

ΕΠΛ362: Τεχνολογία Λογισμικού II

(μετάφραση στα ελληνικά των διαφανειών του βιβλίου Software Engineering, 9/E, Ian Sommerville, 2011)



Ενότητα 1 (κεφάλαια 3 και 23.4) – Ευέλικτη Ανάπτυξη Λογισμικού

Οι διαφάνειες αυτές έχουν συμπληρωματικό και επεξηγηματικό χαρακτήρα και σε καμία περίπτωση δεν υποκαθιστούν το βιβλίο

Γιάννης Α. Παπαδόπουλος
Τμήμα Πληροφορικής
Πανεπιστήμιο Κύπρου

1

Περιεχόμενα



- ✦ Ευέλικτοι (agile) μέθοδοι.
- ✦ Προσχεδιασμένη (plan-driven) και ευέλικτη ανάπτυξη.
- ✦ Ακραίος (extreme) προγραμματισμός.
- ✦ Ευέλικτη διαχείριση έργου.
- ✦ Χρήση ευέλικτων μεθόδων σε έργα μεγάλης κλίμακας.

Ενότητα 1 (Κεφάλαια 3 και 23.4) – Ευέλικτη Ανάπτυξη Λογισμικού

2

Γρήγορη ανάπτυξη λογισμικού



- ✦ Στις μέρες μας συχνά η πιο σημαντική απαίτηση για ένα λογισμικό σύστημα είναι η γρήγορη ανάπτυξη και παράδοσή του.
 - Οι επιχειρήσεις λειτουργούν σε ένα ταχύτατα μεταβαλλόμενο περιβάλλον και είναι πρακτικά αδύνατο να παραχθεί ένα πλήρες και συνεπές σύνολο απαιτήσεων για το εκάστοτε σύστημα.
 - Το λογισμικό πρέπει να εξελίσσεται γρήγορα για να παρακολουθεί τις συχνές αλλαγές στις ανάγκες της επιχείρησης.
- ✦ Γρήγορη ανάπτυξη λογισμικού.
 - Ο καθορισμός των προδιαγραφών και ο σχεδιασμός και ανάπτυξη του λογισμικού λαμβάνουν χώρα ταυτόχρονα.
 - Το σύστημα αναπτύσσεται σαν μία σειρά από επαυξήσεις με τους τελικούς χρήστες να εμπλέκονται στην αξιολόγηση της κάθε επαύξησης.
 - Οι διασυνδέσεις χρήστη της εφαρμογής συχνά αναπτύσσονται με τη χρήση ενός αλληλεπιδραστικού περιβάλλοντος.

Ενότητα 1 (Κεφάλαια 3 και 23.4) – Ευέλικτη Ανάπτυξη Λογισμικού

3

Ευέλικτοι μέθοδοι



✦ Η δυσανεμία την οποία προκαλούσαν οι πρόσθετες καθυστερήσεις των μεθόδων ανάπτυξης λογισμικού των δεκαετιών του '80 και '90 οδήγησε στη δημιουργία των ευέλικτων μεθόδων. Αυτές:

- Επικεντρώνονται στον κώδικα και όχι στο σχεδιασμό.
- Βασίζονται σε μία επαναληπτική προσέγγιση για την ανάπτυξη του λογισμικού.
- Έχουν στόχο τη γρήγορη παράδοση λειτουργήσιμου λογισμικού και την ταχεία εξέλιξη του για την ικανοποίηση των μεταβαλλόμενων απαιτήσεων.

✦ Ο σκοπός των ευέλικτων μεθόδων είναι να ελαττώσουν το κόστος στη διαδικασία ανάπτυξης λογισμικού (π.χ. με τον περιορισμό σε τεκμηρίωση) και να μπορούν να ανταποκριθούν γρήγορα σε αλλαγές των απαιτήσεων χωρίς πολύ επιπλέον δουλειά.

Το μανιφέστο της ευέλικτης ανάπτυξης



✦ Ανακαλύπτουμε καλύτερους τρόπους για την ανάπτυξη λογισμικού καθώς δημιουργούμε το λογισμικό αλλά και βοηθώντας άλλους να κάνουν το ίδιο. Από αυτή τη διαδικασία, μάθαμε να εκτιμούμε περισσότερο:

- Τους ανθρώπους και τις συνεργασίες μεταξύ τους από διεργασίες και εργαλεία.
- Λειτουργήσιμο λογισμικό από περιεκτική και λεπτομερή τεκμηρίωση.
- Καλή συνεργασία με τους πελάτες από διαπραγμάτευση συμβολαίων.
- Δυνατότητα ανταπόκρισης σε αλλαγές από το να ακολουθείται ένα προμελετημένο σχέδιο.

✦ Παρόλο που αναγνωρίζουμε την αξία των εννοιών στα δεξιά, εκτιμούμε περισσότερο αυτές στα αριστερά.

Αρχές των ευέλικτων μεθόδων



Αρχή	Περιγραφή
Συμμετοχή του πελάτη.	Οι πελάτες πρέπει να συμμετέχουν στενά σε όλη τη διαδικασία της ανάπτυξης. Ο ρόλος τους είναι να παρέχουν νέες απαιτήσεις συστήματος και να καθορίζουν την προτεραιότητα τους, καθώς και να αξιολογούν τις επαναλήψεις του συστήματος.
Βαθμιαία παράδοση.	Το λογισμικό αναπτύσσεται σε επουξήσεις και ο πελάτης καθορίζει τις απαιτήσεις που θα συμπεριληφθούν σε κάθε επαύξηση.
Ανθρώπινο, όχι διαδικασίες.	Πάντοτε πρέπει να προσδιορίζονται και να αξιοποιούνται οι ικανότητες της ομάδας ανάπτυξης. Τα μέλη της ομάδας πρέπει να έχουν το ελεύθερο να αναπτύξουν τους δικούς τους τρόπους εργασίας, χωρίς προδιαγεγραμμένες διαδικασίες.
Πρόβλεψη για μεταβολές.	Να αναμένεται ότι οι απαιτήσεις του συστήματος θα αλλάξουν και για αυτό το λόγο το σύστημα πρέπει να έχει σχεδιασθεί με τρόπο που να μπορεί να αντιμετωπίσει εύκολα αυτές τις αλλαγές.
Διατήρηση απλότητας.	Επικέντρωση στην απλότητα τόσο όσον αφορά το λογισμικό που αναπτύσσεται όσο και στη διαδικασία ανάπτυξης που ακολουθείται. Όταν είναι δυνατόν, πρέπει να γίνεται συνειδητή προσπάθεια να εξοικονομείται η πολυπλοκότητα του συστήματος.

