



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης  
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών  
Τομέας Ηλεκτρονικής και Υπολογιστών

# ΕΞΟΡΥΞΗ ΔΕΔΟΜΕΝΩΝ ΤΕΧΝΟΛΟΓΙΑΣ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΕΠΑΝΑΧΡΗΣΙΜΟΠΟΙΗΣΗ ΛΟΓΙΣΜΙΚΟΥ

## Θεμιστοκλής Διαμαντόπουλος

Παρουσίαση – Υποστήριξη Διδακτορικής Διατριβής



Εργαστήριο Επεξεργασίας  
Πληροφορίας και Υπολογισμών

**issel**

Ομάδα Ευφών Συστημάτων και  
Τεχνολογίας Λογισμικού

# Περιεχόμενα

- Εισαγωγή
- Εξόρυξη Απαιτήσεων
- Εξόρυξη Πηγαίου Κώδικα
- Αξιολόγηση Ποιότητας
- Συμπεράσματα

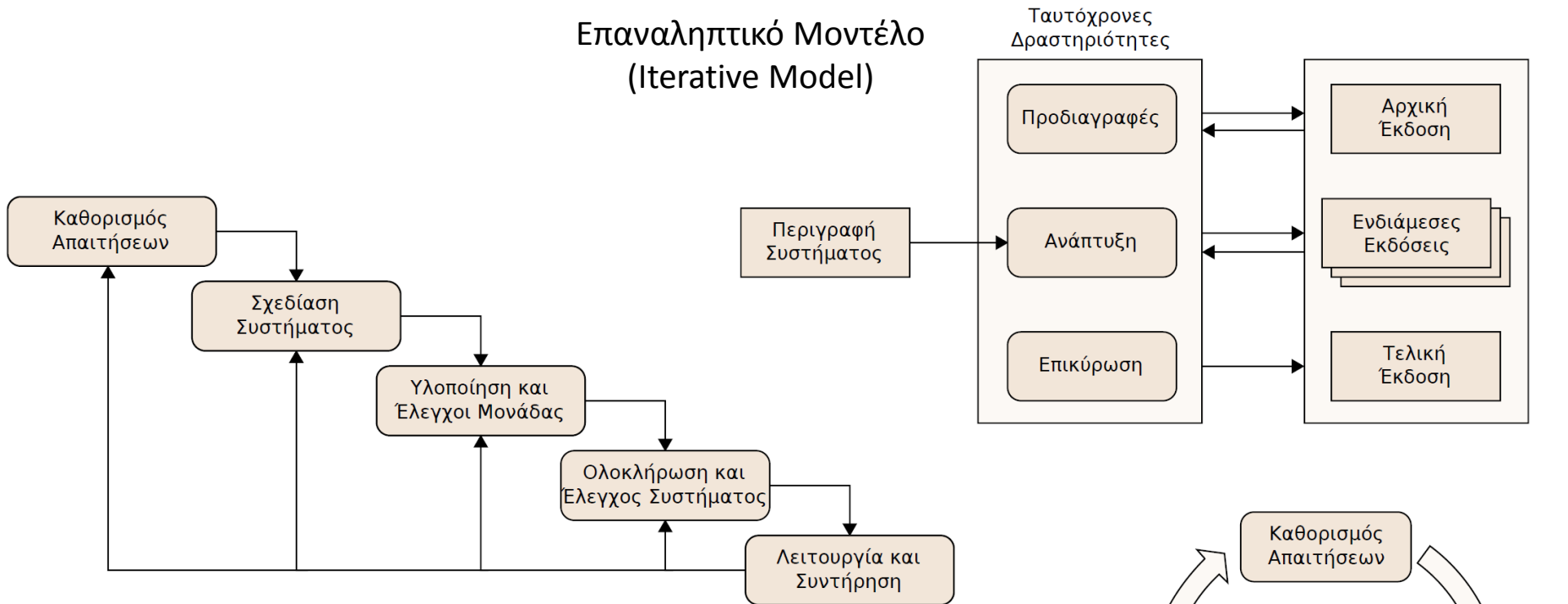
# Εισαγωγή

# Η Τεχνολογία Λογισμικού σήμερα

- Το λογισμικό βρίσκεται παντού
  - Πληροφοριακά συστήματα, διαδικτυακές εφαρμογές, εφαρμογές κινητών συσκευών, κ.α.
- ... όμως η ανάπτυξη και συντήρησή του έχει σημαντικό κόστος
  - 250 δις \$/έτος (Standish Group. The CHAOS Report (1994))
- ... και συχνά προκύπτουν αστοχίες
  - 19% των έργων λογισμικού αποτυγχάνουν
  - 52% κοστίζουν περισσότερο από ότι προβλεπόταν και/ή παραδίδονται εκπρόθεσμα

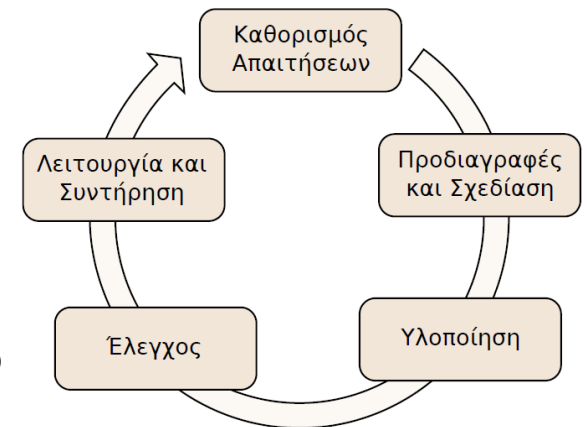
# Μοντέλα Ανάπτυξης Λογισμικού

## Επαναληπτικό Μοντέλο (Iterative Model)



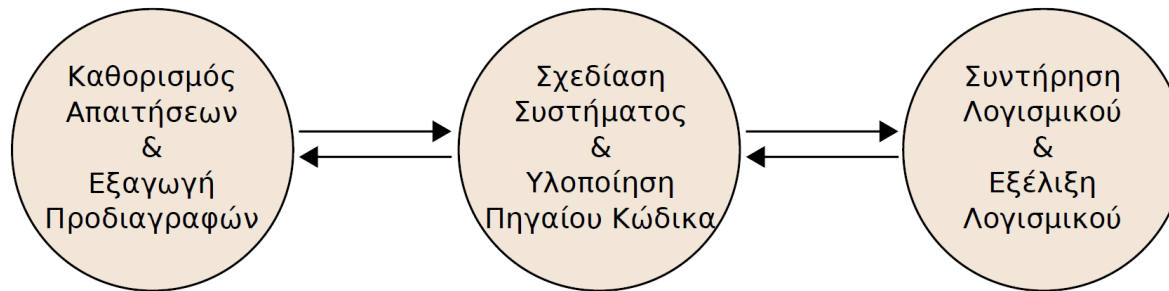
## Μοντέλο Καταρράκτη (Waterfall Model)

## Ευέλικτο Μοντέλο (Agile Model)



# Δεδομένα Τεχνολογίας Λογισμικού (1/2)

- Δραστηριότητες που αλληλοσυνδέονται



- Κάθε δραστηριότητα παράγει δεδομένα προς αξιοποίηση



# Δεδομένα Τεχνολογίας Λογισμικού (2/2)

- Κάποια ενδεικτικά στατιστικά...
  - Υπηρεσίες Φιλοξενίας Κώδικα<sup>1</sup>

	Github	Bitbucket	SourceForge
# Χρήστες	24M	5M	3.7M
# Έργα	69M	-	500K

- Κοινότητες Ερωταπαντήσεων
  - Π.χ. Stack Overflow (> 8M Χρήστες)<sup>2</sup>
- Και πολλές άλλες υπηρεσίες
  - CodeProject, npm registry, ProgrammableWeb, Bugzilla, ...

Stack Overflow Stats

Questions	16M
Answers	25M
Comments	67M
Tags	52K

<sup>1</sup> [https://en.wikipedia.org/wiki/Comparison\\_of\\_source\\_code\\_hosting\\_facilities](https://en.wikipedia.org/wiki/Comparison_of_source_code_hosting_facilities)

<sup>2</sup> <https://data.stackexchange.com/>

# Επαναχρησιμοποίηση Λογισμικού

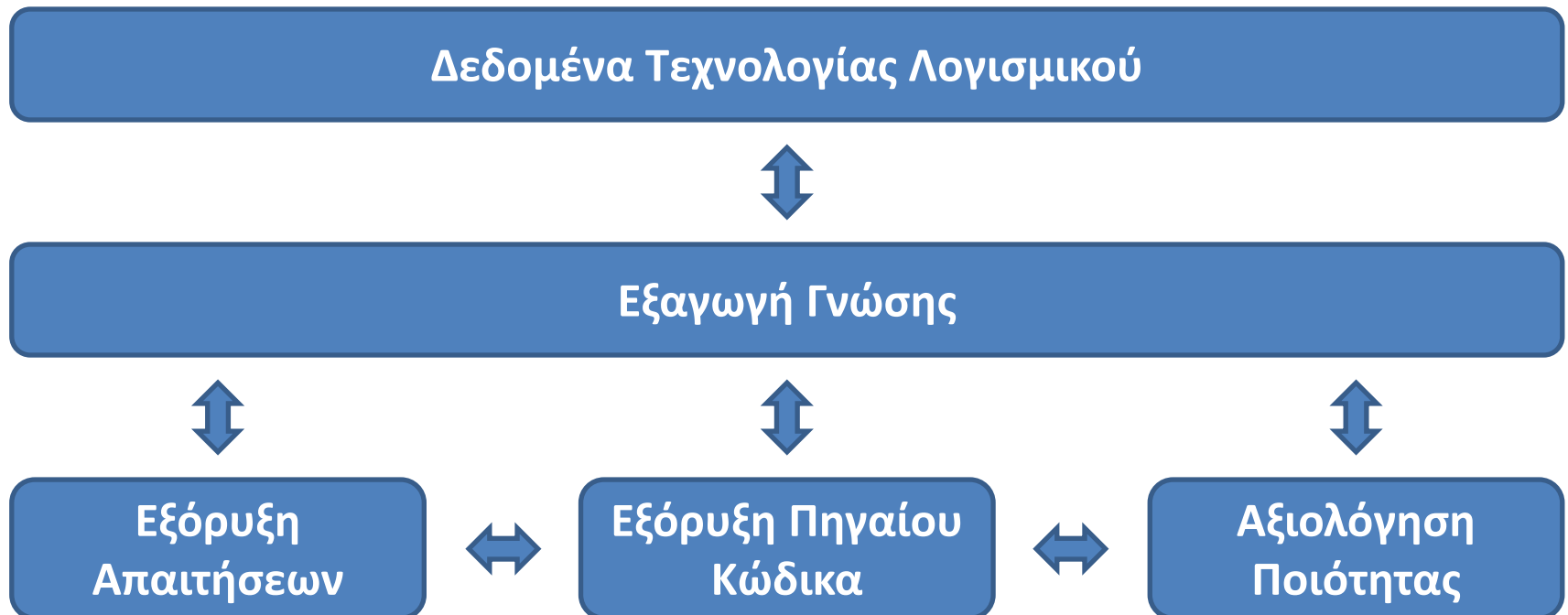
- Τι είναι Επαναχρησιμοποίηση Λογισμικού;
  - “Η συστηματική αξιοποίηση υπαρχόντων αντικειμένων λογισμικού κατά τη διαδικασία κατασκευής νέου λογισμικού ή η ενσωμάτωση τους σε νέο λογισμικό”
- Ποια είναι τα οφέλη;
  - Μείωση Χρόνου και Ανθρωποπροσπάθειας
  - Ταχύτερος Χρόνος Διάθεσης στην Αγορά
  - Υψηλή Ποιότητα του Τελικού Προϊόντος



# Στόχος Διατριβής

“Εφαρμογή τεχνικών εξόρυξης δεδομένων σε δεδομένα τεχνολογίας λογισμικού για τη διευκόλυνση των διαφόρων φάσεων του κύκλου ζωής του λογισμικού”

## Επαναχρησιμοποίηση



# Εξόρυξη Απαιτήσεων

Εξόρυξη Πηγαίου Κώδικα

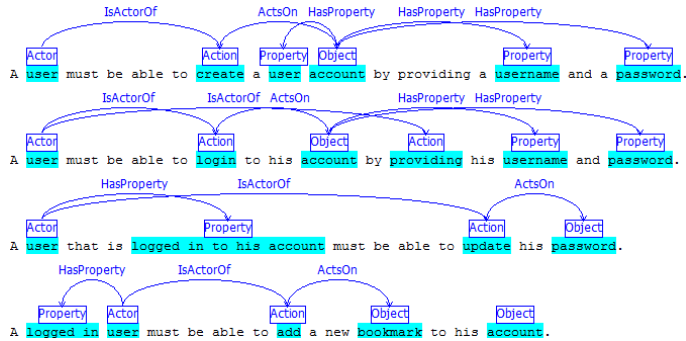
Αξιολόγηση Ποιότητας

# Επισκόπηση (1/2)

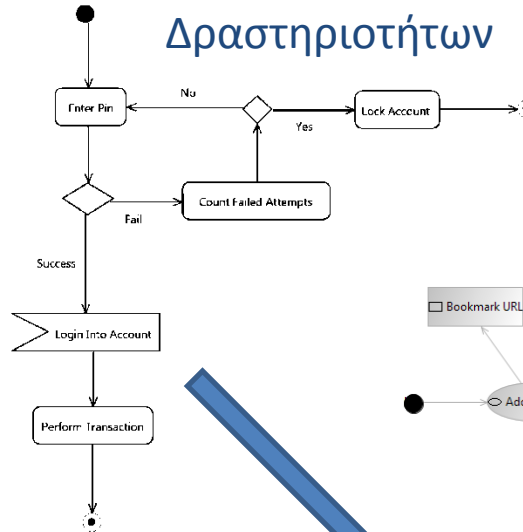
- Εξόρυξη Απαιτήσεων Χρήστη για τον **Καθορισμό Απαιτήσεων** και την **Εξαγωγή Προδιαγραφών**
- Δεδομένα σε πολλαπλές μορφές
  - Λειτουργικές Απαιτήσεις σε Φυσική Γλώσσα
  - Διαγράμματα UML Σεναρίων Χρήσης και Δραστηριοτήτων
- Μεθοδολογία
  - Σχεδιασμός οντολογιών για απαιτήσεις
  - Κατασκευή εργαλείων για το σχολιασμό (annotation) και την εισαγωγή αντικειμένων στις οντολογίες
  - Ανάπτυξη μοντέλου εξαγωγής προδιαγραφών
  - Εφαρμογή τεχνικών εξόρυξης για επαναχρησιμοποίηση απαιτήσεων (κανόνες συσχετίσεων, τεχνικές αντιστοίχισης)

# Επισκόπηση (2/2)

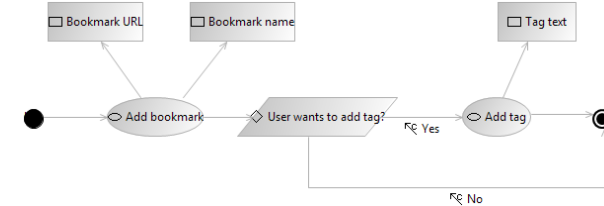
## Λειτουργικές Απαιτήσεις



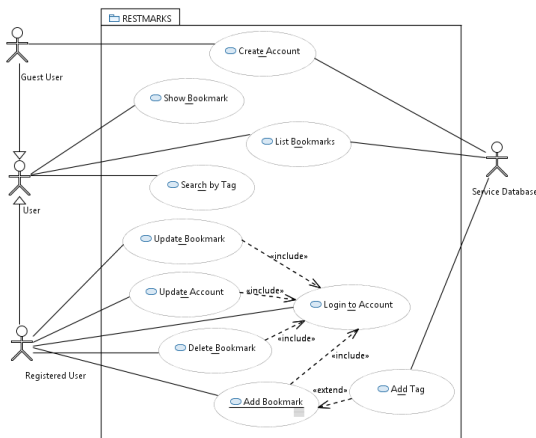
## Διαγράμματα Δραστηριοτήτων



## Γραφικά Σενάρια



## Διαγράμματα Σεναρίων Χρήσης



Στατική Οντολογία

Δυναμική Οντολογία

Προδιαγραφές

# Εξόρυξη Απαιτήσεων

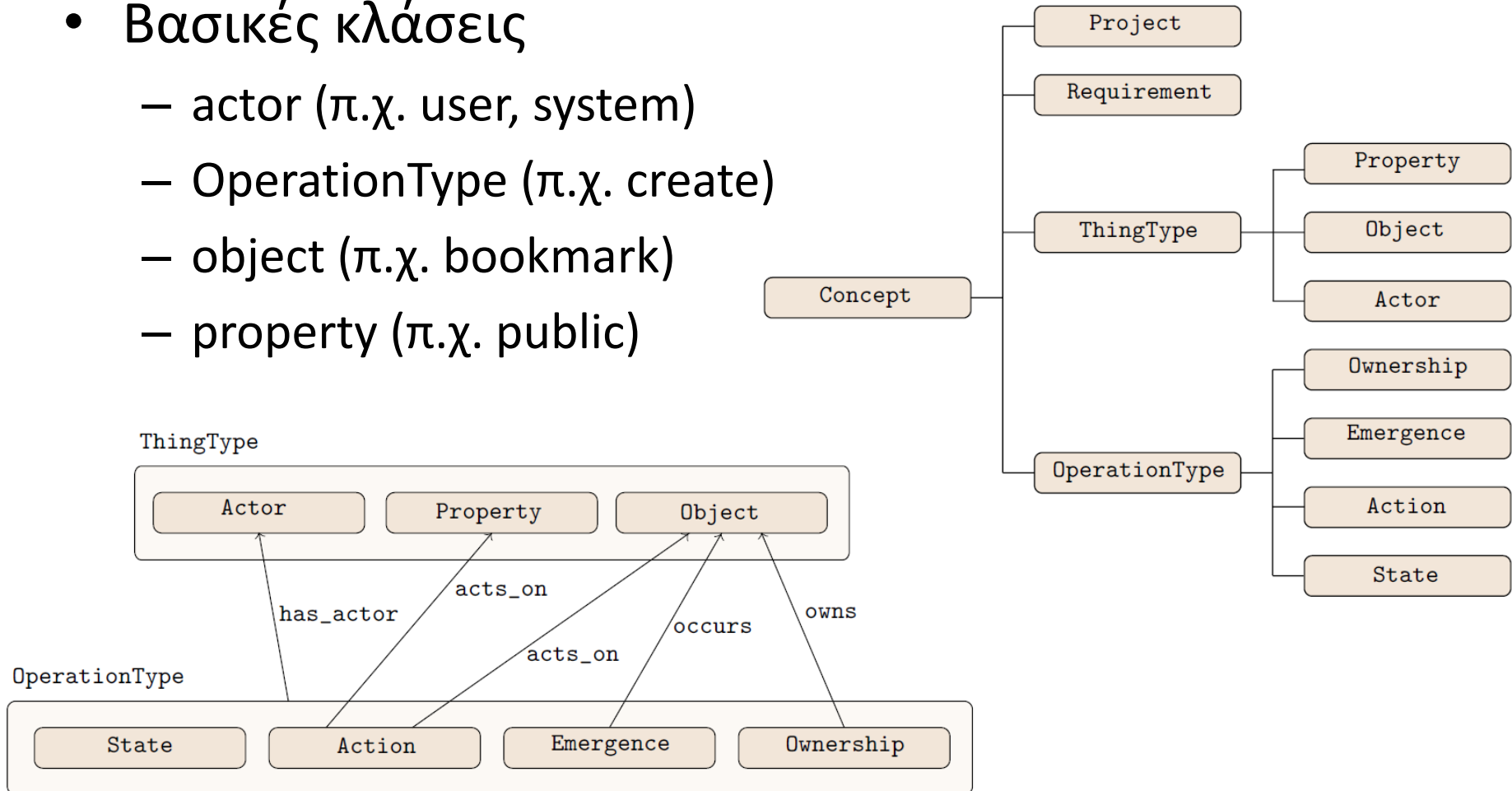
- Μοντελοποίηση Απαιτήσεων
- Εξόρυξη Λειτουργικών Απαιτήσεων & Διαγραμμάτων UML

