

Tuya Android Smart Life App SDK Sample

ΑΝΑΛΥΣΗ ΠΟΙΟΤΗΤΑΣ ΚΩΔΙΚΑ
ΕΡΓΑΣΙΑ ΕΞΑΜΗΝΟΥ 2022-2023 KRISTI CAMI 3882

ΕΡΓΑΛΕΙΑ

- ▶ Για την αξιολόγηση μετρικών του κώδικα χρησιμοποιήθηκαν :
 - IntelliJ IDEA Edu version 2022.3.3
 - Code Metrics Plugin

ΜΕΤΡΗΣΕΙ ΕΡΓΑΛΕΙΟΥ

Το εργαλείο Code metric plugin παρέχει τις εξής μετρήσεις :

1. Total Cyclomatic Complexities
2. Average Cyclomatic Complexity
3. Average Operation Complexity
4. Maximum Cyclomatic Complexity
5. Total Cyclomatic Complexity of the methods
6. Cognitive Complexity
7. Essential Complexity

TOTAL CYCLOMATIC COMPLEXITIES

- ▶ Η μετρική **Total Cyclomatic Complexity** μετρά τον αριθμό των ανεξάρτητων μονοπατιών που μπορούν να παρουσιαστούν σε ένα κομμάτι κώδικα. Κάθε μονοπάτι αντιστοιχεί σε μια διαφορετική ακολουθία αποφάσεων (if-else statements, loops, switch-cases κλπ.) που μπορεί να εκτελεστεί κατά τη διάρκεια της εκτέλεσης του προγράμματος. Επίσης χρησιμοποιείται για να αξιολογήσει τον βαθμό δυσκολίας της κατανόησης, της δοκιμής και της συντήρησης του κώδικα.
- ▶ Στο πλαίσιο του εργαλείου Code Metrics Plugin στο IntelliJ IDEA, το total cyclomatic complexity είναι ένας μετρητής που υπολογίζει τη συνολική πολυπλοκότητα του κώδικα μιας μεθόδου ή μιας κλάσης. Υπολογίζεται συνήθως με βάση τον αριθμό των συνθηκών, των επαναλήψεων και των ανακατευθύνσεων ροής ελέγχου που περιέχει ο κώδικας. Όταν οι μετρήσεις είναι ανάμεσα στο 1-10 η κατανόηση είναι εύκολη.

AVERAGE CYCLOMATIC COMPLEXITY

- ▶ Ορίζουμε το **Average Cyclomatic Complexity** ως την Συνολική Πολυπλοκότητα που έχει το πρόγραμμα δια το άθροισμα των μεθόδων που υπάρχουν σε κάθε κλάση. Όταν οι μετρήσεις είναι ανάμεσα στο 1-10 το πρόγραμμα είναι κατανοητό και διευκολύνετε η δημιουργία των test. Όταν περνάει το 10 μεγαλώνει η πολυπλοκότητα και ο κώδικας γίνεται πιο «σύνθετος».
- ▶ Η μετρική αυτή μπορεί να χρησιμοποιηθεί για να προσδιορίσει τον μέσο βαθμό πολυπλοκότητας των μεθόδων ή των κλάσεων στον κώδικα. Μια υψηλή τιμή μπορεί να υποδηλώνει ότι ο κώδικας έχει περισσότερες σύνθετες μεθόδους ή κλάσεις που απαιτούν περισσότερες αποφάσεις και πιθανώς να είναι δυσκολότερος στην κατανόηση και στη συντήρηση.