Εφαρμογή των ευέλικτων μεθόδων



- ✦ Στην ανάπτυξη προϊόντων όπου μία εταιρία λογισμικού αναπτύσσει για πώληση ένα προϊόν μικρού ή μεσαίου μεγέθους.
- ✦ Στην ανάπτυξη κατά παραγγελία ενός συστήματος εσωτερικά σε ένα οργανισμό, όπου υπάρχει σαφής δέσμευση από τον πελάτη να εμπλακεί στη διαδικασία ανάπτυξης του συστήματος αυτού και δεν υπάρχουν πολλοί εξωτερικοί κανόνες και κανονισμοί που επηρεάζουν το λογισμικό.
- ✦ Λόγω της έμφασης που δίνουν οι ευέλικτες μέθοδοι σε μικρές και ενιαίες ομάδες, είναι δύσκολη η εφαρμογή τους στην ανάπτυξη μεγάλων συστημάτων.

Ενότητα 1 (Κεφάλαια 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού

7

Προβλήματα των ευέλικτων μεθόδων



- ✦ Είναι δύσκολη η διατήρηση του ενδιαφέροντος των πελατών που εμπλέκονται στη διαδικασία.
- ✦ Το ταμπεραμένο κάποιων μελών της ομάδας ανάπτυξης μπορεί να μην ταιριάζει στην έντονη συμμετοχή που συνηθίζεται στις ευέλικτες μεθόδους.
- ✦ Η απόδοση προτεραιοτήτων σε μεταβολές μπορεί να αποδειχθεί εξαιρετικά δύσκολη υπόθεση σε συστήματα όπου υπάρχουν πολλοί ενδιαφερόμενοι.
- ✦ Η διατήρηση της απλότητας απαιτεί επιπλέον εργασία.
- ✦ Οι συμβάσεις μπορεί να αποτελέσουν πρόβλημα (κάτι που ισχύει και σε άλλες προσεγγίσεις βαθμιαίας ανάπτυξης).

Ενότητα 1 (Κεφάλαια 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού

8

Ευέλικτες μέθοδοι και συντήρηση λογισμικού



- ✦ Οι περισσότεροι από τους οργανισμούς ξοδεύουν περισσότερο χρόνο για τη συντήρηση υπάρχοντος λογισμικού, παρά για τη δημιουργία καινούργιου. Επομένως, αν οι ευέλικτες μέθοδοι είναι να επιτύχουν, θα πρέπει να υποστηρίζουν συντήρηση όπως και ανάπτυξη.
- ✦ Εδώ υπάρχουν δύο θέματα-κλειδιά:
 - Είναι τα συστήματα που δημιουργούνται με χρήση ευέλικτων μεθόδων εύκολα συντηρήσιμα, δεδομένου ότι έμφαση δίνεται στην ελαχιστοποίηση της τεκμηρίωσης κατά την ανάπτυξη τους;
 - Μπορούν οι ευέλικτοι μέθοδοι να χρησιμοποιηθούν αποτελεσματικά για την εξέλιξη ενός συστήματος που καθίσταται αναγκαία αν αλλάξουν οι απαιτήσεις του πελάτη;
- ✦ Αν η αρχική ομάδα ανάπτυξης δεν μπορεί να διατηρηθεί, μπορεί να δημιουργηθούν προβλήματα.

Ενότητα 1 (Κεφάλαια 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού

9

Προσχεδιασμένη και ευέλικτη ανάπτυξη



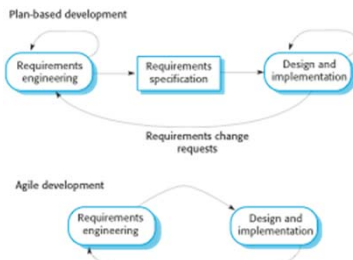
❖ Προσχεδιασμένη ανάπτυξη.

- Μία προσχεδιασμένη προσέγγιση στην τεχνολογία λογισμικού βασίζεται στην ύπαρξη ξεχωριστών φάσεων ανάπτυξης, όπου τα αποτελέσματα της κάθε φάσης είναι σχεδιασμένα εκ των προτέρων.
- Δεν είναι κατ' ανάγκη εφαρμογή του μοντέλου του καταρράκτη.
- Επαναλήψεις υπάρχουν μέσα στις διαδικασίες με χρήση επίσημων εγγράφων για τη μεταφορά πληροφοριών από τη μία φάση μιας διαδικασίας στην επόμενη.

❖ Ευέλικτη ανάπτυξη.

- Ο καθορισμός των προδιαγραφών, ο σχεδιασμός, υλοποίηση και έλεγχος του λογισμικού λαμβάνουν χώρα ταυτόχρονα και τα αποτελέσματα από αυτή τη διαδικασία ανάπτυξης αποφασίζονται μέσω μιας διαδικασίας διαπραγμάτευσης.

Προσχεδιασμένος και ευέλικτος καθορισμός



Τεχνικά και οργανωτικά θέματα και θέματα σχετικά με ανθρώπους (α)



❖ Τα περισσότερα έργα έχουν στοιχεία και από προσχεδιασμένες και από ευέλικτες διεργασίες. Η επίτευξη ισορροπίας εξαρτάται από τα ακόλουθα:

- Είναι σημαντικό να υπάρχει πρώτα λεπτομερής καθορισμός αναγκών και σχεδιασμός ενός συστήματος, πριν ξεκινήσει η υλοποίησή του; Αν ναι, τότε μάλλον πρέπει να προτιμηθεί η προσχεδιασμένη ανάπτυξη.
- Είναι ρεαλιστική η εφαρμογή μιας στρατηγικής όπου το σύστημα παραδίδεται στους πελάτες σαν σειρά επαυξήσεων και αυτοί δίνουν γρήγορα κάθε φορά σχόλια για βελτίωση; Αν ναι, τότε η χρήση ευέλικτων μεθόδων μπορεί να είναι προτιμητέα.
- Πόσο μεγάλο είναι το σύστημα που αναπτύσσεται; Οι ευέλικτες μέθοδοι είναι αποτελεσματικές κυρίως όταν η ομάδα ανάπτυξης είναι μικρή σε αριθμό ατόμων και αυτά βρίσκονται στον ίδιο χώρο και μπορούν να επικοινωνούν συχνά. Για μεγάλα συστήματα που χρειάζονται μεγάλες ομάδες ανάπτυξης, πιθανόν κατανεμημένες, η προσχεδιασμένη ανάπτυξη ίσως είναι καλύτερη.