# Μοντελοποίηση Στατικών Απαιτήσεων

- Απαιτήσεις σε φυσική γλώσσα, π.χ.:
  - “The user must be able to create an account”
- Εξαγωγή δραστών (actors), ενεργειών (actions), αντικειμένων (objects) και ιδιοτήτων (properties)
- Αποθήκευση σε οντολογία για εξαγωγή προδιαγραφών και επαναχρησιμοποίηση

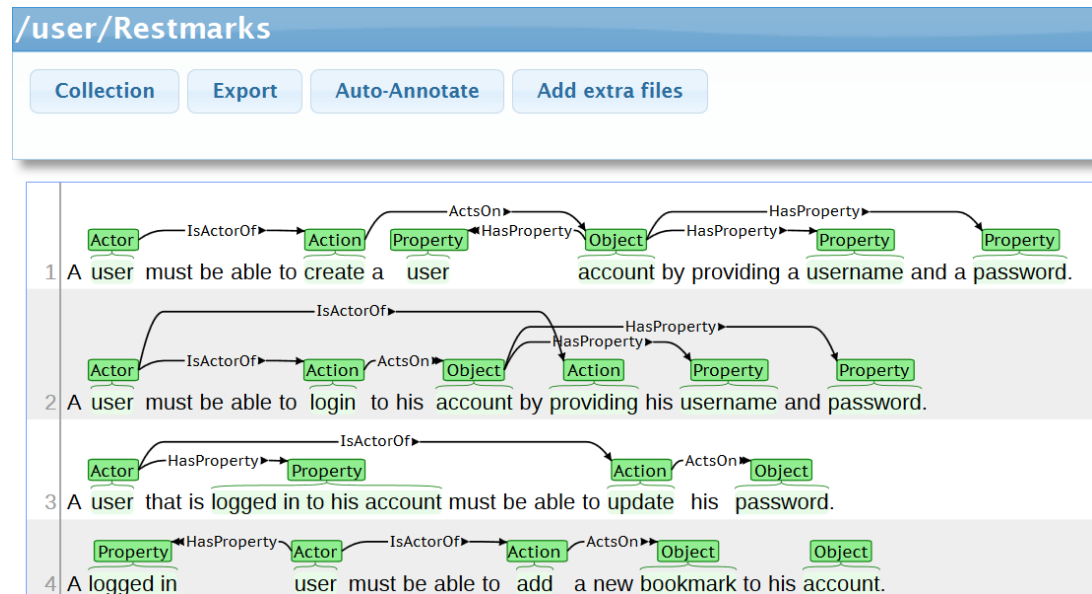
# Οντολογία για Στατική Όψη

- Βασικές κλάσεις
  - actor (π.χ. user, system)
  - OperationType (π.χ. create)
  - object (π.χ. bookmark)
  - property (π.χ. public)



# Εξαγωγή Σημασιολογικών Ρόλων

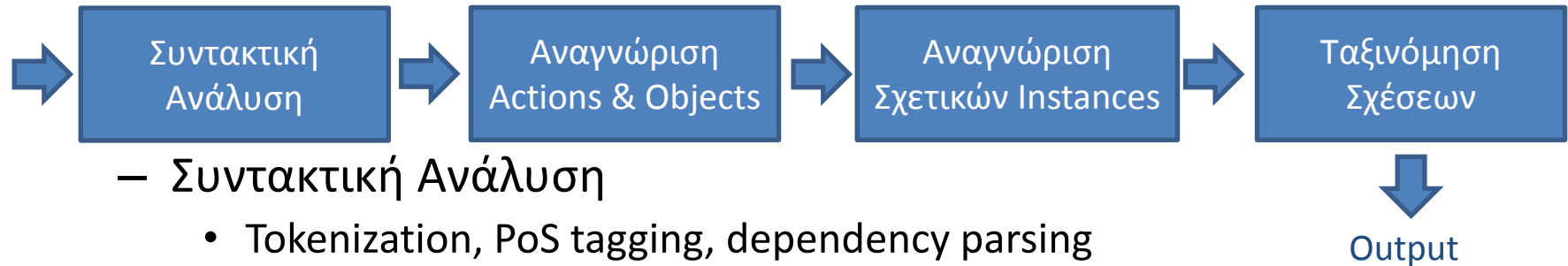
- Μεθοδολογία Σχολιασμών:
  - [Actor] IsActorOf [Action]
  - [Action] ActsOn [Object/Property]
  - [Actor/Object/Property] HasProperty [Property]
- Αυτόματος σχολιασμός
  - Σύνδεση με σημασιολογικό αναλυτή



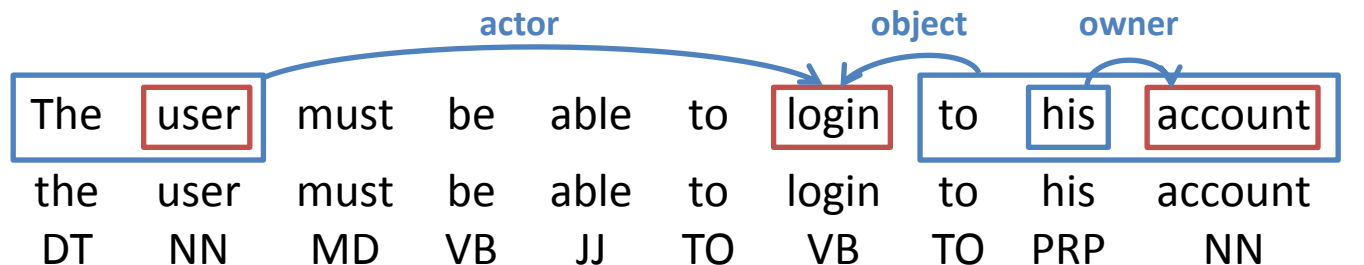


# Σημασιολογικός Αναλυτής NLP

- NLP parser
  - Αρχιτεκτονική pipeline

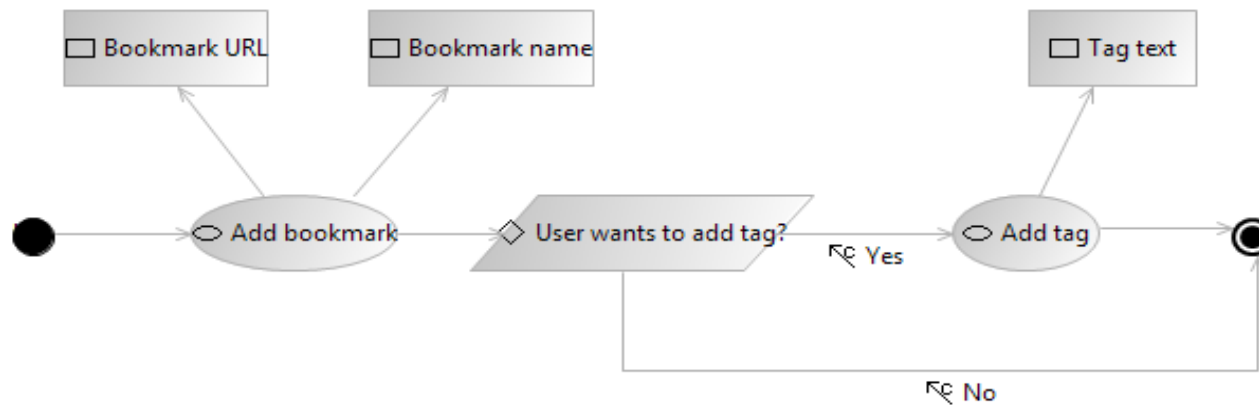


- Συντακτική Ανάλυση
  - Tokenization, PoS tagging, dependency parsing
- Σημασιολογική Ανάλυση
  - Αναγνώριση ενεργειών, αντικειμένων και σχετικών instances
  - Ταξινόμηση σχέσεων μεταξύ instances



# Μοντελοποίηση Δυναμικών Απαιτήσεων

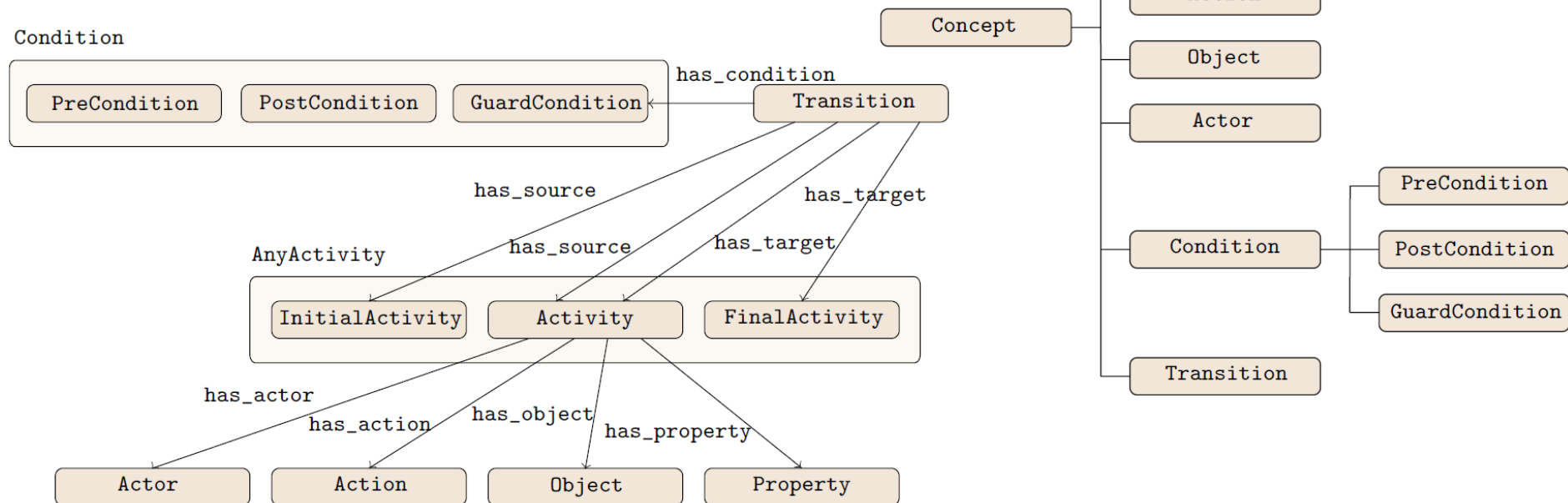
- Απαιτήσεις ως γραφικά σενάρια:



- Εξαγωγή δραστηριοτήτων (activities με actions και objects), ιδιοτήτων (properties), μεταβάσεων (transitions) και συνθηκών (conditions)
- Αποθήκευση σε οντολογία για εξαγωγή προδιαγραφών και επαναχρησιμοποίηση

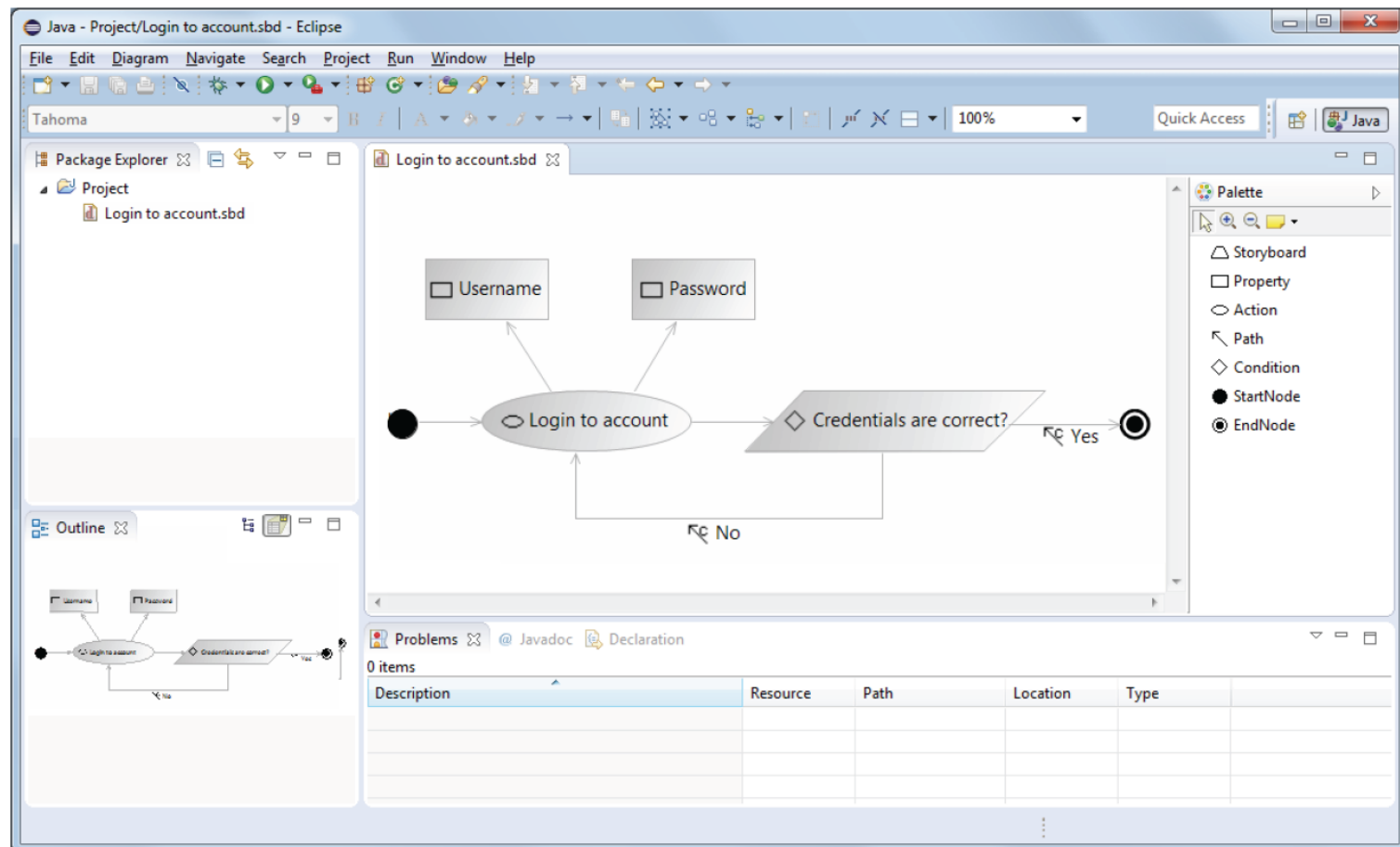
# Οντολογία για Δυναμική Όψη

- Βασικές κλάσεις
  - `Activity` (π.χ. create bookmark)
  - `Property` (π.χ. bookmark name)
  - `Transition` με/χωρίς `Condition` (π.χ. user wants to add bookmark?)



# Κατασκευή Δυναμικών Απαιτήσεων

- Εργαλείο Storyboard Creator



# Εξαγωγή Προδιαγραφών

- Σενάριο Υπηρεσίας RESTful
  - Restmarks
    - Κοινωνικό δίκτυο για σελιδοδείκτες (bookmarks)
    - Λειτουργικές απαιτήσεις
      - Π.χ. The user must be able to add a bookmark.
    - Γραφικά σενάρια
      - Π.χ. Create account, Search bookmark by tag
- ```
- !!Resource
Name: account
IsAlgorithmic: false
CRUDActivities: [C, R, U, D]
Properties: [username, password]
RelatedResources: [bookmark]
```
- ```
- !!Resource
Name: bookmark
IsAlgorithmic: false
CRUDActivities: [C, R, U, D]
Properties: [url, scope]
RelatedResources: [tag]
```
- ```
- !!Resource
Name: tag
IsAlgorithmic: false
CRUDActivities: [C, R, U, D]
Properties: [name, description]
RelatedResources: [tagSearch]
```
- ```
- !!Resource
Name: tagSearch
IsAlgorithmic: true
CRUDActivities: []
Properties: []
RelatedResources: []
```

# Εξόρυξη Απαιτήσεων

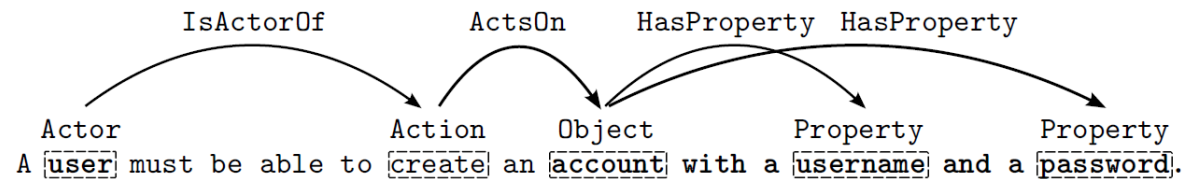
- Μοντελοποίηση Απαιτήσεων
- Εξόρυξη Λειτουργικών Απαιτήσεων & Διαγραμμάτων UML

# Εξόρυξη Λειτουργικών Απαιτήσεων (1/5)

- Σύνολο λειτουργικών απαιτήσεων
  - 325 απαιτήσεις από έργα λογισμικού και εργασίες φοιτητών
- Εξαγωγή κανόνων συσχετίσεων (association rule analysis)
  - Αξιολόγηση με βάση support και confidence
  - Αντιστοίχιση όρων μεταξύ απαιτήσεων
- Εφαρμογή κανόνων σε νέα έργα για προτάσεις απαιτήσεων

# Εξόρυξη Λειτουργικών Απαιτήσεων (2/5)

## 1) Σχολιασμένες λειτουργικές απαιτήσεις



## 2) Εξαγωγή κανόνων συσχέτισης

`user_IsActorOf_create`  
`create_ActsOn_account`  
`account_HasProperty_username`  
`account_HasProperty_password`

Association Rule	Support Confidence	
<code>[provide, product] → [system, provide]</code>	0.167	1.0
<code>[system, validate] → [user, login]</code>	0.1	1.0
<code>[user, buy] → [system, provide]</code>	0.1	1.0
<code>[administrator, add] → [administrator, delete]</code>	0.167	0.833
<code>[user, logout] → [user, login]</code>	0.167	0.833
<code>[user, add] → [user, delete]</code>	0.133	0.8
<code>[user, access] → [user, view]</code>	0.1	0.75
<code>[edit, product] → [add, product]</code>	0.1	0.75
<code>[administrator, delete] → [administrator, add]</code>	0.167	0.714
<code>[user, contact] → [user, search]</code>	0.133	0.5

## 3) Προτάσεις Απαιτήσεων

The user must be able to add a bookmark.

