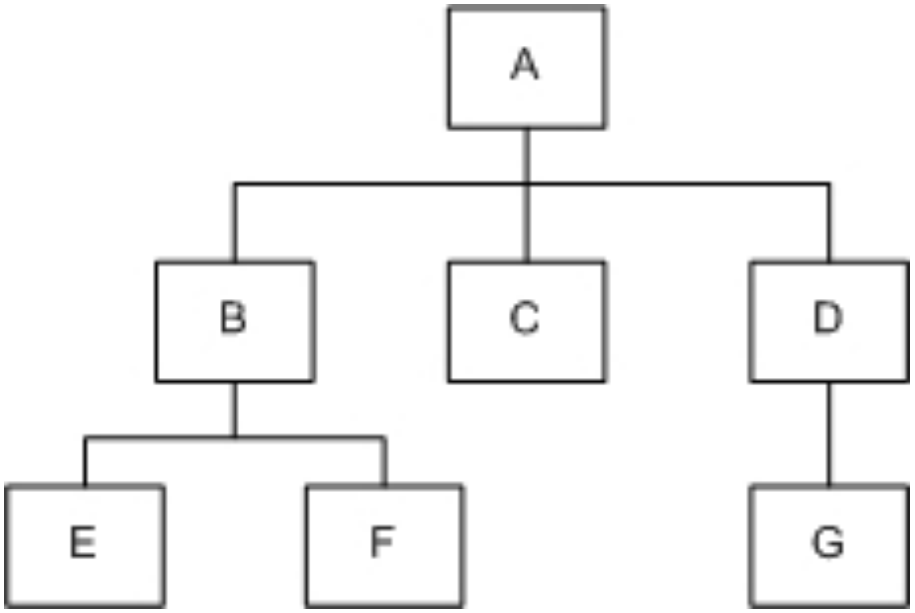
Έλεγχος Συνένωσης και Διασφάλιση Ποιότητας

περιεχόμενα παρουσίασης

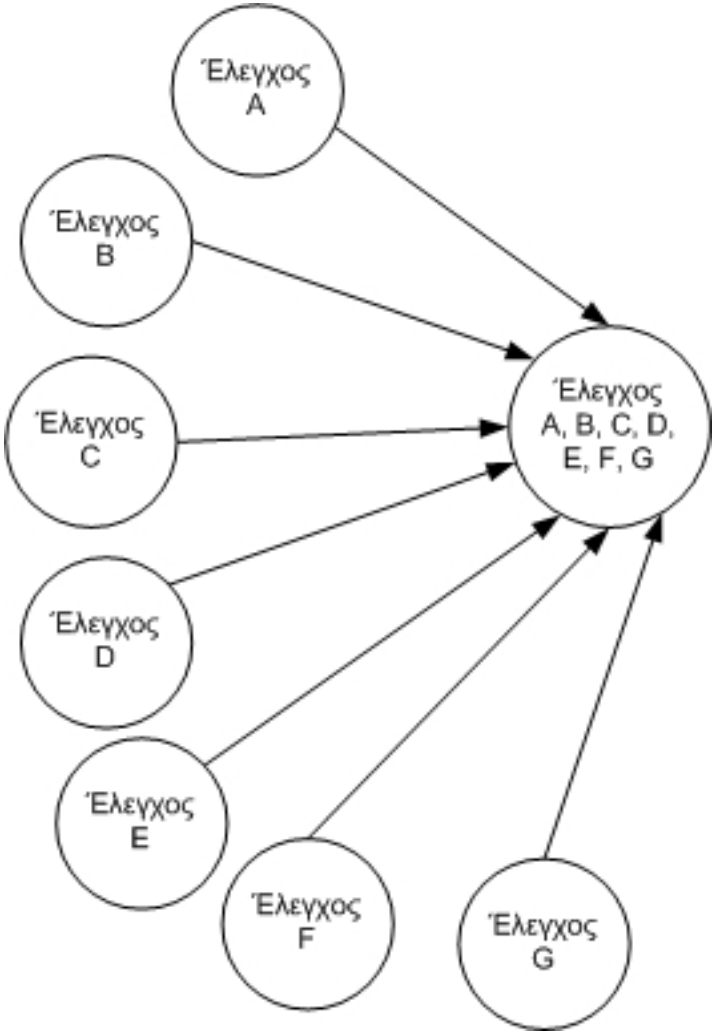
- Έλεγχος συνένωσης
- Συνένωση και οικοδόμηση
- Ημερήσια οικοδόμηση
- Συνεχής συνένωση
- Σχετικές επιδόσεις μεθόδων διασφάλισης ποιότητας
- Μετρικές (μεγέθους, πολυπλοκότητας, συνεκτικότητας και σύζευξης)