AVERAGE OPERATION COMPLEXITY

- ▶ Η μετρική **Average Operation Complexity** αναφέρεται στο μέσο όρο της πολυπλοκότητας των λειτουργιών (operations) μιας μεθόδου ή ενός τμήματος κώδικα. Η πολυπλοκότητα μιας λειτουργίας μπορεί να μετρηθεί με βάση τον αριθμό των ενεργειών (actions) που απαιτούνται για την εκτέλεσή της.
- ▶ Η **Average Operation Complexity** υπολογίζεται παίρνοντας τον μέσο όρο της πολυπλοκότητας των λειτουργιών σε ένα σύνολο κώδικα. Μια υψηλή τιμή υποδηλώνει ότι οι λειτουργίες είναι πιο πολύπλοκες, με περισσότερες ενέργειες που απαιτούνται για την εκτέλεσή τους. Αυτό μπορεί να υποδείξει πιθανά προβλήματα σε σχέση με την κατανοητότητα και τη συντηρησιμότητα του κώδικα.

MAXIMUM CYCLOMATIC COMPLEXITY

- ▶ Η **Maximum Cyclomatic Complexity** αναφέρεται στο μέγιστο επίπεδο κυκλοματικής πολυπλοκότητας που παρουσιάζεται σε ένα σύνολο κώδικα. Η κυκλοματική πολυπλοκότητα μετρά τον αριθμό των ανεξάρτητων μονοπατιών που μπορούν να υπάρχουν σε ένα τμήμα κώδικα, με βάση τις ροές ελέγχου που περιλαμβάνονται, όπως συνθήκες, επαναλήψεις και ανακατευθύνσεις.
- ▶ Η μέγιστη κυκλοματική πολυπλοκότητα υποδεικνύει το υψηλότερο επίπεδο πολυπλοκότητας που παρουσιάζεται σε ένα σύνολο κώδικα. Αυτό μπορεί να σημαίνει ότι υπάρχουν μέθοδοι ή τμήματα κώδικα με πολύπλοκες δομές ελέγχου που απαιτούν περισσότερες αποφάσεις και μονοπάτια για την εκτέλεσή τους.

TOTAL CYCLOMATIC COMPLEXITY OF THE METHODS

- ▶ Η **Total cyclomatic complexity of the methods** αναφέρεται στο άθροισμα των κυκλοματικών πολυπλοκοτήτων όλων των μεθόδων ενός συνόλου κώδικα. Η κυκλοματική πολυπλοκότητα μετρά τον αριθμό των ανεξάρτητων μονοπατιών που μπορούν να υπάρχουν μέσα σε μια μέθοδο, βάσει των δομών ροής ελέγχου όπως συνθήκες, επαναλήψεις και εναλλαγές.

COGNITIVE COMPLEXITY

- ▶ Η **Cognitive Complexity** είναι μια μετρική που αξιολογεί το επίπεδο γνωστικών απαιτήσεων που απαιτούνται για την κατανόηση ενός κομματιού κώδικα. Διαφέρει από την κυκλοματική πολυπλοκότητα καθώς λαμβάνει υπόψη παράγοντες όπως η εμπειρία των προγραμματιστών.

ESSENTIAL COMPLEXITY

- ▶ Ο όρος **Essential Complexity** αναφέρεται στη βασική ή ουσιώδη πολυπλοκότητα ενός συστήματος ή ενός κομματιού κώδικα. Πρόκειται για την πολυπλοκότητα που πηγάζει από τη φύση του προβλήματος που πρέπει να λυθεί και δεν μπορεί να αποφευχθεί πλήρως.
- ▶ Δεν είναι αποτέλεσμα κακού σχεδιασμού ή ατέλειας των προγραμματιστών, αλλά από την πολυπλοκότητα της ίδιας της πραγματικής πρόκλησης που πρέπει να επιλυθεί.

TUYA METRICS

- Εκτελώντας την μετρική **Average Cyclomatic Complexity** περνούμε ως αποτέλεσμα 5,11.

