum, φαίνεται πως αντιδρούν καλύτερα στην εμφάνιση τυχαίου σφάλματος το καθένα με διαφορετικό τρόπο. Πιο συγκεκριμένα, η XP παρουσιάζει την μικρότερη διακύμανση και τυπική απόκλιση στη διάρκεια από όλα τα μοντέλα, στοιχείο που δείχνει ότι αν προκύψει σφάλμα το μοντέλο μπορεί να το διαχειριστεί καλύτερα. Αυτό συμβαίνει λόγω της εβδομαδιαίας επανάληψης των σταδίων που χαρακτηρίζει το μοντέλο με αποτέλεσμα να δίνεται η δυνατότητα κάλυψης κάποιου λάθους σε μικρό χρονικό διάστημα. Η Scrum εμφανίζει μικρότερη διακύμανση και τυπική απόκλιση από τις traditional μεθόδους αλλά μεγαλύτερη από την XP στοιχείο που δικαιολογείται από το γεγονός ότι ο έλεγχος πραγματοποιείται σε μηνιαίες επαναλήψεις.

**Πίνακας 11: Στατιστικά στοιχεία για τη συνολική διάρκεια**

Statistics	V-Model	Spiral	XP	Scrum
Mean (ημέρες)	177,97	172,60	133,28	106,42
Median (ημέρες)	147,04	156,44	132,55	95,72
Mode (ημέρες)	144,00	141,12	119,20	95,72
Standard Deviation	50,57	37,97	6,80	15,33
Variance	2.557,80	1.442,00	46,25	235,09
Minimum (ημέρες)	143,1	141,12	119,20	95,72
Maximum (ημέρες)	511,16	429,77	169,55	223,28
Range Width (ημέρες)	368,06	288,65	50,35	127,56

Ο Πίνακας 12 παρουσιάζει το ποσοστό εμφάνισης των τιμών της διάρκειας για το κάθε μοντέλο.

Βάσει των παρακάτω στατιστικών διαφαίνεται πως τα δύο traditional μοντέλα εμφανίζουν μεγάλες τιμές διάρκειας σε ποσοστό βεβαιότητας από 60% και πάνω, με διαφορά περίπου 30 εργάσιμων ημερών από τα agile, στοιχείο που δείχνει πως η

ύπαρξη σφάλματος στα traditional μοντέλα έχει μεγάλο αντίκτυπο στην διάρκεια του έργου.

Όσον αφορά τα agile μοντέλα, φαίνεται πως οι πιθανές τιμές διάρκειας της Scrum είναι σε όλα τα ποσοστά βεβαιότητας μέχρι το 90% μικρότερες από αυτές της XP. Τόσο η XP όσο και η Scrum εμφανίζουν μικρό ρυθμό αύξησης. Τέλος, όσον αφορά τη Scrum φαίνεται ότι σε ποσοστό από 90 έως 100% διπλασιάζει τις τιμές της διάρκειας.

**Πίνακας 12: Συνολική διάρκεια σε αντιστοίχιση με το ποσοστό βεβαιότητας**

Ποσοστά βεβαιότητας	V-Model (ημέρες)	Spiral (ημέρες)	XP (ημέρες)	Scrum (ημέρες)
0%	143,1	141,12	119,2	95,71625
10%	144	141,12	125,15	95,71625
20%	144	141,12	127,45	95,71625
30%	144	142,1	129,25	95,71625
40%	144	148,96	130,9	95,71625
50%	147,03	156,43	132,55	95,71625
60%	169,48	168,1	134,35	103,29375
70%	192,84	186,57	136,35	111,56375
80%	217,07	207,91	138,7	119,1675
90%	250,39	228,25	142,3	126,40125
100%	511,16	429,77	169,55	223,2775

### 5.2.2.3 Κόστος development

Στον Πίνακα 13 που ακολουθεί παρουσιάζονται συγκεντρωτικά κάποια στατιστικά στοιχεία και κάποιες χαρακτηριστικές τιμές των κατανομών. Πιο συγκεκριμένα αναλύοντας τα παρακάτω στοιχεία μπορούμε να εξάγουμε διάφορα συμπεράσματα για το πως διαχειρίζεται το κόστος του development το κάθε μοντέλο με την ύπαρξη σφάλματος.

Αρχικά προκύπτει ότι η μικρότερη τιμή που έχουν όλα τα μοντέλα στην διάρκεια του development κυμαίνεται στις ίδιες τιμές, γεγονός που είναι λογικό αφού το development στις περιπτώσεις που όλα πάνε καλά για ένα προκαθορισμένο μέγεθος έργου θα έπρεπε σε όλα τα μοντέλα να διαρκεί το ίδιο.

Στη συνέχεια, φαίνεται ότι το V-Model και η Spiral παρουσιάζουν μεγαλύτερο εύρος τιμών στο κόστος του development γεγονός που αποδεικνύει πάλι ότι δεν αντιδρούν επιθυμητά στις περιπτώσεις σφαλμάτων.

Για τα agile μοντέλα, XP και Scrum, σύμφωνα με τον πίνακα 13 προκύπτει ότι έχουν μικρότερο εύρος τιμών από τα traditional με μικρότερο εύρος συνολικά να το εμφανίζει η XP. Αυτό σημαίνει ότι η XP έχει τη δυνατότητα πιο γρήγορης επαναφοράς στην περίπτωση τυχαίου σφάλματος σε σχέση με τα άλλα μοντέλα και έχει την μικρότερη μέγιστη τιμή διάρκειας.

**Πίνακας 13: Στατιστικά στοιχεία για το κόστος development**

Στατιστικά	V-Model	Spiral	XP	Scrum
Mean (ευρώ)	183.841,71	170.276,29	141.780,66	141.905,04
Median (ευρώ)	154.461,60	152.344,80	140.987,14	127.637,10
Mode (ευρώ)	129.600,00	128.250,00	127.636,37	127.637,10
Standard Deviation	67.536,12	47.460,62	7.294,98	20.551,10
Variance	4.561.128.086,97	2.252.510.019,94	53.216.778,19	422.347.731,34
Minimum (ευρώ)	129.600,00	127.904,40	127.636,37	127.637,10
Maximum (ευρώ)	620.892,00	481.723,20	180.943,20	298.679,40
Range Width (ευρώ)	491.292,00	353.818,80	53.306,83	171.042,30

Ο Πίνακας 14 παρουσιάζει τις τιμές του κόστους development σε σύγκριση με το ποσοστό βεβαιότητας. Βάσει του συγκεκριμένου πίνακα εξάγονται διάφορα συμπεράσματα για το πως επηρεάζεται το κόστος του development σε κάθε μοντέλο όταν εμφανίζεται σφάλμα.

Για τα traditional μοντέλα, προκύπτει ότι ο ρυθμός αύξησης του κόστους είναι μεγαλύτερος σε σύγκριση με τα agile μοντέλα καθώς παρόλο που αρχικά όλα τα μοντέλα έχουν αντίστοιχες τιμές κόστους τελικά το V-Model και το Spiral καταλήγουν να παίρνουν πολύ μεγαλύτερες τιμές κόστους.

Αναλυτικότερα για τα agile μοντέλα, η Scrum, όπως έχει αναφερθεί και παραπάνω, έχει μεγάλη πιθανότητα να πετύχει την μικρότερη τιμή της. Ωστόσο η τιμή του σταδίου του development φαίνεται από το 70% και μετά να είναι μικρότερη στην XP, πράγμα που σημαίνει ότι το κόστος που αφιερώνεται στο development σε περίπτωση λάθους σε κάποιες περιπτώσεις αντιμετωπίζεται καλύτερα από την XP σε σύγκριση με τη Scrum.

**Πίνακας 14: Κόστος development σε αντιστοίχιση με το ποσοστό βεβαιότητας**

Ποσοστά βεβαιότητας	V-Model (ευρώ)	Spiral (ευρώ)	XP (ευρώ)	Scrum (ευρώ)
0%	129.600,00	127.904,40	127.636,37	127.637,10
10%	129.600,00	128.250,00	132.983,35	127.637,10
20%	129.600,00	128.250,00	135.459,50	127.637,10
30%	129.600,00	134.568,00	137.409,39	127.637,10
40%	129.600,00	142.246,80	139.194,33	127.637,10
50%	154.461,60	152.344,80	140.985,17	127.637,10
60%	182.952,00	166.214,71	142.891,86	137.610,90
70%	211.312,80	186.213,60	145.071,50	148.797,00
80%	239.133,60	211.118,40	147.616,37	158.930,10
90%	276.544,80	236.930,41	151.514,19	168.741,90
100%	620.892,00	481.723,20	180.943,20	298.679,40

#### 5.2.2.4 Διάρκεια development

Στον Πίνακα 15 παρουσιάζονται τα χαρακτηριστικά των κατανομών που έχουν προκύψει για τη διάρκεια του development σε κάθε μοντέλο. Το συμπέρασμα που εξάγεται από τα παρακάτω στατιστικά είναι το κατά πόσο επηρεάζεται η διάρκεια του development στο εκάστοτε μοντέλο από την ύπαρξη σφάλματος

Βάσει των στατιστικών των traditional μοντέλων φαίνεται πως η ύπαρξη σφάλματων οδηγεί σε μεγάλη διακύμανση της πιθανής διάρκειας του έργου σε αντίθεση με τα agile μοντέλα όπου το εύρος των τιμών είναι αρκετά μικρότερο.

**Πίνακας 15: Στατιστικά στοιχεία για τη διάρκεια development**

Statistics	V-Model	Spiral	XP	Scrum
Mean (ημέρες)	85,111903	78,65344	65,63977	65,6967795
Median (ημέρες)	71,51	70,405	65,2725	59,09125
Mode (ημέρες)	60	59,11	59,09125	59,09125
Standard Deviation	31,26672422	21,98565791	3,377322305	9,514398375
Variance	977,6080433	483,37	11,41	90,52
Minimum (ημέρες)	60	59,11	59,09125	59,09125
Maximum (ημέρες)	287,45	222,93	83,77	138,2775
Range Width (ημέρες)	227,45	163,82	24,67875	79,18625



Στον Πίνακα 16 φαίνονται συγκεντρωτικά τα ποσοστά βεβαιότητας για τη διάρκεια του development στο κάθε μοντέλο.

Το βασικότερο συμπέρασμα που προκύπτει είναι πως τα traditional μοντέλα εμφανίζουν πολύ μεγάλες τιμές στη διάρκεια για ποσοστό βεβαιότητας από 60% και πάνω. Όσον αφορά τις agile μεθόδους, φαίνεται πως στη Scrum η διάρκεια του development επηρεάζεται σε μεγαλύτερο βαθμό από την ύπαρξη σφάλματος σε σύγκριση με την XP.

**Πίνακας 16: Διάρκεια development σε αντιστοίχιση με το ποσοστό βεβαιότητας**

Ποσοστά βεβαιότητας	V-Model (ημέρες)	Spiral (ημέρες)	XP (ημέρες)	Scrum (ημέρες)
0%	60,00	59,11	59,09	59,09
10%	60,00	59,11	61,57	59,09
20%	60,00	59,11	62,71	59,09
30%	60,00	62,18	63,62	59,09
40%	60,00	65,72	64,44	59,09
50%	71,51	70,40	65,27	59,09
60%	84,70	76,78	66,15	63,71
70%	97,83	86,03	67,16	68,89
80%	110,71	97,55	68,34	73,58
90%	128,03	109,60	70,15	78,12
100%	287,45	222,93	83,77	138,28

#### 5.2.2.5 Κόστος των υπόλοιπων φάσεων

Στον Πίνακα 17 που ακολουθεί παρουσιάζονται συγκεντρωτικά κάποια στατιστικά στοιχεία και κάποιες χαρακτηριστικές τιμές των κατανομών. Πιο συγκεκριμένα αναλύοντας τα παρακάτω στοιχεία μπορούμε να εξάγουμε διάφορα συμπεράσματα για το πως αυξάνεται το κόστος των υπόλοιπων φάσεων σε περίπτωση σφάλματος.

Βάσει των στατιστικών προκύπτει ότι τα traditional μοντέλα, δηλαδή το V-Model και η Spiral, εμφανίζουν αρκετά μεγαλύτερο εύρος τιμών σε σύγκριση με τα agile μοντέλα. Επιπλέον, τη μικρότερη διακύμανση και τυπική απόκλιση την εμφανίζει το κόστος που προκύπτει για την XP, το συγκεκριμένο στοιχείο οφείλεται στην εβδομαδιαία επανάληψη των διαδικασιών του μοντέλου, χαρακτηριστικό που έχει παρατηρηθεί σε όλες τις κατανομές κόστους της XP.

Όσον αφορά τη μικρότερη τιμή κόστους που προκύπτει για κάθε μοντέλο φαίνεται πως η XP έχει τη μεγαλύτερη τιμή σε σύγκριση με τα υπόλοιπα μοντέλα. Αυτό το χαρακτηριστικό οφείλεται και πάλι στο γεγονός ότι η XP έχει κάποιες εβδομαδιαίες προκαθορισμένες διαδικασίες των οποίων το κόστος αυξάνει και το συνολικό κόστος.

**Πίνακας 17: Στατιστικά στοιχεία για το κόστος υπόλοιπων σταδίων**

Statistics	V-Model	Spiral	XP	Scrum
Mean (ευρώ)	112.500,7	121.213,9271	141.785,175	94.622,034
Median (ευρώ)	98.160,00	109.285,4	140.990,925	85.110,00
Mode (ευρώ)	98.160,00	101.620,00	127.642,5	85.110,00
Standard Deviation	28.940,81317	26.503,63438	7.295,113987	13.484,18367
Variance	837.570.667	702.442.635,6	53.218.688,08	181.823.209,36
Minimum (ευρώ)	96.216,00	101.393,2	127.642,5	85.110,00
Maximum (ευρώ)	310.720,00	688.368,4	180.947,85	197.100,00
Range Width (ευρώ)	214.504,00	586.975,2	53.305,35	111.990,00

Ο Πίνακας 18 παρουσιάζει τις τιμές τους κόστους σε σύγκριση με το ποσοστό βεβαιότητας για τα υπόλοιπα στάδια των μοντέλων.

Το βασικότερο συμπέρασμα που προκύπτει για το V-Model είναι πως το κόστος των υπόλοιπων φάσεων είναι αρκετά μικρότερο από το κόστος του development, στοιχείο που δείχνει πως η ύπαρξη σφάλματος επηρεάζει σε μεγαλύτερο βαθμό το κόστος του development. Όσον αφορά τη Spiral, φαίνεται πως οι διαδικασίες που έχει το μοντέλο επηρεάζονται σε μεγάλο βαθμό από την ύπαρξη σφάλματος. Τέλος, τα δύο agile μοντέλα εμφανίζουν στο ποσοστό βεβαιότητας του κόστους των υπόλοιπων φάσεων την ίδια συμπεριφορά με αυτή του συνολικού κόστους και του κόστους development. Πιο συγκεκριμένα, η XP διαχειρίζεται το σφάλμα έχοντας πολύ μικρό εύρος τιμών, ενώ η Scrum εμφανίζει πολύ μικρό κόστος μέχρι το ποσοστό βεβαιότητας 90%.

**Πίνακας 18: Κόστος υπόλοιπων σταδίων σε αντιστοίχιση με το ποσοστό βεβαιότητας**

Ποσοστά βεβαιότητας	V-Model (ευρώ)	Spiral (ευρώ)	XP (ευρώ)	Scrum (ευρώ)
0%	96.216,00	101.393,2	127.642,50	85.110,00
10%	96.216,00	101.393,2	132.988,50	85.110,00
20%	98.160,00	101.620,00	135.466,35	85.110,00
30%	98.160,00	101.620,00	137.416,50	85.110,00
40%	98.160,00	103.929,20	139.198,50	85.110,00
50%	98.160,00	109.284,40	140.987,85	85.110,00

Ποσοστά βεβαιότητας	V-Model (ευρώ)	Spiral (ευρώ)	XP (ευρώ)	Scrum (ευρώ)
60%	102.000,00	118.538,40	142.897,50	92.040,00
70%	115.040,00	131.682,80	145.078,35	98.970,00
80%	131.560,00	146.843,20	147.622,50	105.630,00
90%	153.840,00	161.865,20	151.517,85	112.020,00
100%	310.720,00	688.368,40	180.947,85	197.100,00

### 5.2.2.6 Διάρκεια των υπόλοιπων φάσεων

Στον Πίνακα 19 φαίνονται συγκεντρωτικά κάποια χαρακτηριστικά των κατανομών που έχουν προκύψει για τη διάρκεια των υπόλοιπων σταδίων του κάθε μοντέλου. Βάσει των παρακάτω στοιχείων θα μπορέσουμε να προσδιορίσουμε το κατά πόσο η διάρκεια των φάσεων επηρεάζεται από την ύπαρξη σφάλματος για το κάθε μοντέλο.

Αρχικά, είναι εμφανές πως τα δύο traditional μοντέλα έχουν πολύ μεγάλο εύρος τιμών στοιχείο που δείχνει πως η ύπαρξη σφάλματος επηρεάζει τη διάρκεια των φάσεων σε μεγάλο βαθμό. Όσον αφορά το XP και το Scrum, φαίνεται πως το εύρος τιμών τους είναι πολύ μικρότερο και πως διαχειρίζονται σε μεγάλο βαθμό την ύπαρξη σφάλματος.

Πίνακας 19: Στατιστικά στοιχεία για τη διάρκεια υπόλοιπων σταδίων

Statistics	V-Model	Spiral	XP	Scrum
Mean (ημέρες)	92,8578	93,942382	67,63891	40,7232125
Median (ημέρες)	84	86,615	67,288125	36,625
Mode (ημέρες)	84	82,01	60,10875	36,625
Standard Deviation	23,76939792	16,80510357	3,424697376	5,820467112
Variance	564,9842776	282,4115061	11,73	33,88
Minimum (ημέρες)	83,1	82,01	60,10375	36,625
Maximum (ημέρες)	257	353,81	85,78	85
Range Width (ημέρες)	173,9	271,8	25,67625	48,375

Στον Πίνακα 20 φαίνονται συγκεντρωτικά τα ποσοστά βεβαιότητας για τη διάρκεια των υπόλοιπων φάσεων στο κάθε μοντέλο.

Το βασικότερο συμπέρασμα που προκύπτει είναι πως τα traditional μοντέλα εμφανίζουν πολύ μεγάλες τιμές στη διάρκεια για ποσοστό βεβαιότητας από 60% και πάνω. Σε αντίθεση, τα agile μοντέλα φαίνεται πως έχουν μικρή αύξηση στη διάρκεια

των φάσεων. Επιπλέον, η Scrum έχει αρχικά μικρότερες τιμές για τη διάρκεια σε σύγκριση με την XP παρόλα αυτά στο 100% του ποσοστού βεβαιότητας η Scrum και η XP φαίνεται να έχουν σχεδόν την ίδια διάρκεια. Το συμπέρασμα που προκύπτει είναι πως η XP έχει αρχικά μεγαλύτερη διάρκεια παρόλα αυτά επηρεάζεται σε μικρότερο βαθμό από το σφάλμα σε σύγκριση με τη Scrum.