συνένωση

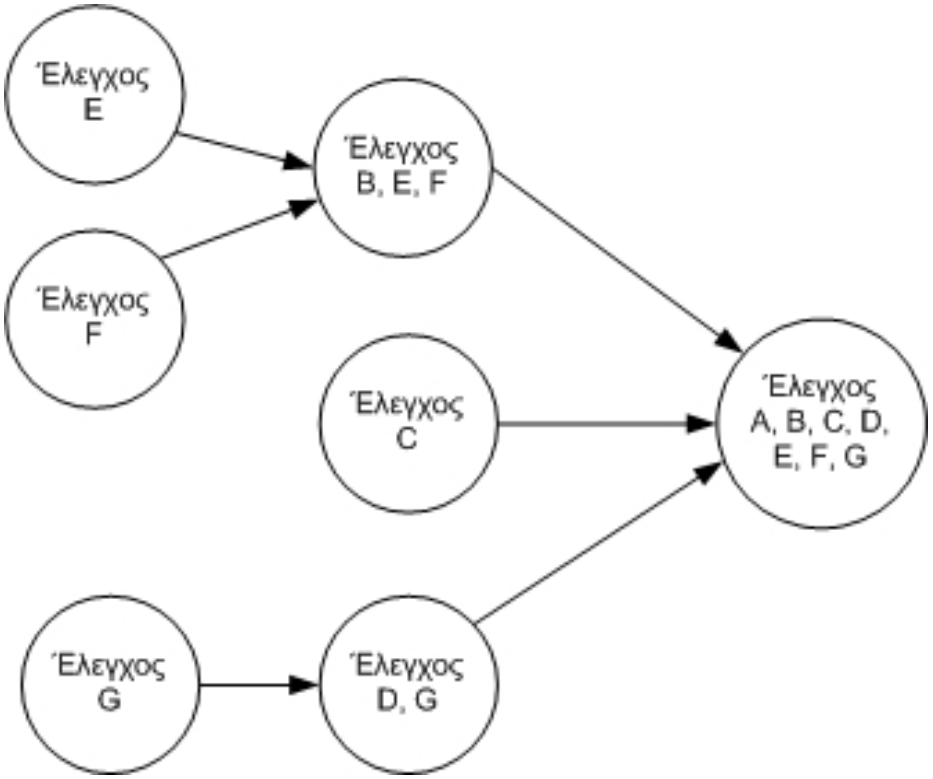


Η συνένωση (integration) είναι η διαδικασία με την οποία οι διαφορετικές μονάδες λογισμικού συνδυάζονται, έτσι ώστε να προκύψει το σύστημα λογισμικού.