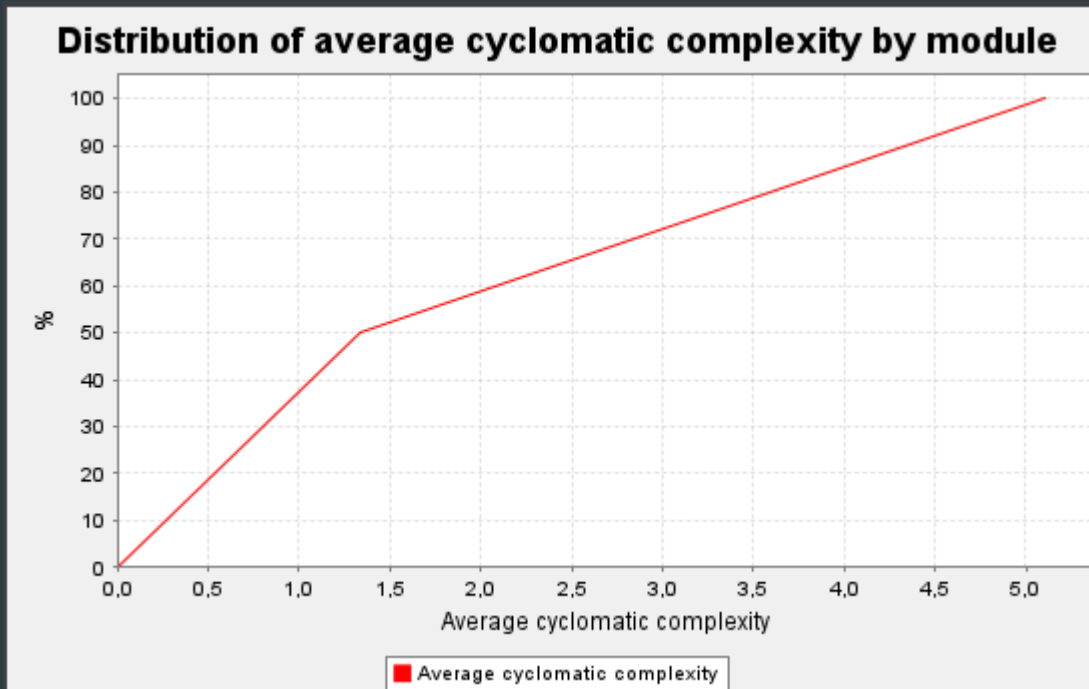
| module | ▼ | v(G)avg | v(G)tot |
|----------------------|---|---------|------------|
| smart-home-java.main | | 5,11 | 189 |
| smart-home-java.test | | 1,33 | 4 |
| Total | | | 193 |
| Average | | 4,83 | 96,50 |

ΣΧΟΛΙΑΣΜΟΣ ΑΠΟΤΕΛΕΣΜΑΤΟΣ

- ▶ Το αποτέλεσμα 5.11 υποδεικνύει την μέση πολυπλοκότητα κάθε μεθόδου ή συνάρτησης στον κώδικά ξεχωριστά.
- ▶ Μια μέση τιμή 5.11 μπορεί να θεωρηθεί αρκετά καλή, αλλά αυτό εξαρτάται από το πλαίσιο του έργου και τις απαιτήσεις ποιότητας του κώδικα. Γενικά, χαμηλότερες τιμές κυκλοματικής πολυπλοκότητας θεωρούνται προτιμότερες, καθώς δείχνουν απλούστερο και πιο ευανάγνωστο κώδικα.