**Πίνακας 20: Διάρκεια υπόλοιπων σταδίων σε αντιστοίχιση με το ποσοστό βεβαιότητας**

Ποσοστά βεβαιότητας	V-Model (ημέρες)	Spiral (ημέρες)	XP (ημέρες)	Scrum (ημέρες)
0%	83,10	82,01	60,10	36,63
10%	83,10	82,01	63,59	36,63
20%	83,10	82,01	64,73	36,63
30%	84,00	82,01	65,64	36,63
40%	84,00	83,26	66,47	36,63
50%	84,00	86,61	67,29	36,63
60%	84,00	91,56	68,19	39,63
70%	94,00	100,39	69,19	42,63
80%	108,00	110,11	70,36	45,50
90%	128,00	119,64	72,17	48,25
100%	257,00	353,81	85,78	85,00

Το συνολικό συμπέρασμα που προκύπτει βάσει όλων των παραπάνω και σε σύγκριση με την περίπτωση μηδενικού σφάλματος είναι πως οι agile μέθοδοι διαχειρίζονται καλύτερα την ύπαρξη σφάλματος κατά τη διαδικασία ανάπτυξης ενός έργου σε αντίθεση με τις traditional μεθόδους.

Το κοινό χαρακτηριστικό των δύο traditional μοντέλων είναι πως οι πιθανές τιμές κόστους και διάρκειας έχουν πολύ μεγάλο εύρος τιμών. Επίσης, σε ποσοστά βεβαιότητας από 60% και πάνω εμφανίζουν και τα δύο μοντέλα αρκετά μεγάλες τιμές κόστους και διάρκειας. Σε αντίθεση, τα agile μοντέλα λόγω των συχνών επαναληπτικών ελέγχων που πραγματοποιούνται εμφανίζουν μικρότερες διακυμάνσεις τόσο στις τιμές του κόστους όσο και της διάρκειας.

### 5.3 Σενάριο με αλλαγή στην πιθανότητα σφάλματος

Στη συγκεκριμένη υποενότητα θα σχολιαστούν τα στατιστικά που έχουν προκύψει για το συνολικό κόστος και τη συνολική διάρκεια του έργου αλλάζοντας την πιθανότητα σφάλματος. Το συμπέρασμα που θα εξαχθεί από τα παρακάτω είναι το πως επηρεάζεται το κάθε μοντέλο από την αύξηση και μείωση της πιθανότητας σφάλματος.

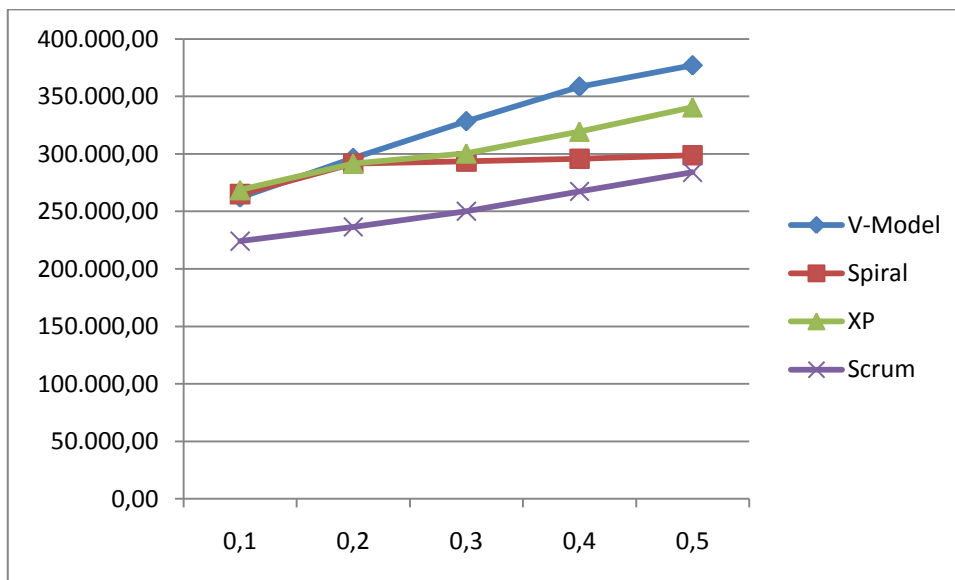
Το βασικό σενάριο, το οποίο έχει αναλυθεί στην υποενότητα 5.2, έχει εκτελεστεί για 65.000 γραμμές κώδικα και με πιθανότητα σφάλματος 0,2. Τα αποτελέσματα που παρατίθενται στη συνέχεια, θα επικεντρωθούν στο mean, maximum και στο ποσοστό βεβαιότητας 60% του κάθε μοντέλου για πιθανότητα σφάλματος από 0,1 έως 0,5 με βήμα 0,1.

Στον Πίνακα 21 παρουσιάζεται το πως επηρεάζεται το mean value της κατανομής του συνολικού κόστους του κάθε μοντέλου αυξάνοντας την πιθανότητα σφάλματος. Αυτό που προκύπτει από τα παρακάτω είναι πως σε όλα τα μοντέλα η αύξηση της πιθανότητας σφάλματος συνεπάγεται κι αύξηση στο συνολικό κόστος. Επιπλέον, προκύπτει ότι το V-Model έχει τη μεγαλύτερη διαφορά μεταξύ των mean τιμών του. Τέλος, προκύπτει ότι σε όλες τις περιπτώσεις το βέλτιστο μοντέλο είναι η Scrum καθώς το mean value του κόστους είναι σε όλες τις περιπτώσεις μικρότερο σε σύγκριση με το αντίστοιχο των υπόλοιπων μοντέλων.

**Πίνακας 21: Mean τιμές κόστους με αλλαγή στην πιθανότητα σφάλματος**

	V-Model (ευρώ)	Spiral (ευρώ)	XP (ευρώ)	Scrum (ευρώ)
Mean για πιθανότητα σφάλματος 0,1	262.037,38	264.975,18	268.632,50	224.099,76
Mean για πιθανότητα σφάλματος 0,2	296.342,41	291.490,21	291.490,21	236.527,08
Mean για πιθανότητα σφάλματος 0,3	328.348,31	293.400,54	300.482,78	250.002,79
Mean για πιθανότητα σφάλματος 0,4	358.426,90	295.701,83	319.428,19	267.392,84
Mean για πιθανότητα σφάλματος 0,5	376.902,57	298.812,69	340.605,81	283.941,70

Στο Σχήμα 10 παρατίθεται το διάγραμμα που προκύπτει για τις παραπάνω τιμές του πίνακα 21.



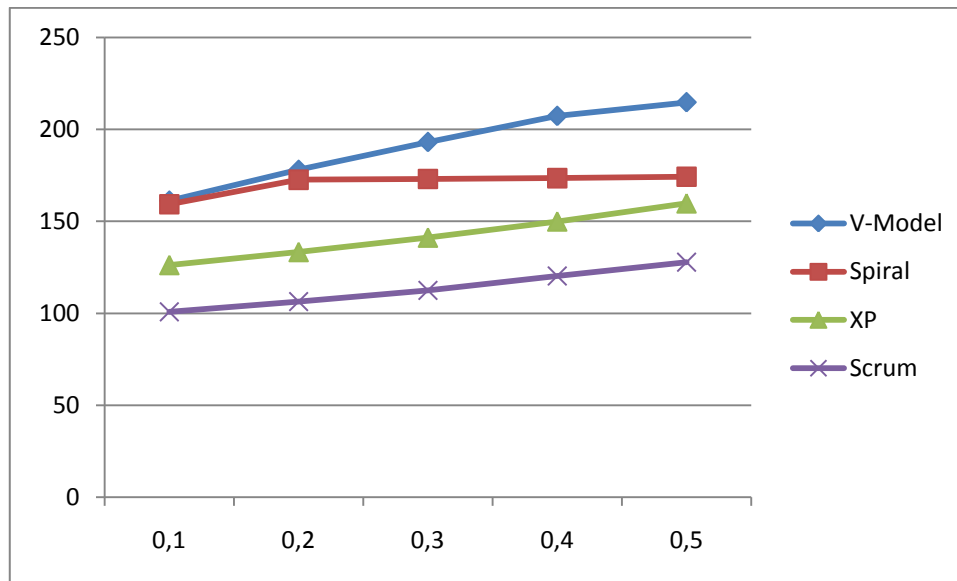
Σχήμα 10: Mean τιμές κόστους με αλλαγή στην πιθανότητα σφάλματος

Στον Πίνακα 22 φαίνεται το πως επηρεάζεται το mean value της κατανομής της συνολικής διάρκειας του κάθε μοντέλου. Βάσει του παρακάτω πίνακα προκύπτει ότι τα αποτελέσματα για το mean value της συνολικής διάρκειας βρίσκονται σε αντιστοιχία με το mean του συνολικού κόστους. Πιο συγκεκριμένα, το mean της διάρκειας στο V-Model φαίνεται να επηρεάζεται σε μεγαλύτερο βαθμό από την αλλαγή στην πιθανότητα σφάλματος σε σύγκριση με τα υπόλοιπα μοντέλα. Τέλος, η Scrum σε όλες τις περιπτώσεις σφάλματος εμφανίζει την μικρότερη mean διάρκεια σε σύγκριση με τα υπόλοιπα μοντέλα.

Πίνακας 22: Mean τιμές διάρκειας με αλλαγή στην πιθανότητα σφάλματος

	V-Model (ημέρες)	Spiral (ημέρες)	XP (ημέρες)	Scrum (ημέρες)
Mean για πιθανότητα σφάλματος 0,1	161,26	159,21	126,20	100,83
Mean για πιθανότητα σφάλματος 0,2	177,97	172,60	133,28	106,42
Mean για πιθανότητα σφάλματος 0,3	193,09	173,04	141,13	112,48
Mean για πιθανότητα σφάλματος 0,4	207,25	173,47	149,91	120,31
Mean για πιθανότητα σφάλματος 0,5	214,68	174,27	159,75	127,75

Στο Σχήμα 11 παρατίθεται το διάγραμμα που προκύπτει για τις παραπάνω τιμές του πίνακα 22.



Σχήμα 11: Μean τιμές διάρκειας με αλλαγή στην πιθανότητα σφάλματος

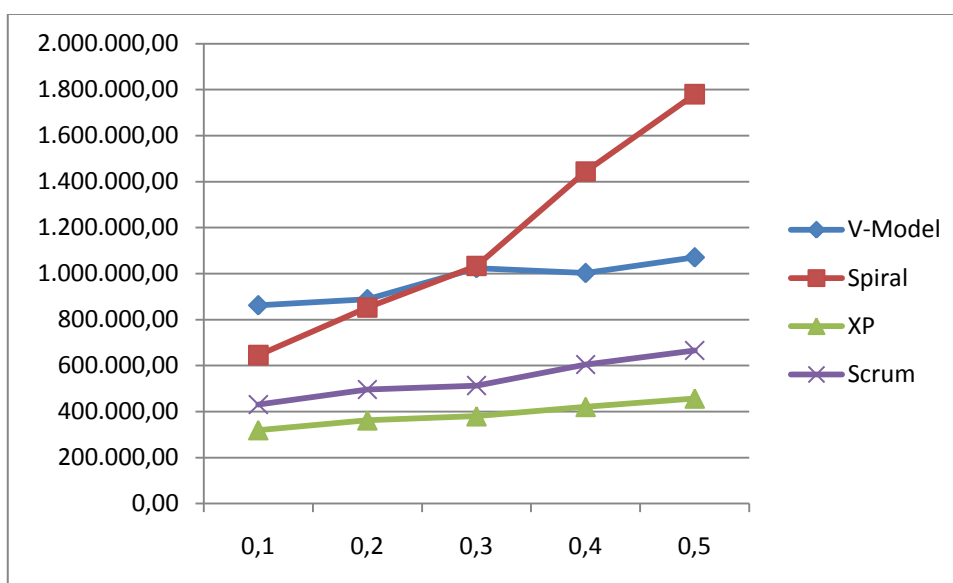
Στον Πίνακα 23 παρουσιάζεται το πως επηρεάζεται το maximum της κατανομής του συνολικού κόστους του κάθε μοντέλου. Προκύπτει ότι το maximum κόστος της κάθε κατανομής επηρεάζεται σε μεγάλο βαθμό από την αλλαγή στην πιθανότητα σφάλματος. Ένα άλλο στοιχείο άξιο σχολιασμού είναι ότι το maximum κόστος της Spiral επηρεάζεται σε μεγάλο βαθμό από την αύξηση στην πιθανότητα εμφάνισης σφάλματος ενώ αντίθετα οι agile μέθοδοι έχουν μικρότερη διαφοροποίηση στις maximum τιμές. Τέλος, η maximum τιμή του V-Model φαίνεται πως επηρεάζεται σε μικρό βαθμό από την αύξηση σφάλματος παρόλα αυτά είναι εμφανές πως το μοντέλο δεν μπορεί εξ αρχής να διαχειριστεί την εμφάνιση σφάλματος καθώς ακόμη και για πιθανότητα εμφάνισης σφάλματος ίση με 0,1 η maximum τιμή που προκύπτει είναι εξαιρετικά μεγάλη.

Πίνακας 23: Maximum τιμές κόστους με αλλαγή στην πιθανότητα σφάλματος

	V-Model (ευρώ)	Spiral (ευρώ)	XP (ευρώ)	Scrum (ευρώ)
Maximum για πιθανότητα σφάλματος 0,1	861.994,40	645.556,01	319.395,51	430.972,20
Maximum για πιθανότητα σφάλματος 0,2	888.332,00	852.636,40	361.891,05	495.779,40
Maximum για πιθανότητα σφάλματος 0,3	1.023.772,80	1.033.486,00	379.803,85	513.271,80

	V-Model (ευρώ)	Spiral (ευρώ)	XP (ευρώ)	Scrum (ευρώ)
Maximum για πιθανότητα σφάλματος 0,4	1.002.273,60	1.444.114,00	420.575,32	604.991,40
Maximum για πιθανότητα σφάλματος 0,5	1.070.410,40	1.780.008,40	457.111,95	665.949,90

Στο Σχήμα 12 παρατίθεται το διάγραμμα που προκύπτει για τις παραπάνω τιμές του πίνακα 23.



Σχήμα 12: Maximum τιμές κόστους με αλλαγή στην πιθανότητα σφάλματος

Στον Πίνακα 24 φαίνεται το πως επηρεάζεται το maximum της κατανομής της συνολικής διάρκειας του κάθε μοντέλου. Όπως είναι αναμενόμενο, ο πίνακας 24 βρίσκεται σε συμφωνία με τα αποτελέσματα και τα συμπεράσματα που έχουμε ήδη εξάγει από τον πίνακα 23.

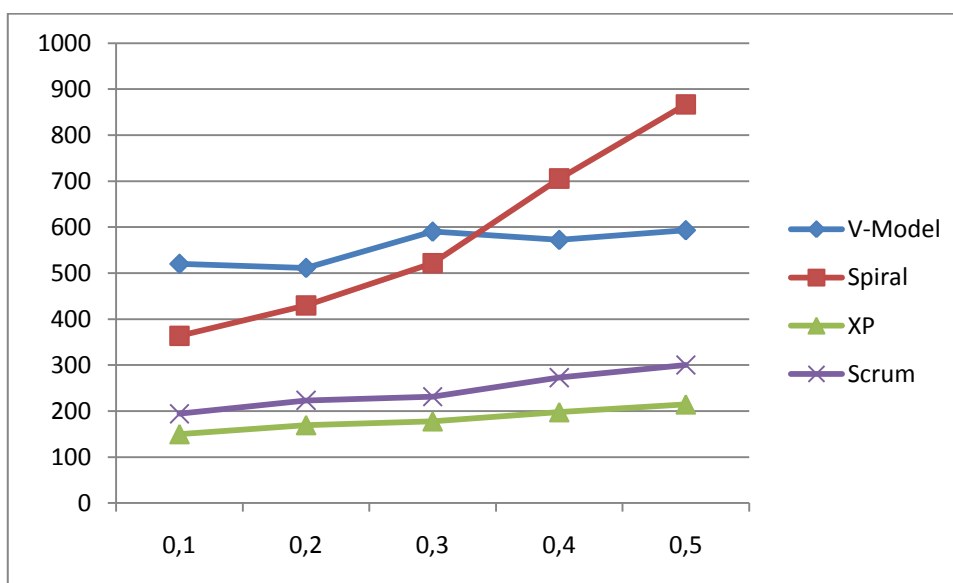
Πίνακας 24: Maximum τιμές διάρκειας (σε ημέρες) με αλλαγή στην πιθανότητα σφάλματος

	V-Model (ημέρες)	Spiral (ημέρες)	XP (ημέρες)	Scrum (ημέρες)
Maximum για πιθανότητα σφάλματος 0,1	520,09	363,49	149,90	194,11
Maximum για πιθανότητα σφάλματος 0,2	511,16	429,77	169,55	223,28
Maximum για πιθανότητα σφάλματος 0,3	590,08	521,69	177,85	231,17
Maximum για πιθανότητα σφάλματος 0,4	572,05	705,51	197,75	272,73



	V-Model (ημέρες)	Spiral (ημέρες)	XP (ημέρες)	Scrum (ημέρες)
Maximum για πιθανότητα σφάλματος 0,5	593,19	866,81	214,65	300,05

Στο Σχήμα 13 παρατίθεται το διάγραμμα που προκύπτει για τις παραπάνω τιμές του πίνακα 24.



Σχήμα 13: Maximum τιμές διάρκειας (σε ημέρες) με αλλαγή στην πιθανότητα σφάλματος

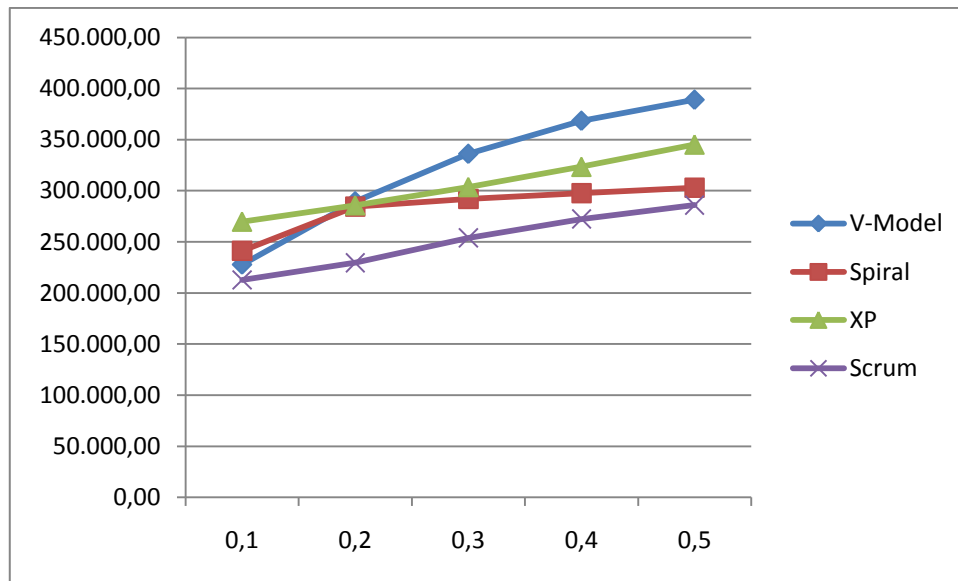
Στον Πίνακα 25 παρουσιάζεται η τιμή του κόστους για ποσοστό βεβαιότητας 60% κάθε μοντέλου. Βάσει των αποτελεσμάτων που έχουν προκύψει φαίνεται πως σε κάθε περίπτωση το μοντέλο Scrum παρουσιάζει την μικρότερη τιμή σε σύγκριση με τα υπόλοιπα.