Τεχνικά και οργανωτικά θέματα και θέματα σχετικά με ανθρώπους (β)



- Τι είδους σύστημα αναπτύσσεται;
 - Οι προσχεδιασμένες προσεγγίσεις μπορεί να είναι αναγκαίες για την ανάπτυξη συστημάτων τα οποία χρειάζονται πολύ ανάλυση πριν την υλοποίησή τους (π.χ. συστήματα πραγματικού χρόνου με πολύπλοκες απαιτήσεις χρονικών περιορισμών).
- Ποιος είναι ο εκτιμώμενος χρόνος ζωής του συστήματος;
 - Συστήματα με μακροζωία μπορεί να χρειάζονται πιο λεπτομερή τεκμηρίωση του σχεδιασμού τους, για να είναι κατορθωτή η μεταφορά των προθέσεων της αρχικής ομάδας ανάπτυξης του συστήματος σε μελλοντικές ομάδες συντήρησής του.
- Τι είδους τεχνολογίες ανάπτυξης λογισμικού είναι διαθέσιμες;
 - Οι ευέλικτες μέθοδοι εξαρτώνται από την ύπαρξη καλών εργαλείων που επιτρέπουν την παρακολούθηση της ανάπτυξης του συστήματος.
- Πως είναι οργανωμένη η ομάδα ανάπτυξης;
 - Αν η ομάδα είναι κατανεμημένη ή μέρος της ανάπτυξης γίνεται με εξωτερική ανάθεση, τότε ίσως πρέπει να δημιουργούνται έγγραφα σχεδίασης για την επικοινωνία μεταξύ των διαφόρων υποομάδων.

Ενότητα 1 (Κεφάλαια 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού

13

Τεχνικά και οργανωτικά θέματα και θέματα σχετικά με ανθρώπους (γ)



- Υπάρχουν οργανωτικά θέματα ή θέματα κουλτούρας που μπορεί να επηρεάσουν την ανάπτυξη του συστήματος;
 - Οι παραδοσιακοί οργανισμοί έχουν κουλτούρα προσχεδιασμένης ανάπτυξης.
- Πόσο καλοί είναι οι σχεδιαστές και προγραμματιστές στην ομάδα ανάπτυξης;
 - Έχει υποστηριχθεί ότι η χρήση ευέλικτων μεθόδων απαιτεί περισσότερη επιδεξιότητα από τη χρήση προσχεδιασμένων μεθόδων όπου οι προγραμματιστές απλά παράγουν κώδικα με βάση ένα λεπτομερές σχέδιο.
- Υπόκειται το σύστημα σε εξωτερικούς ελέγχους και ρυθμίσεις;
 - Αν το σύστημα πρέπει να εγκριθεί από εξωτερικό ρυθμιστή (π.χ. όπως ο οργανισμός FAA εγκρίνει λογισμικό που έχει να κάνει με τη λειτουργία ενός αεροσκάφους), τότε κατά πάσα πιθανότητα θα πρέπει να δημιουργηθεί αναλυτική τεκμηρίωση.

Ενότητα 1 (Κεφάλαια 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού

14

Ακραίος προγραμματισμός



- ✧ Ίσως η πιο γνωστή και πιο ευρέως διαδεδομένη ευέλικτη μέθοδος.
- ✧ Ο ακραίος προγραμματισμός υιοθετεί μία «ακραία» προσέγγιση της βαθμιαίας ανάπτυξης.
 - Μπορεί να δομηθούν πολλές νέες εκδόσεις κάθε μέρα.
 - Οι επαυξήσεις παραδίδονται στους πελάτες κάθε 2 βδομάδες.
 - Για κάθε δόμηση πρέπει να εκτελεστούν όλες οι δοκιμές, ενώ η δόμηση αυτή γίνεται αποδεκτή μόνο αν όλες οι δοκιμές εκτελεστούν σωστά.

Ενότητα 1 (Κεφάλαια 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού

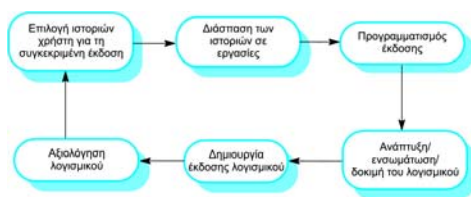
15

Ακραίος προγραμματισμός και αρχές των ευέλικτων μεθόδων



- ✦ Η βαθμιαία ανάπτυξη υποστηρίζεται μέσω μικρών και συχνών εκδόσεων του συστήματος.
- ✦ Η συμμετοχή του πελάτη συνεπάγεται την πλήρη απασχόλησή του στην ομάδα ανάπτυξης.
- ✦ Η αρχή «άνθρωποι, όχι διαδικασίες» υποστηρίζεται με τον προγραμματισμό σε ζευγάρια, τη συλλογική ιδιοκτησία του κώδικα του συστήματος και μία διαδικασία ανάπτυξης η οποία δεν απαιτεί εξοντωτική εργασία.
- ✦ Οι μεταβολές υποστηρίζονται μέσω τακτικών εκδόσεων του συστήματος.
- ✦ Η διατήρηση της απλότητας υποστηρίζεται μέσω της συνεχούς ανασύνθεσης του κώδικα.

Ο κύκλος εκδόσεων του ακραίου προγραμματισμού



Πρακτικές του ακραίου προγραμματισμού (α)



Αρχή ή πρακτική	Περιγραφή
Βαθμιαίος προγραμματισμός.	Οι απαιτήσεις καταγράφονται σε κάρτες ιστοριών και οι ιστορίες που θα συμπεριληφθούν σε μία έκδοση καθορίζονται από το διαθέσιμο χρόνο και τη σχετική τους προτεραιότητα. Δείτε τις εικόνες 3.5 και 3.6.
Εκδόσεις περιορισμένης ευρύτητας.	Πρώτα αναπτύσσεται το ελάχιστο σύνολο λειτουργικότητας που είναι χρήσιμο στην επιχείρηση. Οι εκδόσεις του συστήματος είναι συχνές και προσθέτουν λειτουργικότητα στην πρώτη έκδοση με βαθμιαίο τρόπο.
Απλός σχεδιασμός.	Πραγματοποιείται άσος σχεδιασμός είναι απαραίτητος για την ικανοποίηση των τρεχουσών απαιτήσεων και όχι περισσότερες.
Ανάπτυξη βάσει δοκιμών.	Χρησιμοποιείται ένα αυτοματοποιημένο πλαίσιο εργασίας για τη συγγραφή δοκιμών κάθε νέου τμήματος λειτουργικότητας, πριν υλοποιηθεί η ίδια η λειτουργικότητα.
Ανασύνθεση (refactoring).	Κάθε προγραμματιστής πρέπει συνεχώς να ανασυνθέτει τον κώδικα, αμέσως μόλις εντοπίζονται πιθανές βελτιώσεις. Με αυτόν τον τρόπο ο κώδικας διατηρείται απλός και συντηρήσιμος.