The user must be able to delete a bookmark.



# Εξόρυξη Λειτουργικών Απαιτήσεων (3/5)

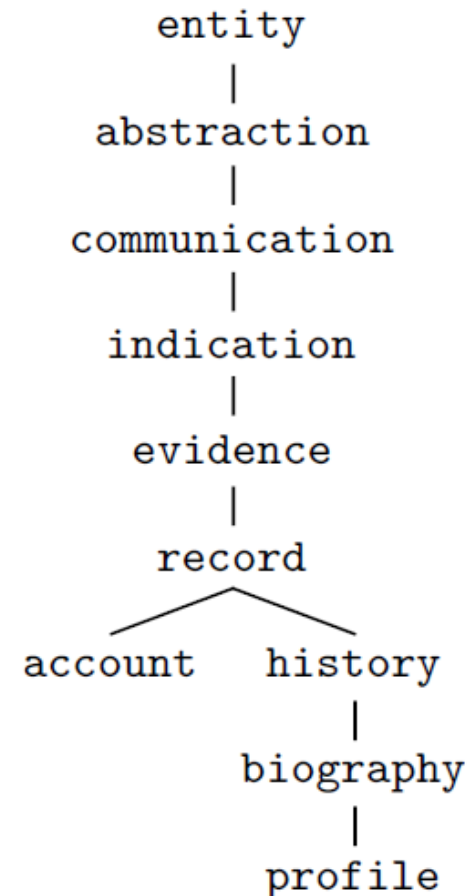
- Σημασιολογική ομοιότητα μεταξύ όρων

- Χρήση WordNet
- Ομοιότητα Lin

Για 2 κλάσεις  $C_1, C_2$ :

$$\text{sim}(C_1, C_2) = \frac{2 \cdot \log P(C_0)}{\log P(C_1) + \log P(C_2)}$$

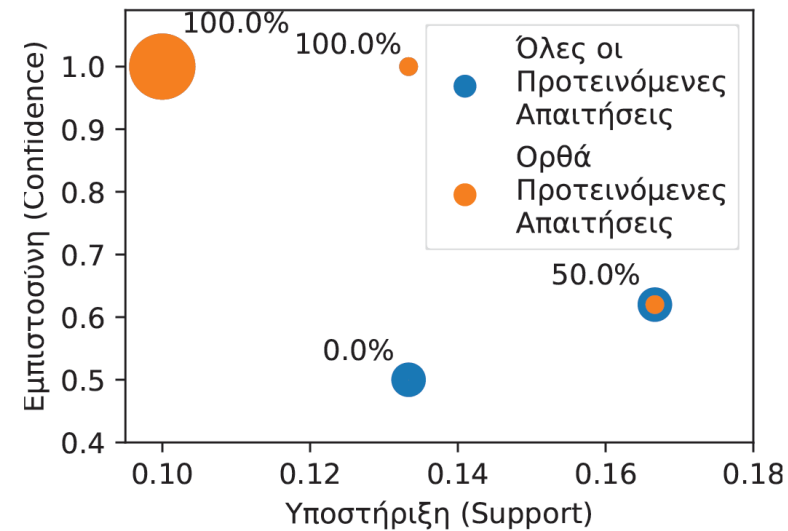
όπου  $C_0$  η πιο ειδική κλάση που τις περιγράφει



# Εξόρυξη Λειτουργικών Απαιτήσεων (4/5)

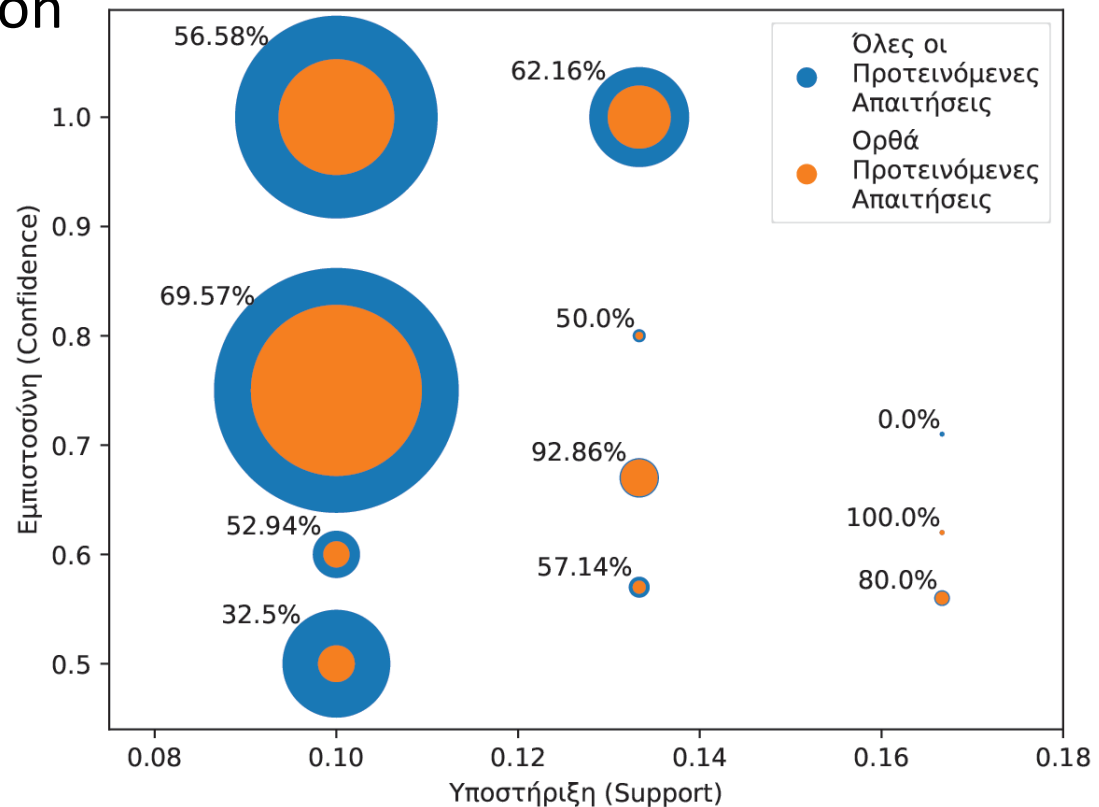
- Αξιολόγηση
  - Προτεινόμενες απαιτήσεις στο έργο Restmarks

Requirement	$\sigma$	c
+ The user must be able to edit bookmark.	0.138	1.0
+ The user must be able to view bookmark.	0.103	1.0
+ The user must be able to view account.	0.103	1.0
+ The user must be able to edit tag.	0.103	1.0
+ The user must be able to edit account.	0.103	1.0
+ The user must be able to logout account.	0.172	0.62
– The user must be able to contact account.	0.172	0.62
– The user must be able to contact bookmark.	0.138	0.5
– The user must be able to stop account.	0.138	0.5



# Εξόρυξη Λειτουργικών Απαιτήσεων (5/5)

- Αξιολόγηση
  - 30 έργα λογισμικού
  - 5-fold cross-validation
    - Επιλογή 5 έργων κάθε φορά (fold) για validation set

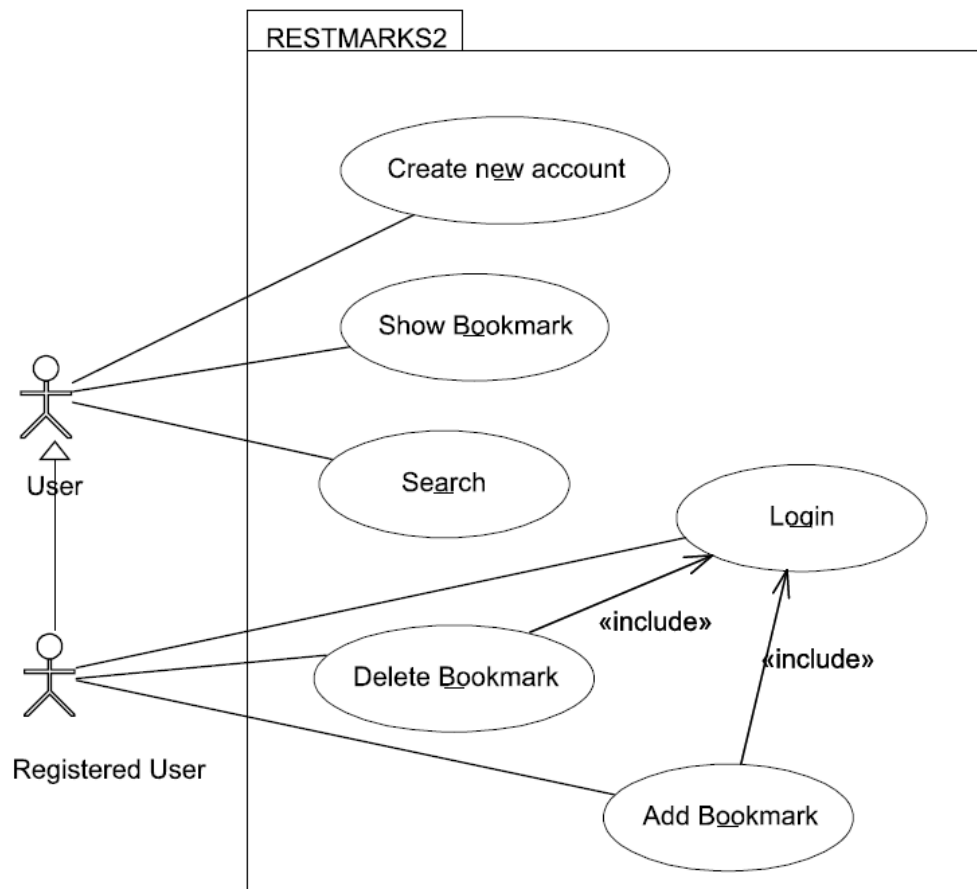


# Εξόρυξη Διαγραμμάτων UML (1/6)

- Σύνολο διαγραμμάτων
  - 65 διαγράμματα σεναρίων χρήσης (use case diagrams)
  - 72 διαγράμματα δραστηριοτήτων (activity diagrams)
- Πρόβλημα: εύρεση παρόμοιων διαγραμμάτων
- Αντιστοίχιση αντικειμένων μοντέλων
  - Σημασιολογική αντιστοίχιση συμβολοσειρών
    - Χρήση WordNet
  - Ροή διαγράμματος

# Εξόρυξη Διαγραμμάτων UML (2/6)

- Εξαγωγή μοντέλων διαγραμμάτων σεναρίων χρήσης



User

Registered User

Delete Bookmark

Show Bookmark

Add Bookmark

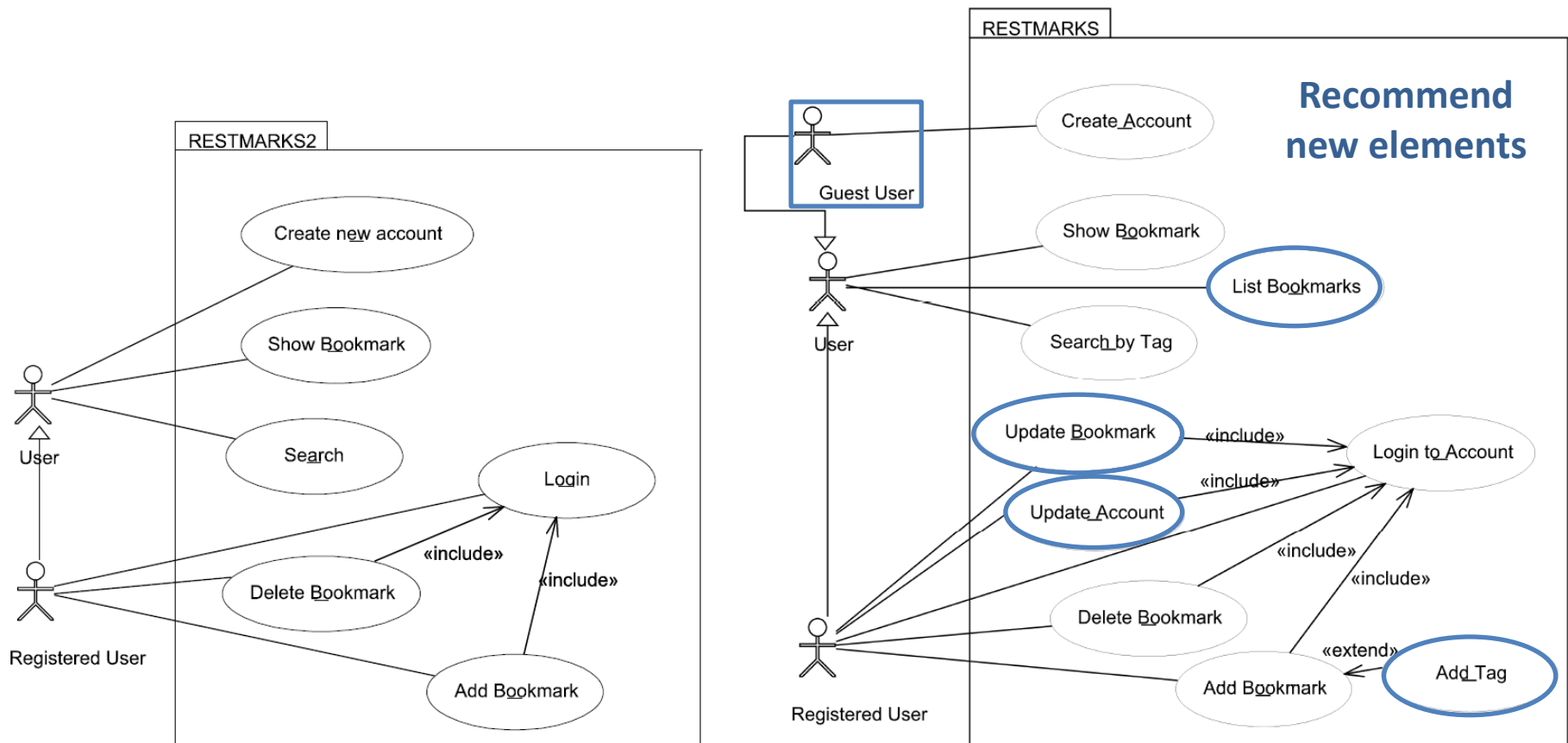
Create new account

Search

Login

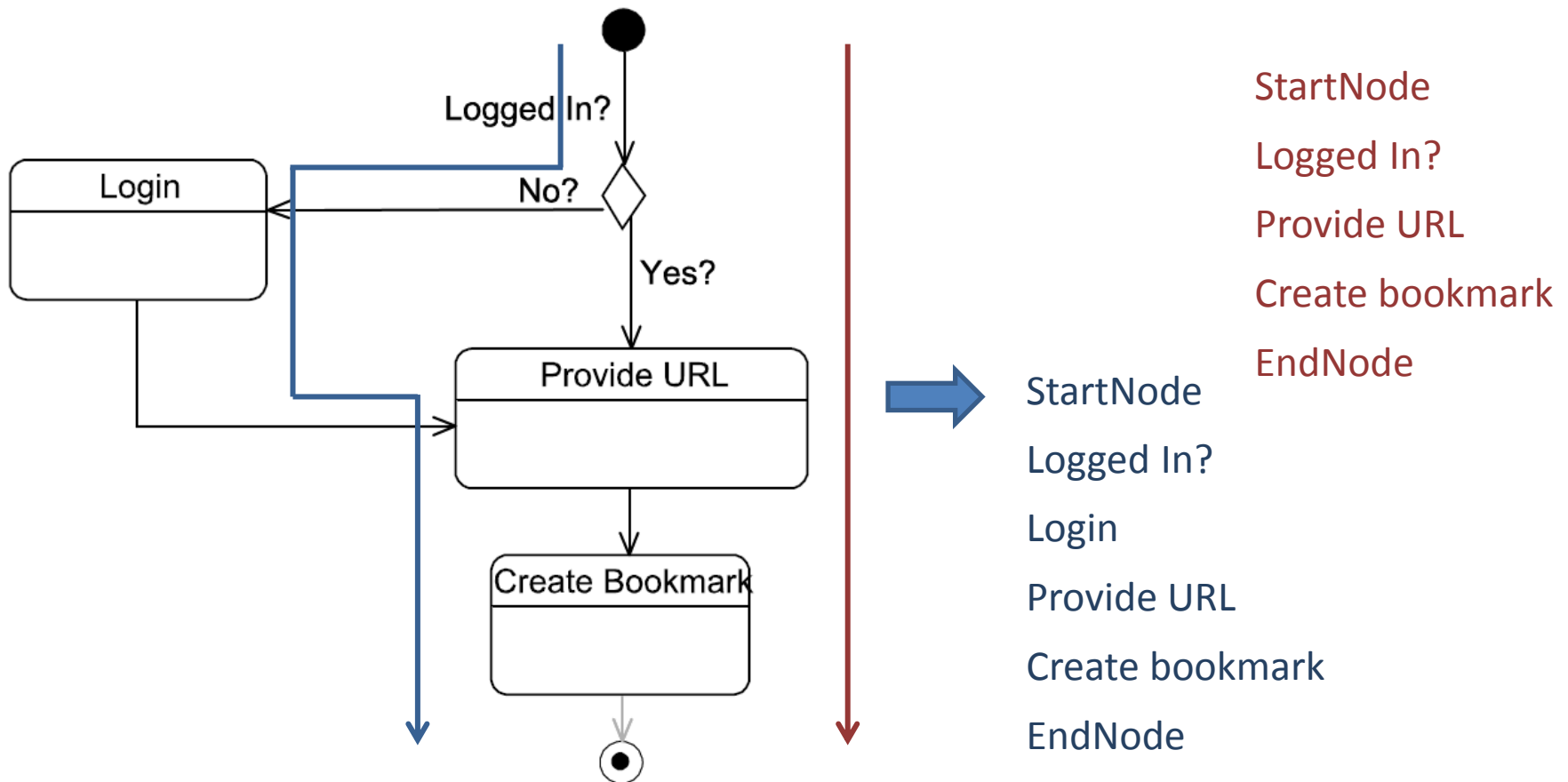
# Εξόρυξη Διαγραμμάτων UML (3/6)