Πίνακας 25: Τιμές κόστους για ποσοστό βεβαιότητας 60% με αλλαγή στην πιθανότητα σφάλματος

	V-Model (ευρώ)	Spiral (ευρώ)	XP (ευρώ)	Scrum (ευρώ)
Certainty 60% για πιθανότητα σφάλματος 0,1	227.760,00	241.194,40	269.782,04	212.747,10
Certainty 60% για πιθανότητα σφάλματος 0,2	289.627,20	284.496,40	285.789,36	229.564,50
Certainty 60% για πιθανότητα σφάλματος 0,3	336.192,80	292.102,40	303.683,46	253.938,30
Certainty 60% για πιθανότητα σφάλματος 0,4	368.572,00	297.664,70	323.484,30	272.403,90

	V-Model (ευρώ)	Spiral (ευρώ)	XP (ευρώ)	Scrum (ευρώ)
Certainty 60% για πιθανότητα σφάλματος 0,5	389.192,80	302.972,00	345.313,55	285.904,80

Στο Σχήμα 14 παρατίθεται το διάγραμμα που προκύπτει για τις παραπάνω τιμές του πίνακα 25.



Σχήμα 14: Τιμές κόστους για ποσοστό βεβαιότητας 60% με αλλαγή στην πιθανότητα σφάλματος

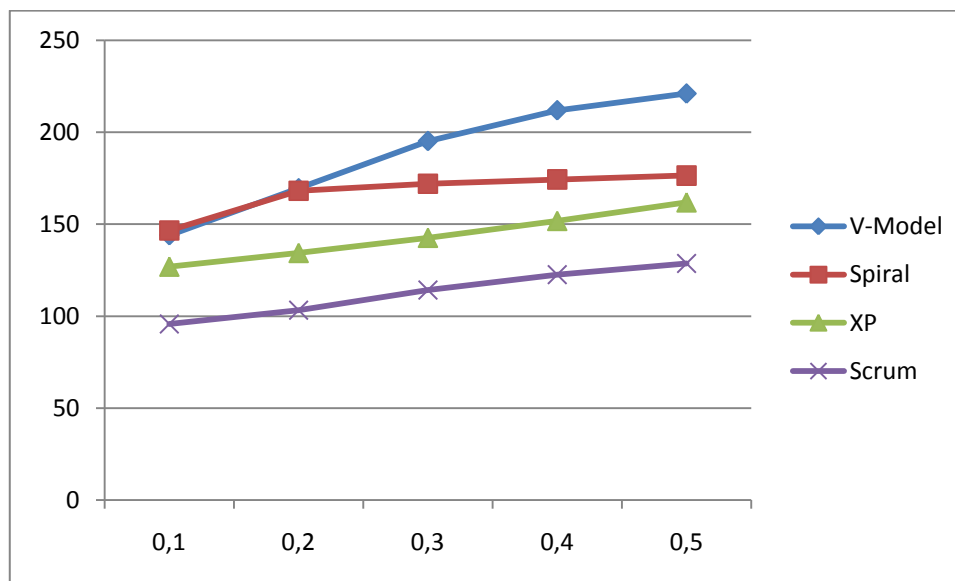
Στον Πίνακα 26 παρουσιάζεται το πως επηρεάζεται το ποσοστό βεβαιότητας 60% της κατανομής της συνολικής διάρκειας του κάθε μοντέλου. Το συμπέρασμα που προκύπτει είναι πως η Scrum σε όλες τις περιπτώσεις σφάλματος έχει την μικρότερη τιμή συνολικής διάρκειας. Τέλος, η Spiral και το V-Model έχουν εξ αρχής μεγαλύτερη συνολική διάρκεια σε σύγκριση με τις δύο agile μεθόδους.

Πίνακας 26: Τιμές διάρκειας (σε ημέρες) για ποσοστό βεβαιότητας 60% με αλλαγή στην πιθανότητα σφάλματος

	V-Model (ημέρες)	Spiral (ημέρες)	XP (ημέρες)	Scrum (ημέρες)
Certainty 60% για πιθανότητα σφάλματος 0,1	144,00	146,53	126,90	95,72
Certainty 60% για πιθανότητα σφάλματος 0,2	169,48	168,10	134,35	103,29
Certainty 60% για πιθανότητα σφάλματος 0,3	195,15	171,91	142,60	114,23

			V-Model (ημέρες)	Spiral (ημέρες)	XP (ημέρες)	Scrum (ημέρες)
Certainty	60%	για				
πιθανότητα	σφάλματος	0,4	211,86	174,31	151,80	122,56
Certainty	60%	για				
πιθανότητα	σφάλματος	0,5	220,95	176,43	161,90	128,63

Στο Σχήμα 15 παρατίθεται το διάγραμμα που προκύπτει για τις παραπάνω τιμές του πίνακα 26.



**Σχήμα 15: Τιμές διάρκειας (σε ημέρες) για ποσοστό βεβαιότητας 60% με αλλαγή στην πιθανότητα σφάλματος**

Το τελικό συμπέρασμα που προκύπτει είναι πως η εμφάνιση σφάλματος είναι πιο εύκολα διαχειρίσιμη από τα agile μοντέλα σε σύγκριση με τα traditional μοντέλα. Τέλος, σε όλες τις περιπτώσεις εμφάνισης σφάλματος, το έργο ολοκληρώνεται σε μικρότερο κόστος και σε μικρότερη διάρκεια όταν χρησιμοποιείται η Scrum επομένως σε περιβάλλοντα όπου η εμφάνιση σφάλματος έχει πιθανότητα εμφάνισης από 0,1 έως 0,5 φαίνεται πως η Scrum είναι η βέλτιστη επιλογή.

#### 5.4 Σενάριο με αλλαγή στον αριθμό γραμμών κώδικα

Στη συγκεκριμένη υποενότητα θα σχολιαστούν τα στατιστικά που έχουν προκύψει για το συνολικό κόστος και τη συνολική διάρκεια του έργου αλλάζοντας τον αριθμό

γραμμών κώδικα προς υλοποίηση. Το συμπέρασμα που εξάγεται από τα παρακάτω είναι το πως συμπεριφέρεται το κάθε μοντέλο για διαφορετικά μεγέθη έργου.

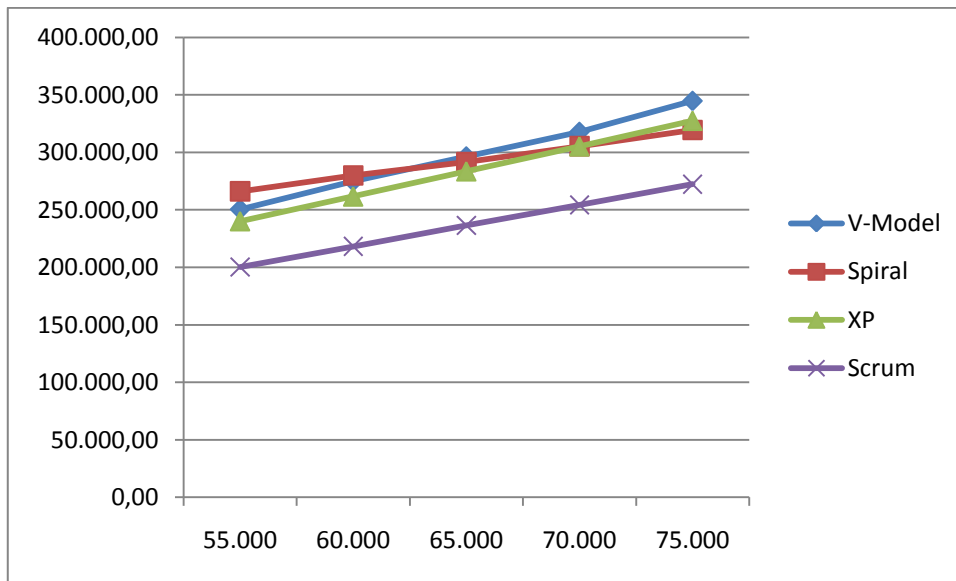
Το βασικό σενάριο, το οποίο έχει αναλυθεί στην υποενότητα 5.2, έχει εκτελεστεί για 65.000 γραμμές κώδικα και πιθανότητα σφάλματος 0,2. Τα αποτελέσματα της συγκεκριμένης υποενότητας θα επικεντρωθούν στο mean, maximum και σε ποσοστό βεβαιότητας 60% του κάθε μοντέλου για γραμμές κώδικα από 55.000 έως 75.000 με βήμα 5.000.

Στον Πίνακα 27 φαίνεται το πως επηρεάζεται το mean της κατανομής του συνολικού κόστους του κάθε μοντέλου. Το συμπέρασμα που προκύπτει είναι ότι η αύξηση των γραμμών κώδικα παρουσιάζει μια μετατόπιση στο mean των μοντέλων. Πιο συγκεκριμένα, φαίνεται πως όλα τα μοντέλα εμφανίζουν σχεδόν την ίδια αύξηση στο mean τους όταν αυξάνονται οι γραμμές κώδικα. Τέλος, η Scrum φαίνεται πως είναι και πάλι η βέλτιστη επιλογή βάσει του mean καθώς παρόλο που ο ρυθμός αύξησης του κόστους είναι σχεδόν ίδιος για όλα τα μοντέλα, η Scrum έχει το μικρότερο συνολικό κόστος σε όλες τις περιπτώσεις.

**Πίνακας 27: Mean τιμές κόστους με αλλαγή στις γραμμές κώδικα**

	V-Model (ευρώ)	Spiral (ευρώ)	XP (ευρώ)	Scrum (ευρώ)
Mean για LOCs 55.000	250.425,28	266.082,48	240.177,97	200.221,34
Mean για LOCs 60.000	274.809,56	279.892,26	261.670,00	218.209,43
Mean για LOCs 65.000	296.342,41	291.490,21	283.565,84	236.527,08
Mean για LOCs 70.000	317.689,20	305.322,33	305.258,60	254.291,76
Mean για LOCs 75.000	344.636,70	319.331,96	327.533,46	272.234,24

Στο Σχήμα 16 παρατίθεται το διάγραμμα που προκύπτει για τις παραπάνω τιμές του πίνακα 27.



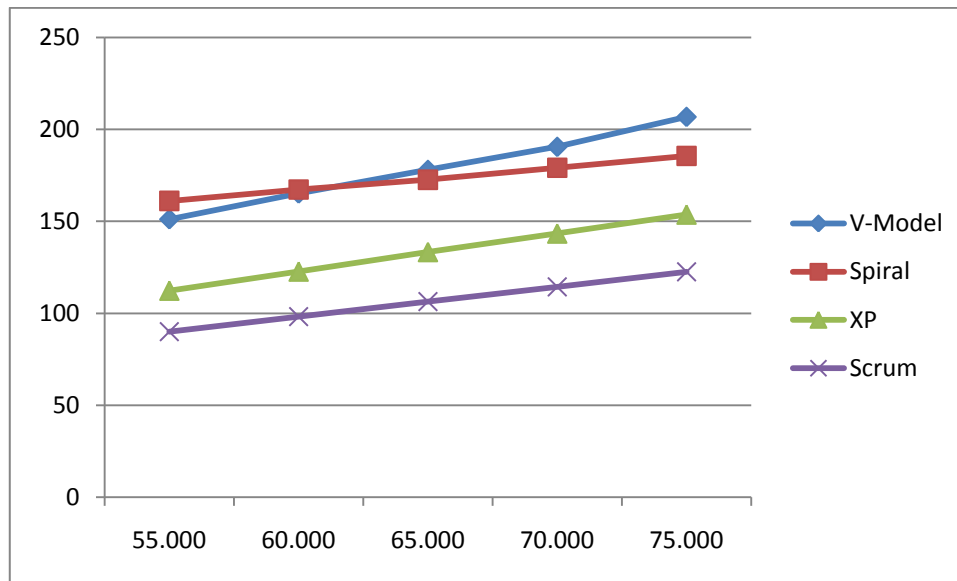
**Σχήμα 16: Mean τιμές κόστους με αλλαγή στις γραμμές κώδικα**

Στον Πίνακα 28 παρατίθεται το πως επηρεάζεται το mean της κατανομής της συνολικής διάρκειας του κάθε μοντέλου. Σε αντιστοιχία με το συνολικό κόστος, ο ρυθμός αύξησης της διάρκειας φαίνεται πως είναι σταθερός και σχεδόν ίδιος για όλα τα μοντέλα όσο αυξάνονται οι γραμμές κώδικα, ωστόσο η Scrum έχει τη μικρότερη συνολική διάρκεια σε όλες τις περιπτώσεις σε σύγκριση με τα υπόλοιπα μοντέλα.

**Πίνακας 28: Mean τιμές διάρκεια με αλλαγή στις γραμμές κώδικα**

	V-Model (ημέρες)	Spiral (ημέρες)	XP (ημέρες)	Scrum (ημέρες)
Mean για LOCs 55.000	151,15	161,07	112,3	90,06
Mean για LOCs 60.000	165,16	167,27	122,68	98,17
Mean για LOCs 65.000	177,97	172,6	133,28	106,42
Mean για LOCs 70.000	190,52	179,06	143,35	114,42
Mean για LOCs 75.000	206,82	185,59	153,66	122,51

Στο Σχήμα 17 παρατίθεται το διάγραμμα που προκύπτει για τις παραπάνω τιμές του πίνακα 28.



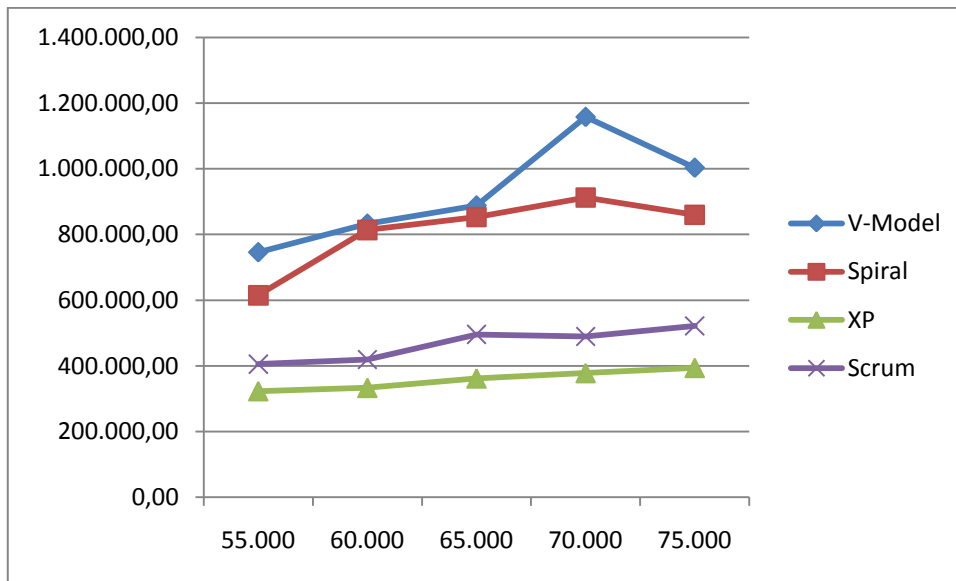
Σχήμα 17: Mean τιμές διάρκεια με αλλαγή στις γραμμές κώδικα

Στον Πίνακα 29 φαίνεται το πως επηρεάζεται το maximum της κατανομής του συνολικού κόστους του κάθε μοντέλου. Το συμπέρασμα που προκύπτει είναι πως η maximum τιμή του κόστους είναι εξαρχής μικρότερη για τα agile μοντέλα και όσο αυξάνονται οι γραμμές κώδικα παραμένει μικρότερη σε σύγκριση με τα traditional μοντέλα. Τέλος, η μικρότερη maximum τιμή κόστους εμφανίζεται σε όλες τις περιπτώσεις για την XP.

Πίνακας 29: Maximum τιμές κόστους με αλλαγή στις γραμμές κώδικα

	V-Model (ευρώ)	Spiral (ευρώ)	XP (ευρώ)	Scrum (ευρώ)
Maximum για LOCs 55.000	745.944,80	614.705,50	323.040,26	405.236,40
Maximum για LOCs 60.000	832.816,00	813.444,40	333.094,56	419.602,20
Maximum για LOCs 65.000	888.332,00	852.636,40	361.891,05	495.779,40
Maximum για LOCs 70.000	1.158.036,00	912.247,51	378.065,77	489.519,00
Maximum για LOCs 75.000	1.003.348,00	859.945,60	393.900,30	521.385,60

Στο Σχήμα 18 παρατίθεται το διάγραμμα που προκύπτει για τις παραπάνω τιμές του πίνακα 29.



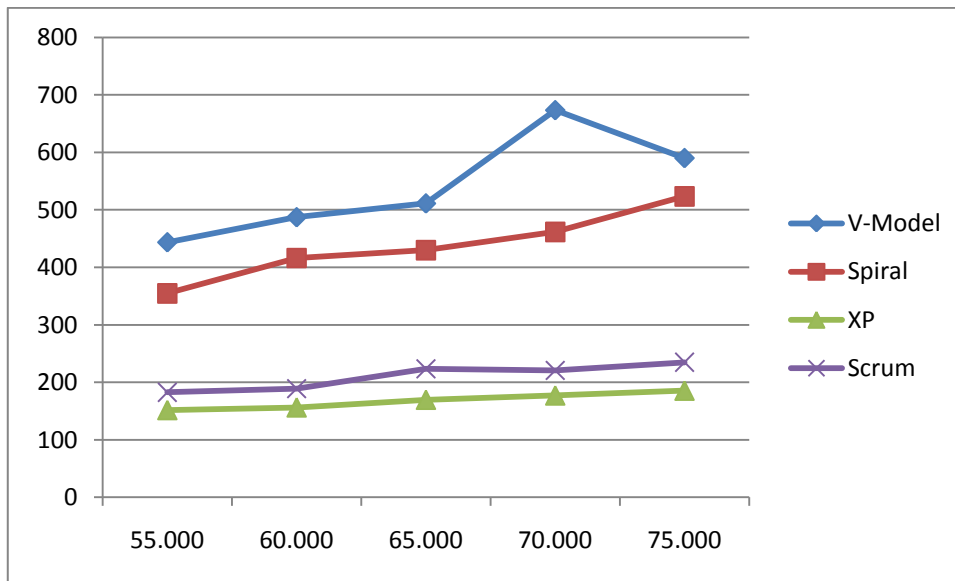
Σχήμα 18: Maximum τιμές κόστους με αλλαγή στις γραμμές κώδικα

Στον Πίνακα 30 φαίνεται το πως επηρεάζεται το maximum της κατανομής της συνολικής διάρκειας του κάθε μοντέλου. Το συμπέρασμα που προκύπτει είναι πως τα traditional μοντέλα παρουσιάζουν εξαρχής μεγαλύτερη τιμή στο maximum σε σύγκριση με τα agile μοντέλα. Τέλος, η μικρότερη maximum τιμή διάρκειας εμφανίζεται σε όλες περιπτώσεις για την XP.

Πίνακας 30: Maximum τιμές διάρκειας (σε ημέρες) με αλλαγή στις γραμμές κώδικα

	V-Model (ημέρες)	Spiral (ημέρες)	XP (ημέρες)	Scrum (ημέρες)
Maximum για LOCs 55.000	443,53	354,61	151,60	182,54
Maximum για LOCs 60.000	487,10	416,07	156,25	188,98
Maximum για LOCs 65.000	511,16	429,77	169,55	223,28
Maximum για LOCs 70.000	673,35	461,56	177,05	220,59
Maximum για LOCs 75.000	590,05	523,36	185,40	234,79

Στο Σχήμα 19 παρατίθεται το διάγραμμα που προκύπτει για τις παραπάνω τιμές του πίνακα 30.