Πρακτικές του ακραίου προγραμματισμού (β)



Προγραμματισμός σε ζευγάρια.	Οι προγραμματιστές εργάζονται ανά δύο, ελέγχοντας ο ένας τη δουλειά του άλλου και παρέχοντας υποστήριξη ώστε να γίνεται πάντοτε καλή δουλειά.
Συλλογική ιδιοκτησία.	Τα ζευγάρια των προγραμματιστών εργάζονται σε όλους τους τομείς του συστήματος, έτσι ώστε να μην δημιουργούνται απομονωμένες νησίδες εξειδίκευσης και όλος ο κώδικας να ανήκει σε όλους. Ο καθένας μπορεί να αλλάξει οπδήποτε.
Συνεχής ενσωμάτωση.	Μόλις ολοκληρωθεί μία εργασία, ενσωματώνεται στο συνολικό σύστημα. Μετά από κάθε τέτοια ενσωμάτωση, πρέπει να εκτελεστούν με επιτυχία όλες οι δοκιμές μονάδας του συστήματος.
Μη εφ'εξουχιστί ρυθμοί εργασίας.	Οι πολλές επιπλέον ώρες εργασίας δε θεωρούνται αποδεκτή πρακτική, καθώς το τελικό αποτέλεσμα είναι να ελαττώνεται η ποιότητα του κώδικα και η παραγωγικότητα να κυμαίνεται σε μέτρια επίπεδα.
Πάντα διαθέσιμος πελάτης.	Η ομάδα του ακραίου προγραμματισμού πρέπει να έχει κάθε στιγμή στη διάθεσή της έναν αντιπρόσωπο του τελικού χρήστη του συστήματος (τον Πελάτη). Σε μία διαδικασία ακραίου προγραμματισμού, ο πελάτης αποτελεί μέλος της ομάδας ανάπτυξης και είναι υπεύθυνος για να προσφορέσει την ομάδα με απαιτήσεις συστήματος προς υλοποίηση.

Ενότητα 1 (Κεφάλαιο 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού 19

Σενάρια απαιτήσεων



- ✦ Στον ακραίο προγραμματισμό ο πελάτης ή τελικός χρήστης θεωρείται μέρος της ομάδας ανάπτυξης και είναι υπεύθυνος για να παίρνει αποφάσεις σε θέματα απαιτήσεων του συστήματος.
- ✦ Όλες οι απαιτήσεις εκφράζονται με τη μορφή σεναρίων ή ιστορίες χρήστη.
- ✦ Γράφονται σε κάρτες και μετά η ομάδα ανάπτυξης τις διασπά σε εργασίες υλοποίησης. Αυτές οι εργασίες αποτελούν τη βάση των χρονικών και οικονομικών εκτιμήσεων.
- ✦ Ο πελάτης επιλέγει τις ιστορίες που θα συμπεριληφθούν στην επόμενη έκδοση με βάση τις προτεραιότητες και τις εκτιμήσεις που αφορούν το χρονοδιάγραμμα.

Ενότητα 1 (Κεφάλαιο 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού 20

Μία ιστορία για χορηγία φαρμάκου



Prescribing medication

The record of the patient must be open for input. Click on the medication field and select either 'current medication', 'new medication' or 'formulary'.

If you select 'current medication', you will be asked to check the dose; if you wish to change the dose, enter the new dose then confirm the prescription.

If you choose 'new medication', the system assumes that you know which medication you wish to prescribe. Type the first few letters of the drug name. You will then see a list of possible drugs starting with these letters. Choose the required medication. You will then be asked to check that the medication you have selected is correct. Enter the dose then confirm the prescription.

If you choose 'formulary', you will be presented with a search box for the approved formulary. Search for the drug required then select it. You will then be asked to check that the medication you have selected is correct. Enter the dose then confirm the prescription.

In all cases, the system will check that the dose is within the approved range and will ask you to change it if it is outside the range of recommended doses.

After you have confirmed the prescription, it will be displayed for checking. Either click 'OK' or 'Change'. If you click 'OK', your prescription will be recorded on the audit database. If you click 'Change', you reenter the 'Prescribing medication' process.

Ενότητα 1 (Κεφάλαιο 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού 21

Παραδείγματα καρτών για χορηγία φαρμάκου



Task 1: Change dose of prescribed drug

Task 2: Formulary selection

Task 3: Dose checking

Dose checking is a safety precaution to check that the doctor has not prescribed a dangerously small or large dose.

Using the formulary id for the generic drug name, lookup the formulary and retrieve the recommended maximum and minimum dose.

Check the prescribed dose against the minimum and maximum. If outside the range, issue an error message saying that the dose is too high or too low. If within the range, enable the 'Confirm' button.

Ενότητα 1 (Κεφάλαιο 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού

22

Ακραίο προγραμματισμός και πραγματοποίηση τροποποιήσεων



- ✦ Μία γενική αρχή της τεχνολογίας λογισμικού είναι ότι ο σχεδιασμός πρέπει να γίνεται λαμβάνοντας υπόψη το ενδεχόμενο πραγματοποίησης αλλαγών. Είναι σκόπιμη η επένδυση χρόνου και προσπάθειας για την πρόβλεψη αλλαγών καθώς αυτό θα μειώσει τα έξοδα σε μεταγενέστερα στάδια του κύκλου ζωής του συστήματος.
- ✦ Όμως, σύμφωνα με τον ακραίο προγραμματισμό, αυτό δεν αξίζει τον κόπο αφού οι διάφορες αλλαγές δεν μπορούν να προβλεφθούν με αξιόπιστο τρόπο.
- ✦ Αντί αυτού, προτείνεται η συνεχής βελτίωση του κώδικα (ανασύνθεση) προκειμένου να διευκολυνθεί η διαδικασία αλλαγών όταν πρόκειται αυτές να υλοποιηθούν.