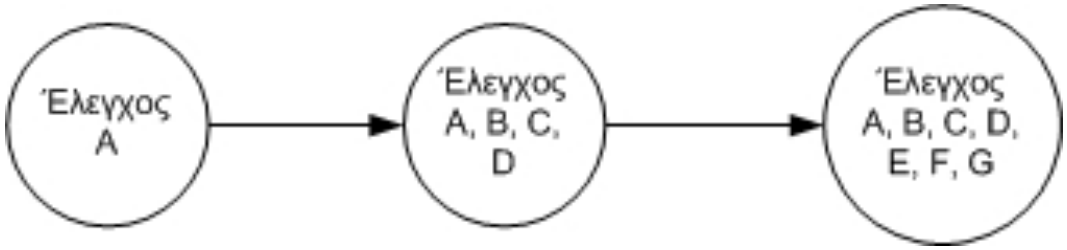
συνένωση big-bang



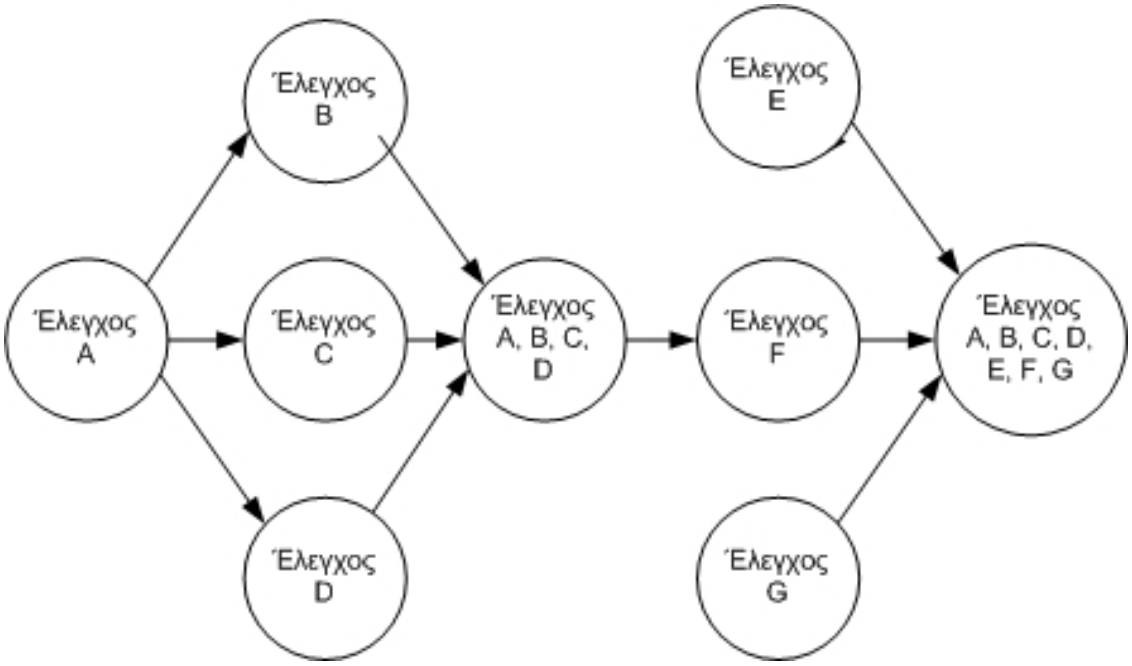
ανοδική συνένωση



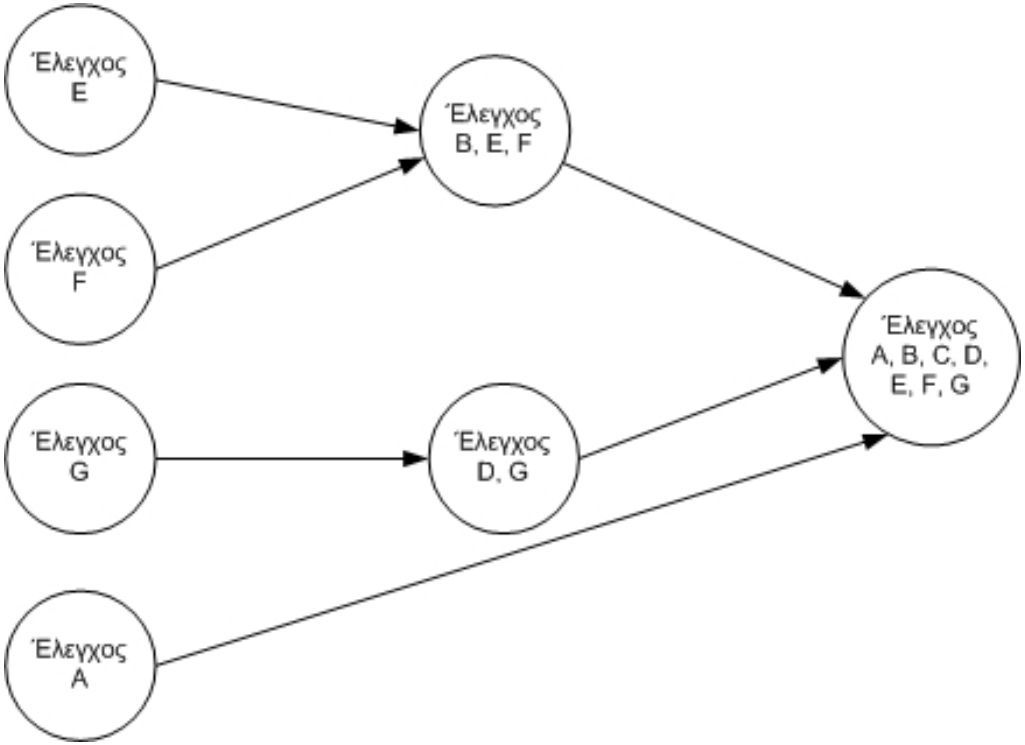
καθοδική συνένωση



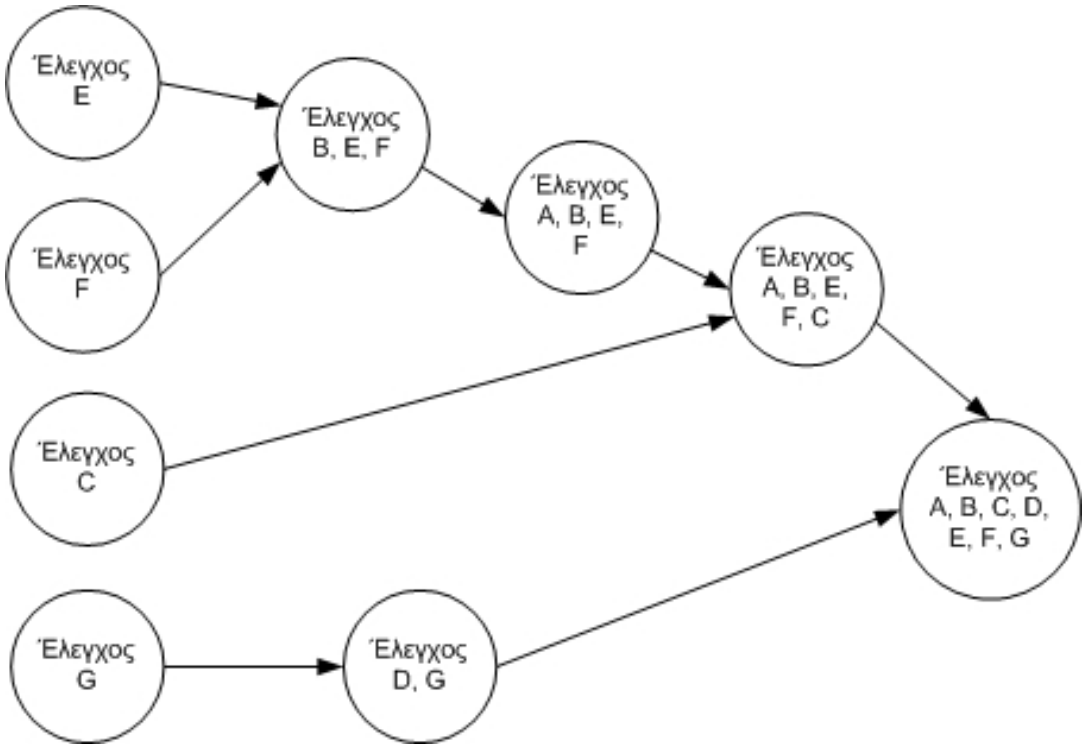
τροποποιημένη καθοδική συνένωση



συνένωση σάντουιτς



συνένωση τροποποιημένου σάντουιτς



συνένωση και οικοδόμηση

- Η συνένωση των μονάδων λογισμικού σχετίζεται άμεσα και με την οικοδόμηση (build) του συστήματος.
- Η οικοδόμηση του συστήματος είναι η μεταγλώττιση και η συνένωση όλου του κώδικα, έτσι ώστε να προκύψει το λογισμικό ως σύστημα.
- Η οικοδόμηση του λογισμικού, όταν αυτό αποτελείται από ένα αρχείο πηγαίου κώδικα και αναπτύσσεται από έναν και μόνο προγραμματιστή, δεν είναι ιδιαίτερα προβληματική.
- Από τη στιγμή όμως που η ανάπτυξη του λογισμικού πραγματοποιείται από μία ομάδα και περιλαμβάνει δεκάδες, εκατοντάδες ή και χιλιάδες αρχεία πηγαίου κώδικα, η οικοδόμηση του λογισμικού, προκειμένου να φθάσει στην εκτελέσιμη μορφή του, είναι αρκετά πολύπλοκη. Η οικοδόμηση του λογισμικού φέρνει και στην επιφάνεια και τα σφάλματα συνένωσης.