- Αντιστοίχιση μοντέλων διαγραμμάτων σεναρίων χρήσης



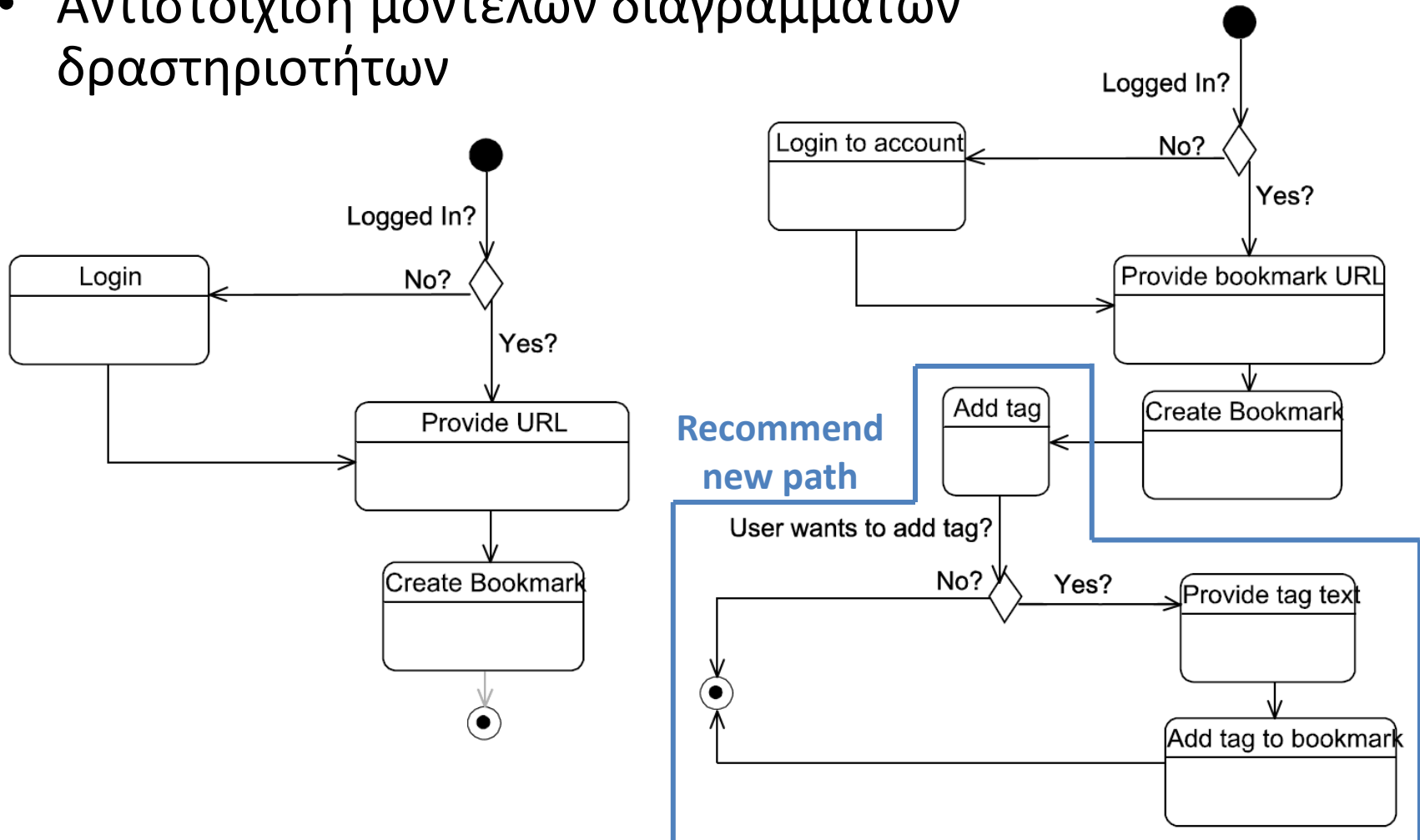
# Εξόρυξη Διαγραμμάτων UML (4/6)

- Εξαγωγή μοντέλων διαγραμμάτων δραστηριοτήτων



# Εξόρυξη Διαγραμμάτων UML (5/6)

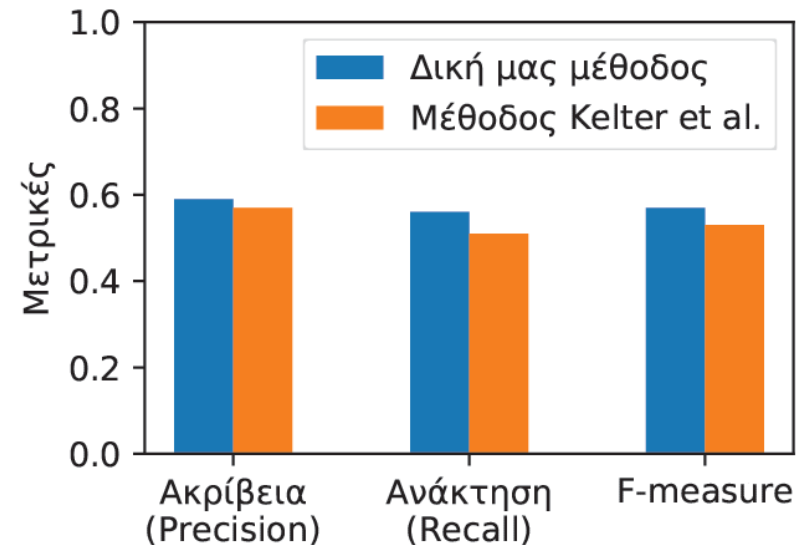
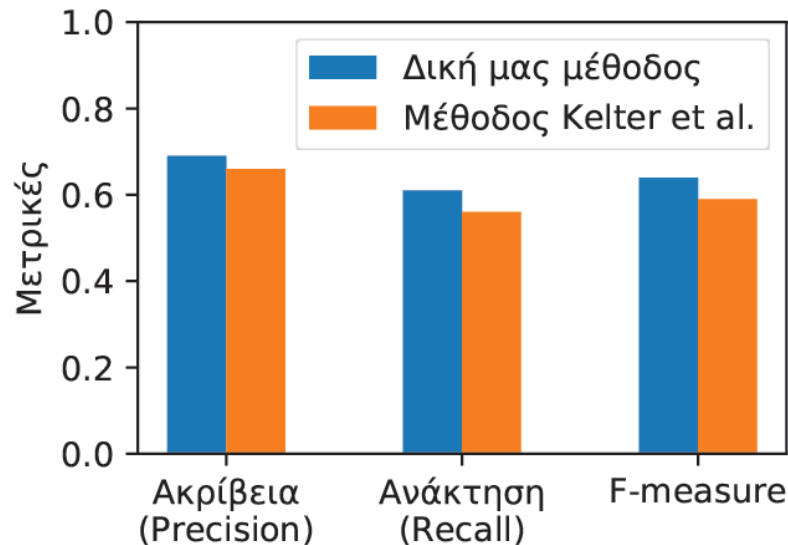
- Αντιστοίχιση μοντέλων διαγραμμάτων δραστηριοτήτων





# Εξόρυξη Διαγραμμάτων UML (6/6)

- Εύρεση παρόμοιων διαγραμμάτων
  - 65 use case diagrams → 2080 ζεύγη
  - 72 activity diagrams → 2556 ζεύγη



Σύγκριση με:

Udo Kelter, Jürgen Wehren, and Jorg Niere. "A Generic Difference Algorithm for UML Models." In: *Software Engineering*. Ed. by Peter Liggesmeyer, Klaus Pohl, and Michael Goedicke. Vol. 64. LNI. GI, 2005, pp. 105–116.

**Εξόρυξη Απαιτήσεων**

**Εξόρυξη Πηγαίου Κώδικα**

**Αξιολόγηση Ποιότητας**

# Επισκόπηση

- Ευρετηριοποίηση Κώδικα (Source Code Indexing)
- Δεδομένα από code repositories και κοινότητες QA
- Επαναχρησιμοποιήσιμες λύσεις σε διαφορετικά επίπεδα (έργων, τμημάτων, snippets)
- Βήματα επαναχρησιμοποίησης
  - Λήψη και αποθήκευση (ευρετηριοποίηση) κώδικα
  - Εξαγωγή/Δημιουργία ερωτήματος προγραμματιστή
  - Αναζήτηση και αντιστοίχιση ερωτήματος

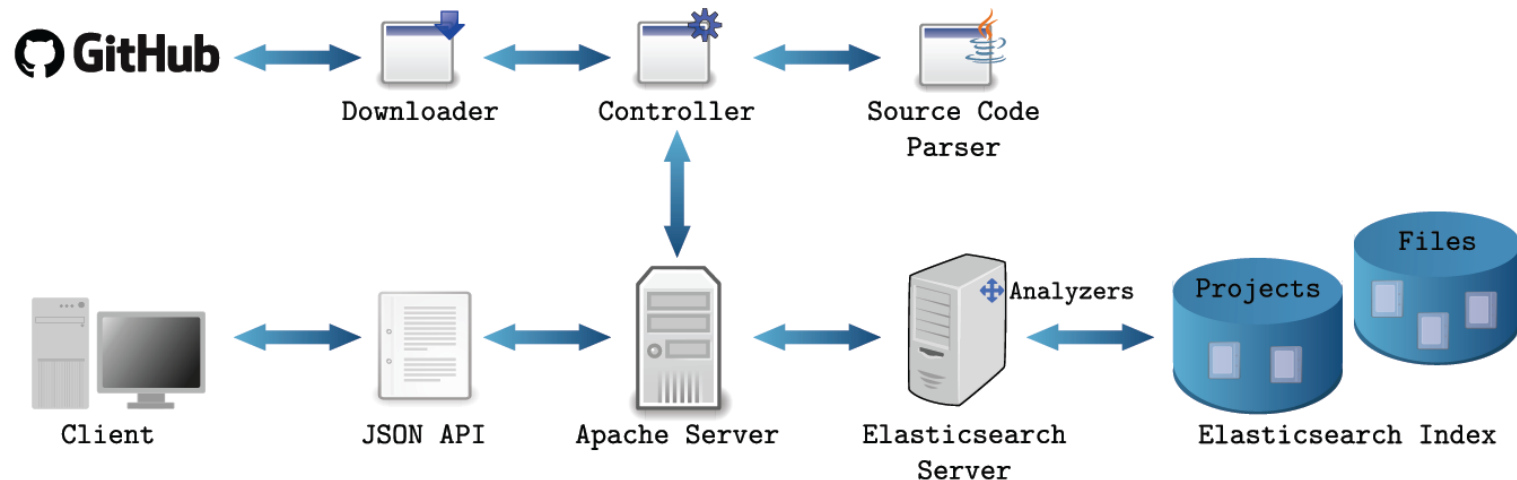


# Εξόρυξη Πηγαίου Κώδικα

- Ευρετηριοποίηση Κώδικα
- Εξόρυξη Κώδικα για Επαναχρησιμοποίηση
- Εξόρυξη API Snippets
- Βελτίωση Ερωταπαντήσεων Υλοποίησης

# Ευρετηριοποίηση Κώδικα

- Η Μηχανή Αναζήτησης Κώδικα AGORA
  - Elasticsearch Index που ενημερώνεται αυτόματα από το GitHub
    - ~3000 πιο δημοφιλή Java projects (με βάση stars/forks)
  - Εξαγωγή σύνταξης κώδικα (κλάσεις, μέθοδοι, παράμετροι, ...)
  - Ανοικτό API για σύνδεση με άλλα συστήματα



# Ερωτήματα της AGORA (1/5)

- Διαφορετικοί τύποι αναζήτησης
  - Simple Search
    - Αναζήτηση σε όλα τα πεδία, υποστηρίζει κανονικές εκφράσεις
  - Advanced Search
    - Αναζήτηση με βάση τη σύνταξη (π.χ. ονόματα κλάσεων, ονόματα και τύποι μεθόδων και παραμέτρων, κτλ.)
  - Snippet Search
    - Αναζήτηση για snippets με βάση την ομοιότητα κώδικα
  - Project Search
    - Αναζήτηση σε επίπεδο έργου (multi-class components)

# Ερωτήματα της AGORA (2/5)

- Παράδειγμα Simple Search

## AGORA Search Engine

static int editdistance

© Copyright AGORA 2017  
[API](#) | [Help](#) | [About](#) | [Disclaimer](#)

Found 953849 results in 0.314 seconds

[App/src/com/dozuki/ifixit/util/EditDistance.java](#)

Project: [iFixit/iFixitAndroid](#)

```
{  
    public static int editDistance(String s, String t) {  
        int m = s.length();  
        int n
```

[Download file](#) [See at GitHub](#)

[core/src/main/java/hudson/util/EditDistance.java](#)

Project: [kohsuke/jenkins](#)

```
) where n=a.length() and m=b.length().  
    */  
    public static int editDistance( String a, String
```

[Download file](#) [See at GitHub](#)

[hudson-core/src/main/java/hudson/util/EditDistance.java](#)

Project: [hudson/hudson-2.x](#)

```
=a.length() and m=b.length().  
    */  
    public static int editDistance( String a, String b
```

[Download file](#) [See at GitHub](#)

# Ερωτήματα της AGORA (3/5)

- Παράδειγμα Advanced Search

Class

Name  Interface

Extends  Implements

Imports  Package

Access   final  abstract

Variables

Variable 1

Name  Type

Access   final  abstract  static

Methods

Method 1

Name  Return type

Access   final  abstract  static

Parameters

Name  Type

Exceptions

Method 2

Name  Return type

Access   final  abstract  static

Parameters

Name  Type

Exceptions

Found 427656 results in 0.19 seconds

[src/dist/edu/umd/cloud9/util/StackOfInts.java](#)

Project: [jintool/Cloud9](#)

```
public class StackOfInts extends ArrayListOfInts{  
    public void push(int i);  
    public int pop();  
}
```

[Download file](#) [See at GitHub](#)

[core/src/main/java/org/modelmapper/internal/util/Stack.java](#)

Project: [jhalterman/modelmapper](#)

```
public class Stack extends ArrayList<T>{  
    public void push(T element);  
    public T pop();  
}
```

[Download file](#) [See at GitHub](#)

[java/Chapter 3/Question3\\_3/Stack.java](#)

Project: [gaylemcd/ctci](#)

```
public class Stack{  
    public boolean push(int v);  
    public int pop();  
}
```

[Download file](#) [See at GitHub](#)

[java/Chapter 3/Question3\\_2/StackWithMin2.java](#)

Project: [gaylemcd/ctci](#)

```
public class StackWithMin2 extends Stack<Integer>{  
    public void push(int value);  
    public Integer pop();  
}
```

[Download file](#) [See at GitHub](#)



# Ερωτήματα της AGORA (4/5)

- Παράδειγμα Snippet Search

```
DocumentBuilderFactory dbFactory =  
DocumentBuilderFactory.newInstance();  
DocumentBuilder dBuilder =  
dbFactory.newDocumentBuilder();  
Document doc = dBuilder.parse();
```

Found 23266 results in 0.01 seconds

[test/com/facebook/buck/apple/xcode/WorkspaceGeneratorTest.java](#)

Project: [facebook/buck](#)

```
() ;  
DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();  
Document workspace = dBuilder.parse
```

[Download file](#) [See at GitHub](#)

[agent/jvm-spring/src/test/java/org/jolokia/jvmagent/spring/config/SpringConfigTest.java](#)

Project: [rhuss/jolokia](#)

```
{  
DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();  
DocumentBuilder dBuilder
```

[Download file](#) [See at GitHub](#)

[org/openhab/binding/insteonplm/internal/device/DeviceCategoryLoader.java](#)

Project: [openhab/openhab](#)

```
= DocumentBuilderFactory.newInstance();  
DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();  
Document doc
```

[Download file](#) [See at GitHub](#)

[org/openhab/binding/insteonplm/internal/device/XMLDeviceFeatureReader.java](#)

Project: [openhab/openhab](#)

```
{  
DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();  
DocumentBuilder dBuilder
```

[Download file](#) [See at GitHub](#)

# Ερωτήματα της AGORA (5/5)

- Παράδειγμα Project Search

Classes

Class

Name  Interface

Extends  Implements

Imports  Package

Class

Name  Interface

Extends  Implements

Imports  Package

Class

Name  Interface

Extends  Implements

Imports  Package

Class

Name  Interface

Extends  Implements

Imports  Package

+ -

Found 1527 results in 0.125 seconds

[multibit](#)

User: keepkey

[See at GitHub](#)

[infinitest](#)

User: infinitest

[See at GitHub](#)

[infinispan](#)

User: Infinispan

[See at GitHub](#)

[citrus](#)

User: webx

[See at GitHub](#)

[neo4j](#)

User: neo4j

[See at GitHub](#)

[astrid](#)

User: todoroo

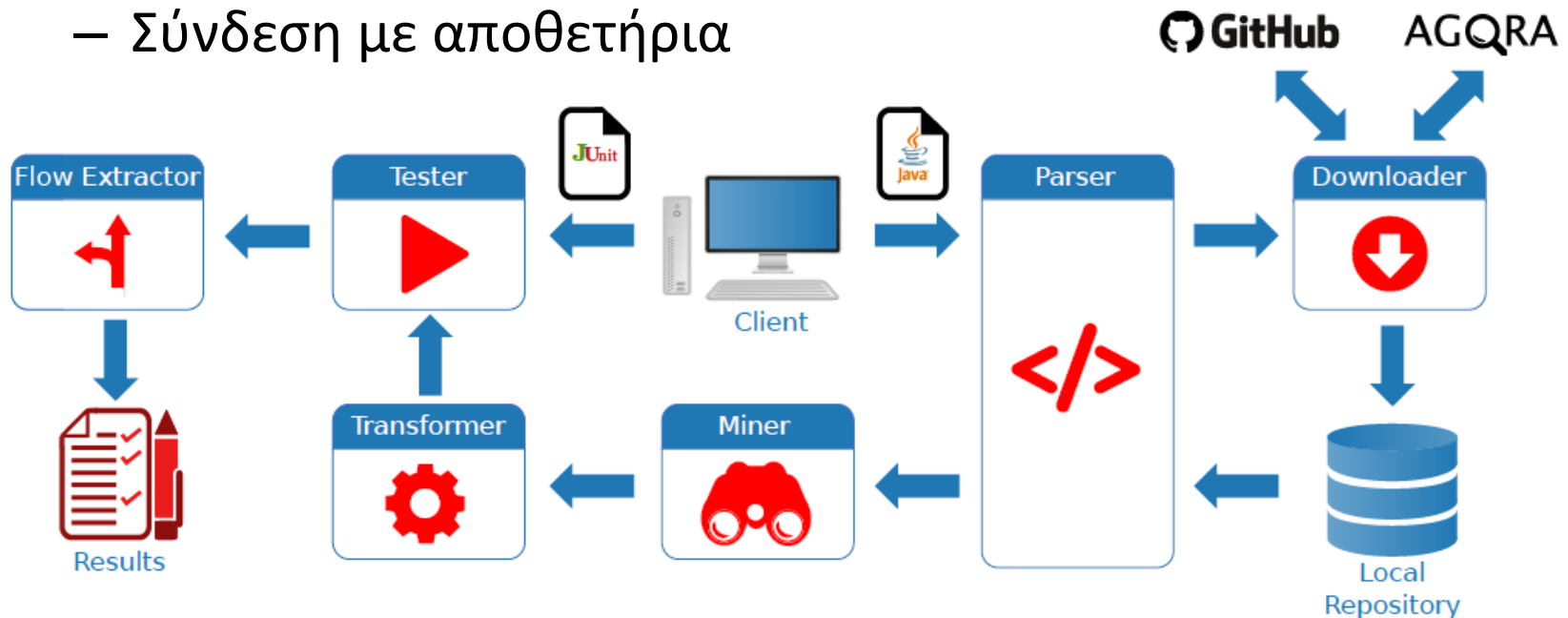
[See at GitHub](#)

# Εξόρυξη Πηγαίου Κώδικα

- Ευρετηριοποίηση Κώδικα
- Εξόρυξη Κώδικα για Επαναχρησιμοποίηση
- Εξόρυξη API Snippets
- Βελτίωση Ερωταπαντήσεων Υλοποίησης