Σχήμα 19: Maximum τιμές διάρκειας (σε ημέρες) με αλλαγή στις γραμμές κώδικα

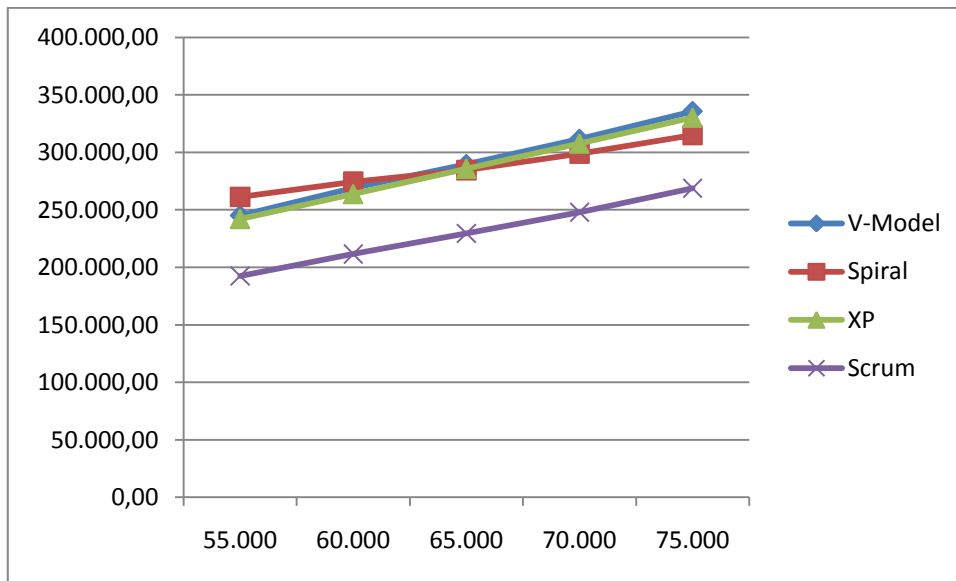
Στον Πίνακα 31 φαίνεται το πως επηρεάζεται το ποσοστό βεβαιότητας 60% της κατανομής του συνολικού κόστους του κάθε μοντέλου. Στον παρακάτω πίνακα είναι εμφανές πως σε ποσοστό βεβαιότητας 60%, η φθηνότερη επιλογή βάσει του τελικού κόστους είναι η Scrum. Όσον αφορά τα υπόλοιπα μοντέλα φαίνεται πως το κόστος που προκύπτει για ποσοστό βεβαιότητας 60% είναι σχεδόν ίδιο παρόλα αυτά αν λάβουμε υπόψη τους προηγούμενους πίνακες, προκύπτει πως το XP διαχειρίζεται καλύτερα την εμφάνιση σφάλματος.

Πίνακας 31: Τιμές κόστους για ποσοστό βεβαιότητας 60% με αλλαγή στις γραμμές κώδικα

	V-Model (ευρώ)	Spiral (ευρώ)	XP (ευρώ)	Scrum (ευρώ)
Certainty 60% για LOCs 55.000	245.150,40	261.166,01	242.127,39	192.441,30
Certainty 60% για LOCs 60.000	268.991,20	274.430,81	263.815,78	211.537,50
Certainty 60% για LOCs 65.000	289.627,20	284.496,40	285.789,36	229.564,50
Certainty 60% για LOCs 70.000	311.548,80	298.735,60	307.647,81	247.794,00
Certainty 60% για LOCs 75.000	335.693,60	314.982,00	330.142,98	268.750,80



Στο Σχήμα 20 παρατίθεται το διάγραμμα που προκύπτει για τις παραπάνω τιμές του πίνακα 31.



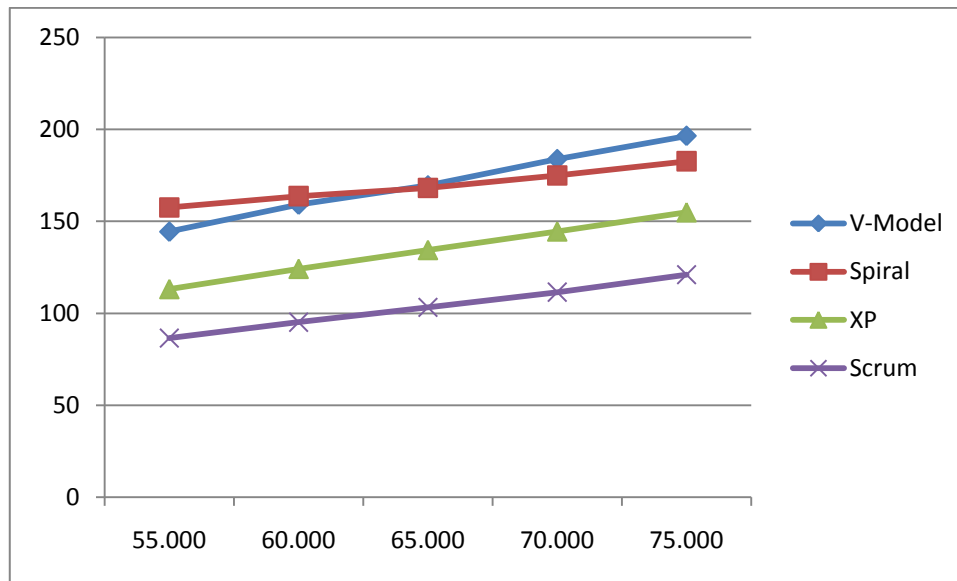
**Σχήμα 20: Τιμές κόστους για ποσοστό βεβαιότητας 60% με αλλαγή στις γραμμές κώδικα**

Στον Πίνακα 32 παρουσιάζεται το πως επηρεάζεται το ποσοστό βεβαιότητας 60% της κατανομής της συνολικής διάρκειας του κάθε μοντέλου. Το κυριότερο χαρακτηριστικό που είναι άξιο σχολιασμού από το συγκεκριμένο πίνακα είναι πως τα δύο traditional μοντέλα φαίνεται να έχουν μεγαλύτερες τιμές διάρκειας από τα agile μοντέλα. Επιπλέον, ανάμεσα στα δύο agile μοντέλα προκύπτει ότι για ποσοστό βεβαιότητας 60% η Scrum είναι η βέλτιστη επιλογή καθώς εμφανίζει τη μικρότερη συνολική διάρκεια.

**Πίνακας 32: Τιμές διάρκειας (σε ημέρες) για ποσοστό βεβαιότητας 60% με αλλαγή στις γραμμές κώδικα**

	V-Model (ημέρες)	Spiral (ημέρες)	XP (ημέρες)	Scrum (ημέρες)
Certainty 60% για LOCs 55.000	144,41	157,48	113,10	86,52
Certainty 60% για LOCs 60.000	159,08	163,69	124,10	95,16
Certainty 60% για LOCs 65.000	169,48	168,10	134,35	103,29
Certainty 60% για LOCs 70.000	183,75	174,88	144,45	111,48
Certainty 60% για LOCs 75.000	196,41	182,65	154,85	120,93

Στο Σχήμα 21 παρατίθεται το διάγραμμα που προκύπτει για τις παραπάνω τιμές του πίνακα 32.



**Σχήμα 21: Τιμές διάρκειας (σε ημέρες) για ποσοστό βεβαιότητας 60% με αλλαγή στις γραμμές κώδικα**

Το συνολικό συμπέρασμα που προκύπτει είναι πως η αλλαγή στο μέγεθος του έργου προκαλεί αύξηση τόσο στη διάρκεια όσο και στο κόστος του έργου παρόλα αυτά ο ρυθμός αύξησης είναι σταθερός. Επιπλέον, φαίνεται πως σε ποσοστό βεβαιότητας 60% η βέλτιστη επιλογή είναι η Scrum ωστόσο το μικρότερο maximum κόστος και η μικρότερη maximum διάρκεια προκύπτουν για την XP.

### 5.5 Σενάριο με αλλαγή στο μέγεθος της ομάδας ανάπτυξης

Στη συγκεκριμένη υποενότητα θα σχολιαστούν τα στατιστικά που έχουν προκύψει για το συνολικό κόστος και τη συνολική διάρκεια του έργου αλλάζοντας τον αριθμό των Seniors και των Juniors που παίρνουν μέρος στη φάση του development. Η αλλαγή των ατόμων επηρεάζει ανάλογα και το ποσοστό σφάλματος όπως έχει ήδη αναφερθεί στην υποενότητα 4.3. Το συμπέρασμα που θα εξαχθεί από τα παρακάτω είναι το πως επηρεάζεται το συνολικό κόστος και η συνολική διάρκεια αλλάζοντας τον αριθμό και την εμπειρία της ομάδας υλοποίησης.

Το βασικό σενάριο, το οποίο έχει ήδη παρουσιαστεί στην υποενότητα 5.2, έχει εκτελεστεί για 65.000 γραμμές κώδικα και πιθανότητα σφάλματος 0,2. Τα αποτελέσματα που παρατίθενται στη συνέχεια επικεντρώνονται στο σχολιασμό του mean, maximum

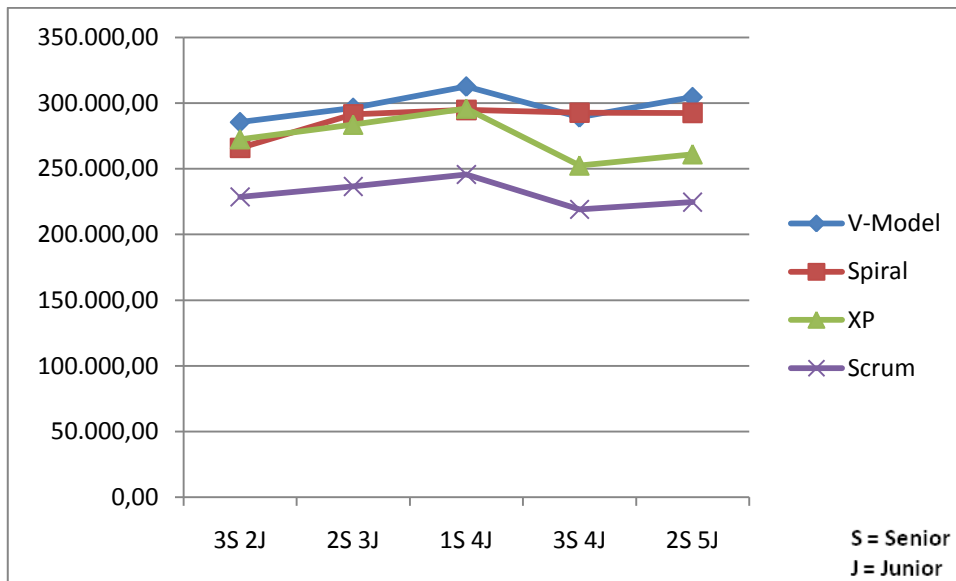
και σε ποσοστό βεβαιότητας 60% του κάθε μοντέλου για αλλαγή των ατόμων υλοποίησης.

Στον Πίνακα 33 παρουσιάζεται το πως επηρεάζεται το mean της κατανομής του συνολικού κόστους του κάθε μοντέλου. Αυτό που προκύπτει από τα παρακάτω είναι πως σε όλα τα μοντέλα δεν παρουσιάζεται σημαντική αυξομείωση στο συνολικό κόστος για την τιμή του mean με την προσθαφαίρεση ατόμων. Το παραπάνω οφείλεται στο γεγονός ότι το κόστος αυξάνεται λόγω των επιπλέον resources αλλά ταυτόχρονα μειώνεται λόγω της μείωσης της διάρκειας του έργου. Τέλος, προκύπτει ότι σε όλες τις περιπτώσεις το βέλτιστο μοντέλο είναι η Scrum καθώς το mean του κόστους είναι σε όλες τις περιπτώσεις μικρότερο σε σύγκριση με το mean των υπόλοιπων μοντέλων.

**Πίνακας 33: Mean τιμές κόστους με αλλαγή στον αριθμό των Seniors και των Juniors**

	V-Model (ευρώ)	Spiral (ευρώ)	XP (ευρώ)	Scrum (ευρώ)
Mean για 3 Seniors και 2 Juniors	285.469,30	265.762,34	272.500,27	228.588,99
Mean για 2 Seniors και 3 Juniors	296.342,41	291.490,21	283.565,84	236.527,08
Mean για 1 Seniors και 4 Juniors	312.515,05	294.729,53	295.818,95	245.600,75
Mean για 3 Seniors και 4 Juniors	289.110,74	292.570,80	252.460,60	219.003,31
Mean για 2 Seniors και 5 Juniors	304.468,58	292.502,83	261.032,12	224.584,34

Στο Σχήμα 22 παρατίθεται το διάγραμμα που προκύπτει για τις παραπάνω τιμές του πίνακα 33.



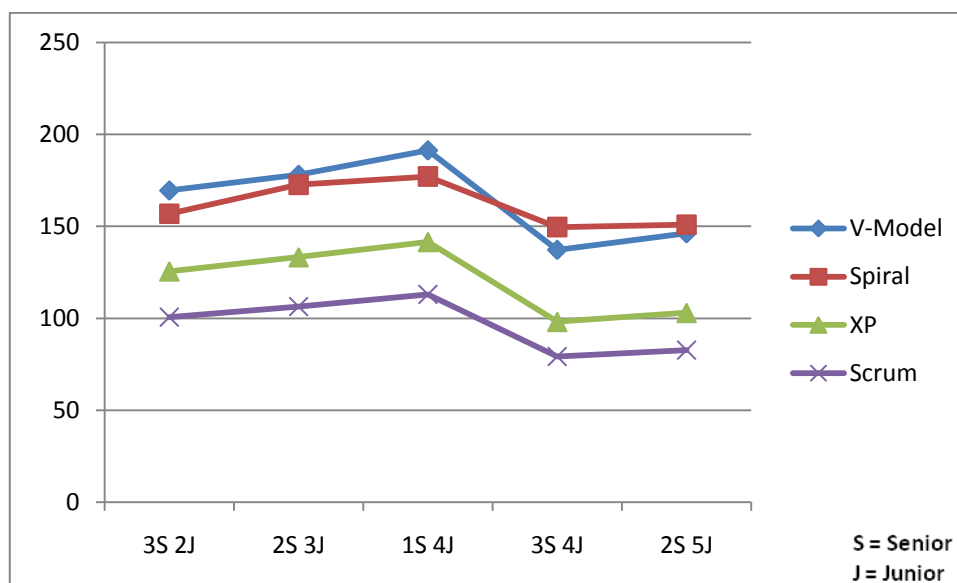
Σχήμα 22: Mean τιμές κόστους με αλλαγή στον αριθμό των Seniors και των Juniors

Στον Πίνακα 34 παρουσιάζεται το πως επηρεάζεται το mean της κατανομής της συνολικής διάρκειας του κάθε μοντέλου. Βάσει του παρακάτω πίνακα προκύπτει ότι τα αποτελέσματα για το mean της συνολικής διάρκειας με προσθαφαίρεση ή εναλλαγή ατόμου παρουσιάζουν διακυμάνσεις σε όλα τα μοντέλα. Η προσθήκη ατόμων βοηθάει στη μείωση της διάρκειας του έργου όπως είναι λογικό και ειδικά η προσθήκη ενός Senior ή η εναλλαγή ενός Senior με ένα Junior.

Πίνακας 34: Mean τιμές διάρκειας (σε ημέρες) με αλλαγή στον αριθμό των Seniors και των Juniors

	V-Model (ημέρες)	Spiral (ημέρες)	XP (ημέρες)	Scrum (ημέρες)
Mean για 3 Seniors και 2 Juniors	169,46	156,82	125,58	100,58
Mean για 2 Seniors και 3 Juniors	177,97	172,60	133,28	106,42
Mean για 1 Seniors και 4 Juniors	191,27	177,08	141,56	113,01
Mean για 3 Seniors και 4 Juniors	137,25	149,60	98,13	79,19
Mean για 2 Seniors και 5 Juniors	146,25	150,89	103,01	82,67

Στο Σχήμα 23 παρατίθεται το διάγραμμα που προκύπτει για τις παραπάνω τιμές του πίνακα 34.



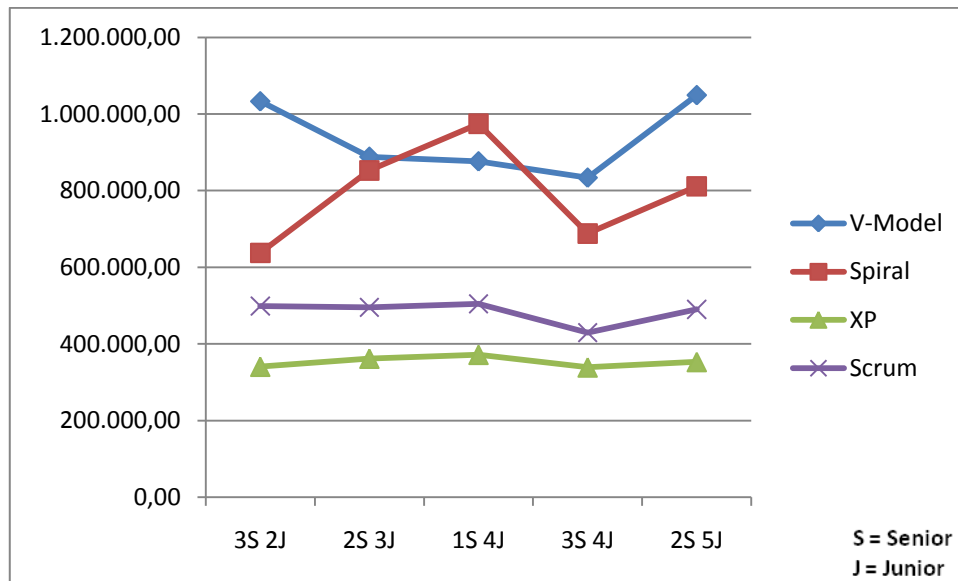
**Σχήμα 23: Mean τιμές διάρκειας (σε ημέρες) με αλλαγή στον αριθμό των Seniors και των Juniors**

Στον Πίνακα 35 παρουσιάζεται το πως επηρεάζεται το maximum της κατανομής του συνολικού κόστους του κάθε μοντέλου. Το συμπέρασμα που προκύπτει είναι πως η maximum τιμή του κόστους είναι μικρότερη για τα agile μοντέλα για όλους τους παρακάτω συνδυασμούς ατόμων υλοποίησης σε σύγκριση με τα traditional μοντέλα. Τέλος, η μικρότερη maximum τιμή κόστους εμφανίζεται σε όλες περιπτώσεις για την XP.

**Πίνακας 35: Maximum τιμές κόστους με αλλαγή στον αριθμό των Seniors και των Juniors**

	V-Model (ευρώ)	Spiral (ευρώ)	XP (ευρώ)	Scrum (ευρώ)
Maximum για 3 Seniors και 2 Juniors	1.033.249,60	637.782,00	340.889,52	498.869,20
Maximum για 2 Seniors και 3 Juniors	888.332,00	852.636,40	361.891,05	495.779,40
Maximum για 1 Seniors και 4 Juniors	876.296,00	974.495,20	371.939,44	504.781,20
Maximum για 3 Seniors και 4 Juniors	833.668,80	688.065,01	339.018,16	429.331,00
Maximum για 2 Seniors και 5 Juniors	1.049.428,00	811.351,60	353.145,10	490.202,00

Στο Σχήμα 24 παρατίθεται το διάγραμμα που προκύπτει για τις παραπάνω τιμές του πίνακα 35.