Ενότητα 1 (Κεφάλαιο 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού

23

Ανασύνθεση



- ✦ Η ομάδα προγραμματισμού κοιτάει συνεχώς για πιθανές βελτιώσεις του κώδικα και τις εφαρμόζει ακόμα και όταν δεν υπάρχει άμεση ανάγκη για αυτές.
- ✦ Αυτό βελτιώνει την κατανόηση του λογισμικού και κατ'επέκταση μειώνει την ανάγκη σε τεκμηρίωση.
- ✦ Είναι πιο εύκολο να γίνουν αλλαγές γιατί ο κώδικας είναι καλά δομημένος και ξεκάθαρος.
- ✦ Όμως κάποιες αλλαγές απαιτούν ανασύνθεση της αρχιτεκτονικής του συστήματος και αυτό είναι μία πιο ακριβή διαδικασία.

Ενότητα 1 (Κεφάλαιο 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού

24

Παραδείγματα ανασύνθεσης



- ✦ Αναδιοργάνωση μίας ιεραρχίας τάξης για απομάκρυνση πανομοιότυπου κώδικα.
- ✦ Νοικοκύρεμα και μετονομασία χαρακτηριστικών και μεθόδων για καλύτερη κατανόησή τους.
- ✦ Αντικατάσταση εμβόλιμου κώδικα με καλέσματα σε μεθόδους που υπάρχουν σε βιβλιοθήκες.

Εκτέλεση δοκιμών στον ακραίο προγραμματισμό



- ✦ Οι δοκιμές αποτελούν κεντρική δραστηριότητα στον ακραίο προγραμματισμό και επιτελούνται κάθε φορά που ο κώδικας υφίσταται κάποια αλλαγή.
- ✦ Τα χαρακτηριστικά των δοκιμών στον ακραίο προγραμματισμό:
 - Ανάπτυξη βάσει δοκιμών.
 - Βαθμιαία ανάπτυξη δοκιμών από σενάρια.
 - Συμμετοχή των χρηστών στην ανάπτυξη και επικύρωση των δοκιμών.
 - Κάθε φορά που δομείται μία νέα έκδοση, χρησιμοποιούνται αυτοματοποιημένα δοκιμαστικά προγράμματα για την εκτέλεση των δοκιμών όλων των συστατικών στοιχείων.

Ανάπτυξη βάσει δοκιμών



- ✦ Η δημιουργία δοκιμών πριν από τη δημιουργία κώδικα συμβάλλει στη διευκρίνιση των απαιτήσεων που πρόκειται να υλοποιηθούν.
- ✦ Οι δοκιμές γράφονται με τη μορφή προγραμμάτων και όχι δεδομένων ώστε να μπορούν να εκτελούνται αυτόματα. Κάθε δοκιμή περιλαμβάνει έναν έλεγχο για τη σωστή εκτέλεσή τους.
 - Συνήθως βασίζεται στην ύπαρξη ενός περιβάλλοντος δοκιμών όπως είναι το JUnit.
- ✦ Κάθε φορά που προστίθενται νέες λειτουργικές δυνατότητες, εκτελούνται αυτόματα όλες οι νέες, αλλά και οι προηγούμενες δοκιμές. Έτσι ελέγχεται το ενδεχόμενο εμφάνισης σφαλμάτων εξαιτίας των νέων δυνατοτήτων.

Συμμετοχή του πελάτη στις δοκιμές



- ✦ Ο σκοπός του πελάτη στη διαδικασία δοκιμών είναι να βοηθήσει στη δημιουργία δοκιμών αποδοχής για τις ιστορίες που θα υλοποιηθούν στην επόμενη έκδοση του συστήματος.
- ✦ Ο πελάτης που είναι μέλος της ομάδας ανάπτυξης παρέχει αυτές τις δοκιμές κατά τη διάρκεια ανάπτυξης του συστήματος. Με αυτόν τον τρόπο ο καινούργιος κώδικας εγκρίνεται και βεβαιώνεται ότι εξυπηρετεί τις ανάγκες του πελάτη.
- ✦ Όμως όσοι παίζουν το ρόλο του πελάτη με τον προαναφερθέντα τρόπο, συνήθως έχουν περιορισμένο διαθέσιμο χρόνο, δεν μπορούν να εμπλακούν αποκλειστικά σε αυτή τη δουλειά και μπορεί να αισθάνονται ότι αρκεί που έδωσαν τις απαιτήσεις και δεν χρειάζεται να αναμειχθούν και στις δοκιμές.

Περιγραφή τεστ δοκιμής για έλεγχο δόσης



Test 4: Dose checking

Input:

1. A number in mg representing a single dose of the drug.
2. A number representing the number of single doses per day.

Tests:

1. Test for inputs where the single dose is correct but the frequency is too high.
2. Test for inputs where the single dose is too high and too low.
3. Test for inputs where the single dose * frequency is too high and too low.
4. Test for inputs where single dose * frequency is in the permitted range.

Output:

OK or error message indicating that the dose is outside the safe range.

Αυτοματοποίηση δοκιμών



- ✦ Με αυτοματοποίηση δοκιμών εννοούμε τη δημιουργία των τεστ δοκιμών σαν εκτελέσιμο κώδικα και πριν την παραγωγή του κώδικα της (επόμενης έκδοσης της) εφαρμογής.
 - Ο κώδικας των τεστ δοκιμών πρέπει να εκτελείται ανεξάρτητα από το υπό υλοποίηση σύστημα, θα πρέπει να εξομοιώνει την εισαγωγή δεδομένων σε αυτό και να ελέγχει ότι το αποτέλεσμα επεξεργασίας των δεδομένων αυτών είναι το αναμενόμενο. Εδώ μπορεί να γίνει χρήση ειδικών περιβαλλόντων (π.χ. Junit) τα οποία επιτρέπουν την ταχεία δημιουργία και εκτέλεση αυτοματοποιημένων δοκιμών.
- ✦ Με την αυτοματοποίηση των δοκιμών, υπάρχει πάντα μία ομάδα από τεστ τα οποία μπορούν γρήγορα και εύκολα να εκτελεσθούν.
 - Με κάθε επαύξηση του συστήματος με νέα λειτουργικότητα, μπορούν να εκτελεστούν τα τεστ δοκιμών και να ανιχνεύσουν αμέσως τυχόν προβλήματα που δημιουργήσε ο νέος κώδικας.