# Εξόρυξη Κώδικα για Επαναχρησιμοποίηση

- Σύστημα Επαναχρησιμοποίησης Κώδικα Mantissa
  - Σύνταξη ερωτήματος ως interface
  - Εξόρυξη με βάση τη σύνταξη κώδικα (κλάσεις, μέθοδοι, ...)
  - Αυτόματος μετασχηματισμός κώδικα
  - Αξιολόγηση λειτουργικότητας (με περιπτώσεις ελέγχου)
  - Σύνδεση με αποθετήρια



# Μηχανισμός Αντιστοίχισης του Mantissa (1/2)

- Αντιστοίχιση μεθόδων
  - Σύγκριση ονομάτων, τύπων επιστροφής, παραμέτρων
  - Κατασκευή διανύσματος υπογραφής

π.χ. *[Stack, [push, void, Object], [pop, Object]]*

*[MyStack, [pushObject, bool, Object], [popObject, Object]]*

---

```
public class Stack {
    public void push(Object o){}
    public Object pop(){}
```

---



---

```
public class MyStack {
    public bool pushObject(Object o){}
    public Object popObject(){}
```

---

Ομοιότητα μεταξύ λιστών/διανυσμάτων

$$Tanimoto(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}|^2 + |\vec{B}|^2 - \vec{A} \cdot \vec{B}}$$

Ομοιότητα μεταξύ συμβολοσειρών (strings)

Διαχωρισμένα σε CamelCase

$$Jaccard(S, T) = \frac{S \cap T}{S \cup T}$$

# Μηχανισμός Αντιστοίχισης του Mantissa (2/2)

- Αντιστοίχιση μεθόδων
  - Σύγκριση ονομάτων, τύπων επιστροφής, παραμέτρων
  - Κατασκευή διανύσματος υπογραφής

π.χ.  $[Stack, [push, void, Object], [pop, Object]]$

$[MyStack, [pushObject, bool, Object], [popObject, Object]]$

---

```
public class Stack {
    public void push(Object o){}
    public Object pop(){}
```

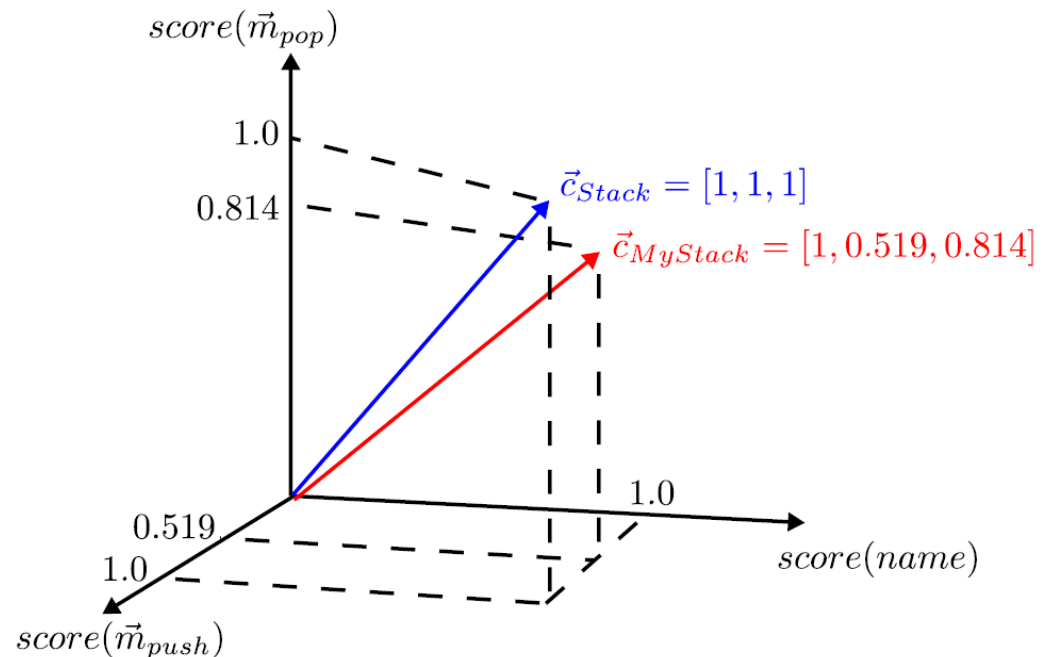
---



---

```
public class MyStack {
    public bool pushObject(Object o){}
    public Object popObject(){}
```

---



# Επιπλέον Χαρακτηριστικά του Mantissa (1/2)

- Αξιολόγηση λειτουργικότητας αποτελεσμάτων
  - Παροχή περιπτώσεων ελέγχου από τον προγραμματιστή
- Μετασχηματισμοί Κώδικα
  - Παράδειγμα `public void setAddress(String address)`

```
public boolean setTheAddress(String address, String postCode) {  
    this.address = address;  
    if (postCode != null) {  
        this.address += ", " + postCode;  
        return true;  
    }  
    return false;  
}
```

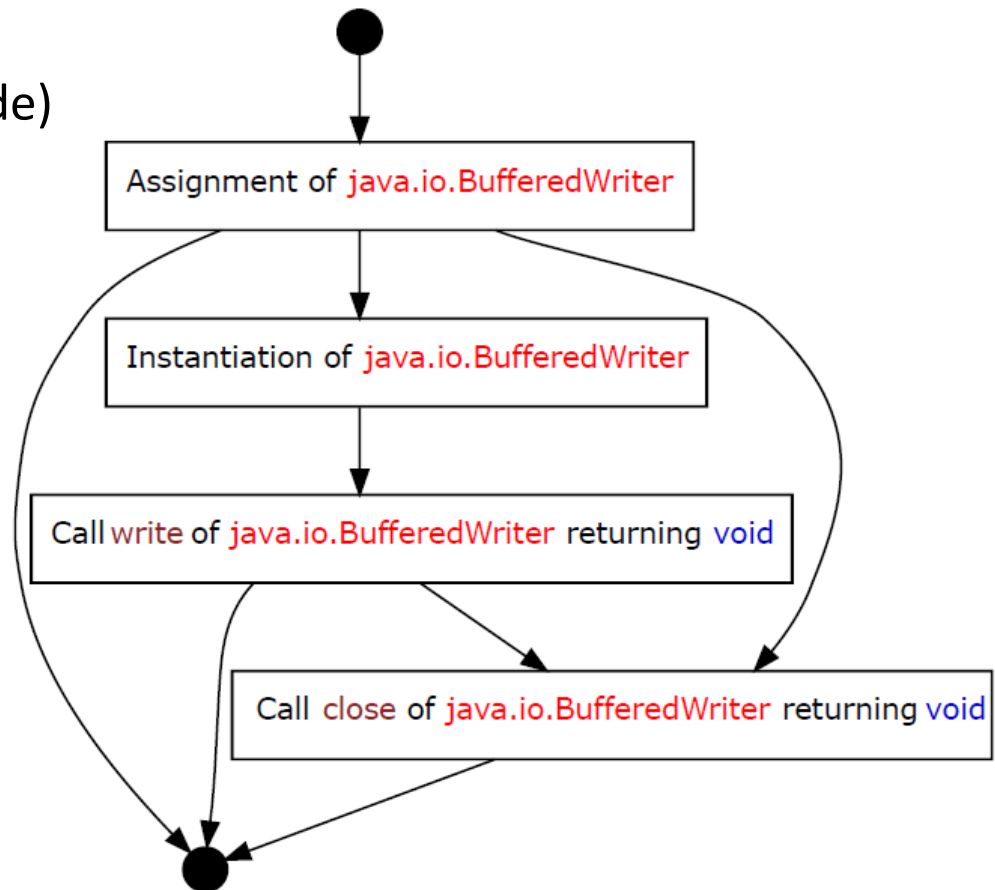


```
public void setAddress(String address) {  
    String postCode = null;  
    this.address = address;  
    if (postCode != null) {  
        this.address += ", " + postCode;  
        return;  
    }  
    return;  
}
```

# Επιπλέον Χαρακτηριστικά του Mantissa (2/2)



- Εξαγωγή ροής κώδικα
  - Επισήμανση τύπων της Java και αναζήτηση εξωτερικών εξαρτήσεων (link to grepcode)

```
BufferedWriter out = null;
try {
    out = new BufferedWriter(
        new FileWriter(filePath));
    out.write(content);
} finally {
    if (out != null)
        out.close();
}
```





# Γραφική Διεπαφή του Mantis

### Code

```
public class Stack{
    public void push(Object o){}
    public Object pop(){}
}
```


### Test

```
import static org.junit.Assert.assertEquals;



import org.junit.Test;
import java.lang.Object;
import java.lang.reflect.*;

public class StackTest{
    ...
}
```

**Source**

AGORA 

**Search**

### Results

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

Score: 0.96   Physical Lines: 27   Compilable: Yes   Test cases: Passed 0 of 1 tests

[Show Result File](#) [Show Original File](#) [Show Control Flow](#) [Show Imports](#) 0

```
package org.jbox2d.particle;

public class StackQueue<T> {

    private T[] m_buffer;
    private int m_front;
    private int m_back;
    private int m_end;

    public StackQueue() {}

    public void reset(T[] buffer) {
        m_buffer = buffer;
        m_front = 0;
        m_back = 0;
        m_end = buffer.length;
    }
    ...
}
```

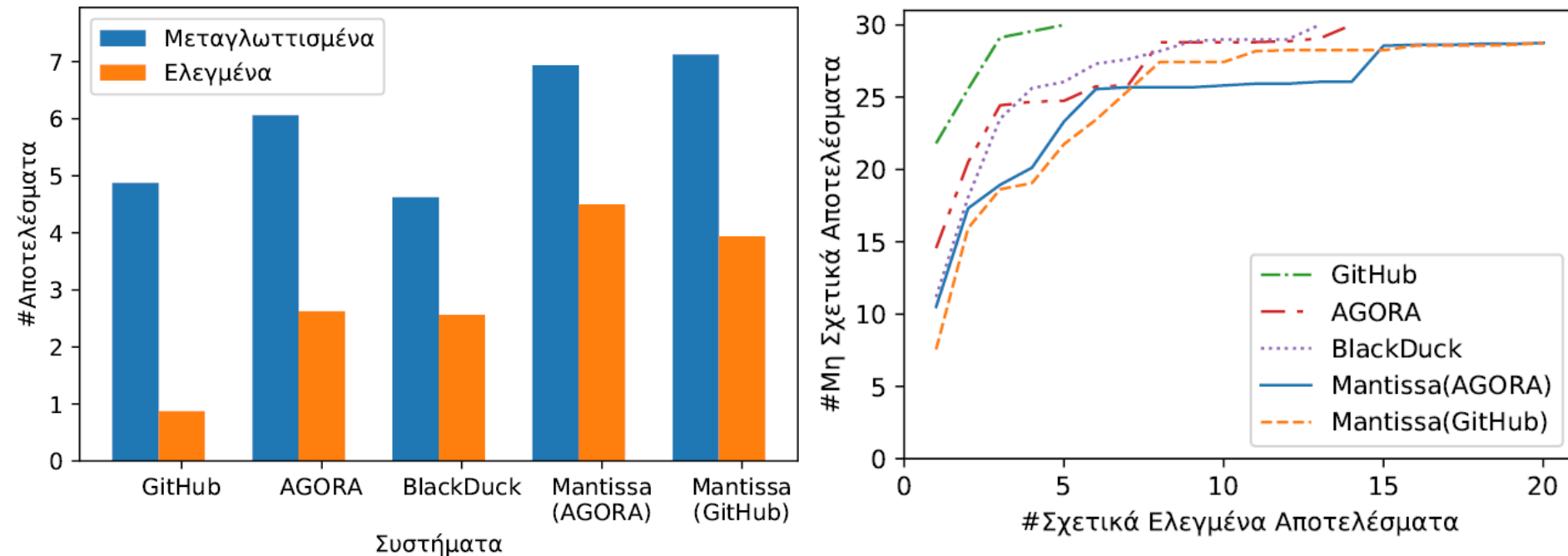
# Αξιολόγηση του Mantissa (1/3)

- Μεθοδολογία αξιολόγησης
  - Σενάριο επαναχρησιμοποίησης τμημάτων κώδικα
  - Ερωτήματα για τμήματα κώδικα
  - Περιπτώσεις ελέγχου

Κλάση	Μέθοδοι
Account	deposit, withdraw, getBalance
Article	setId, getId, setName, getName, setPrice, getPrice
Calculator	add, subtract, divide, multiply
ComplexNumber	ComplexNumber, add, getRealPart, getImaginaryPart
CreditCardValidator	isValid
Customer	setAddress, getAddress
Gcd	gcd
Md5	md5Sum
Mortgage	setRate, setPrincipal, setYears, getMonthlyPayment
Movie	Movie, getTitle
Prime	checkPrime
Sort	sort
Spreadsheet	put, get
Stack	push, pop
Stack2	pushObject, popObject
Tokenizer	tokenize

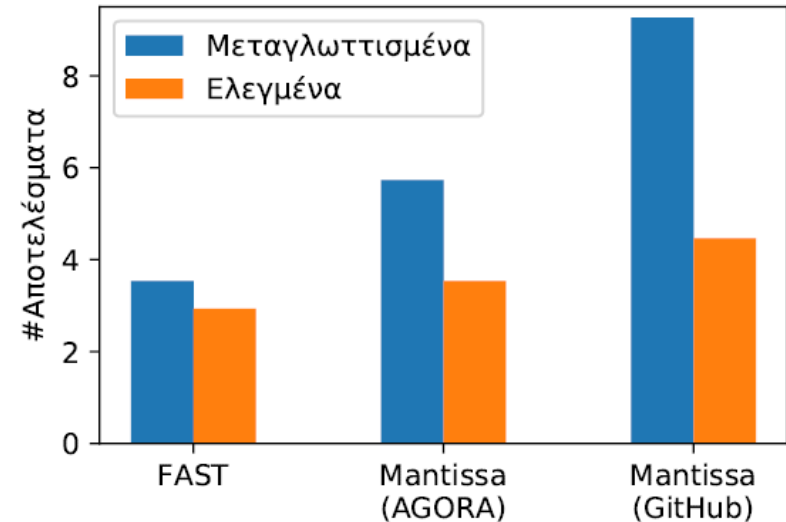
# Αξιολόγηση του Mantissa (2/3)

- Αξιολόγηση με Μηχανές Αναζήτησης Κώδικα
  - Αριθμός αποτελεσμάτων που μεταγλωττίζονται
  - Αριθμός αποτελεσμάτων που περνάνε τον έλεγχο
  - Μήκος αναζήτησης (search length)



# Αξιολόγηση του Mantissa (3/3)

- Αξιολόγηση με Συστήματα Επαναχρησιμοποίησης
  - Συστήματα FAST και Code Conjurer



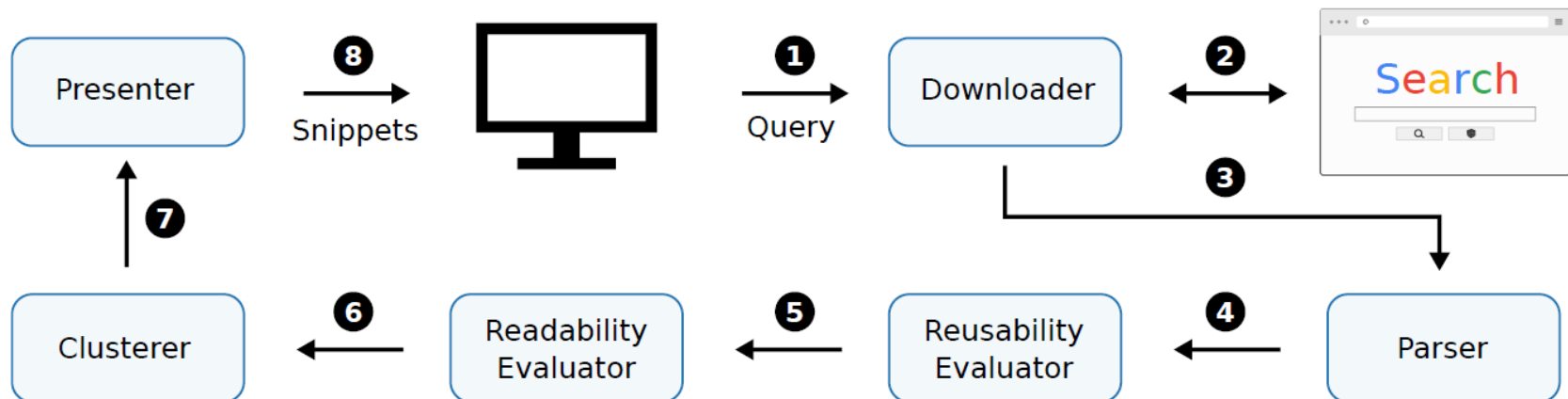
Ερώτημα	Code Conjurer				Mantissa			
	Interface		Adaptation		AGORA		GitHub	
	Ελεγμένα	Χρόνος	Ελεγμένα	Χρόνος	Ελεγμένα	Χρόνος	Ελεγμένα	Χρόνος
Calculator	1	19s	22	20h 24m	0	25s	0	2m 35s
ComplexNumber	0	3s	30	1m 19s	4	35s	7	2m 8s
Matrix	2	23s	26	5m 25s	0	31s	0	2m 51s
MortgageCalculator	0	4s	15	3h 19m	0	31s	0	2m 18s
Spreadsheet	0	3s	4	15h 13m	1	23s	1	2m 28s
Stack	30	26m	30	18h 23m	14	30s	10	2m 53s
Μέσοι όροι	5.500	4m 28s	21.167	9h 34m 17s	3.167	29s	3.000	2m 32s

# Εξόρυξη Πηγαίου Κώδικα

- Ευρετηριοποίηση Κώδικα
- Εξόρυξη Κώδικα για Επαναχρησιμοποίηση
- Εξόρυξη API Snippets
- Βελτίωση Ερωταπαντήσεων Υλοποίησης

# Εξόρυξη API Snippets

- Σύστημα Επαναχρησιμοποίησης Κώδικα CodeCatch
  - Σύνταξη ερωτήματος σε φυσική γλώσσα
    - Λήψη αποτελεσμάτων της μηχανής αναζήτησης Google
  - Αξιολόγηση επαναχρησιμοποιησιμότητας και αναγνωσιμότητας
  - Εξόρυξη σε επίπεδο κλήσεων API και ομαδοποίηση υλοποιήσεων



# Λειτουργία του CodeCatch (1/3)

- Εξαγωγή κλήσεων μεθόδων (API calls)
- Έλεγχος αναγνωσιμότητας κώδικα (Buse & Weimer\*)
- **Αξιολόγηση επαναχρησιμοποιησιμότητας**
  - Συχνότητα εμφάνισης κλήσεων API στο ευρετήριο της AGORA

Παράδειγμα για ένα ερώτημα “How to read a CSV file”

```
String line = "";  
BufferedReader br = null;  
try {  
    br = new BufferedReader(new FileReader("test.csv"));  
    while((line = br.readLine()) != null) {  
        String[] data = line.split(",");  
    }  
    br.close();  
} catch (Exception e) {  
    System.err.println("CSV file cannot be read: " + e);  
}
```