ημερήσια οικοδόμηση

- Η ημερήσια οικοδόμηση είναι μία στρατηγική, με την οποία το λογισμικό οικοδομείται καθημερινά.
- Η οικοδόμηση συνοδεύεται από ένα γρήγορο έλεγχο (smoke test) ο οποίος, αν και δεν είναι εξαντλητικός, θεωρείται ως επαρκής για να διαπιστωθεί η υγεία της οικοδόμησης. Εάν η οικοδόμηση του συστήματος αποτύχει, τότε η ανάκαμψη σε σωστή οικοδόμηση θεωρείται για την ομάδα ανάπτυξης ως πρώτη προτεραιότητα.
- Επειδή η ημερήσια οικοδόμηση γίνεται καθημερινά, είναι απαραίτητη η αυτοματοποίησή της.

πλεονεκτήματα ημερήσιας οικοδόμησης

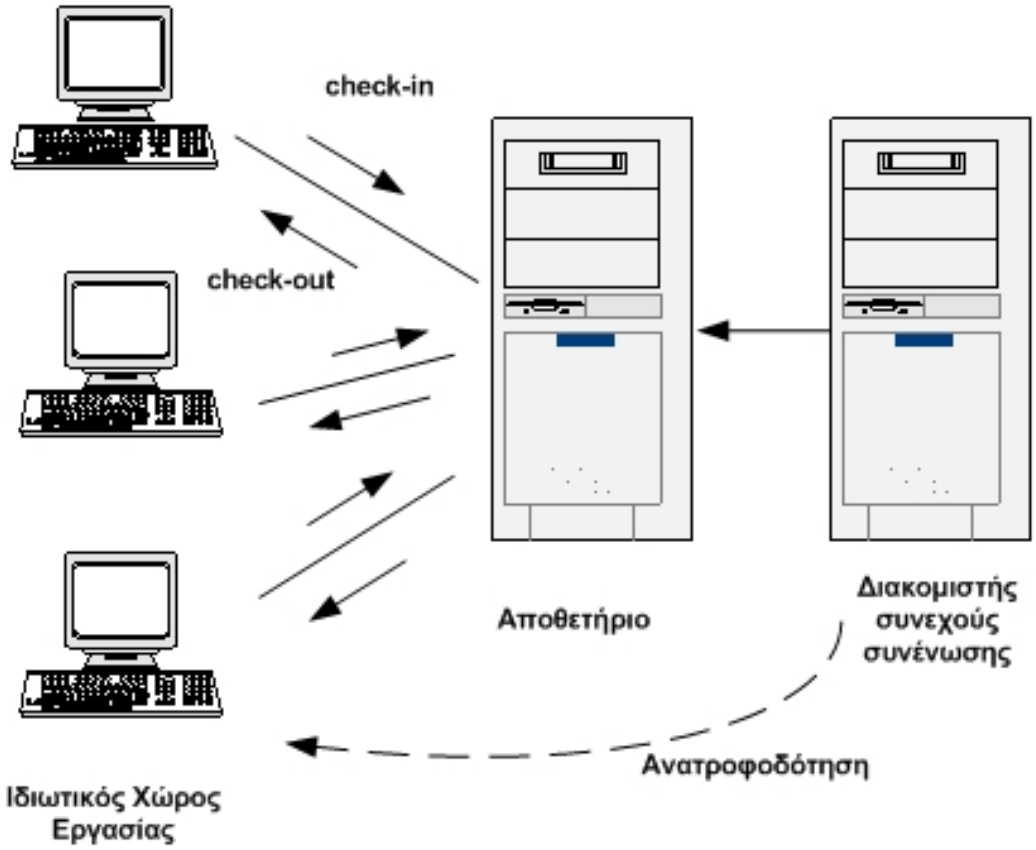
Η ημερήσια οικοδόμηση παρέχει πολλά πλεονεκτήματα, όπως:

- Μείωση κινδύνων.
- Διατηρεί την υψηλή ποιότητα του λογισμικού.
- Η ημερήσια οικοδόμηση γίνεται καθημερινά άρα και ο γρήγορος έλεγχος θα πρέπει να εξασκείται καθημερινά.
- Η ημερήσια οικοδόμηση παρέχει συνεχή πληροφόρηση για την κατάσταση του έργου.