Δυσκολίες στη δημιουργία τεστ δοκιμών στον ακραίο προγραμματισμό



- ❖ Οι προγραμματιστές συχνά προτιμούν να γράφουν κώδικα από το να τον ελέγχουν και μερικές φορές τα τεστ που δημιουργούν είναι ελλιπή και δεν καλύπτουν όλες τις πιθανές περιπτώσεις.
- ❖ Μερικά τεστ είναι δύσκολο να δημιουργηθούν σταδιακά. Π.χ. σε μία πολύπλοκη διεπαφή είναι συχνά δύσκολο να δημιουργηθούν δοκιμές μονάδας που να ελέγχουν τον τρόπο παρουσίασης των πληροφοριών και την αλληλεξάρτηση μεταξύ των διαφόρων οθονών.
- ❖ Είναι επίσης δύσκολο να εκτιμηθεί αν μία ομάδα δοκιμών είναι πλήρης. Αν και μπορεί να υπάρχουν πολλά τεστ, δεν είναι σίγουρο ότι έχουν καλυφτεί όλες οι πιθανότητες.

Προγραμματισμός σε ζευγάρια (α)



- ❖ Στον ακραίο προγραμματισμό, οι προγραμματιστές οργανώνονται σε ζευγάρια και κάθε ζευγάρι κάθεται μαζί για να δημιουργήσει κώδικα.
- ❖ Αυτό βοηθάει στο να ανήκει ο κώδικας σε όλους και στη διάχυση της τεχνογνωσίας σε όλη την ομάδα.
- ❖ Επιτρέπει επίσης την ανεπίσημη επιθεώρηση του κώδικα ο οποίος ελέγχεται από περισσότερα του ενός άτομα.
- ❖ Προτρέπει την ανασύνθεση του κώδικα διότι θα ωφελήσει όλη την ομάδα.
- ❖ Μετρήσεις έχουν δείξει ότι η παραγωγικότητα με χρήση ζευγαριών είναι παρόμοια με αυτή δύο ατόμων τα οποία εργάζονται ανεξάρτητα.

Προγραμματισμός σε ζευγάρια (β)



- ❖ Το κάθε ζευγάρι κάθεται μπροστά στον ίδιο Η/Υ για να εργασθεί.
- ❖ Τα ζευγάρια δημιουργούνται δυναμικά έτσι ώστε όλα τα μέλη μίας ομάδας θα βρεθούν κάποια στιγμή να αποτελούν ζευγάρια στη διάρκεια της ανάπτυξης ενός έργου.
- ❖ Ο διαμοιρασμός της γνώσης που λαμβάνει χώρα μέσα σε ένα ζευγάρι είναι σημαντικός γιατί μειώνει τα ρίσκα όταν κάποια μέλη της ομάδας φύγουν.
- ❖ Ο προγραμματισμός σε ζευγάρια δεν είναι κατ' ανάγκη μη αποδοτικός και υπάρχουν αποδείξεις ότι η αποδοτικότητα ενός ζευγαριού είναι υψηλότερη από αυτή δύο προγραμματιστών που δουλεύουν ανεξάρτητα.

Ευέλικτη διαχείριση έργου



- ❖ Η βασική ευθύνη του διαχειριστή έργου είναι να διαχειρίζεται την ανάπτυξη του έργου έτσι ώστε το λογισμικό να παραδοθεί στον προκαθορισμένο χρόνο και κόστος.
- ❖ Ο καθιερωμένος τρόπος που αυτό επιτυγχάνεται είναι με προσχεδίασμα. Οι διαχειριστές δημιουργούν ένα πλάνο εργασίας το οποίο δείχνει τι πρέπει να παραδοθεί, πότε πρέπει να παραδοθεί και ποιοι πρέπει να εργασθούν σε κάθε παραδοτέο του έργου.
- ❖ Η ευέλικτη διαχείριση έργου καθιστά αναγκαία την υιοθέτηση μιας διαφορετικής προσέγγισης, που να είναι κατάλληλη για ανάπτυξη με βάση προσαυξήσεις και ευέλικτες μεθόδους.
- ❖ Η απόφαση για τη λειτουργικότητα της κάθε επαύξησης και το τι αυτή θα περιλαμβάνει, δεν αποφασίζεται εκ των προτέρων αλλά δυναμικά με βάση τη τρέχουσα πρόοδο του έργου και των προτεραιοτήτων και αναγκών των πελατών οι οποίες αλλάζουν.

Ενότητα 1 (Κεφάλαια 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού

34

Διαχείριση έργου στον ακραίο προγραμματισμό



Ενότητα 1 (Κεφάλαια 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού

35

Διαχείριση έργου με βάση ιστορίες



- ❖ Ο καθορισμός των προδιαγραφών του συστήματος στον ακραίο προγραμματισμό γίνεται με βάση ιστορίες του τελικού χρήστη που αντανακλούν τα χαρακτηριστικά που πρέπει να έχει το σύστημα.
- ❖ Η ομάδα ανάπτυξης μελετάει και αναλύει τις ιστορίες και τις οργανώνει σε σειρά με βάση τον προβλεπόμενο χρόνο ανάπτυξής τους.
- ❖ Κατόπιν επιλέγονται και βελτιώνονται οι ιστορίες που θα υλοποιηθούν με βάση τα χαρακτηριστικά που πρέπει να υποστηρίξει η επόμενη έκδοση του συστήματος.
- ❖ Ο αριθμός των ιστοριών που επιλέγονται για υλοποίηση κάθε φορά θα πρέπει να αντανακλάει και το χρόνο μέχρι την παράδοση της επόμενης έκδοσης του συστήματος (συνήθως 2 με 3 βδομάδες).

Ενότητα 1 (Κεφάλαια 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού

36

Scrum



✦ Η μέθοδος Scrum είναι μία γενική ευέλικτη μέθοδος που δίνει έμφαση στην επαναληπτική ανάπτυξη και όχι ειδικά σε ευέλικτες πρακτικές.

✦ Υπάρχουν τρεις φάσεις στη μέθοδο Scrum:

- Στην αρχική φάση τίθενται οι γενικοί στόχοι του έργου και σχεδιάζεται η αρχιτεκτονική λογισμικού του συστήματος.
- Ακολουθεί μία σειρά από ταχείς κύκλους ανάπτυξης, όπου σε κάθε κύκλο αναπτύσσεται μία προσαύξηση του συστήματος.
- Στην τελευταία φάση ολοκληρώνονται οι ενέργειες του έργου με τη δημιουργία τεκμηρίωσης και την ανάλυση των εμπειριών που αποκτήθηκαν κατά την ανάπτυξη του έργου.