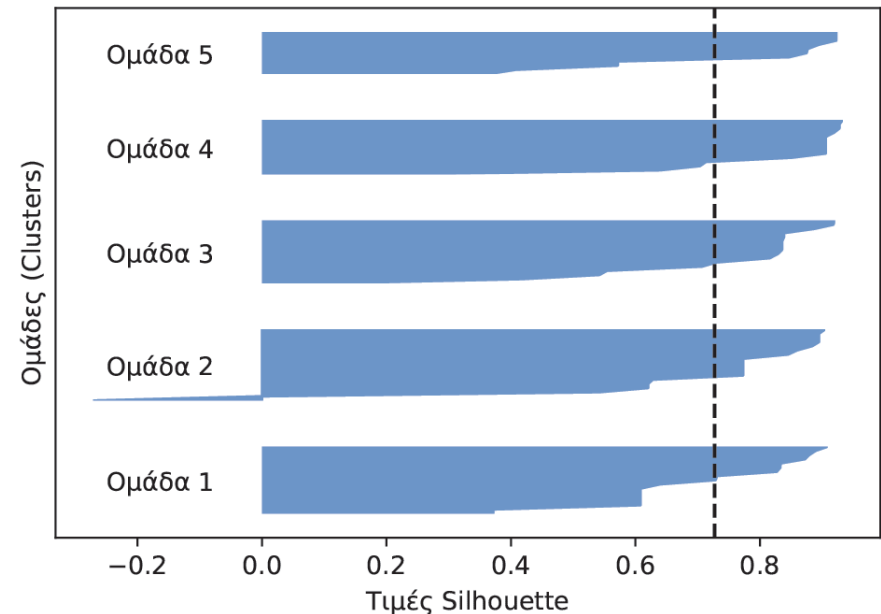
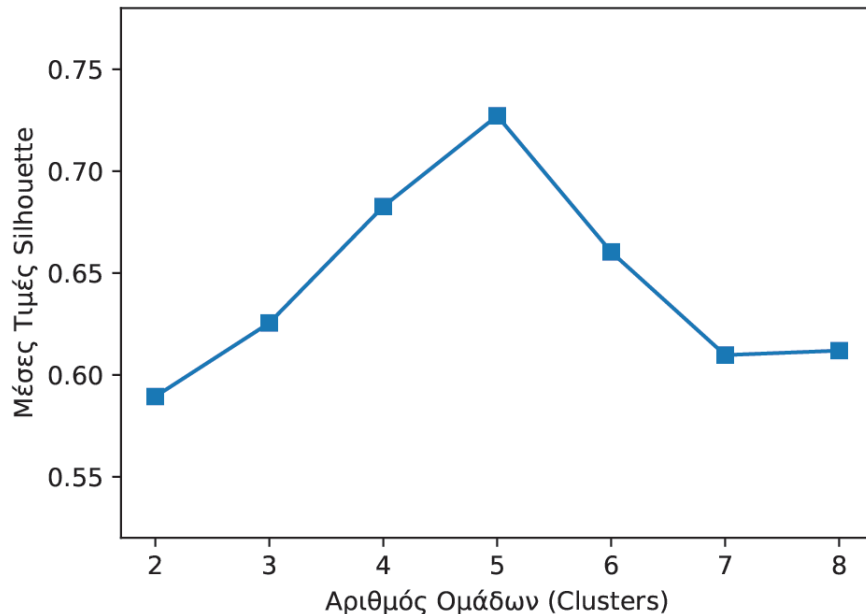
```
Scanner scanner = null;  
try{  
    scanner = new Scanner(new File("test.csv"));  
    scanner.useDelimiter(",");  
    while(scanner.hasNext()) {  
        System.out.print(scanner.next() + " ");  
    }  
    scanner.close();  
} catch (Exception e) {  
    System.err.println("CSV file cannot be read: " + e);  
}
```

\* Raymond P. L. Buse and Westley R. Weimer. “Learning a Metric for Code Readability”. In: IEEE Trans. Softw. Eng. 36.4 (2010), pp. 546–558.

# Λειτουργία του CodeCatch (2/3)

- Ομαδοποίηση snippets με βάση τις κλήσεις API
  - Κατασκευή εγγράφων και ομαδοποίηση με βάση το cosine similarity
  - Χρήση K-Means με 2, 3, ..., 8 ομάδες
  - Επιλογή αριθμού ομάδων (clusters) με βάση το Silhouette

Παράδειγμα για ένα ερώτημα “How to read a CSV file”





# Λειτουργία του CodeCatch (3/3)

- Κατάταξη με βάση την επαναχρησιμοποιησιμότητα
- Παρουσίαση Αποτελεσμάτων (για ένα ερώτημα “How to read a CSV file”)

## Cluster 1

Score: 35.61%

Imports API calls

- ✓ java.io.BufferedReader
- ✓ java.io.FileReader
- ✓ java.util.ArrayList
- ✓ java.io.IOException
- ✓ java.util.List

Explore Cluster 1

## Cluster 2

Score: 30.21%

Imports API calls

- ✓ java.util.Scanner
- ✓ java.util.List
- ✓ java.io.File
- ✓ java.util.ArrayList
- ✓ java.io.FileNotFoundException

Explore Cluster 2

## Cluster 3

Score: 19.37%

Imports API calls

- ✓ AM.\_\_init\_\_
- ✓ FileWriter.\_\_init\_\_
- ✓ CSV.\_\_init\_\_
- ✓ CSV.hasNext
- ✓ CSV.next

Explore Cluster 3

API Score: 0.29

Centroid Distance: 0.10

Readability: Low

Position: 6

Order in page: 4

Number of API calls: 4

Lines of Code: 14

Show invocations - Go to snippet webpage

```
public class InsertValuesIntoTestDb {

    public static void main(String[] args) throws Exception {
        String splitBy = ",";
        BufferedReader br = new BufferedReader(new FileReader("test.csv"));
        while((line = br.readLine()) != null) {
            String[] b = line.split(splitBy);
            System.out.println(b[0]);
        }
        br.close();
    }
}
```

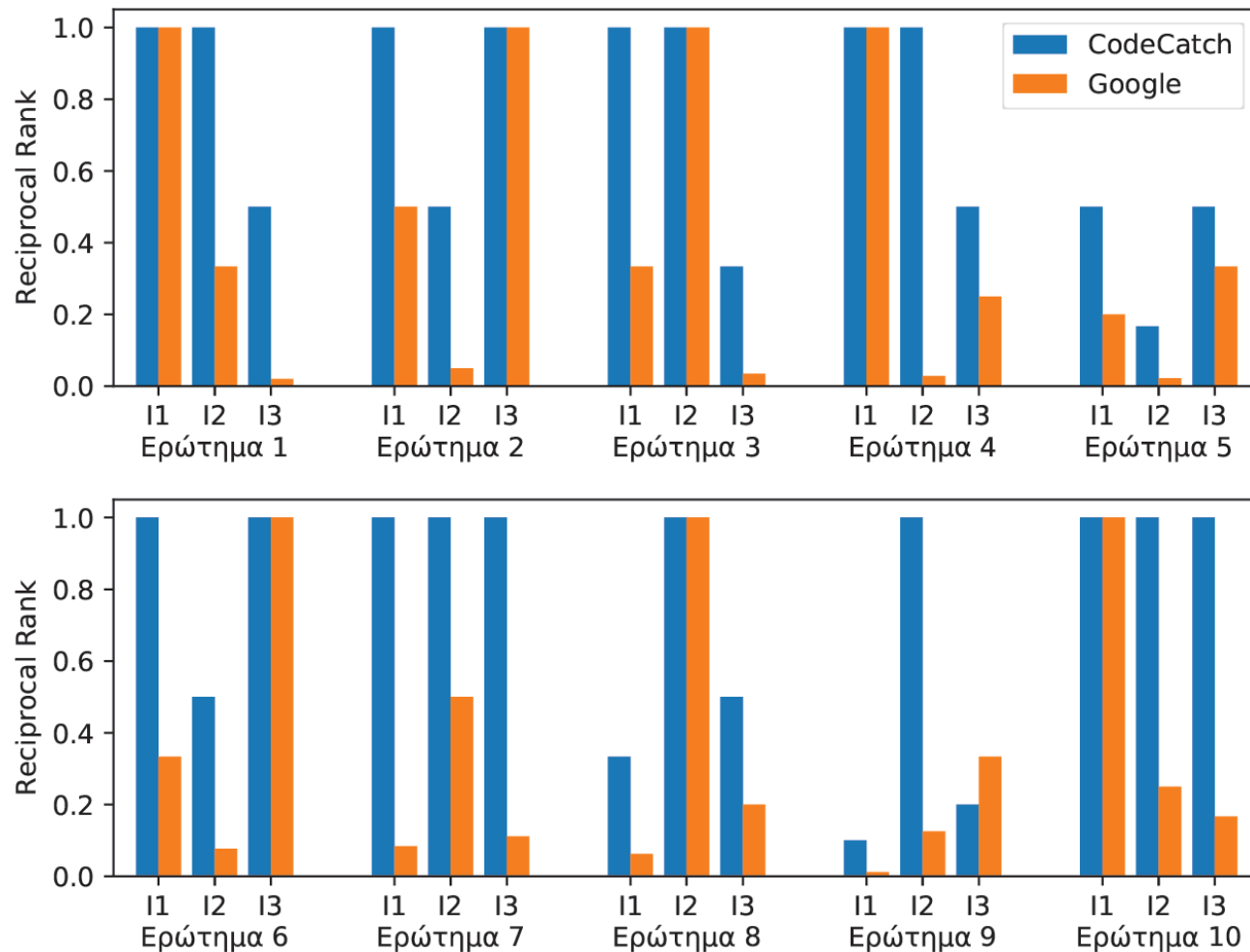
# Αξιολόγηση του CodeCatch (1/2)

- Αξιολόγηση με τη Μηχανή Αναζήτησης Google
  - Άξονας αξιολόγησης:
    - ευκολία εύρεσης διαφορετικών υλοποιήσεων
  - Μέτρηση Reciprocal Rank για κάθε υλοποίηση ξεχωριστά

A/A	Ερώτημα	# Ομάδες	# Snippets
1	How to read CSV file	5	76
2	How to generate MD5 hash code	5	65
3	How to send packet via UDP	5	34
4	How to split string	4	22
6	How to upload file to FTP	4	31
5	How to send email	5	79
7	How to initialize thread	6	51
8	How to connect to a JDBC database	5	42
9	How to read ZIP archive	6	82
10	How to play audio file	6	45

# Αξιολόγηση του CodeCatch (2/2)

- Υψηλότερο Reciprocal Rank σε μια ομάδα → Εύκολη εύρεση καλύτερης υλοποίησης
- Υψηλότερο Reciprocal Rank συνολικά (ανά ερώτημα) → Εύκολη εύρεση διαφορετικών υλοποιήσεων



# Εξόρυξη Πηγαίου Κώδικα

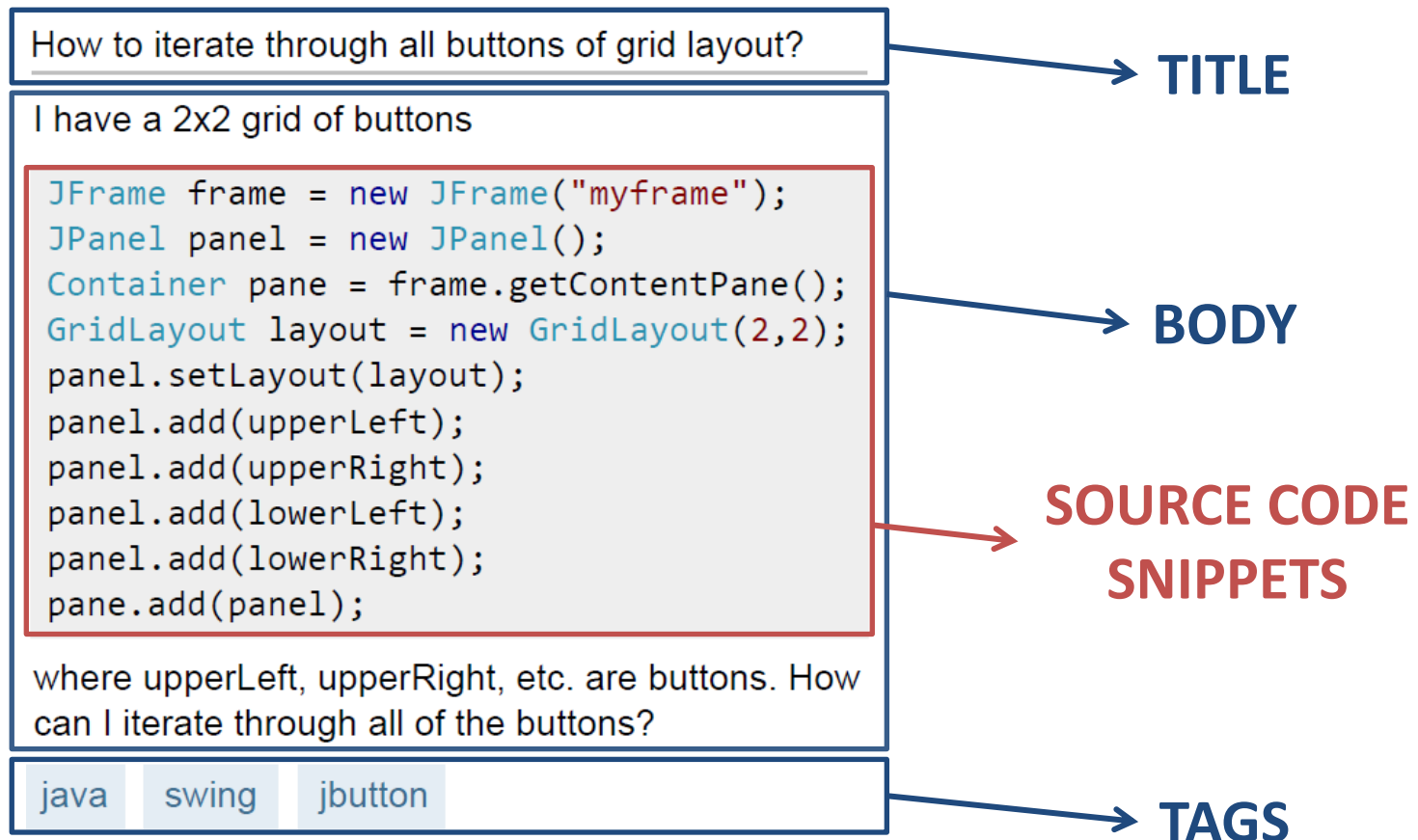
- Ευρετηριοποίηση Κώδικα
- Εξόρυξη Κώδικα για Επαναχρησιμοποίηση
- Εξόρυξη API Snippets
- Βελτίωση Ερωταπαντήσεων Υλοποίησης

# Βελτίωση Ερωταπαντήσεων Υλοποίησης (1/2)

- Αναζήτηση χρήσιμων ερωτήσεων στο Stack Overflow
- Σύνολο δεδομένων MSR
- Στοιχεία ερωτήματος: title, body, snippets, tags
- Τακτική “searching before posting”
  - Αναζήτηση με βάση το κείμενο της ερώτησης;
  - Αναζήτηση με βάση κώδικα;
- Εύρεση παρόμοιων snippets με αυτά του προγραμματιστή
  - Βελτίωση υπάρχουσας υλοποίησης

# Βελτίωση Ερωταπαντήσεων Υλοποίησης (2/2)

- Αναζήτηση χρήσιμων ερωτήσεων στο Stack Overflow
- Στοιχεία ερωτήματος: title, body, snippets, tags



# Ευρετηριοποίηση Ερωτήσεων (1/2)

- Εξαγωγή και αποθήκευση στοιχείων ερώτησης
  - Ευρετηριοποίηση
  - Πεδία κειμένου *Title* και *Body* → ευρετηριοποίηση ως tokens
    - Αφαίρεση stop words

How to iterate through all  
buttons of grid layout?



iterate, through, buttons,  
grid, layout

- Ευρετηριοποίηση *Tags* ως πεδία πίνακα/συνόλου

{java, swing, jbutton}

# Ευρετηριοποίηση Ερωτήσεων (2/2)

- Ανάλυση και Ευρετηριοποίηση Snippets

- Δύο περάσματα (passes) στο AST

- Εξαγωγή τύπων
- Ακολουθίες κώδικα
  - assignments (AM)
  - functions calls (FC)
  - class instantiations (CI)

**PASS 1**

Μεταβλητή	Τύπος
frame	JFrame
panel	JPanel
pane	Container
layout	GridLayout

```
JFrame frame = new JFrame("myframe");
JPanel panel = new JPanel();
Container pane = frame.getContentPane();
GridLayout layout = new GridLayout(2,2);
panel.setLayout(layout);
panel.add(upperLeft);
panel.add(upperRight);
panel.add(lowerLeft);
panel.add(lowerRight);
pane.add(panel);
```

**PASS 2**

CI_JFrame
CI_JPanel
FC_Container
CI_GridLayout
FC_void

...



# Αντιστοίχιση Ερωτήσεων (1/2)

- Αντιστοίχιση *Titles/Bodies*
  - TF-IDF, Vector Space Model

$$score(d_1, d_2) = \frac{d_1 \cdot d_2}{|d_1| \cdot |d_2|} = \frac{\sum_1^N tfidf(ti, d_1) \cdot tfidf(ti, d_2)}{\sum_1^N tfidf^2(ti, d_1) \cdot \sum_1^N tfidf^2(ti, d_2)}$$

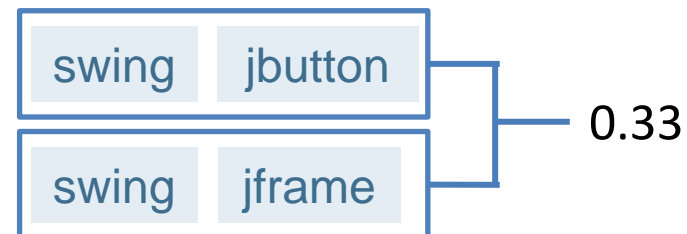
How to iterate through all buttons of grid layout?

How do you set the position of button in grid layout?