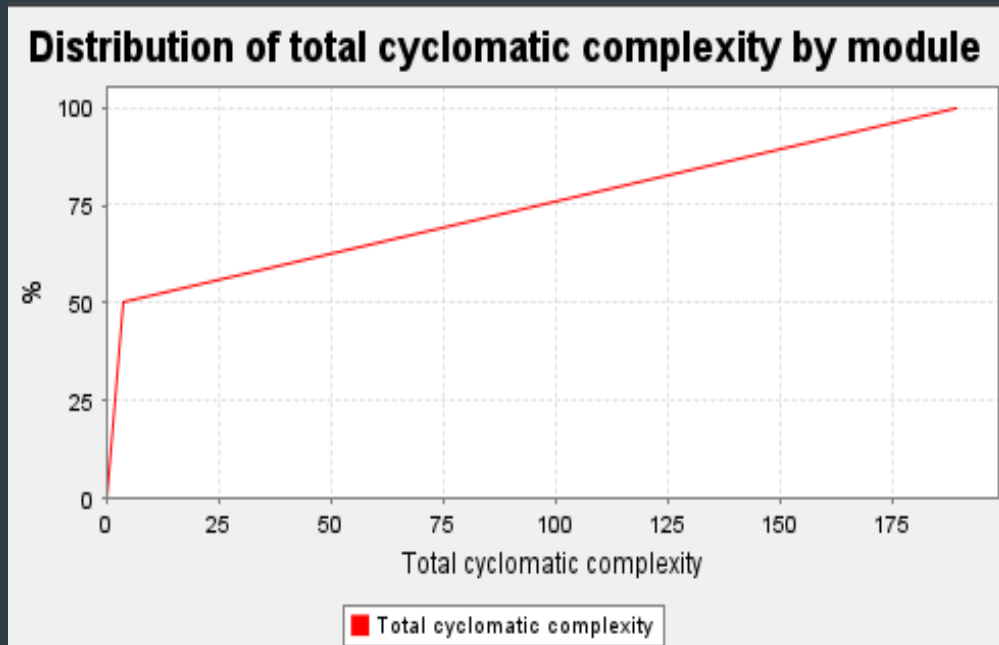
ΤΡΟΠΟΙ ΒΕΛΤΙΩΣΕΙΣ

- Αφού το αποτέλεσμα κρίθηκε αρκετά καλό στα πλαίσια του συγκεκριμένου έργου, δεν υπήρξαν προσπάθειες για την περεταίρω βελτιώσσει του.



TUYA METRICS

- Εκτελώντας την μετρική **Total Cyclomatic Complexities** στο project Tuya περνούμε ως αποτέλεσμα 189.



ΣΧΟΛΙΑΣΜΟΣ ΑΠΟΤΕΛΕΣΜΑΤΟΣ

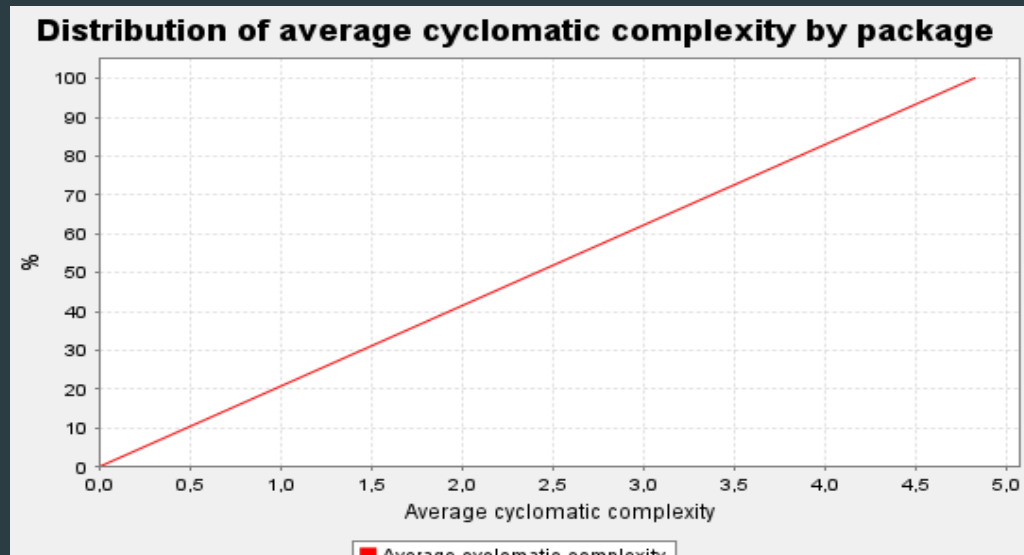
- ▶ Όταν λέμε ότι η συνολική κυκλοματική πολυπλοκότητα του κώδικά είναι 189, αυτό σημαίνει ότι υπάρχουν 189 σημεία στον κώδικα όπου ο ελεγκτής ροής κάνει αποφάσεις. Με άλλα λόγια, υπάρχουν 189 σημεία στον κώδικα που χρησιμοποιούν συνθήκες όπως if, switch, while, for κ.λπ.
- ▶ Ο αριθμός 189 υποδεικνύει ότι ο κώδικάς έχει πολλές αποφάσεις και πολλαπλά μονοπάτια εκτέλεσης