Η διαδικασία Scrum



Ο ταχύς κύκλος (α)



✦ Κάθε ταχύς κύκλος έχει σταθερή χρονική διάρκεια, συνήθως 2-4 εβδομάδες και αντιστοιχεί στην ανάπτυξη της επόμενης έκδοσης ενός συστήματος στον ακραίο προγραμματισμό.

✦ Το σημείο εκκίνησης για κάθε κύκλο είναι οι εργασίες που έχουν συσσωρευτεί για να γίνουν, από την προηγούμενη έκδοση του έργου.

✦ Η βάση επιλογής εμπλέκει όλα τα μέλη της ομάδας που συνεργάζονται με τον πελάτη για την επιλογή των χαρακτηριστικών και της λειτουργικότητας που θα αναπτυχθεί στον τρέχοντα κύκλο.

Ο ταχύς κύκλος (β)



- ✦ Όταν υπάρξει συμφωνία για αυτά, τότε η ομάδα οργανώνεται για την ανάπτυξη του λογισμικού. Σε αυτή τη φάση, η ομάδα ανάπτυξης δεν έχει επαφή με τον πελάτη και τον οργανισμό και η μόνη επαφή της με το περιβάλλον είναι μέσω ενός συντονιστή (Scrum master).
- ✦ Ο συντονιστής προστατεύει την ομάδα από εξωτερικές ενοχλήσεις και αποσπάσεις της προσοχής.
- ✦ Μετά την ολοκλήρωση του τρέχοντος κύκλου, επιθεωρείται η εργασία που έχει γίνει και παρουσιάζεται στους ενδιαφερόμενους. Κατόπιν, ξεκινάει ο επόμενος κύκλος.

Ομαδική δουλειά στη μέθοδο Scrum



- ✦ Σκοπός του συντονιστή είναι να οργανώνει συναντήσεις, ελέγχει τη δουλειά που πρέπει να γίνει, καταγράφει αποφάσεις, εκτιμά την πρόοδο που γίνεται και αποτελεί το σημείο επαφής της ομάδας με τους πελάτες και τη διεύθυνση.
- ✦ Όλα τα μέλη της ομάδας ανάπτυξης έχουν σύντομες καθημερινές συναντήσεις όπου μοιράζονται πληροφορίες, περιγράφουν την πρόοδο που έκαναν από την προηγούμενη συνάντηση, αναφέρουν τυχόν προβλήματα και το τι θα κάνουν την επόμενη μέρα.
 - Επομένως το κάθε μέλος της ομάδας ξέρει τι γίνεται και αν υπάρχουν προβλήματα είναι δυνατός ο επανασχεδιασμός κάποιων δραστηριοτήτων.

Πλεονεκτήματα της μεθόδου Scrum



- ✦ Το υπό παραγωγή προϊόν διασπάται σε ένα σύνολο από διαχειρίσιμα και κατανοητά κομμάτια.
- ✦ Οι ασταθείς απαιτήσεις δεν εμποδίζουν την πρόοδο του έργου.
- ✦ Όλα τα μέλη της ομάδας είναι ενήμερα για ότι συμβαίνει και κατ'επέκταση βελτιώνεται η επικοινωνία μεταξύ των μελών της ομάδας.
- ✦ Οι πελάτες παραλαμβάνουν έγκαιρα την επόμενη προσαύξηση και αποκτούν γνώση για το πώς το προϊόν λειτουργεί.
- ✦ Δημιουργείται κλίμα εμπιστοσύνης μεταξύ ομάδας και πελατών όπου όλοι αναμένουν ότι το έργο θα ολοκληρωθεί με επιτυχία.

Χρήση ευέλικτων μεθόδων σε έργα μεγάλης κλίμακας



- ✦ Οι ευέλικτες μέθοδοι έχουν αποδειχθεί επιτυχείς για έργα μικρής ή μεσαίας κλίμακας τα οποία μπορούν να υλοποιηθούν από μικρές ομάδες όπου όλα τα μέλη της ομάδας βρίσκονται στον ίδιο γεωγραφικό χώρο.
- ✦ Υπάρχει η άποψη ότι η επιτυχία αυτή προέρχεται από το γεγονός ότι η οργάνωση της ομάδας με τον προαναφερθέντα τρόπο επιτρέπει καλύτερη επικοινωνία μεταξύ των μελών της.
- ✦ Για μεγάλα έργα, οι ευέλικτες μέθοδοι πρέπει να αντιμετωπίσουν την περίπτωση να υπάρχουν πολλαπλές ομάδες ανάπτυξης καταμεμημένες σε διαφορετικούς γεωγραφικούς χώρους.

Ενότητα 1 (Κεφάλαιο 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού

43

Ανάπτυξη έργων μεγάλης κλίμακας (α)



- ✦ Τα μεγάλα έργα είναι συνήθως ένα σύνολο από επί μέρους έργα, τα οποία επικοινωνούν μεταξύ τους και αναπτύσσονται από διαφορετικές ομάδες, οι οποίες συχνά έχουν τη βάση τους σε διαφορετικούς γεωγραφικούς χώρους και μερικές φορές και με μεγάλη διαφορά ώρας μεταξύ τους.
- ✦ Λόγω της αλληλεπίδρασης ενός επί μέρους συστήματος με άλλα, δεν είναι εύκολο να υποστεί το σύστημα αυτό αλλαγές γιατί μπορεί να επηρεάσει αρνητικά την επικοινωνία του με τα υπόλοιπα. Κατ' επέκταση, το σύστημα δεν είναι ευέλικτο και δεν είναι εύκολη η αυξητική ανάπτυξή του.
- ✦ Όταν ένας αριθμός επί μέρους συστημάτων ενσωματώνονται σε ένα σύνθετο σύστημα, ένα μεγάλο μέρος του φόρτου εργασίας επικεντρώνεται στη διαμόρφωση του συστήματος παρά στην ανάπτυξη του αρχικού κώδικα.

Ενότητα 1 (Κεφάλαιο 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού

44

Ανάπτυξη έργων μεγάλης κλίμακας (β)



- ✦ Συχνά τα μεγάλα συστήματα και οι διεργασίες ανάπτυξής τους περιορίζονται από εξωτερικούς κανόνες και κανονισμούς, που με τη σειρά του περιορίζει τα περιθώρια ανάπτυξής τους.
- ✦ Τα μεγάλα συστήματα χρειάζονται πολύ χρόνο για συμβάσεις προμηθειών και υλοποίηση. Είναι δύσκολο να κρατηθεί ομοιογένεια στην ομάδα ανάπτυξης γιατί σε τέτοιο μεγάλο χρονικό διάστημα κάποια μέλη της ομάδας θα αποχωρήσουν και αντικατασταθούν με άλλα.
- ✦ Τα μεγάλα συστήματα συχνά έχουν διαφορετικά είδη ενδιαφερομένων και είναι πρακτικά αδύνατο όλοι αυτοί να εμπλακούν στη φάση ανάπτυξης του συστήματος.