συνεχής συνένωση

- Επειδή ακριβώς η συνένωση και η οικοδόμηση που τη συνοδεύει είναι μία επίπονη διαδικασία, πολλοί προτείνουν την ακόμα πιο συχνή συνένωση και οικοδόμηση του λογισμικού.
- Η συνεχής συνένωση (continuous integrationμας προτείνει καταρχήν ότι ένας προγραμματιστής θα πρέπει να υποβάλλει τον κώδικα (check-in) πολλές φορές την ημέρα.
- Αντιστοίχως το λογισμικό θα πρέπει να οικοδομείται επίσης πολλές φορές την ημέρα.
- Η βασική ιδέα της συνεχούς συνένωσης είναι απόρροια μίας εκ των βασικότερων αρχών της τεχνολογίας λογισμικού, ότι όσο νωρίτερα διαπιστωθεί ένα σφάλμα, τόσο λιγότερο κοστίζει η διόρθωσή του.
- Είναι προφανές ότι στη συνεχή συνένωση οι ανάγκες αυτοματοποίησης είναι πιο επιτακτικές.

συνεχής συνένωση



σχετικές επιδόσεις μεθόδων διασφάλισης ποιότητας

Μέθοδος	**Ελάχιστο Ποσοστό**	**Μέγιστο Ποσοστό**
Άτυπες ανασκοπήσεις κώδικα (περιηγήσεις, προγραμματισμός κατά ζεύγη)	20%	35%
Τυπικές ανασκοπήσεις κώδικα (επιθεωρήσεις)	45%	70%
Έλεγχος μονάδας	15%	50%
Έλεγχος Συνένωσης	25%	40%
Έλεγχος παλινδρόμησης	15%	30%
Έλεγχος Συστήματος	25%	55%
Έλεγχος beta μικρής εμβέλειας (<10)	25%	40%
Έλεγχος beta μεγάλης εμβέλειας (>1000)	60%	85%

μετρικές

- Μετρική (metric) είναι η ποσοτικοποιημένη μέτρηση του βαθμού με τον οποίο ένα σύστημα, μία συνιστώσα ή διαδικασία κατέχει κάποιο συγκεκριμένο χαρακτηριστικό [IEEE 90].
- Με άλλα λόγια η μετρική μάς προσφέρει μία ποσοτικοποίηση κάποιου χαρακτηριστικού του λογισμικού.
- Έχουμε για παράδειγμα το χαρακτηριστικό της πολυπλοκότητας του λογισμικού.
- Θα πρέπει να ορίσουμε μία μετρική (ή μετρικές) με την οποία να μπορούμε να ποσοτικοποιήσουμε αυτό το χαρακτηριστικό.

κατηγοριοποίηση μετρικών

Μία κατηγοριοποίηση των μετρικών για την ανάπτυξη λογισμικού είναι :

- Μετρικές του προϊόντος (product metrics). Είναι μετρικές που σχετίζονται με το προϊόν που αναπτύσσεται.
- Μετρικές του έργου (project metrics). Είναι μετρικές που αφορούν το συγκεκριμένο έργο ανάπτυξης.
- Μετρικές της διαδικασίας (process metrics). Μετρικές που αφορούν την ίδια τη διαδικασία ανάπτυξης λογισμικού και αφορούν πολλά έργα ανάπτυξης στην πάροδο του χρόνου.

μέγεθος

- Η απλούστερη μετρική είναι ο αριθμός των γραμμών κώδικα (Lines of Code ή LOC) ή χιλιάδων γραμμών κώδικα (KLOC).
- Η μέτρηση των γραμμών κώδικα (χωρίς τα σχόλια και κενές γραμμές) γίνεται σε επίπεδο μονάδας (μεθόδου ή κλάσης) ή ακόμα και στο σύνολο του λογισμικού.
- Σε επίπεδο μεθόδου είναι μία ένδειξη για την πολυπλοκότητα και την κατανοησιμότητα της μεθόδου.
- Σε επίπεδο του συνόλου του λογισμικού είναι ένα κριτήριο για να κατατάξουμε την πολυπλοκότητα και το μέγεθος του λογισμικού που αναπτύσσουμε.

πολυπλοκότητα

- Μία παλαιά μετρική που έχει αποδείξει την αξία της στο πέρασμα του χρόνου είναι η κυκλωματική πολυπλοκότητα (cyclomatic complexity – CC).
- Βασικό κριτήριο για την εκτίμηση της πολυπλοκότητας μίας μονάδας λογισμικού (π.χ. μίας μεθόδου) είναι η διακλάδωση στη ροή ελέγχου.
- Προτάσεις if, switch, for και while διακλαδώνουν τη ροή ελέγχου.
- Το αποτέλεσμα είναι η αύξηση της πολυπλοκότητας του λογισμικού.