ΤΡΟΠΟΙ ΒΕΛΤΙΩΣΕΙΣ

- ▶ Για να μειωθεί το **Total Cyclomatic Complexitie**, εξετάστηκε αναλυτικά ο κώδικας και έγιναν προσπάθειες για την απλοποίησή του. Μερικές τεχνικές που χρησιμοποιήθηκαν ήταν η ανάλυση και διάσπαση μεγάλων συναρτήσεων σε μικρότερες και η μείωση της εμβέλειας των μεταβλητών.
- ▶ Παρόλα αυτά η αλλαγές που έγιναν δεν επηρέασαν το τελικό αποτέλεσμα της μετρικής, και το αποτέλεσμα συνέχισε να βγαίνει 189.

TUYA METRICS

- Οι μετρήσεις για το **Average Operation Complexity** είναι 4,83 όπως φαίνεται και στην παρακατω γραφική παράσταση.



ΣΧΟΛΙΑΣΜΟΣ ΑΠΟΤΕΛΕΣΜΑΤΟΣ

- ▶ Έχοντας μια μέση πολυπλοκότητα λειτουργίας 4.83, υποδεικνύετε ότι οι λειτουργίες στον κώδικά έχουν μια μέτρια πολυπλοκότητα. Αυτό μπορεί να υποδεικνύει ότι οι λειτουργίες διαθέτουν μια συγκεκριμένη λογική ή αριθμό επιμέρους ενεργειών που πρέπει να εκτελεστούν.

ΤΡΟΠΟΙ ΒΕΛΤΙΩΣΕΙΣ

- ▶ Για την βελτίωση της συγκεκριμένης μετρικής εξετάστηκαν η λειτουργίες με την υψηλότερη πολυπλοκότητα και επανεξετάστηκε ο κώδικας τους. Έγιναν προσπάθειες για την ανάλυση της λογικής του κώδικα καθώς και προσπάθειες για την διάσπαση τους σε μικρότερες και πιο απλές λειτουργίες.

TUYA METRICS

- Το **Maximum Cyclomatic Complexity** για το πρόγραμμα μας είναι 11,09 και η **Total Cyclomatic Complexity of the methods** είναι 15,27.

| Method metrics | | Class metrics | Package metrics | Module metrics | Project metrics |
|------------------------------|--|---------------|-----------------|----------------|-----------------|
| class | | OCavg ▼ | OCmax | WMC | |
| com.example.ReportState | | 1,33 | 2 | 4 | |
| com.example.SmarHomeEndToEi | | 1,33 | 2 | 4 | |
| com.example.SmartHomeServlet | | 1,25 | 2 | 5 | |
| com.example.FakeAuthServlet | | 1,00 | 1 | 2 | |
| com.example.LoginServlet | | 1,00 | 1 | 2 | |
| com.example.SmartHomeCreateS | | 1,00 | 1 | 3 | |
| com.example.SmartHomeDeleteS | | 1,00 | 1 | 3 | |
| Total | | | | 168 | |
| Average | | 4,20 | 11,09 | 15,27 | |

ΣΧΟΛΙΑΣΜΟΣ ΑΠΟΤΕΛΕΣΜΑΤΟΣ

- ▶ Μια τιμή **Maximum Cyclomatic Complexity** της τάξεως του 11,09 δεν είναι υπερβολικά υψηλή. Σύμφωνα με διάφορα συμβατικά πρότυπα, κάτι πάνω από 10 θεωρείται ως υψηλή πολυπλοκότητα, και πιθανότατα χρειάζεται αναθεώρηση για να διευκολυνθεί η αναγνωσιμότητα και η συντηρησιμότητα.
- ▶ Ένας δείκτης **Total Cyclomatic Complexity** της τάξης του 15,27 είναι σχετικά υψηλός. Αυτός ο αριθμός αντικατοπτρίζει τη συνολική πολυπλοκότητα όλων των μεθόδων που εξετάζονται, δηλαδή τον αριθμό των διαφορετικών διαδρομών που μπορεί να ακολουθήσει το πρόγραμμα κατά την εκτέλεση αυτών των μεθόδων.