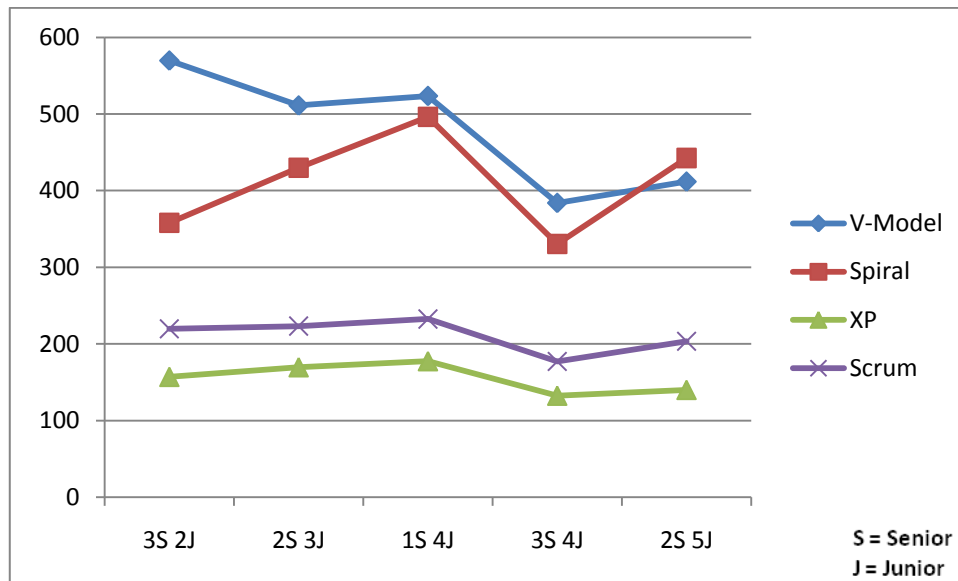
Σχήμα 24: Maximum τιμές κόστους με αλλαγή στον αριθμό των Seniors και των Juniors

Στον Πίνακα 36 παρουσιάζεται το πως επηρεάζεται το maximum της κατανομής της συνολικής διάρκειας του κάθε μοντέλου. Το συμπέρασμα που προκύπτει είναι πως τα traditional μοντέλα παρουσιάζουν εξαρχής μεγαλύτερη maximum τιμή σε σύγκριση με τα agile μοντέλα για όλους τους συνδυασμούς ατόμων υλοποίησης. Τέλος, η μικρότερη maximum τιμή διάρκειας εμφανίζεται σε όλες περιπτώσεις για την XP.

Πίνακας 36: Maximum τιμές διάρκειας (σε ημέρες) με αλλαγή στον αριθμό των Seniors και των Juniors

	V-Model (ημέρες)	Spiral (ημέρες)	XP (ημέρες)	Scrum (ημέρες)
Maximum για 3 Seniors και 2 Juniors	569,79	357,99	156,95	219,74
Maximum για 2 Seniors και 3 Juniors	511,16	429,77	169,55	223,28
Maximum για 1 Seniors και 4 Juniors	523,58	496,23	177,45	232,58
Maximum για 3 Seniors και 4 Juniors	383,97	330,50	132,40	177,24
Maximum για 2 Seniors και 5 Juniors	411,80	442,55	139,95	203,54

Στο Σχήμα 25 παρατίθεται το διάγραμμα που προκύπτει για τις παραπάνω τιμές του πίνακα 36.



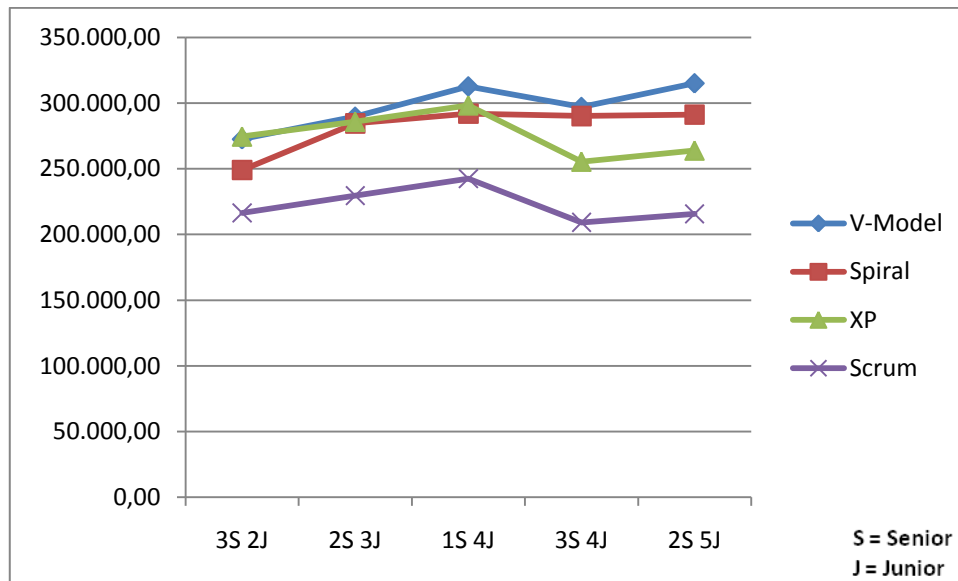
**Σχήμα 25: Maximum τιμές διάρκειας (σε ημέρες) με αλλαγή στον αριθμό των Seniors και των Juniors**

Στον Πίνακα 37 παρουσιάζεται το πως επηρεάζεται το ποσοστό βεβαιότητας 60% της κατανομής του συνολικού κόστους του κάθε μοντέλου. Σε ποσοστό βεβαιότητας 60% το μοντέλο με τη χαμηλότερη τιμή κόστους είναι η Scrum. Όσον αφορά τα υπόλοιπα μοντέλα φαίνεται πως το κόστος που προκύπτει για ποσοστό βεβαιότητας 60% είναι σχεδόν ίδιο παρόλα αυτά αν λάβουμε υπόψη τους προηγούμενους πίνακες, προκύπτει πως η XP διαχειρίζεται καλύτερα την εμφάνιση σφάλματος.

**Πίνακας 37: Τιμές κόστους για ποσοστό βεβαιότητας 60% με αλλαγή στον αριθμό των Seniors και των Juniors**

	V-Model (ευρώ)	Spiral (ευρώ)	XP (ευρώ)	Scrum (ευρώ)
Certainty 60% για 3 Seniors και 2 Juniors	272.563,20	249.108,41	274.516,98	216.451,20
Certainty 60% για 2 Seniors και 3 Juniors	289.627,20	284.496,40	285.789,36	229.564,50
Certainty 60% για 1 Seniors και 4 Juniors	312.712,00	291.958,20	298.279,05	242.529,00
Certainty 60% για 3 Seniors και 4 Juniors	297.041,60	290.081,00	255.409,54	209.186,40
Certainty 60% για 2 Seniors και 5 Juniors	315.079,20	291.184,00	263.863,62	215.674,00

Στο Σχήμα 26 παρατίθεται το διάγραμμα που προκύπτει για τις παραπάνω τιμές του πίνακα 37.



**Σχήμα 26: Τιμές κόστους για ποσοστό βεβαιότητας 60% με αλλαγή στον αριθμό των Seniors και των Juniors**

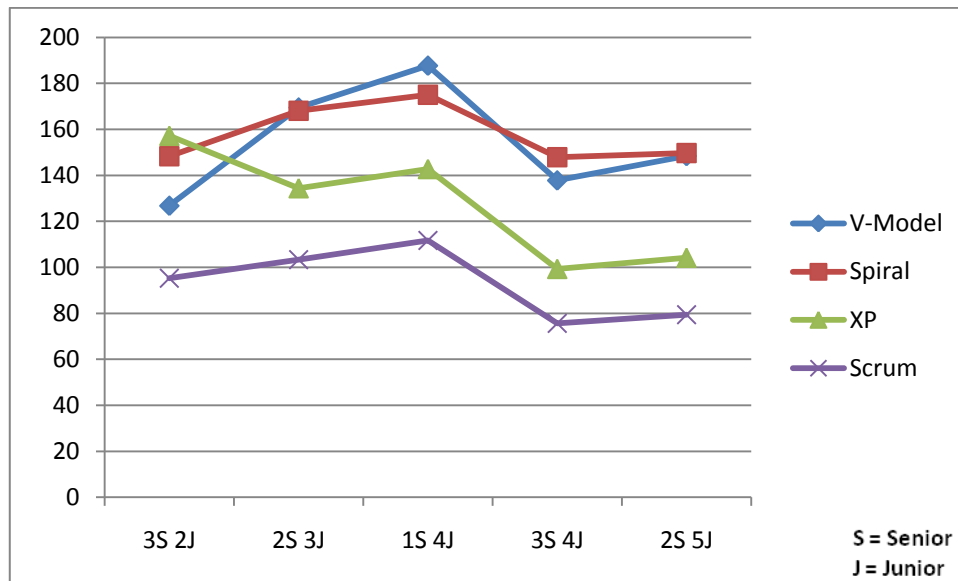
Στον Πίνακα 38 παρουσιάζεται το πως επηρεάζεται το ποσοστό βεβαιότητας 60% της κατανομής της συνολικής διάρκειας του κάθε μοντέλου. Προκύπτει ότι για ποσοστό βεβαιότητας 60% η Scrum εμφανίζει τη χαμηλότερη τιμή.

**Πίνακας 38: Τιμές διάρκειας (σε ημέρες) για ποσοστό βεβαιότητας 60% με αλλαγή στον αριθμό των Seniors και των Juniors**

	V-Model (ημέρες)	Spiral (ημέρες)	XP (ημέρες)	Scrum (ημέρες)
Certainty 60% για 3 Seniors και 2 Juniors	126,80	148,31	157,16	95,25
Certainty 60% για 2 Seniors και 3 Juniors	169,48	168,10	134,35	103,29
Certainty 60% για 1 Seniors και 4 Juniors	187,67	175,01	142,70	111,64
Certainty 60% για 3 Seniors και 4 Juniors	137,80	147,85	99,25	75,66
Certainty 60% για 2 Seniors και 5 Juniors	148,41	149,70	104,10	79,38



Στο Σχήμα 27 παρατίθεται το διάγραμμα που προκύπτει για τις παραπάνω τιμές του πίνακα 38.



**Σχήμα 27: Τιμές διάρκειας (σε ημέρες) για ποσοστό βεβαιότητας 60% με αλλαγή στον αριθμό των Seniors και των Juniors**

Το συνολικό συμπέρασμα που προκύπτει είναι ότι η προσθαφαίρεση των ατόμων υλοποίησης σε συνδυασμό με την τροποποίηση της πιθανότητας σφάλματος ανάλογα με την εμπειρία των ατόμων δεν προκαλεί μεγάλες διακυμάνσεις στις τιμές του κόστους σε κάθε μοντέλο, ενώ αντίθετα, όσον αφορά τη διάρκεια προκύπτει αισθητή μείωση από την αύξηση του μεγέθους της ομάδας. Τέλος, προκύπτει ότι η προσθήκη senior σε σχέση με την προσθήκη junior μειώνει πιο πολύ την τελική διάρκεια γεγονός που είναι λογικό καθώς αν προστεθεί ένας πιο έμπειρος θα μπορεί να υλοποιήσει μεγαλύτερο όγκο και τελικά να μειωθεί η διάρκεια του έργου. Επίσης, φαίνεται πως σε ποσοστό βεβαιότητας 60% η βέλτιστη επιλογή είναι η Scrum, ωστόσο το μικρότερο maximum κόστος και η μικρότερη maximum διάρκεια προκύπτουν για την XP.

## 5.6 Αποτελέσματα sensitivity ανάλυσης

Τα sensitivity charts που έχουν παραχθεί έχουν σαν στόχο την μελέτη των μεταβλητών και τον υπολογισμό της ποσοστιαίας τους επιρροής σε κάθε μοντέλο. Πιο συγκεκριμένα, η μελέτη των sensitivity charts έχει μεγάλο ενδιαφέρον όταν επικεντρώνεται στο κάθε μοντέλο ξεχωριστά καθώς ο υπολογισμός της επιρροής της κάθε μεταβλητής πραγματοποιείται λαμβάνοντας υπόψη και τις υπόλοιπες μεταβλητές

για τις οποίες έχει οριστεί assumption. Στις ενότητες που ακολουθούν, θα σχολιαστούν τα αποτελέσματα τόσο ανά μεταβλητή περιβάλλοντος όσο και ανά μοντέλο με σκοπό την εξαγωγή συμπερασμάτων για τις μεταβλητές που επηρεάζουν σε μεγαλύτερο βαθμό τα τελικά αποτελέσματα.

### 5.6.1 Σενάριο με σταθερή ομάδα

Αρχικά, έχουν παραχθεί sensitivity charts μέσω των οποίων θα εξαχθούν συμπεράσματα για τις μεταβλητές που επηρεάζουν το τελικό κόστος και την τελική διάρκεια χωρίς αλλαγή στο μέγεθος της ομάδας.

Τα assumptions που έχουν οριστεί αφορούν τις εξής μεταβλητές του συστήματος:

- Πιθανότητα εμφάνισης σφάλματος (fault probability): οι τιμές κυμαίνονται από 0 έως 0,5 σε ομοιόμορφη κατανομή.
- Μέγεθος του έργου (LOC): οι τιμές κυμαίνονται από 55.000 έως 75.000 σε ομοιόμορφη κατανομή.
- Γραμμές κώδικα που παράγει ανά ημέρα ένας senior (Senior LOC per day): οι τιμές ακολουθούν κανονική κατανομή με μέση τιμή τις 250 γραμμές ανά ημέρα και τυπική απόκλιση 30 γραμμές.
- Γραμμές κώδικα που παράγει ανά ημέρα ένας junior (Junior LOC per day): οι τιμές ακολουθούν κανονική κατανομή με μέση τιμή τις 200 γραμμές ανά ημέρα και τυπική απόκλιση 30 γραμμές.

Ο Πίνακας 39 παρουσιάζει το πόσο επηρεάζεται το κόστος που προκύπτει για το κάθε μοντέλο από την αλλαγή της πιθανότητας σφάλματος. Βάσει του παρακάτω πίνακα προκύπτει ότι το μοντέλο που επηρεάζεται σε μεγαλύτερο βαθμό από την αλλαγή στην πιθανότητα σφάλματος είναι το V-Model στοιχείο που είναι αναμενόμενο καθώς δεν έχει διαδικασίες που να διαχειρίζονται την αλλαγή στην πιθανότητα σφάλματος. Οι agile μεθόδοι, δηλαδή η XP και η Scrum, φαίνεται πως επηρεάζονται σε αντίστοιχο βαθμό. Όσον αφορά το κόστος της Spiral φαίνεται πως επηρεάζεται σε μικρότερο βαθμό. Τέλος, προκύπτει ότι η αύξηση πιθανότητας σφάλματος αυξάνει το τελικό κόστος.

Πίνακας 39: Επιρροή της πιθανότητας σφάλματος στο τελικό κόστος

	Fault Probability
V-Model	52.9%

	Fault Probability
Spiral	14.0%
XP	28.0%
Scrum	22.9%

Ο Πίνακας 40 παρουσιάζει το πόσο επηρεάζεται το κόστος που προκύπτει για το κάθε μοντέλο από την αλλαγή των γραμμών κώδικα. Αρχικά, φαίνεται πως το Spiral επηρεάζεται σε μεγάλο βαθμό από την αλλαγή στο μέγεθος του έργου. Το στοιχείο αυτό δικαιολογεί την χαμηλή επιρροή που είχε η αλλαγή της πιθανότητας σφάλματος στο κόστος του Spiral καθώς ο υπολογισμός της επιρροής της κάθε μεταβλητής γίνεται λαμβάνοντας υπόψη και την επιρροή των υπόλοιπων μεταβλητών του περιβάλλοντος. Σε αντιστοιχία με τα παραπάνω, το V-Model, του οποίου το κόστος επηρεάζεται σε μεγάλο βαθμό από την αλλαγή σφάλματος (πίνακας 39), φαίνεται πως επηρεάζεται σε μικρότερο βαθμό από την αλλαγή στις γραμμές κώδικα σε σύγκριση με τα υπόλοιπα μοντέλα. Επιπλέον, όσον αφορά την XP και τη Scrum φαίνεται ότι και τα δύο μοντέλα ανταποκρίνονται με παρόμοιο τρόπο. Τέλος, και στα τέσσερα μοντέλα η αύξηση των γραμμών κώδικα συνεπάγεται και αύξηση στο κόστος.

**Πίνακας 40: Επιρροή των γραμμών κώδικα στο τελικό κόστος**

	LOC
V-Model	21.6%
Spiral	40.5%
XP	33.4%
Scrum	35.8%

Ο Πίνακας 41 παρουσιάζει το πόσο επηρεάζεται το κόστος που προκύπτει για το κάθε μοντέλο από την αλλαγή των γραμμών κώδικα που παράγει ένας senior μέσα σε μία ημέρα. Το συμπέρασμα που προκύπτει είναι πως και στα τέσσερα μοντέλα η αύξηση των γραμμών κώδικα που παράγει ένας senior developer μειώνουν σε μικρό ποσοστό το τελικό κόστος.

**Πίνακας 41: Επιρροή γραμμών κώδικα senior developer στο τελικό κόστος**

	Senior LOC per day
V-Model	-8.0%
Spiral	-14.8%

	Senior LOC per day
XP	-12.6%
Scrum	-13.2%

Ο Πίνακας 42 παρουσιάζει το πόσο επηρεάζεται το κόστος που προκύπτει για το κάθε μοντέλο από την αλλαγή των γραμμών κώδικα που παράγει ένας junior μέσα σε μία ημέρα. Το συμπέρασμα που προκύπτει είναι πως και στα τέσσερα μοντέλα η αύξηση των γραμμών κώδικα που παράγει ένας junior developer μειώνουν το τελικό κόστος. Επιπλέον, συγκρίνοντας τα αποτελέσματα που έχουν προκύψει στον πίνακα 41 και 42 προκύπτει ότι η αλλαγή της αποδοτικότητας των junior developers επηρεάζει σε μεγαλύτερο βαθμό το κόστος σε σχέση με την αλλαγή της αποδοτικότητας των senior developers.

**Πίνακας 42: Επιρροή γραμμών κώδικα junior developer στο τελικό κόστος**

	Junior LOC per day
V-Model	-17.6%
Spiral	-30.7%
XP	-26.1%
Scrum	-28.1%

Οι πίνακες που ακολουθούν παρουσιάζουν την επιρροή των μεταβλητών του περιβάλλοντος στην τελική διάρκεια του εκάστοτε μοντέλου. Βάσει των αποτελεσμάτων προκύπτει ότι οι μεταβλητές εμφανίζουν παρόμοια ποσοστιαία επιρροή τόσο στο συνολικό κόστος όσο και στη συνολική διάρκεια του έργου.