0.632

- Αντιστοίχιση *Tags*
  - Jaccard Index

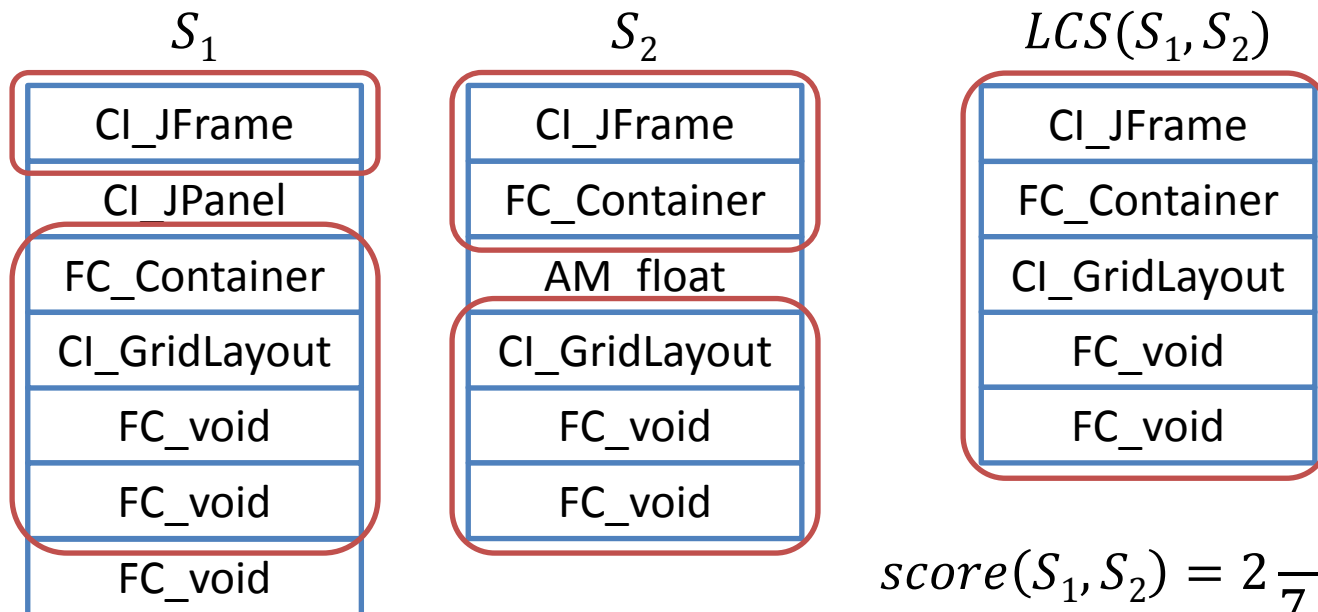
$$J(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$$



# Αντιστοίχιση Ερωτήσεων (2/2)

- Αντιστοίχιση *Snippets*
  - Longest Common Subsequence

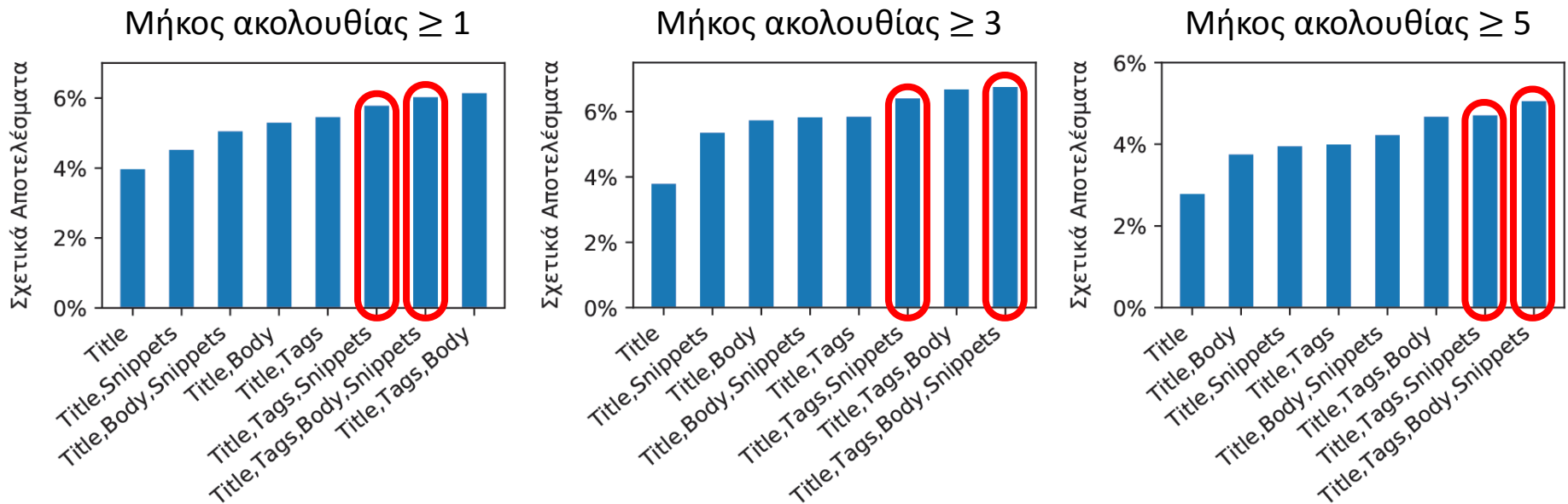
$$score(S_1, S_2) = 2 \frac{|LCS(S_1, S_2)|}{|S_1| + |S_2|}$$



$$score(S_1, S_2) = 2 \frac{5}{7 + 6} = 0.769$$

# Αξιολόγηση Αποτελεσμάτων

- Ground truth: linked ερωτήσεις του Stack Overflow
- Ποσοστό linked ερωτήσεων στις πρώτες 20
  - Διαφορετικά ελάχιστα μήκη ακολουθιών snippets



- Συμπέρασμα: χρήση snippets κατά την αναζήτηση

**Εξόρυξη Απαιτήσεων**

**Εξόρυξη Πηγαίου Κώδικα**

**Αξιολόγηση Ποιότητας**

# Επισκόπηση

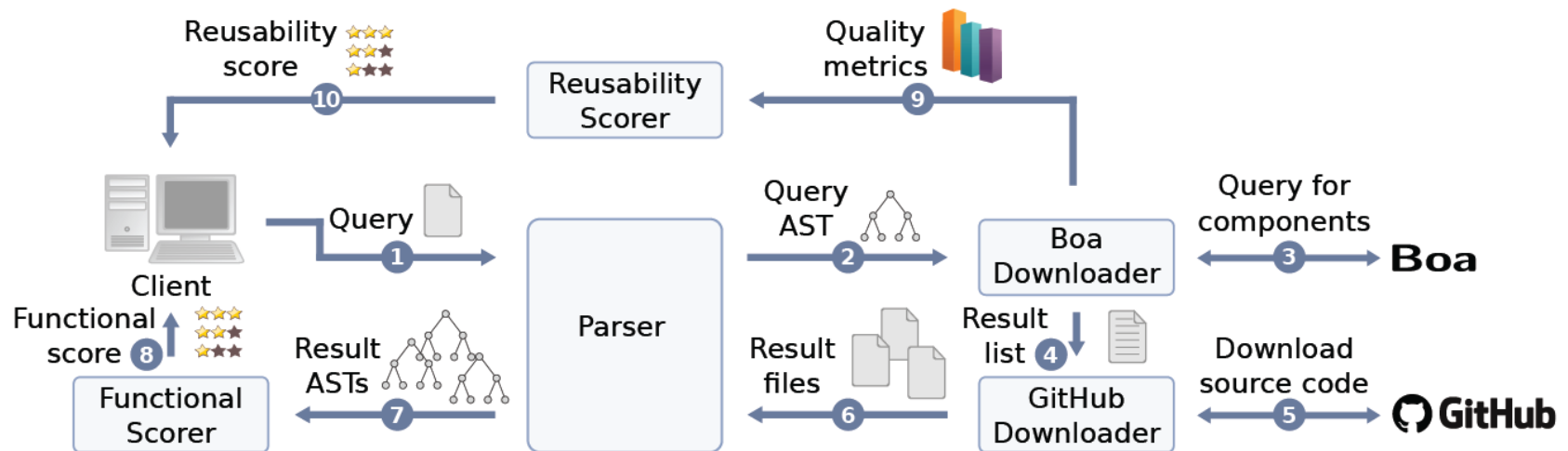
- Συστήματα Επαναχρησιμοποίησης Κώδικα
  - Εύρεση χρήσιμων τμημάτων κώδικα
  - Δυνατότητα/Καταλληλότητα επαναχρησιμοποίησης των τμημάτων
- Μετρικές Επαναχρησιμοποιησιμότητας
- Μοντέλο Επαναχρησιμοποιησιμότητας
  - Βασισμένο σε εμπειρογνώμονες (expert-based)
  - Βασισμένο στην προτίμηση από τους προγραμματιστές

# Αξιολόγηση Ποιότητας

- Προτάσεις Επαναχρησιμοποιήσιμου Κώδικα
- Αξιολόγηση Επαναχρησιμοποιησιμότητας Κώδικα

# Προτάσεις Επαναχρησιμοποίησης Κώδικα

- Σύστημα Επαναχρησιμοποίησης Κώδικα QualBoa
  - Σύνδεση με Boa (σύστημα indexing από MSR) και GitHub
  - Λήψη κώδικα και εξαγωγή μετρικών στατικής ανάλυσης
  - Εξόρυξη τμημάτων κώδικα
  - Αξιολόγηση επαναχρησιμοποιησιμότητας



# Εξόρυξη Τμημάτων Κώδικα (1/2)

- Υπολογισμός λειτουργικότητας για κάθε αποτέλεσμα
  - Υπογραφές κώδικα
    - π.χ. *[Stack, [push, void, Object], [pop, Object]]*
  - Ομοιότητα μεταξύ λιστών/διανυσμάτων
    - Tanimoto coefficient

$$\text{Tanimoto}(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}|^2 + |\vec{B}|^2 - \vec{A} \cdot \vec{B}}$$

- Ομοιότητα μεταξύ συμβολοσειρών (strings)
  - Διαχωρισμός strings (camelCase) και Jaccard index

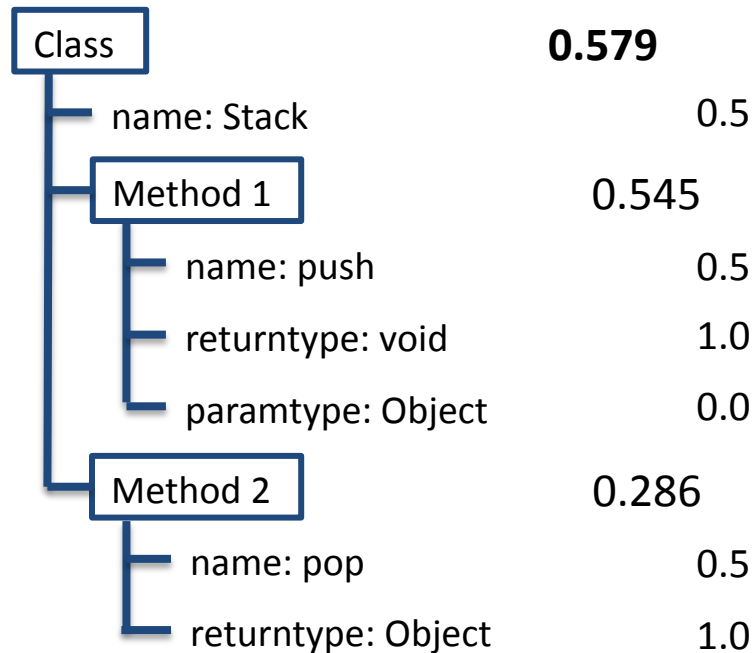
$$\text{Jaccard}(S, T) = \frac{S \cap T}{S \cup T}$$



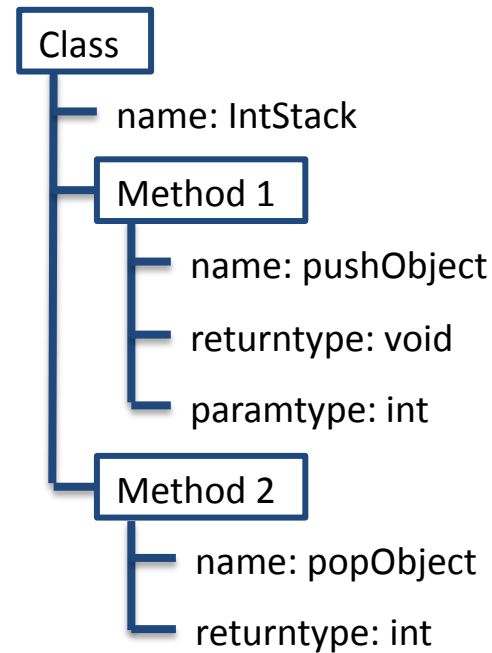
# Εξόρυξη Τμημάτων Κώδικα (2/2)

- Παράδειγμα αντιστοίχισης υπογραφών

```
public class Stack{
    public void push(Object element);
    public Object pop();
}
```



```
public class IntStack{
    public void pushObject(int elem);
    public int popObject();
}
```



# Αξιολόγηση Επαναχρησιμοποιησιμότητας

- Μοντέλο Επαναχρησιμοποιησιμότητας

Μετρικές Ποιότητας	Όρια	Τμηματοπο ισησιμότητα	Συντηρησ ιμότητα	Χρηστικ ότητα	Κατανοη σιμότητα	Επαναχρησιμο ποιησιμότητα
Average LoC per Method	> 30		x		x	x
Average Cyclomatic Complexity	> 8		x		x	x
Coupling Between Objects	> 20	x		x		x
Lack of Cohesion in Methods	> 20	x				x
Average Block Depth	> 3				x	x
Efferent Couplings	> 20	x		x		x
Number of Public Fields	> 10		x	x		x
Number of Public Methods	> 30		x	x		x
#Μετρικές ανά Χαρακτηριστικό Ποιότητας:		3	4	4	3	8

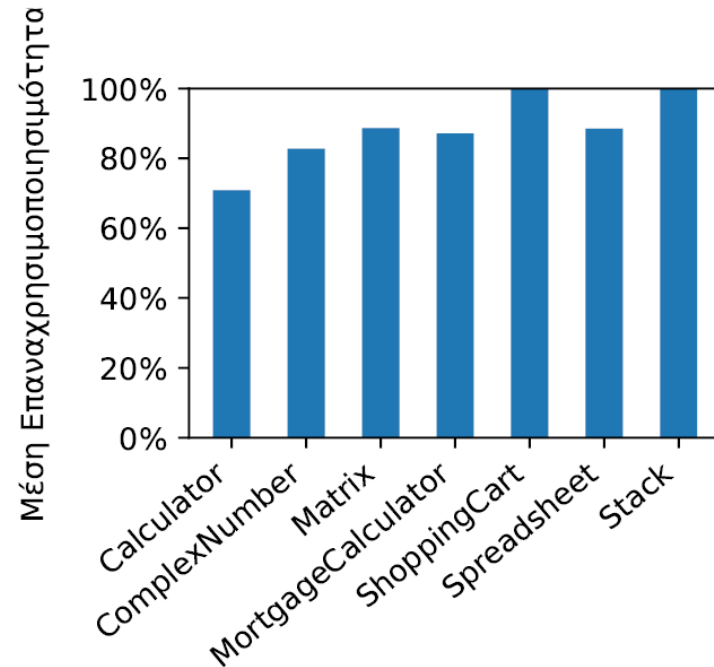
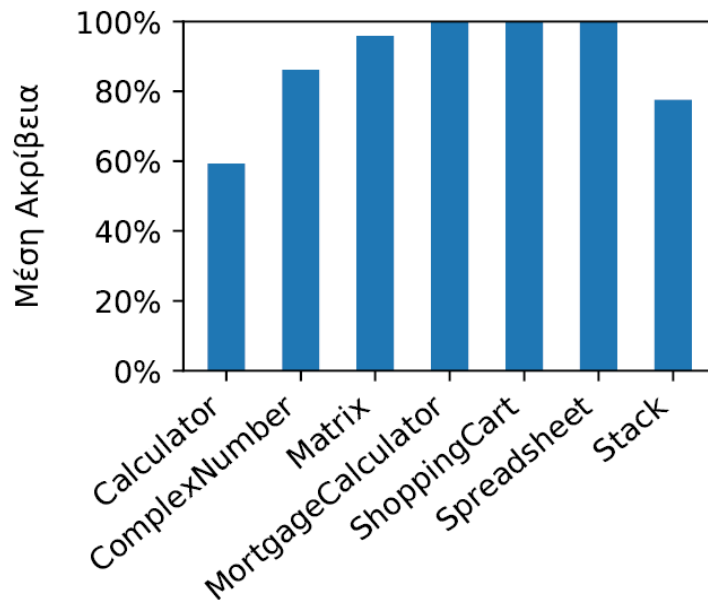
# Αξιολόγηση του QualBoa (1/2)

- Σύνολο από ερωτήματα για τμήματα κώδικα

Ερώτημα	#Σχετικά Αποτελέσματα	Μέση Ακρίβεια	Επαναχρησι-μοποιησιμότητα
Calculator	18	59.27%	70.83%
ComplexNumber	15	86.18%	82.76%
Matrix	10	95.88%	88.68%
MortgageCalculator	7	100.00%	87.17%
ShoppingCart	13	100.00%	100.00%
Spreadsheet	2	100.00%	88.54%
Stack	22	77.59%	100.00%
Μέσες τιμές	12.43	88.42%	88.28%

# Αξιολόγηση του QualBoa (2/2)

- Αποτελέσματα που καλύπτουν τη λειτουργικότητα για όλα τα ερωτήματα
  - > 10 αποτελέσματα σε 5/7 ερωτήματα
- Μέση επαναχρησιμοποιησιμότητα > 87.5%
  - Κατά μέσο όρο ξεπερνά τα όρια 1 στις 8 μετρικές

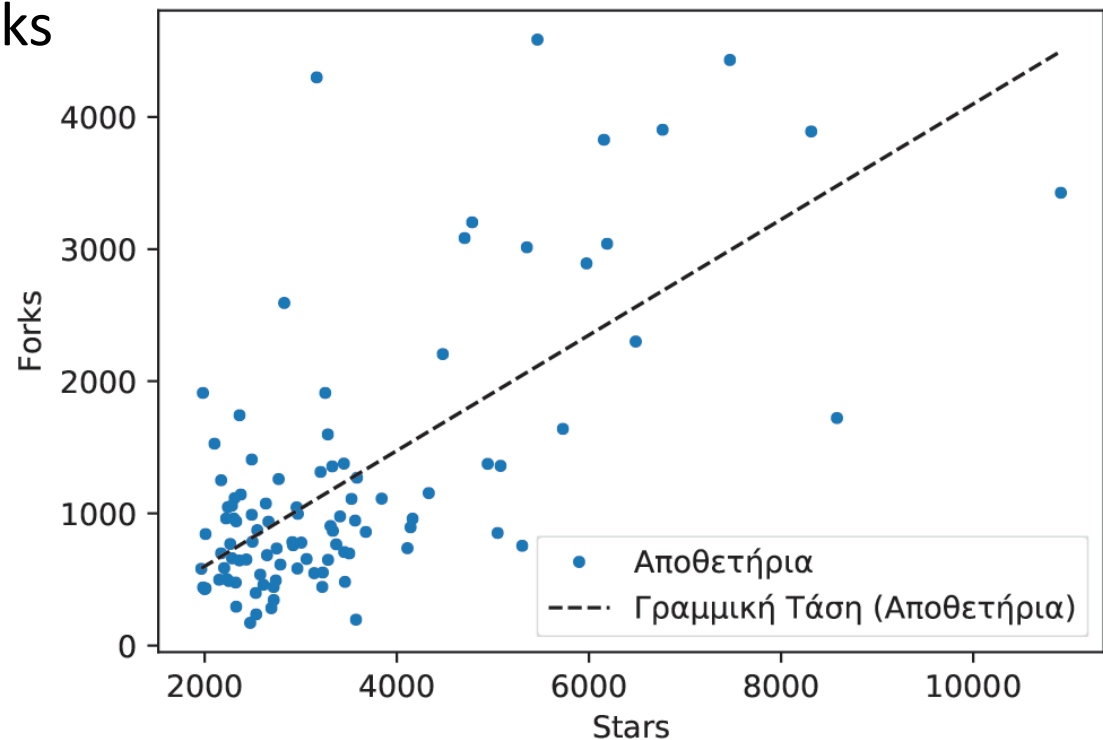


# Αξιολόγηση Ποιότητας

- Προτάσεις Επαναχρησιμοποιήσιμου Κώδικα
- Αξιολόγηση Επαναχρησιμοποιησιμότητας Κώδικα