ΤΡΟΠΟΙ ΒΕΛΤΙΩΣΕΙΣ

- ▶ Όπως και στην προσπάθεια βελτιώσεις των προηγούμενων μετρικών, έτσι και εδώ θα χρειαστεί να:
 1. Διαιρέσουμε τις μεθόδους μας σε μικρότερες, πιο διαχειρίσιμες μεθόδους.
 2. Αποφυγή της χρήσης υπερβολικών ελεγκτικών δομών (όπως if, switch, while, for, κ.λπ.) μέσα σε μία μέθοδο.
 3. Και τέλος αναθεώρηση της λογικής των μεθόδων.

ΤΥΥΑ METRICS

► Cognitive Complexity :

Το πρόγραμμα έχει ως επί τον πλείστον χαμηλή πολυπλοκότητα. Σε 3 μεθόδους ξεπερνάει το όριο του 10. Πιο συγκεκριμένα στην μέθοδο Map έχει 102 πολυπλοκότητα. Στην μέθοδο ExecuteResponse έχει πολυπλοκότητα 23. Στην μέθοδο updateDevise έχει πολυπλοκότητα 17. Ο μέσος όρος πολυπλοκότητας των μεθόδων συνολικά είναι 4,78. Αυτή η τιμή μπορεί αν θεωρηθεί ως μια πολύ καλή μέτρηση .

| method | ▼ | CogC | ev(G) | iv(G) | v(G) |
|---|---|------|-------|-------|------|
|   com.example.MyDataStore.execute(String, String, Execution) | | 102 | 19 | 44 | 105 |
|   com.example.MySmartHomeApp.onExecute(ExecuteRequest, Map<?, ?>) | | 23 | 7 | 9 | 9 |
|   com.example.MyDataStore.updateDevice(String, String, Map<String, Object>, Map<Str | | 17 | 1 | 12 | 12 |
|   com.example.MySmartHomeApp.onSync(SyncRequest, Map<?, ?>) | | 12 | 1 | 8 | 8 |
|   com.example.SmartHomeUpdateServlet.doPost(HttpServletRequest, HttpServletResponse | | 5 | 1 | 6 | 6 |

TUYA METRICS

► Essential Complexity:

Παρατηρείτε ότι όπου οι μέθοδοι παρουσιάζουν μεγάλη πολυπλοκότητα παρουσιάζουν επίσης και μεγάλη έλλειψη οργάνωσης της δομής. Συγκεκριμένα η μέθοδος `Map` έχει τιμή 19 και η μέθοδος `ExecuteResponse` έχει τιμή 7. Όμως ο μέσος όρος είναι 1,65. Το αποτέλεσμα βρίσκεται κοντά στο 1 άρα σαν συνολική εικόνα έχει καλή συνοχή.



ΣΥΜΠΕΡΑΣΜΑ

- ▶ Σαν γενική εικόνα το έργο παρουσιάζει μια καλή συνοχή και συνεκτικότητα τόσο στις κλάσεις όσο και στις μεθόδους των κλάσεων.
- ▶ Το μόνο πρόβλημα που παρουσιάζετε είναι ότι έχει μεγάλη τιμή στο **Average Cyclomatic Complexity**. Ένας υψηλός δείκτης μπορεί να είναι δείγμα περίπλοκου κώδικα και δύσκολης συντηρησιμότητας, αφού περισσότερα μονοπάτια εκτέλεσης συχνά σημαίνουν περισσότερες περιπτώσεις που πρέπει να δοκιμαστούν και να κατανοηθούν.