**Πίνακας 43: Επιρροή της πιθανότητας σφάλματος στη τελική διάρκεια**

	Fault Probability
V-Model	40.2%
Spiral	12.8%
XP	27.9%
Scrum	22.8%

**Πίνακας 44: Επιρροή των γραμμών κώδικα στη τελική διάρκεια**

	LOC
V-Model	27.3%

	LOC
Spiral	41.0%
XP	33.4%
Scrum	35.8%

**Πίνακας 45: Επιρροή γραμμών κώδικα senior developer στη τελική διάρκεια**

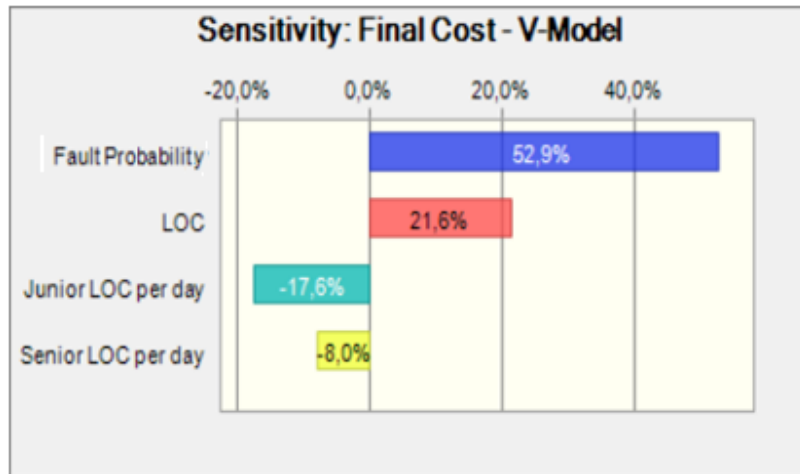
	Senior LOC per day
V-Model	-10.3%
Spiral	-15.1%
XP	-12.6%
Scrum	-13.2%

**Πίνακας 46: Επιρροή γραμμών κώδικα senior developer στη τελική διάρκεια**

	Junior LOC per day
V-Model	-22.2%
Spiral	-31.1%
XP	-26.1%
Scrum	-28.2%

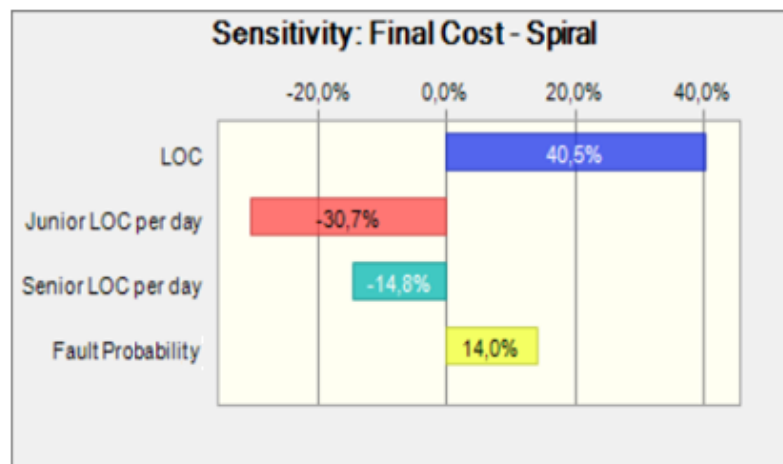
Στη συνέχεια παρατίθενται τα sensitivity charts που έχουν προκύψει για την ποσοστιαία επιρροή της κάθε μεταβλητής στο τελικό κόστος του κάθε μοντέλου.

Το Σχήμα 28 παρουσιάζει την ποσοστιαία επιρροή στο κόστος του V-Model για τις μεταβλητές στις οποίες έχει οριστεί assumption. Βάσει του παρακάτω διαγράμματος προκύπτει ότι η αύξηση της πιθανότητας σφάλματος και των γραμμών κώδικα προς υλοποίηση αυξάνουν το τελικό κόστος. Αντίθετα, η αύξηση της αποδοτικότητας των εργαζομένων μειώνει το τελικό κόστος. Επιπλέον, η μεταβλητή που αυξάνει σε μεγαλύτερο βαθμό το κόστος είναι η αύξηση της πιθανότητας σφάλματος ενώ η μεταβλητή που μειώνει σε μεγαλύτερο βαθμό το κόστος είναι η αύξηση της αποδοτικότητας των junior εργαζομένων.



Σχήμα 28: Sensitivity Chart για το συνολικό κόστος του V-Model

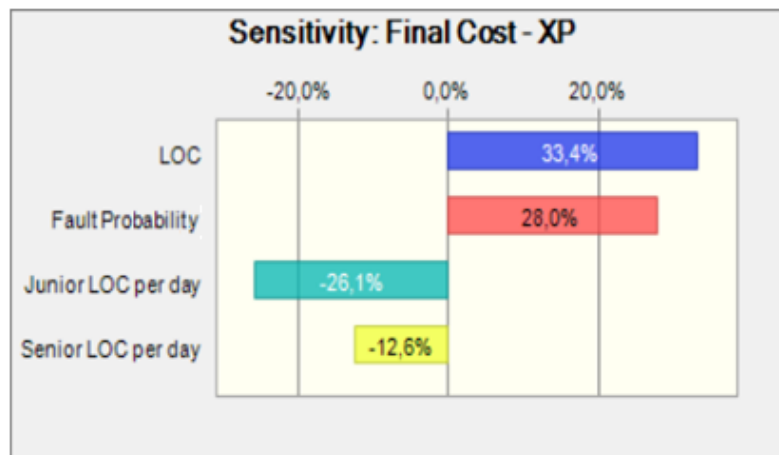
Το Σχήμα 29 παρουσιάζει την ποσοστιαία επιρροή στο κόστος του Spiral για τις μεταβλητές στις οποίες έχει οριστεί assumption. Βάσει του παρακάτω διαγράμματος προκύπτει ότι η αύξηση της πιθανότητας σφάλματος και των γραμμών κώδικα προς υλοποίηση αυξάνουν το τελικό κόστος. Αντίθετα, η αύξηση της αποδοτικότητας των εργαζομένων μειώνει το τελικό κόστος. Επιπλέον, η μεταβλητή που αυξάνει σε μεγαλύτερο βαθμό το κόστος είναι η αύξηση των γραμμών κώδικα ενώ η μεταβλητή που μειώνει σε μεγαλύτερο βαθμό το κόστος είναι η αύξηση της αποδοτικότητας των junior εργαζομένων.



Σχήμα 29: Sensitivity Chart για το συνολικό κόστος του Spiral

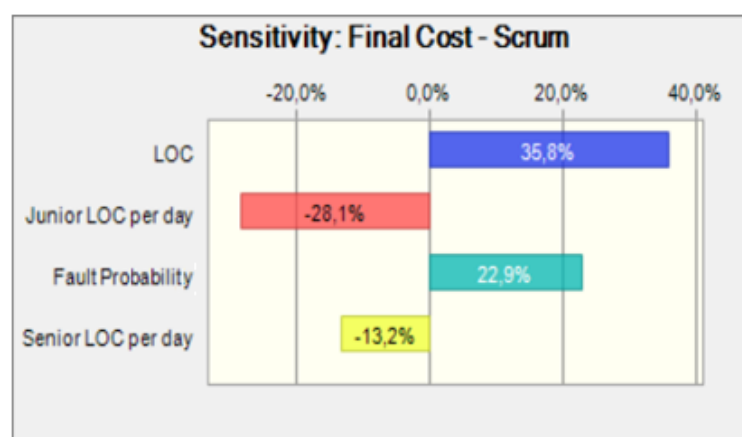
Το Σχήμα 30 παρουσιάζει την ποσοστιαία επιρροή στο κόστος της XP για τις μεταβλητές στις οποίες έχει οριστεί assumption. Τα αποτελέσματα του διαγράμματος δείχνουν ότι και πάλι η αύξηση της αποδοτικότητας μειώνει το κόστος. Επιπλέον, την

μεγαλύτερη επιρροή στην αύξηση του κόστους την παρουσιάζει η αύξηση των γραμμών κώδικα ενώ την μεγαλύτερη μείωση η αύξηση της αποδοτικότητας των junior εργαζομένων.



Σχήμα 30: Sensitivity Chart για το συνολικό κόστος της XP

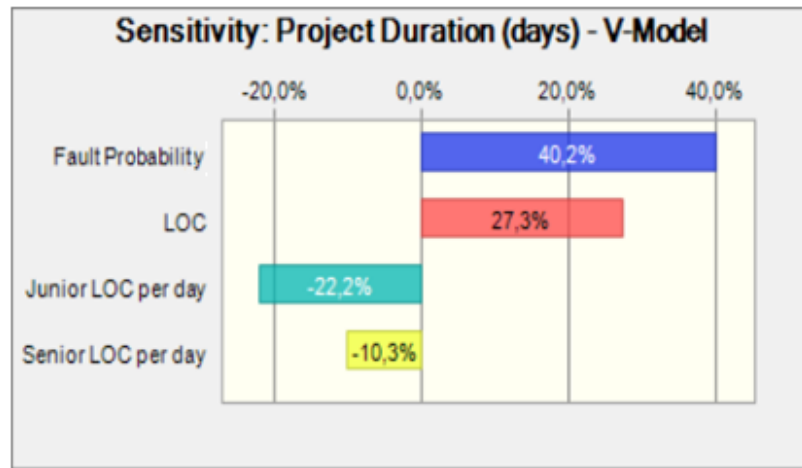
Το Σχήμα 31 παρουσιάζει την ποσοστιαία επιρροή στο κόστος της Scrum για τις μεταβλητές στις οποίες έχει οριστεί assumption. Αρχικά, φαίνεται ότι η αύξηση της πιθανότητας σφάλματος και των γραμμών κώδικα προς υλοποίηση αυξάνουν το τελικό κόστος ενώ η αύξηση της αποδοτικότητας των εργαζομένων το μειώνει. Τέλος, η μεταβλητή που αυξάνει σε μεγαλύτερο βαθμό το κόστος είναι η αύξηση των γραμμών κώδικα ενώ η μεταβλητή που μειώνει σε μεγαλύτερο βαθμό το κόστος είναι η αύξηση της αποδοτικότητας των junior εργαζομένων.



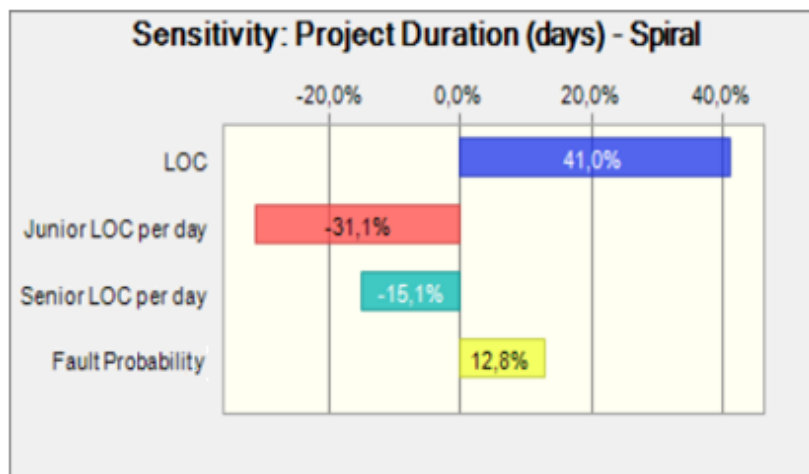
Σχήμα 31: Sensitivity Chart για το συνολικό κόστος της Scrum

Στη συνέχεια παρατίθενται τα sensitivity charts που έχουν προκύψει για την ποσοστιαία επιρροή της κάθε μεταβλητής στην τελική διάρκεια του κάθε μοντέλου. Το

συμπέρασμα που προκύπτει από τα παρακάτω διαγράμματα σε συνδυασμό με τον σχολιασμό που έχουμε ήδη πραγματοποιήσει για τα διαγράμματα του κόστους είναι πως οι μεταβλητές επηρεάζουν σε ίδιο ποσοστό το τελικό κόστος και την τελική διάρκεια στο κάθε μοντέλο.

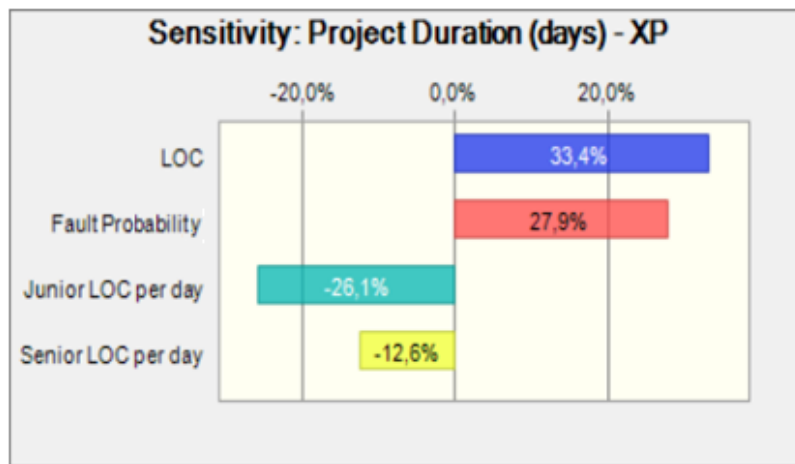


Σχήμα 32: Sensitivity Chart για τη συνολική διάρκεια του V-Model

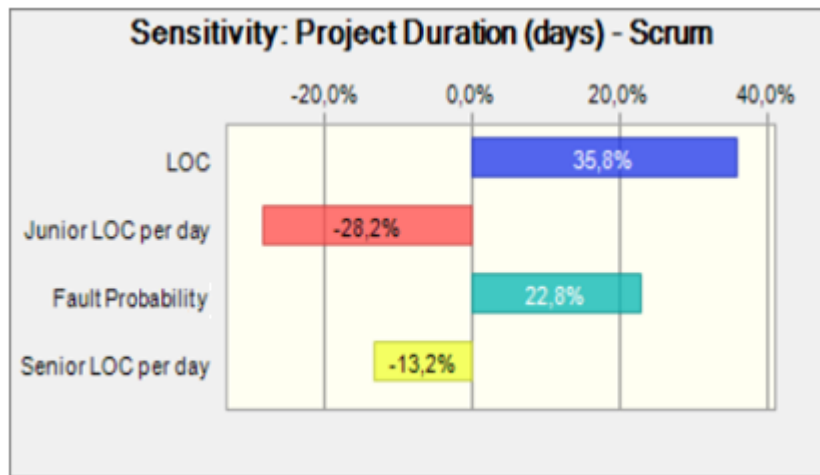


Σχήμα 33: Sensitivity Chart τη συνολική διάρκεια του Spiral





Σχήμα 34: Sensitivity Chart τη συνολική διάρκεια της XP



Σχήμα 35: Sensitivity Chart τη συνολική διάρκεια της Scrum

Το γενικό συμπέρασμα που προκύπτει είναι πως όταν υπάρχει σταθερή ομάδα εργασίας, η αύξηση της αποδοτικότητας μειώνει τόσο το τελικό κόστος όσο και την τελική διάρκεια. Τέλος, η αύξηση του μεγέθους του έργου και της πιθανότητας εμφάνισης σφάλματος έχουν αρνητική επίδραση στο τελικό κόστος και στην τελική διάρκεια όλων των μοντέλων.

### 5.6.2 Σενάριο με αλλαγή στο μέγεθος της ομάδας

Στη συγκεκριμένη υποενοότητα έχουν παραχθεί sensitivity charts τα οποία θα βοηθήσουν στον προσδιορισμό των μεταβλητών που επηρεάζουν το τελικό κόστος και την τελική διάρκεια όταν μεταβάλλεται το μέγεθος της ομάδας.

Τα assumptions που έχουν οριστεί αφορούν τις εξής μεταβλητές του συστήματος:

- Μέγεθος του έργου (LOC): οι τιμές κυμαίνονται από 55.000 έως 75.000 σε ομοιόμορφη κατανομή.
- Γραμμές κώδικα που παράγει ανά ημέρα ένας senior (Senior LOC per day): οι τιμές ακολουθούν κανονική κατανομή με μέση τιμή τις 250 γραμμές ανά ημέρα και τυπική απόκλιση 30 γραμμές.
- Γραμμές κώδικα που παράγει ανά ημέρα ένας junior (Junior LOC per day): οι τιμές ακολουθούν κανονική κατανομή με μέση τιμή τις 200 γραμμές ανά ημέρα και τυπική απόκλιση 30 γραμμές.
- Αριθμός senior εργαζομένων (Number of Seniors): οι τιμές κυμαίνονται διακριτά από 1 έως 3 σε ομοιόμορφη κατανομή.
- Αριθμός junior εργαζομένων (Number of Juniors): οι τιμές κυμαίνονται διακριτά από 2 έως 5 σε ομοιόμορφη κατανομή.

Ο Πίνακας 47 παρουσιάζει το πόσο επηρεάζεται το κόστος που προκύπτει για το κάθε μοντέλο από την αλλαγή των γραμμών κώδικα. Βάσει των παρακάτω προκύπτει ότι μοντέλο που επηρεάζεται σε μεγαλύτερο βαθμό από την αύξηση γραμμών κώδικα είναι το Spiral ενώ σε μικρότερο βαθμό η XP.

**Πίνακας 47: Επιρροή των γραμμών κώδικα στο τελικό κόστος**

	LOC
V-Model	38.2%
Spiral	43,0%
XP	29,7%
Scrum	34,1%

Ο Πίνακας 48 παρουσιάζει το πόσο επηρεάζεται το κόστος που προκύπτει για το κάθε μοντέλο από την αλλαγή των γραμμών κώδικα που παράγει ένας senior μέσα σε μία ημέρα. Βάσει του παρακάτω πίνακα προκύπτει ότι η αύξηση των γραμμών που παράγει ένας senior μέσα σε μία μέρα μειώνει το κόστος σε όλα τα μοντέλα.

**Πίνακας 48: Επιρροή γραμμών κώδικα senior developer στο τελικό κόστος**

	Senior LOC per day
V-Model	-12.8%

	Senior LOC per day
Spiral	-14,7%
XP	-8,4%
Scrum	-9,4%

Ο Πίνακας 49 παρουσιάζει το πόσο επηρεάζεται το κόστος που προκύπτει για το κάθε μοντέλο από την αλλαγή των γραμμών κώδικα που παράγει ένας junior μέσα σε μία ημέρα. Βάσει του παρακάτω πίνακα προκύπτει ότι η αύξηση των γραμμών που παράγει ένας junior μέσα σε μία μέρα μειώνει το κόστος σε όλα τα μοντέλα. Επιπλέον, σε σύγκριση με τα αποτελέσματα του πίνακα 48 προκύπτει ότι η αύξηση της παραγωγικότητας των junior εργαζομένων επηρεάζουν σε μεγαλύτερο βαθμό το κόστος.

**Πίνακας 49: Επιρροή γραμμών κώδικα junior developer στο τελικό κόστος**

	Junior LOC per day
V-Model	-34.5%
Spiral	-39,6%
XP	-25,3%
Scrum	-29,7%

Με δεδομένο ότι η αλλαγή του αριθμού των seniors επηρεάζει και την πιθανότητα εμφάνισης σφάλματος λόγω της συσχέτισης που έχουμε ορίσει, ο Πίνακας 50 δείχνει την αλλαγή που προκύπτει στο τελικό κόστος όταν αλλάζοντας τον αριθμό των senior ατόμων επηρεάζεται και η πιθανότητα εμφάνισης σφάλματος. Βάσει των παρακάτω προκύπτει ότι η προσθήκη senior εργαζομένων σε μια ομάδα μειώνει το τελικό κόστος σε όλα μοντέλα εκτός της Spiral.

**Πίνακας 50: Επιρροή του αριθμού των senior developers στο τελικό κόστος**

	Number of Seniors
V-Model	-13.3%
Spiral	0,0%
XP	-23,3%
Scrum	-17,9%

Ο Πίνακας 51 παρουσιάζει το πόσο επηρεάζεται το κόστος που προκύπτει για το κάθε μοντέλο από την αλλαγή του αριθμού των junior developers σε σχέση με την πιθανότητα εμφάνισης σφάλματος. Βάσει των παρακάτω προκύπτει ότι η προσθήκη junior εργαζομένων σε μια ομάδα μειώνει το τελικό κόστος σε όλα μοντέλα εκτός της Spiral παρόλα αυτά η μείωση που προκύπτει από την προσθήκη junior είναι μικρότερη σε σύγκριση με την μείωση που προκύπτει από την προσθήκη senior. Το χαρακτηριστικό αυτό μπορεί να δικαιολογηθεί από το γεγονός ότι η προσθήκη junior αυξάνει σε μεγαλύτερο βαθμό την πιθανότητα εμφάνισης σφάλματος.