Ενότητα 1 (Κεφάλαιο 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού

45

Επέκταση των ευέλικτων μεθόδων



- ✦ Η επέκταση των ευέλικτων μεθόδων γίνεται προς δυο κατευθύνσεις:
 - Τη χρήση των ευέλικτων μεθόδων για την ανάπτυξη μεγάλων λογισμικών συστημάτων τα οποία δεν μπορούν να υλοποιηθούν από μία μικρή ομάδα.
 - Το πώς οι ευέλικτες μέθοδοι μπορούν να εφαρμοσθούν μέσα σε ένα μεγάλο οργανισμό με πολλά χρόνια εμπειρίας στην ανάπτυξη λογισμικού.
- ✦ Όταν αυξάνουμε την κλίμακα χρήσης των ευέλικτων μεθόδων θα πρέπει να διατηρούμε τις βασικές αρχές τους.
 - Ευέλικτος σχεδιασμός, συχνή παραγωγή νέων εκδόσεων, συνεχής ενσωμάτωση, ανάπτυξη με βάση δοκιμές και καλή επικοινωνία μέσα στις ομάδες ανάπτυξης.

Χρήση ευέλικτων μεθόδων για την ανάπτυξη μεγάλων λογισμικών συστημάτων



- ✦ Για μεγάλα συστήματα δεν είναι δυνατόν να επικεντρωθούν οι εργασίες μόνο στην ανάπτυξη του κώδικα και πρέπει να γίνει αρκετή δουλειά σε σχεδιασμό και τεκμηρίωση του συστήματος.
- ✦ Πρέπει να αναπτυχθούν καλοί τρόποι επικοινωνίας μεταξύ των ομάδων που δυνατόν να περιλαμβάνει συχνές ιδεατές συναντήσεις μέσω τηλεφώνου ή βίντεο.
- ✦ Λόγω του μεγέθους τους είναι αδύνατη η συνεχής ενσωμάτωση των επί μέρους αναπτυσσόμενων μερών σε ένα ολοκληρωμένο σύστημα με τη μέθοδο της επαύξησης. Όμως είναι σημαντικό να υπάρχουν συχνές νέες εκδόσεις του συστήματος.

Εφαρμογή ευέλικτων μεθόδων σε μεγάλους οργανισμούς



- ✦ Οι διαχειριστές έργων που δεν έχουν εμπειρία στη χρήση ευέλικτων μεθόδων μπορεί να είναι διστακτικοί στο να πάρουν το ρίσκο αυτής της νέας προσέγγισης.
- ✦ Οι μεγάλοι οργανισμοί συχνά έχουν συγκεκριμένες διαδικασίες προτύπων και εξασφάλισης ποιότητας, οι οποίες αναμένονται να εφαρμόζονται από όλα τα έργα. Αυτές οι διαδικασίες συχνά είναι γραφειοκρατικές και κατ' επέκταση δεν συνάδουν με το πνεύμα των ευέλικτων μεθόδων.
- ✦ Οι ευέλικτες μέθοδοι φαίνεται να λειτουργούν καλύτερα σε ομάδες όπου τα μέλη τους έχουν υψηλά προσόντα. Όμως σε μεγάλους οργανισμούς, ενδέχεται να υπάρχουν άτομα με διαφορετικά προσόντα και ικανότητες.
- ✦ Μπορεί να υπάρχει αρνητική κουλτούρα χρήσης των ευέλικτων μεθόδων σε οργανισμούς που με τα χρόνια έχουν αναπτύξει πιο συμβατικές μεθόδους.

Κύρια σημεία (α)



- ❖ Οι ευέλικτοι μέθοδοι είναι αυξητικής μορφής μέθοδοι ανάπτυξης που βασίζονται σε γρήγορη παραγωγή, συχνές παραδόσεις νέων εκδόσεων του λογισμικού, μείωση του κόστους παραγωγής, δημιουργία υψηλής ποιότητας κώδικα και συνεχή εμπλοκή του πελάτη στη διαδικασία παραγωγής.
- ❖ Η απόφαση για το αν θα χρησιμοποιηθεί μία ευέλικτη ή μία προσχεδιασμένη μέθοδος θα πρέπει να εξαρτάται από το είδος του λογισμικού που θα αναπτυχθεί, τις ικανότητες της ομάδας ανάπτυξης και την κουλτούρα του οργανισμού στον οποίο θα αναπτυχθεί το σύστημα.
- ❖ Ο ακραίος προγραμματισμός είναι μία γνωστή ευέλικτη μέθοδος που συνδυάζει μία ομάδα καλών προγραμματιστικών πρακτικών όπως συχνή παραγωγή νέων εκδόσεων, συνεχής βελτίωση του κώδικα και συμμετοχή του πελάτη στην ομάδα ανάπτυξης.

Ενότητα 1 (Κεφάλαιο 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού

49

Κύρια σημεία (β)



- ❖ Ένα ιδιαίτερα δυνατό χαρακτηριστικό του ακραίου προγραμματισμού είναι η δημιουργία αυτοματοποιημένων τεστ δοκιμών για χαρακτηριστικά του συστήματος, πριν αυτά τα τελευταία υλοποιηθούν. Τα τεστ αυτά πρέπει να εκτελεστούν με επιτυχία όταν επαυξηθεί το σύστημα με τα νέα χαρακτηριστικά.
- ❖ Η μέθοδος Scrum είναι μία ευέλικτη μέθοδος που παρέχει ένα πλαίσιο διαχείρισης ενός έργου. Βασίζεται στην ύπαρξη ενός αριθμού ταχέων κύκλων ανάπτυξης, οι οποίοι είναι σταθεροί χρονικοί περίοδοι μέσα στους οποίους δημιουργούνται οι επαυξήσεις του συστήματος.
- ❖ Η εφαρμογή των ευέλικτων μεθόδων σε μεγάλης κλίμακας συστήματα είναι δύσκολη για ένα αριθμό από λόγους που έχουν να κάνουν με τη φύση των έργων προς υλοποίηση και των οργανισμών που εμπλέκονται.

Ενότητα 1 (Κεφάλαιο 3 και 23.4) — Ευέλικτη Ανάπτυξη Λογισμικού

50