πολυπλοκότητα

- Η κυκλωματική πολυπλοκότητα μετρά τον αριθμό των διαφορετικών μονοπατιών στη ροή μίας μονάδας.
- Περισσότερα μονοπάτια σημαίνει και μεγαλύτερη πολυπλοκότητα.
- Ο τύπος υπολογισμού της κυκλωματικής πολυπλοκότητας είναι:
  CC = P + 1
  Όπου P ο αριθμός των Boolean αποφάσεων σε μια μονάδα.
- Μία σειριακή εκτέλεση εντολών μας δίνει CC=1 και κάθε εντολή ροής ελέγχου προσθέτει κατά ένα.
- Ένας τρόπος υπολογισμού για κώδικα σε Java είναι να ξεκινήσουμε από ένα και να προσθέτουμε κάθε φορά που συναντούμε προτάσεις if, for, while, case και catch ή τους τελεστές &&, || και ?.

πολυπλοκότητα

- Στο αντικειμενοστρεφές λογισμικό η κυκλωματική πολυπλοκότητα εφαρμόζεται στο επίπεδο μίας μεθόδου.
- Μία μετρική για την πολυπλοκότητα μίας κλάσης είναι οι Σταθμισμένες Μέθοδοι ανά Κλάση (Weighted Methods per Class – WMC). Υπολογίζεται ως εξής:

$$WMC = \sum_{i=1}^{n} ci$$

- Η WMC υπολογίζεται ως το άθροισμα των μετρικών της πολυπλοκότητας κάθε μεθόδου για όλες τις μεθόδους (όπου n ο αριθμός των μεθόδων ανά κλάση και ci η πολυπλοκότητα κάθε μεθόδου).
- Η χρήση της μετρικής είναι γενική και δεν ορίζει ποια είναι η μετρική ci της πολυπλοκότητας κάθε μεθόδου. Είναι προφανές ότι, αν θεωρήσουμε ως ci την κυκλωματική πολυπλοκότητα μίας μεθόδου, έχουμε μία μετρική πολυπλοκότητας στο επίπεδο της κλάσης.

συνεκτικότητα

- Για την αντικειμενοστρεφή σχεδίαση μία κλάση πρέπει να απεικονίζει μία αφαίρεση.
- Η έλλειψη συνεκτικότητας σημαίνει ότι η κλάση μπορεί να αναπαριστά δύο αφαιρέσεις και ίσως θα πρέπει να αλλάξει η σχεδίαση σε περισσότερες της μίας κλάσης.
- Θα εξετάσουμε δύο μετρικές για τη έλλειψη συνεκτικότητας των μεθόδων μίας κλάσης (Lack of Cohesion in Methods – LCOM).
- Η βασική ιδέα των παρακάτω μετρικών είναι να συσχετίσουμε τις μεθόδους μίας κλάσης με τα πεδία της.
- Μία κλάση θεωρείται συνεκτική, εάν οι μέθοδοι της κλάσης χρησιμοποιούν μεγάλο μέρος των πεδίων της.

συνεκτικότητα

- Η πρώτη μετρική LCOM1, λαμβάνει υπόψη της δύο όρους P και Q.
- Ο όρος P είναι ο αριθμός των ζευγών των μεθόδων της κλάσης που δεν έχουν κάποιο κοινό πεδίο.
- Ο όρος Q είναι ο αριθμός των ζευγών των μεθόδων της κλάσης που χρησιμοποιούν έστω και ένα κοινό πεδίο.
- Η μετρική LCOM1 υπολογίζεται ως εξής:
  LCOM1 = P – Q εάν P > Q, ή 0 εάν P<=Q
- Υψηλές τιμές της μετρικής σημαίνει και χαμηλότερη συνεκτικότητα. Αυτό με τη σειρά του σημαίνει ότι οι μέθοδοι είναι ανόμοιες, επειδή δε χρησιμοποιούν κοινά πεδία της κλάσης.

παράδειγμα LCOM1

- Ας θεωρήσουμε, για παράδειγμα, ότι μία κλάση έχει τα πεδία f1, f2, f3, και f4 και τις μεθόδους M1, M2, M3, και M4. Τα πεδία που χρησιμοποιεί κάθε μέθοδος είναι:

M1	f1, f2
M2	f3
M3	f2, f4
M4	f1

- Για να υπολογίσουμε του όρους P και Q, παίρνουμε όλα τα ζεύγη των μεθόδων και εξετάζουμε εάν έχουν κοινά πεδία. Ο όρος P είναι 4, επειδή τα ζεύγη M1-M2, M2-M3, M2-M4, M3-M4 δεν έχουν κάποιο κοινό πεδίο. Ο όρος Q είναι 2, επειδή τα ζεύγη M1-M3 και M1-M4 χρησιμοποιούν κοινά πεδία.
- Η μετρική LCOM1 είναι 2.