**Πίνακας 51: Επιρροή του αριθμού των junior developers στο τελικό κόστος**

	Number of Juniors
V-Model	-2.0%
Spiral	2,8%
XP	-13,3%
Scrum	-8,9%

Ο Πίνακας 52 παρουσιάζει το πόσο επηρεάζεται η διάρκεια που προκύπτει για το κάθε μοντέλο από την αλλαγή των γραμμών κώδικα. Βάσει των παρακάτω είναι εμφανές ότι και τα τέσσερα μοντέλα συμπεριφέρονται με τον ίδιο τρόπο στην αύξηση των γραμμών κώδικα. Το συμπέρασμα που προκύπτει είναι πως η αύξηση των γραμμών αυξάνει σε μικρό βαθμό τη διάρκεια του έργου.

**Πίνακας 52: Επιρροή των γραμμών κώδικα στη τελική διάρκεια**

	LOC
V-Model	12,6%
Spiral	10,4%
XP	12,0%
Scrum	11,9%

Ο Πίνακας 53 παρουσιάζει το πόσο επηρεάζεται η διάρκεια που προκύπτει για το κάθε μοντέλο από την αλλαγή των γραμμών κώδικα που παράγει ένας senior μέσα σε μία ημέρα. Το συμπέρασμα που εξάγεται είναι πως και τα τέσσερα μοντέλα συμπεριφέρονται με τον ίδιο τρόπο στην αύξηση των γραμμών κώδικα που παράγει ένας senior εργαζόμενος και πως η αύξηση των γραμμών μειώνει σε μικρό βαθμό τη διάρκεια του έργου.

**Πίνακας 53: Επιρροή γραμμών κώδικα senior developer στη τελική διάρκεια**

	Senior LOC per day
V-Model	-4,3%
Spiral	-3,9%
XP	-3,6%
Scrum	-3,3%

Ο Πίνακας 54 παρουσιάζει το πόσο επηρεάζεται η διάρκεια που προκύπτει για το κάθε μοντέλο από την αλλαγή των γραμμών κώδικα που παράγει ένας junior μέσα σε μία ημέρα. Από τα δεδομένα του πίνακα φαίνεται ότι και τα τέσσερα μοντέλα συμπεριφέρονται με τον ίδιο τρόπο στην αύξηση των γραμμών κώδικα που παράγει ένας junior εργαζόμενος. Επιπλέον, προκύπτει ότι η αλλαγή στην παραγωγικότητα των junior εργαζομένων μειώνει σε μεγαλύτερο βαθμό τη διάρκεια σε σύγκριση με την αλλαγή στην παραγωγικότητα των seniors (πίνακας 53). Το παραπάνω δικαιολογείται από το γεγονός ότι οι junior εργαζόμενοι είναι περισσότεροι άρα παράγουν μεγαλύτερο όγκο κώδικα αθροιστικά.

**Πίνακας 54: Επιρροή γραμμών κώδικα junior developer στη τελική διάρκεια**

	Junior LOC per day
V-Model	-10,9%
Spiral	-9,5%
XP	-10,2%
Scrum	-10,1%

Ο Πίνακας 55 παρουσιάζει το πόσο επηρεάζεται η διάρκεια που προκύπτει για το κάθε μοντέλο από την αλλαγή του αριθμού των senior developers σε σχέση με την πιθανότητα εμφάνισης σφάλματος. Είναι εμφανές ότι σε όλα τα μοντέλα η αύξηση του αριθμού των seniors μειώνει σε μεγάλο βαθμό την τελική διάρκεια.

**Πίνακας 55: Επιρροή του αριθμού των senior developers στη τελική διάρκεια**

	Number of Seniors
V-Model	-39,9%
Spiral	-38,2%
XP	-39,3%
Scrum	-39,0%

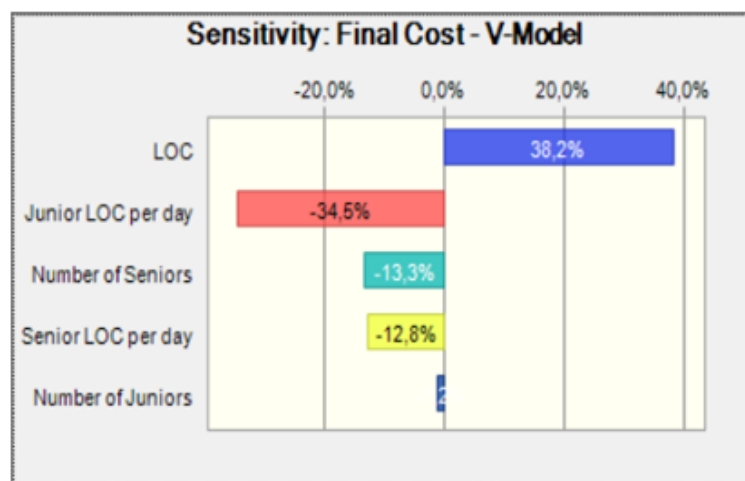
Ο Πίνακας 56 παρουσιάζει το πόσο επηρεάζεται η διάρκεια που προκύπτει για το κάθε μοντέλο από την αλλαγή του αριθμού των senior developers σε συσχέτιση με την πιθανότητα εμφάνισης σφάλματος. Βάσει των παρακάτω προκύπτει ότι η αύξηση του του αριθμού των juniors μειώνει σε μεγάλο βαθμό την τελική διάρκεια. Επιπλέον, η προσθήκη junior εργαζομένου φαίνεται πως μειώνει τη διάρκεια του έργου λιγότερο σε σχέση με την προσθήκη senior (πίνακας 55), στοιχείο που μπορεί να δικαιολογηθεί από το γεγονός ότι οι seniors παράγουν περισσότερες γραμμές κώδικα μέσα σε μία ημέρα.

**Πίνακας 56: Επιρροή του αριθμού των junior developers στη τελική διάρκεια**

	Number of Juniors
V-Model	-32,3%
Spiral	-38,0%
XP	-34,9%
Scrum	-35,6%

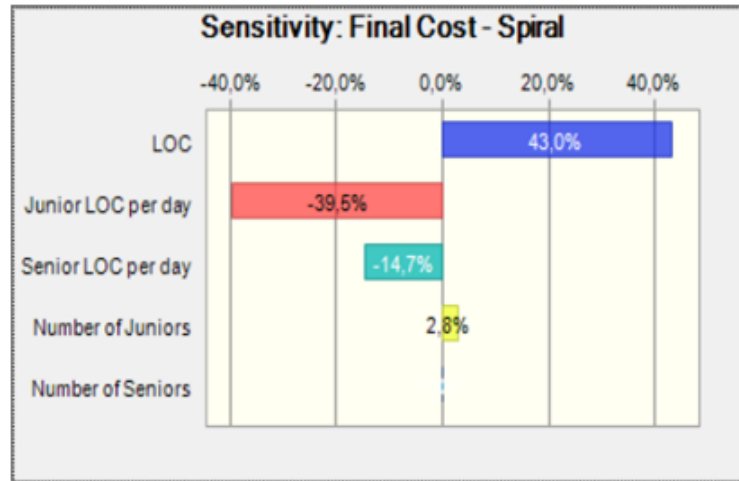
Στη συνέχεια παρατίθενται τα sensitivity charts που έχουν προκύψει για την ποσοστιαία επιρροή της κάθε μεταβλητής στο τελικό κόστος του κάθε μοντέλου.

Το Σχήμα 36 παρουσιάζει την ποσοστιαία επιρροή στο κόστος του V-Model για τις μεταβλητές στις οποίες έχει οριστεί assumption. Από το παρακάτω διάγραμμα προκύπτει ότι η μεταβλητή που αυξάνει σε μεγαλύτερο βαθμό το κόστος είναι η αύξηση των γραμμών κώδικα ενώ η μεταβλητή που μειώνει σε μεγαλύτερο βαθμό το κόστος είναι η αύξηση της παραγωγικότητας των junior εργαζομένων,



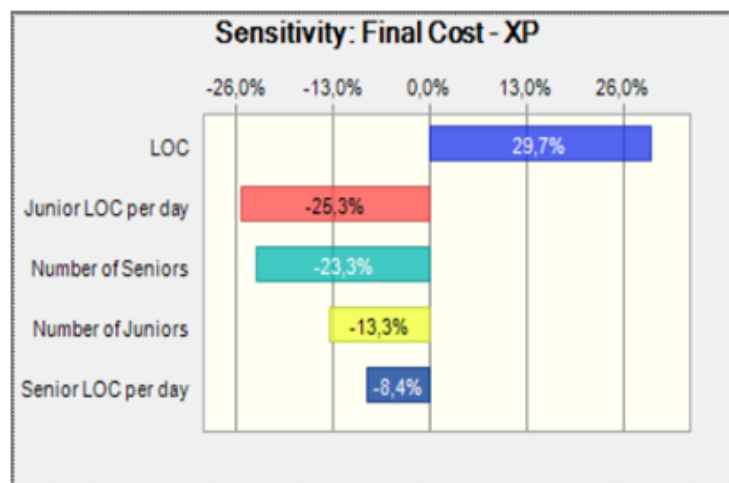
**Σχήμα 36: Sensitivity Chart για το συνολικό κόστος του V-Model**

Το Σχήμα 37 παρουσιάζει την ποσοστιαία επιρροή στο κόστος του Spiral για τις μεταβλητές στις οποίες έχει οριστεί assumption. Βάσει των παρακάτω είναι εμφανές ότι η αύξηση των γραμμών κώδικα αυξάνουν σε μεγάλο βαθμό το τελικό κόστος και αντίστοιχα η αύξηση της παραγωγικότητας των juniors το μειώνει σε μεγάλο βαθμό.



Σχήμα 37: Sensitivity Chart για το συνολικό κόστος του Spiral

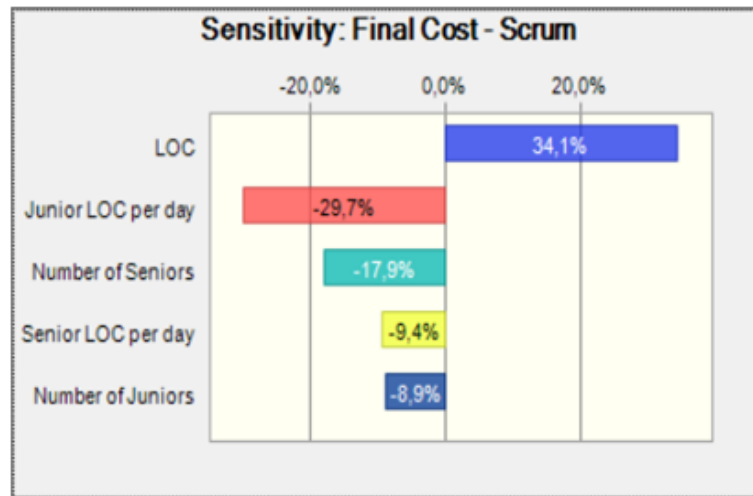
Το Σχήμα 38 παρουσιάζει την ποσοστιαία επιρροή στο κόστος της XP για τις μεταβλητές στις οποίες έχει οριστεί assumption. Βάσει του παρακάτω, προκύπτει ότι όλες οι μεταβλητές μειώνουν το κόστος της XP εκτός από την αύξηση γραμμών κώδικα η οποία αυξάνει σε μεγάλο ποσοστό το τελικό κόστος.



Σχήμα 38: Sensitivity Chart για το συνολικό κόστος της XP

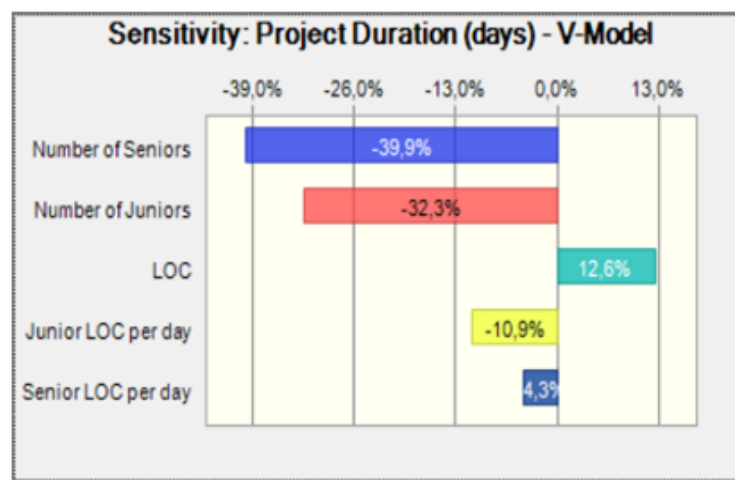
Το Σχήμα 39 παρουσιάζει την ποσοστιαία επιρροή στο κόστος της Scrum για τις μεταβλητές στις οποίες έχει οριστεί assumption. Σε αντιστοιχία με τα παραπάνω, το

κόστος της Scrum αυξάνει σε μεγάλο βαθμό όταν αυξάνονται οι γραμμές κώδικα ενώ όλες οι υπόλοιπες μεταβλητές το μειώνουν.



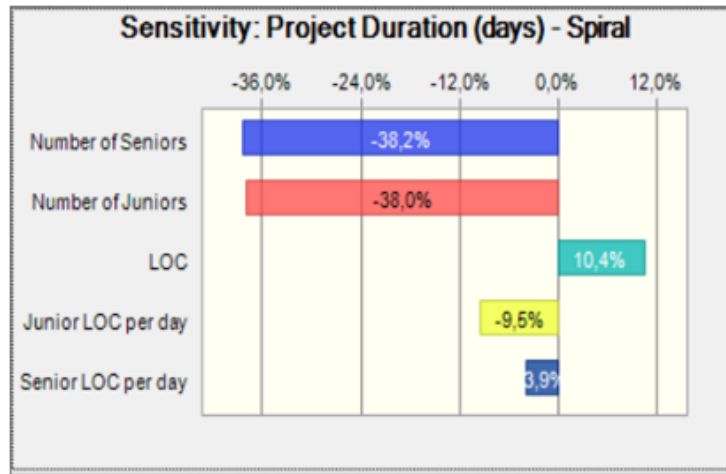
Σχήμα 39: Sensitivity Chart για το συνολικό κόστος της Scrum

Στη συνέχεια παρατίθενται τα sensitivity charts που έχουν προκύψει για την ποσοστιαία επιρροή της κάθε μεταβλητής στην τελική διάρκεια του κάθε μοντέλου. Το συμπέρασμα που προκύπτει από τα παρακάτω διαγράμματα είναι πως η διάρκεια και των τεσσάρων μοντέλων επηρεάζεται με τον ίδιο τρόπο από τις μεταβλητές του περιβάλλοντος. Πιο συγκεκριμένα, για όλα τα μοντέλα προκύπτει ότι η αύξηση του μεγέθους της ομάδας και της παραγωγικότητας των εργαζομένων μειώνουν σε μεγάλο βαθμό την τελική διάρκεια του έργου. Τέλος, σε όλα τα μοντέλα η αύξηση των γραμμών κώδικα προκαλεί μικρή αύξηση στη συνολική διάρκεια του έργου.

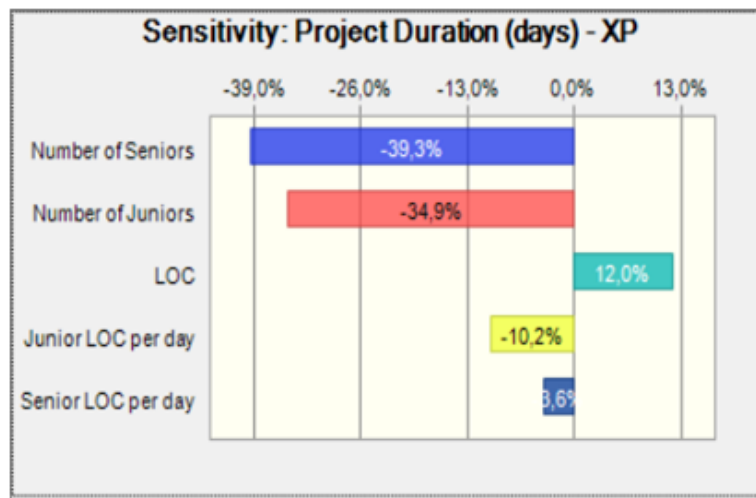


Σχήμα 40: Sensitivity Chart για τη συνολική διάρκεια του V-Model

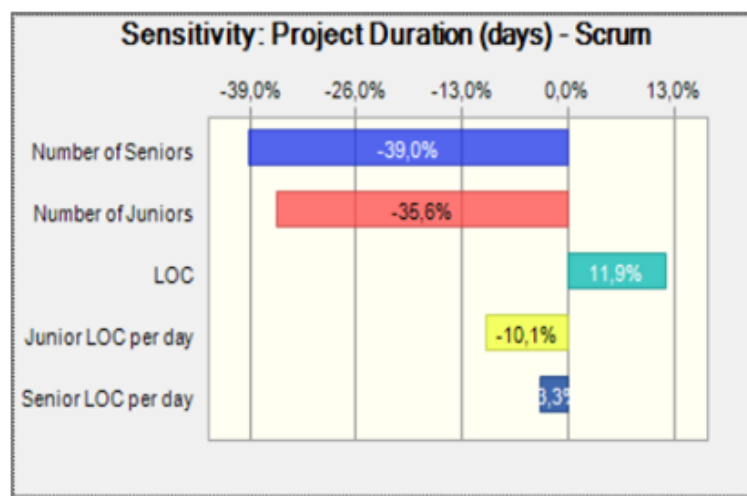




Σχήμα 41: Sensitivity Chart τη συνολική διάρκεια του Spiral



Σχήμα 42: Sensitivity Chart τη συνολική διάρκεια της XP



Σχήμα 43: Sensitivity Chart τη συνολική διάρκεια της Scrum

Το γενικό συμπέρασμα που προκύπτει από το παραπάνω είναι πως η αύξηση του μεγέθους της ομάδας επηρεάζει σε μικρό βαθμό το τελικό κόστος παρόλα αυτά σε όλες τις περιπτώσεις μειώνει σε μεγάλο βαθμό την τελική διάρκεια. Επιπλέον, σημαντικό εργαλείο για τη μείωση τόσο του κόστους όσο και της τελικής διάρκειας αποτελεί η αύξηση της παραγωγικότητας των μελών της ομάδας.

## 6. ΣΥΜΠΕΡΑΣΜΑΤΑ

Ο σκοπός της εργασίας είναι να διερευνηθεί ο τρόπος με τον οποίο συγκεκριμένα μοντέλα ανάπτυξης λογισμικού αντιμετωπίζουν την εμφάνιση τυχαίων σφαλμάτων στην ανάπτυξη λογισμικού με στόχο τη βελτίωση της διαδικασίας ανάπτυξης και την καλύτερη κοστολόγηση του έργου. Πιο συγκεκριμένα, εξετάστηκε η περίπτωση ανάπτυξης ενός έργου μεσαίου μεγέθους κάνοντας χρήση δύο traditional μοντέλων (V-Model, Spiral) και δύο agile (XP, Scrum).

Αρχικά, εκτελέστηκε σενάριο με ύπαρξη μηδενικού σφάλματος. Το συμπέρασμα που εξάγεται είναι πως σε περιβάλλοντα χωρίς σφάλμα τα traditional και τα agile μοντέλα έχουν αντίστοιχη συμπεριφορά. Αναλυτικότερα, η Scrum εμφανίζει τις μικρότερες τιμές κόστους, επίσης τόσο το V-Model όσο και η Spiral είναι φθηνότερες από την XP. Τέλος, στο σενάριο μηδενικού σφάλματος είναι εμφανές πως η διαφοροποίηση που εμφανίζεται μεταξύ των τελικών τιμών κόστους και διάρκειας οφείλεται στο κόστος και στη διάρκεια των περιφερειακών φάσεων που ορίζει η θεωρία του κάθε μοντέλου καθώς το κόστος και ο χρόνος που δαπανάται για το development του έργου είναι ίδιος για όλα τα μοντέλα.

Βάσει των αποτελεσμάτων του βασικού σεναρίου προκύπτει ότι οι agile μέθοδοι διαχειρίζονται καλύτερα την ύπαρξη σφάλματος κατά τη διαδικασία ανάπτυξης ενός έργου σε αντίθεση με τις traditional μεθόδους. Αναλυτικότερα, τα δύο traditional μοντέλα εμφανίζουν πιθανές τιμές κόστους και διάρκειας με μεγάλο εύρος τιμών. Επίσης, σε ποσοστά βεβαιότητας από 60% και πάνω εμφανίζουν και τα δύο μοντέλα αρκετά μεγάλες τιμές κόστους και διάρκειας. Σε αντίθεση, τα agile μοντέλα λόγω των συχνών επαναληπτικών ελέγχων που πραγματοποιούνται εμφανίζουν μικρότερες διακυμάνσεις τόσο στις τιμές του κόστους όσο και της διάρκειας.

Επιπρόσθετα, ελέγχθηκε η συμπεριφορά των μοντέλων σε περιπτώσεις όπου η πιθανότητα εμφάνισης σφάλματος κυμαίνεται από 0,1 έως 0,5. Αρχικά, για τα traditional μοντέλα επιβεβαιώνεται το συμπέρασμα που έχει εξαχθεί από το βασικό σενάριο καθώς όσο αυξάνεται η πιθανότητα εμφάνισης σφάλματος τόσο πιο δύσκολα το διαχειρίζονται. Το παραπάνω χαρακτηριστικό δικαιολογείται από το γεγονός ότι τα traditional μοντέλα πραγματοποιούν τον έλεγχο της υλοποίησης χωρίς τη χρήση επαναληπτικών διαδικασιών. Όσον αφορά την XP, προκύπτει ότι διαχειρίζεται την εμφάνιση σφάλματος έχοντας το μικρότερο εύρος τιμών σε σύγκριση με τα υπόλοιπα μοντέλα, στοιχείο που οφείλεται στον εβδομαδιαίο επαναληπτικό έλεγχο που ορίζει η θεωρία της, ωστόσο σε

ποσοστά βεβαιότητας έως και 90% είναι ακριβότερη και πιο χρονοβόρα σε σύγκριση με τη Scrum. Συμπερασματικά, προκύπτει ότι με τη χρήση του μοντέλου Scrum το έργο ολοκληρώνεται σε μικρότερο κόστος και διάρκεια κατά το 90% των περιπτώσεων.

Τα επόμενα σενάρια που εκτελέστηκαν αφορούν την αλλαγή κάποιων παραμέτρων εισόδου με σκοπό να ελεγχθεί το πως αυτές οι παράμετροι επηρεάζουν το τελικό αποτέλεσμα των μοντέλων. Αρχικά, εκτελέστηκε σενάριο με αλλαγή του μέγεθος του έργου προς υλοποίηση. Το συμπέρασμα που προκύπτει είναι ότι σε όλα τα μοντέλα η αλλαγή στο μέγεθος του έργου προκαλεί αύξηση τόσο στη διάρκεια όσο και στο κόστος του έργου. Επιπλέον, φαίνεται πως σε ποσοστό βεβαιότητας 60% η βέλτιστη επιλογή είναι η Scrum ωστόσο το μικρότερο maximum κόστος και η μικρότερη maximum διάρκεια προκύπτουν για την XP. Το τελευταίο σενάριο που εκτελέστηκε αφορά την αλλαγή του μεγέθους ή της εμπειρίας της ομάδας υλοποίησης. Το συμπέρασμα που εξάγεται βάσει των αποτελεσμάτων είναι ότι η προσθαφαίρεση των ατόμων της ομάδας ανάπτυξης δεν επηρεάζει σε μεγάλο βαθμό το τελικό κόστος, σε αντίθεση με την τελική διάρκεια στην οποία εμφανίζεται αισθητή μείωση. Επίσης, προκύπτει ότι η προσθήκη senior εργαζομένου σε σχέση με την προσθήκη junior μειώνει πιο πολύ την τελική διάρκεια γεγονός που είναι λογικό καθώς ένας senior εργαζόμενος θα μπορεί να υλοποιήσει μεγαλύτερο όγκο δουλειάς και τελικά να μειωθεί η διάρκεια του έργου. Τέλος, φαίνεται πως σε ποσοστό βεβαιότητας 60%, η βέλτιστη επιλογή είναι η Scrum, ωστόσο το μικρότερο maximum κόστος και η μικρότερη maximum διάρκεια προκύπτουν για την XP.

Τέλος, πραγματοποιήθηκε μία sensitivity ανάλυση. Βάσει των αποτελεσμάτων της, είναι δυνατό να εντοπιστούν οι μεταβλητές περιβάλλοντος οι οποίες επηρεάζουν σε μεγαλύτερο βαθμό το τελικό αποτέλεσμα κόστους και διάρκειας. Σε περιπτώσεις όπου υπάρχει σταθερή ομάδα εργασίας, η αύξηση του μεγέθους του έργου και της πιθανότητας εμφάνισης σφάλματος έχουν αρνητική επίδραση στο τελικό κόστος και στην τελική διάρκεια όλων των μοντέλων. Όσον αφορά περιπτώσεις όπου η ομάδα εργασίας διαφοροποιείται, προκύπτει ότι η αύξηση του μεγέθους της ομάδας επηρεάζει σε μικρό βαθμό το τελικό κόστος παρόλα αυτά σε όλες τις περιπτώσεις μειώνει σε μεγάλο βαθμό την τελική διάρκεια. Τέλος, σε κάθε περίπτωση η αύξηση της αποδοτικότητας των μελών της ομάδας οδηγεί σε μείωση τόσο του τελικού κόστους όσο και της τελικής διάρκειας.

Εν κατακλείδι, βάσει όλων των παραπάνω εξάγεται το συμπέρασμα ότι οι agile μέθοδοι, λόγω των επαναληπτικών μηχανισμών ελέγχου, έχουν τη δυνατότητα να διαχειρίζονται καλύτερα το σφάλμα. Επιπλέον, η αύξηση της παραγωγικότητας της ομάδας υλοποίησης μειώνει σε κάθε περίπτωση το τελικό κόστος και την τελική διάρκεια.

## 7. ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΕΚΤΑΣΕΙΣ

Στα πλαίσια της συγκεκριμένης εργασίας μελετήθηκε το πως ανταποκρίνονται δύο traditional και δύο agile μοντέλα στην ανάπτυξη ενός έργου μεσαίου μεγέθους. Η μελέτη επικεντρώθηκε στο πως επηρεάζεται το κάθε μοντέλο από την ύπαρξη τυχαίου σφάλματος, από την αλλαγή της πιθανότητας εμφάνισής του, από την αύξηση των ανθρώπινων πόρων ή την αλλαγή εμπειρίας τους κι από την αλλαγή των γραμμών κώδικα προς υλοποίηση. Η συγκεκριμένη ενότητα επικεντρώνεται στους τρόπους με τους οποίους μπορεί να προεκταθεί η συγκεκριμένη μελέτη ώστε να καλύπτει περισσότερα περιβάλλοντα ανάπτυξης λογισμικού.

Αρχικά, για την ανάπτυξη έργων μεσαίου μεγέθους υπάρχουν κι άλλα μοντέλα που μπορούν να αναλυθούν στο επίπεδο που αναλύθηκε το V-Model, το Spiral, η XP και η Scrum όπως για παράδειγμα το Incremental και το RAD. Ακόμη, εκτός από την ανάλυση μοντέλων που επικεντρώνονται στην ανάπτυξη μεσαίων έργων λογισμικού, θα μπορούσαν να μελετηθούν και μοντέλα που προορίζονται για την ανάπτυξη έργων μικρού μεγέθους, όπως για παράδειγμα το μοντέλο Waterfall αλλά και μεγάλου μεγέθους, όπως για παράδειγμα το μοντέλο Iterative.

Μια ακόμη πιθανή προέκταση της συγκεκριμένης έρευνας είναι η μελέτη επιπλέον μεταβλητών που επηρεάζουν την ανάπτυξη ενός έργου. Στην υπάρχουσα μελέτη, η αλλαγή εμπειρίας στους ανθρώπινους πόρους επηρεάζει το ποσοστό εμφάνισης σφάλματος παρόλα αυτά το μέγεθος του σφάλματος που προκύπτει είναι τυχαίο και δεν επηρεάζεται από την εμπειρία της ομάδας. Θα μπορούσε να μελετηθεί ακόμη το πως επηρεάζεται το έργο από τη διακύμανση στην απόδοση του κάθε εργαζόμενου. Επιπλέον, στη μελέτη που έχει πραγματοποιηθεί έχει δοθεί βάση στην αύξηση των συνολικών γραμμών κώδικα λόγω σφάλματος κι όχι λόγω αλλαγής απαιτήσεων. Με δεδομένο ότι η αλλαγή των απαιτήσεων είναι συχνή στον τομέα της ανάπτυξης λογισμικού, θα μπορούσε η συγκεκριμένη μελέτη να προεκταθεί με τέτοιο τρόπο ώστε να εξαχθούν συμπεράσματα και για το πως αντιδράει το κάθε μοντέλο στην αλλαγή των απαιτήσεων από τον πελάτη.

**ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ**

<b>Ξενόγλωσσος όρος</b>	<b>Ελληνικός όρος</b>
Project	Έργο
Case study	Εξεταζόμενο σενάριο
Sensitivity Analysis	Ανάλυση Ευαισθησίας
Senior developer	Πρεσβύτερος προγραμματιστής
Junior developer	Νεότερος προγραμματιστής
Chart	Διάγραμμα
Life Cycle Costing	Ανάλυση Κόστους Κύκλου Ζωής
System Development Life Cycle	Κύκλος Ζωής Ανάπτυξης Λογισμικού
Waterfall Model	Μοντέλο Καταρράκτη
Incremental Model	Επαυξητικό Μοντέλο
Iterative Model	Επαναληπτικό Μοντέλο
Verification	Επαλήθευση
Validation	Εγκυρότητα
Development	Υλοποίηση
Developer	Προγραμματιστής
Coding	Κωδικοποίηση
Design	Σχεδιασμός
Testing	Έλεγχος
Deployment	Εγκατάσταση λογισμικού
User Acceptance Test	Έλεγχος Αποδοχής Χρήστη
System Integration Test	Έλεγχος Ενσωμάτωσης Συστήματος
System testing	Έλεγχος Συστήματος
Functional specifications	Προδιαγραφές λειτουργικότητας
High Level Design	Σχεδιασμός Υψηλού Επιπέδου
Detailed design	Λεπτομερής σχεδιασμός
Unit testing	Δοκιμή μονάδας
Test cases	Περιπτώσεις ελέγχου
Prototyping model	Μοντέλο Πρωτοτύπου
Prototype	Πρωτότυπο
Meeting	Συνάντηση
Effort	Προσπάθεια
Project Manager	Διαχειριστής έργου
Sprint	Επανάληψη 30 ημερών
Product Owner	Ιδιοκτήτης προϊόντος
Team	Ομάδα ανάπτυξης
Product Backlog	Κατάλογος εκκρεμοτήτων προϊόντος
Features	Χαρακτηριστικά προϊόντος
Releases	Εκδόσεις κυκλοφορίας λογισμικού
Release backlog	Κατάλογος εκκρεμοτήτων κυκλοφορίας
Sprint backlog	Κατάλογος εκκρεμοτήτων επανάληψης

<b>Ξενόγλωσσος όρος</b>	<b>Ελληνικός όρος</b>
Daily scrum	Ημερήσια σύσκεψη
Stand-up meetings	Σύσκεψη με όρθια τα άτομα
Sprint review	Σύσκεψη ανασκόπησης
Sprint retrospective	Σύσκεψη αναθεώρησης
Risk Management	Διαχείριση κινδύνου
Stakeholders	Ενδιαφερόμενοι
Business Plan	Επιχειρηματικό πλάνο
Assumptions	Παραδοχές
Agile software model	Ευκίνητο μοντέλο λογισμικού
Traditional software model	Παραδοσιακό μοντέλο λογισμικού
Requirements	Προδιαγραφές
Gathering	Συλλογή
Resources	Πόροι
Reconsideration meeting	Συνάντηση αναθεώρησης
Review meeting	Συνάντηση ανασκόπησης
Training	Εκπαίδευση
Determine objectives	Καθορισμός στόχων
Alternatives	Εναλλακτικές
Constraints	Περιορισμοί
Evaluate alternatives	Αξιολόγηση εναλλακτικών
Identify	Εξακρίβωση
Resolve risks	Επίλυση κινδύνων
Verify next level product	Βεβαίωση προϊόντος επόμενου σταδίου
Plan next phases	Σχεδιασμός επόμενων φάσεων
Mean	Μέσος όρος
Median	Διάμεσος
Mode	Επικρατούσα τιμή
Standard Deviation	Τυπική απόκλιση
Variance	Διακύμανση
Minimum	Ελάχιστο
Maximum	Μέγιστο
Range Width	Εύρος πλάτους
Certainty	Βεβαιότητα
Assumption	Υπόθεση
Fault probability	Πιθανότητα σφάλματος



## ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΑ

ICT	Information and Communications Technology
SDLC	Software Development Life Cycle
UAT	User Acceptance Test
SIT	System Integration Test
HLD	High Level Design
XP	Extreme Programming
RM	Risk Management
PM	Project Manager
LOC	Lines of Code
RAD	Rapid Application Development
RUP	Rational Unified Process
κλπ	Και λοιπά
ΕΚΠΑ	Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

## ΑΝΑΦΟΡΕΣ

- [1] David G. Woodward, "Life cycle costing--theory, information acquisition and application", International Journal of Project Management, Vol. 15, No. 6, pp. 335-344, 1997.
- [2] Mohamed Sami Abd El-Satar "Software Development Life Cycle Models and Methodologies", 2012; <https://melsatar.wordpress.com/2012/03/15/software-development-life-cycle-models-and-methodologies/> [Προσπελάστηκε 04/12/15]
- [3] Harish Rohil and Syan Manisha, "Ananalysis of Agile and Traditional Approach for Software development", International Journal of Latest Trends in Engineering and Technology ,Vol. 1(4),1-3, 2012.
- [4] Highsmith, J., "Agile Software Development Ecosystems", Addison Wesley, 2002.
- [5] Agile manifesto; <http://www.agilemanifesto.org/principles.html> [Προσπελάστηκε 02/11/15]
- [6] Investopedia, "Risk Management"; <http://www.investopedia.com/terms/r/riskmanagement.asp> [Προσπελάστηκε 10/01/16]
- [7] Steve Easterbrook, "Software Lifecycles", University of Toronto Department of Computer Science, 2001; <http://www.cs.toronto.edu/~sme/CSC444F/slides/L04-Lifecycles.pdf> [Προσπελάστηκε 31/01/16]
- [8] Sonali Mathur, Shaily Malik, "Advances in the V-Model", International Journal of Computer Applications, 2010; <http://www.ijcaonline.org/journal/number12/pxc387425.pdf> [Προσπελάστηκε 5/12/15]
- [9] Boehm, B. "A Spiral Model of Software Development and Enhancement", Computer (May 1988), pp.61-72.
- [10] Scowet Technologies, "Spiral Model Development", 2009; <http://www.scowet.com/process.html> [Προσπελάστηκε 5/12/15]
- [11] Barry W. Boehm, "A Spiral Model of Software Development and Enhancement", 1988; <http://csse.usc.edu/TECHRPTS/1988/usccse88-500/usccse88-500.pdf> [Προσπελάστηκε 5/12/15]
- [12] Kent Beck, "Test-Driven Development by Example.", Boston, MA: Addison Wesley, 2002.

- [13] Kent Beck, "Extreme Programming Explained — Second Edition", 2004;  
<http://ptgmedia.pearsoncmg.com/images/9780321278654/samplepages/9780321278654.pdf> [Προσπελάσθηκε 5/12/15]
- [14] Cockburn A. and Williams L., "The Costs and Benefits of Pair Programming, in Extreme Programming Examined", Addison Wesley, 2001.
- [15] Jeff Sutherland, "The Scrum Papers: Nuts, Bolts and Origins of Agile Framework" Jan 2011; <http://www.scruminc.com/scrumpapers.pdf> [Προσπελάσθηκε 5/12/15]
- [16] Schwaber, K., "Agile Project Management with Scrum", Microsoft Press, 2004.
- [17] "Οδηγός Απόκτησης Γνώσεων σχετικά με τη Διαχείριση Έργων", του Ινστιτούτου Διαχείρισης Έργων (Project Management Institute)
- [18] Wikipedia, "Διαχείριση Κινδύνου", Wikipedia: The free encyclopedia, Aug. 2014; [https://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%B1%CF%87%CE%B5%CE%AF%CF%81%CE%B9%CF%83%CE%B7\\_%CE%9A%CE%B9%CE%BD%CE%B4%CF%8D%CE%BD%CE%BF%CF%85](https://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%B1%CF%87%CE%B5%CE%AF%CF%81%CE%B9%CF%83%CE%B7_%CE%9A%CE%B9%CE%BD%CE%B4%CF%8D%CE%BD%CE%BF%CF%85) [Προσπελάσθηκε 9/2/15]
- [19] Tharwon Arnuphaptrairong, "Top Ten Lists of Software Project Risks: Evidence from the Literature Syrvey, March 2011; [http://www.iaeng.org/publication/IMECS2011/IMECS2011\\_pp732-737.pdf](http://www.iaeng.org/publication/IMECS2011/IMECS2011_pp732-737.pdf) [Προσπελάσθηκε 19/2/2015]
- [20] Wikipedia, "Monte Carlo (μέθοδος)", Wikipedia: The free encyclopedia, Aug. 2015; [https://el.wikipedia.org/wiki/Monte\\_Carlo\\_%28%CE%BC%CE%AD%CE%B8%CE%BF%CE%B4%CE%BF%CF%82%29](https://el.wikipedia.org/wiki/Monte_Carlo_%28%CE%BC%CE%AD%CE%B8%CE%BF%CE%B4%CE%BF%CF%82%29) [Προσπελάσθηκε 14/2/2015]
- [21] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S., "Global Sensitivity Analysis: The Primer. John Wiley & Sons", 2008
- [22] Pannell, D. J., "Sensitivity Analysis of Normative Economic Models: Theoretical Framework and Practical Strategies", 1997
- [23] Bahremand, A., De Smedt F, "Distributed Hydrological Modeling and Sensitivity Analysis in Torysa Watershed, Slovakia", 2008
- [24] Hakan Erdogmus, "Comparative evaluation of software development strategies based on Net Prive Value", March 1999