συνεκτικότητα

- Μία δεύτερη μέτρηση της έλλειψης συνεκτικότητας η LCOM2 έχει διαφορετικό τρόπο υπολογισμού.
- Έστω:
  M: Το σύνολο των μεθόδων μίας κλάσης όπου ο αριθμός των μεθόδων ορίζεται ως m.
  F: Το σύνολο των πεδίων της κλάσης.
  p(f) ο αριθμός των μεθόδων της κλάσης που χρησιμοποιούν το πεδίο f το οποίο ανήκει στο F.
  ap ο μέσος αριθμητικών των p(f) ως προς το F
- LCOM2 = (ap – m) / (1 – m)
- Η μετρική LCOM2 κινείται στο πεδίο [0-2]. Η τέλεια συνεκτικότητα είναι, όταν όλες οι μέθοδοι χρησιμοποιούν όλα τα πεδία της κλάσης και η τιμή της LCOM2 είναι 0. Εάν η LCOM2 είναι μεγαλύτερη του 1, έχουμε ένδειξη για χαμηλή συνεκτικότητα.

παράδειγμα: LCOM2

Για την προηγούμενη περίπτωση έχουμε:

m=4

p(f1) = 2

p(f2) = 2

p(f3) = 1

p(f4) = 1

ap=6/4

Η LCOM2 υπολογίζεται: LCOM2 = 0,83

σύζευξη

- Οι απλούστερες μετρικές της σύζευξης είναι το fan-in και το fan-out.
- Για μία μονάδα λογισμικού το fan-in υπολογίζεται ως ο αριθμός των μονάδων που χρησιμοποιούν τη συγκεκριμένη μονάδα.
- Το fan-out αποτυπώνει την αντίστροφη σχέση. Τον αριθμό των μονάδων λογισμικού που χρησιμοποιεί η συγκεκριμένη μονάδα.
- Για το αντικειμενοστρεφές λογισμικό οι δύο μετρικές χρησιμοποιούνται σε επίπεδο κλάσης.
- Έτσι για μία κλάση υπολογίζουμε το fan-in ως τον αριθμό των κλάσεων που τη χρησιμοποιούν και το fan-out ως τον αριθμό των κλάσεων που χρησιμοποιεί η κλάση.

σύζευξη

- Υψηλό fan-out θεωρείται ως ένδειξη υψηλής σύζευξης και πιθανή ένδειξη για τροποποίηση της σχεδίασης.
- Το fan-out το συναντούμε και ως μετρική για τη σύζευξη μεταξύ αντικειμένων (Coupling Between Objects – CBO).
- Η αξιολόγηση υψηλού fan-in είναι πιο σύνθετη.
- Αν για παράδειγμα επιδιώκουμε να μεγιστοποιήσουμε την επαναχρησιμοποίηση μίας κλάσης, τότε υψηλό fan-in σημαίνει και υψηλό βαθμό επαναχρησιμοποίησης και δε θα πρέπει να θεωρείται αναγκαστικά ως κακό σημάδι.

σύζευξη

- Οι μετρικές του fan-in και fan-out μπορούν να γενικευτούν και σε επίπεδο πακέτων [Martin 03]. Έτσι ορίζουμε:
  - Φυγόκεντρη Σύζευξη (Efferent Coupling-Ce). Είναι ο αριθμός των κλάσεων εντός του πακέτου οι οποίες εξαρτώνται από κλάσεις εκτός του πακέτου.
  - Κεντρομόλος Σύζευξη (Afferent Coupling – Ca). Είναι ο αριθμός των κλάσεων εκτός του πακέτου οι οποίες εξαρτώνται από τις κλάσεις εντός του πακέτου.
- Μπορούμε να συνδυάσουμε τις δύο μετρικές Ca και Ce για να υπολογίσουμε την αστάθεια (instability) I ενός πακέτου ως εξής:

  $I = Ce / (Ca + Ce)$

- Η μετρική I της αστάθειας κινείται στο [0-1], όπου το 0 δείχνει τη μέγιστη ευστάθεια και το 1 τη μέγιστη αστάθεια.