# Αξιολόγηση Επαναχρησιμοποιησιμότητας Κώδικα

- Μοντέλο επαναχρησιμοποιησιμότητας με βάση την προτίμηση από τους προγραμματιστές
  - Αποφυγή expert-based μεθοδολογίας
  - Χρήση πληροφορίας από GitHub stars και forks
  - Μετρικές στατικής ανάλυσης



# Μετρικές Επαναχρησιμοποιησιμότητας Κώδικα

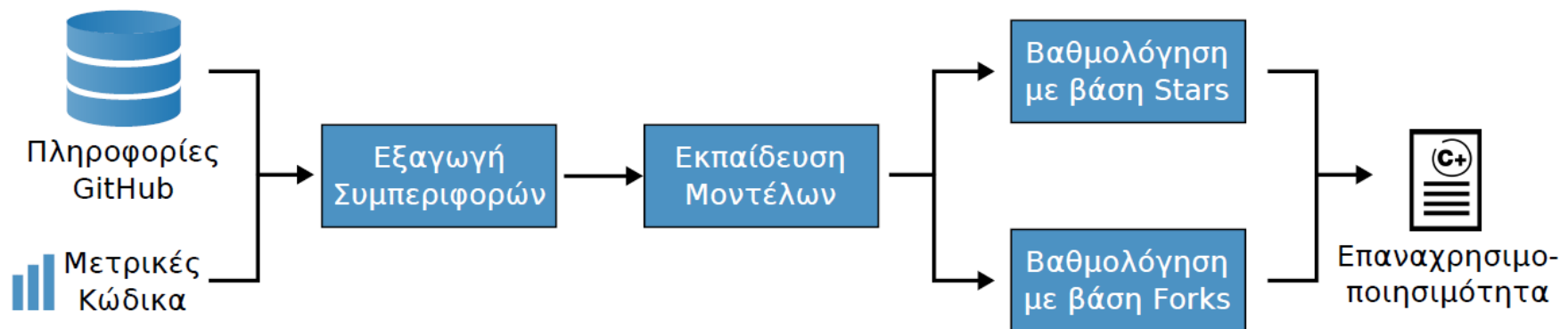
- 53 μετρικές στατικής ανάλυσης σε κατηγορίες
  - Πολυπλοκότητα (π.χ. Weighted Methods per Class)
  - Σύζευξη (π.χ. Coupling Between Objects)
  - Συνοχή (π.χ. Lack of Cohesion in Methods)
  - Τεκμηρίωση (π.χ. API Documentation)
  - Κληρονομικότητα (π.χ. Depth of Inheritance Tree)
  - Μέγεθος (π.χ. Lines of Code)

Κατηγορίες	Σχετικά με Forks			Σχετικά με Stars
	Κατανοησιμότητα	Δυν. Εκμάθησης	Λειτουργικότητα	Ελκυστικότητα
Πολυπλοκότητα	IP	IP	IP	P
Σύζευξη	IP	IP	IP	P
Συνοχή	P	P	P	P
Τεκμηρίωση	P	P	–	–
Κληρονομικότητα	IP	IP	–	P
Μέγεθος	IP	IP	IP	P

P: Ανάλογη Σχέση  
IP: Αντιστρόφως Ανάλογη Σχέση

# Μεθοδολογία Αξιολόγησης Επαναχρησιμοποιησιμότητας

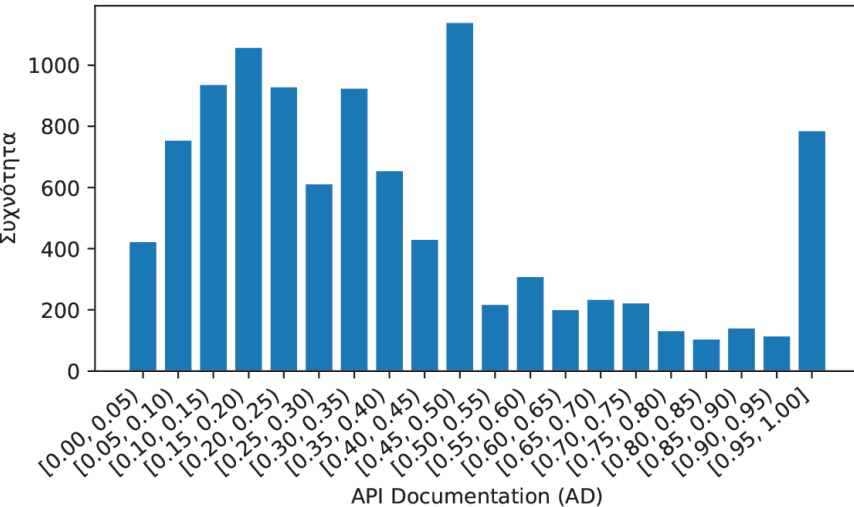
- Εξαγωγή συμπεριφορών από μετρικές στατικής ανάλυσης
  - Αντιστοίχιση μετρικών με πληροφορία stars/forks
- Εκπαίδευση μοντέλων
  - Είσοδος: μετρικές στατικής ανάλυσης
  - Έξοδος: επαναχρησιμοποιησιμότητα με βάση stars/forks
- Τελική έξοδος: τιμή επαναχρησιμοποιησιμότητας



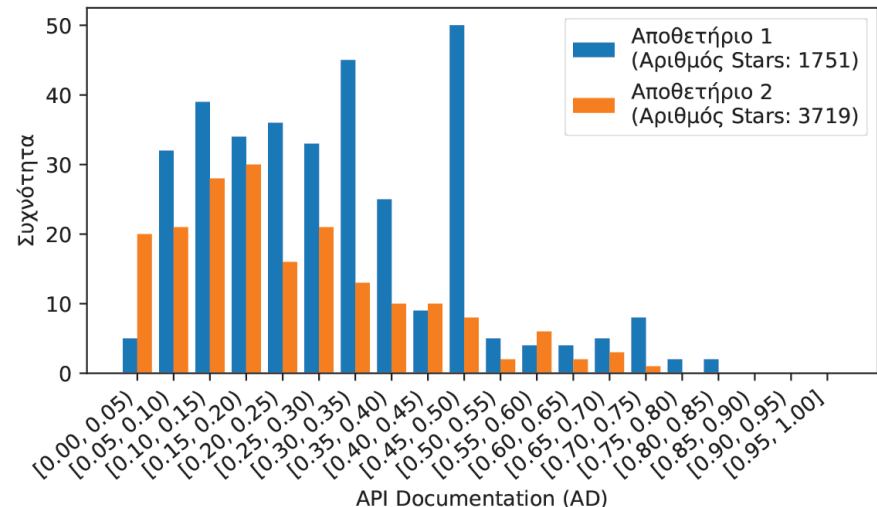


# Εξαγωγή Συμπεριφορών από Μετρικές Στατικής Ανάλυσης

1) Binning με Doane Formula



2) Συνδυασμός τιμών μετρικών

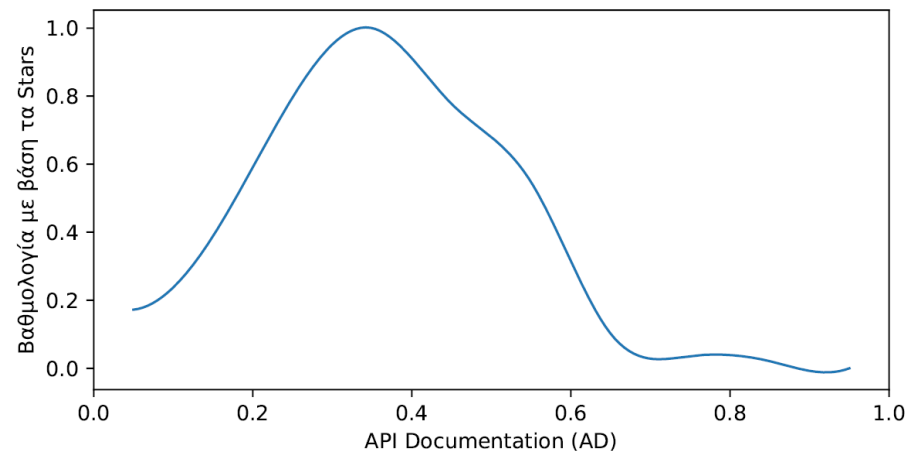


3) Εξαγωγή επαναχρησιμοποιησιμότητας

$$RS_{Metric}(i) = \sum_{repo=1}^N freq_{p.u.}(i) \cdot \log(S(repo))$$

$$RF_{Metric}(i) = \sum_{repo=1}^N freq_{p.u.}(i) \cdot \log(F(repo))$$

όπου  $freq_{p.u.}(i) = \frac{F_i}{\sum_{i=1}^N F_i}$



# Εκπαίδευση Μοντέλων

- Εξαγωγή τελικής τιμής επαναχρησιμοποιησιμότητας

$$RS_{Final} = \frac{\sum_{j=1}^k RS_{metric}(j) \cdot corr(metric_j, stars)}{\sum_{j=1}^k corr(metric_j, stars)} \quad \begin{array}{l} \times 1 \\ \text{Reusability} \\ \text{Score} \end{array}$$

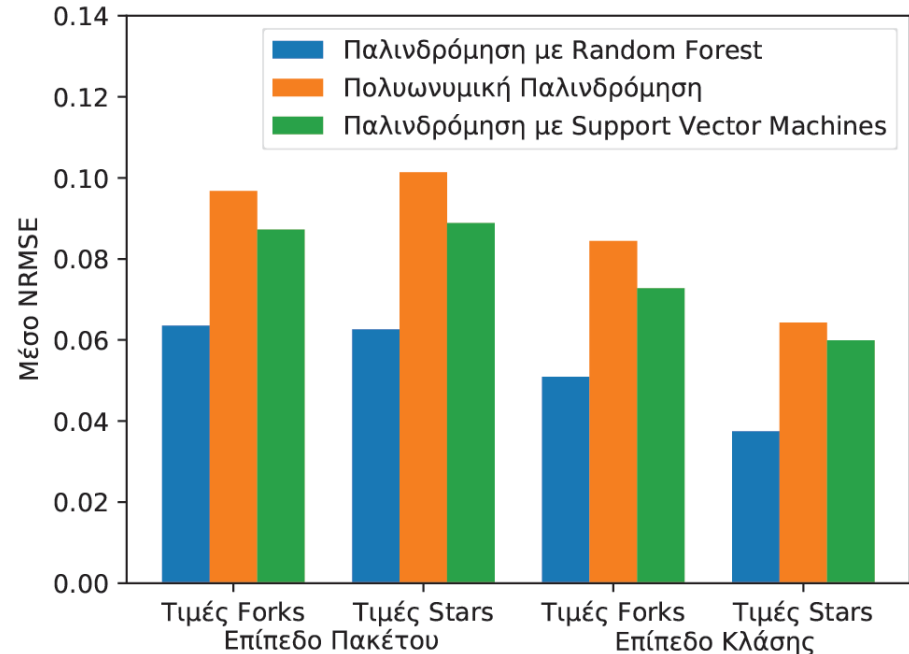
$$RF_{Final} = \frac{\sum_{j=1}^k RF_{metric}(j) \cdot corr(metric_j, forks)}{\sum_{j=1}^k corr(metric_j, forks)} \quad \begin{array}{l} \times 3 \end{array}$$

- Εκπαίδευση μοντέλων

- Random Forest
- Πολυωνυμική παλινδρόμηση
- Support Vector Machines

- Επιλογή με βάση το NRMSE

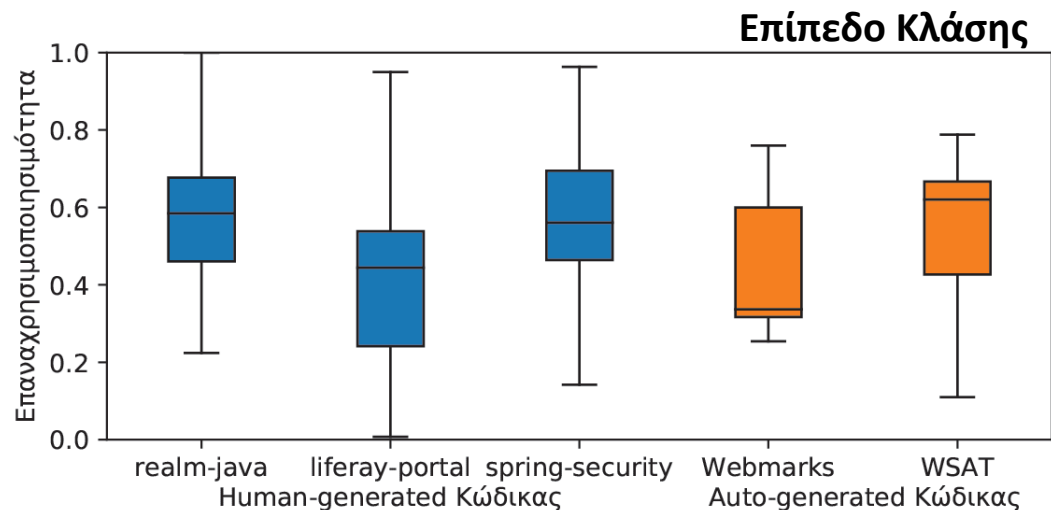
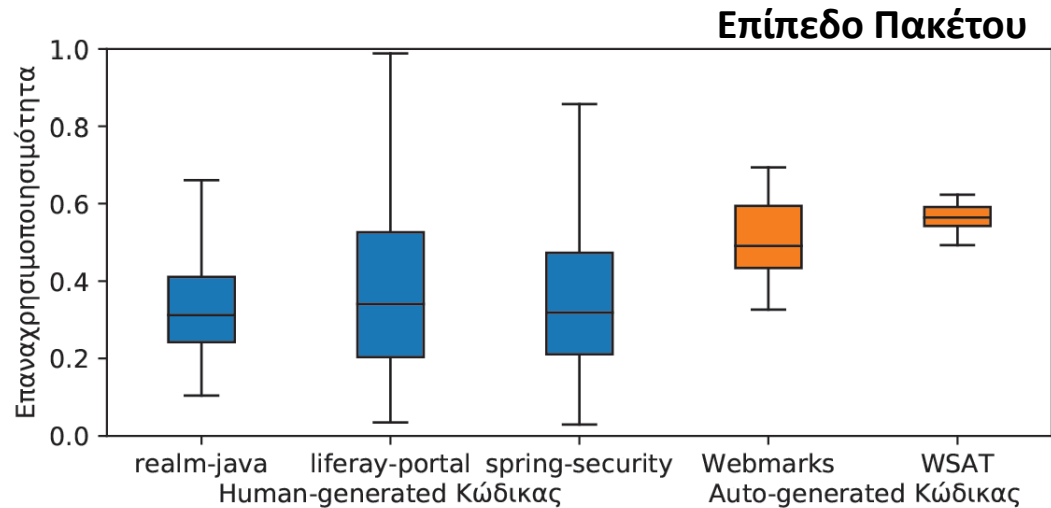
$$NRMSE = \sqrt{\frac{1}{N} \cdot \frac{\sum_{i=1}^N (\hat{y}_i - y_i)^2}{\sum_{i=1}^N (\bar{y} - y_i)^2}}$$



# Αξιολόγηση (1/2)

- 5 έργα λογισμικού
  - **Human-generated**
  - και **Auto-generated**

Όνομα Έργου	#Πακέτα	#Κλάσεις
realm-java	137	3859
liferay-portal	1670	3941
spring-security	543	7099
Webmarks	9	27
WSAT	20	127



## Αξιολόγηση (2/2)

- Τιμές Μετρικών για Κλάσεις και Πακέτα με διαφορετική Επαναχρησιμοποιησιμότητα

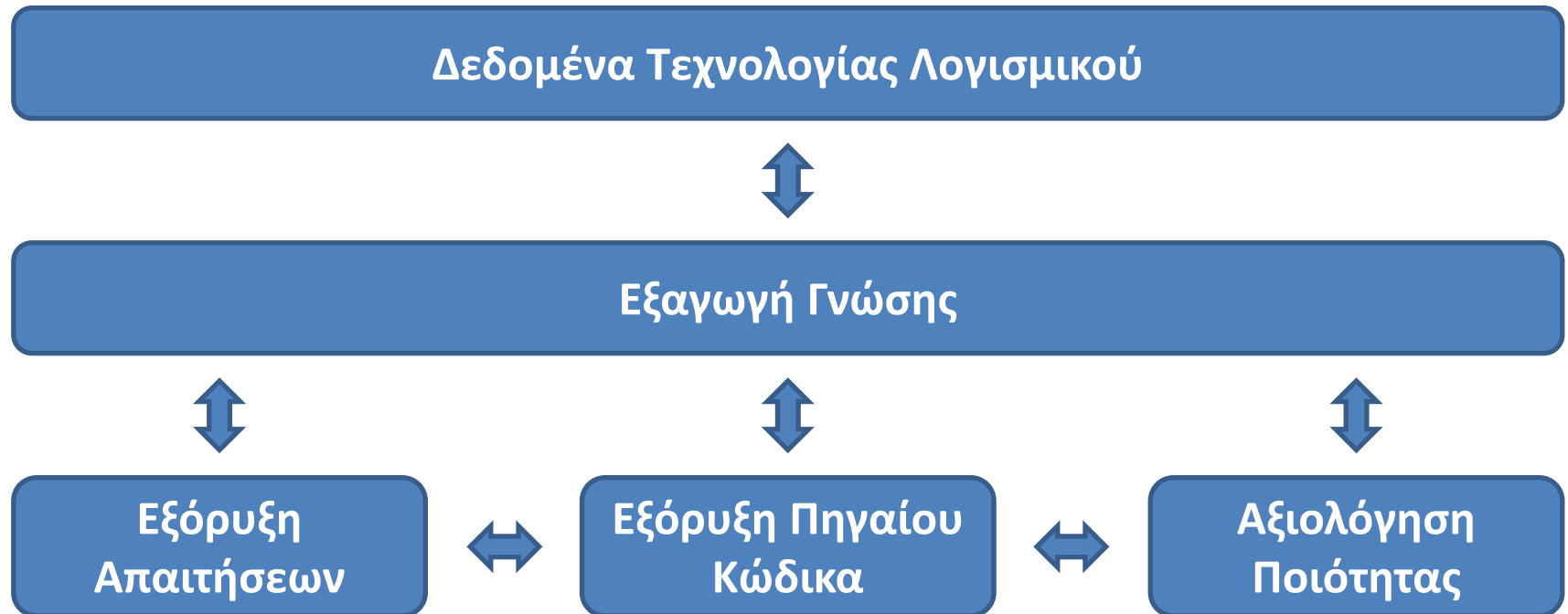
Μετρικές	Κλάσεις		Πακέτα	
	Κλάση 1	Κλάση 2	Πακέτο 1	Πακέτο 2
WMC	14	12	–	–
CBO	12	3	–	–
LCOM5	2	11	–	–
CD (%)	20.31%	10.2%	41.5%	0.0%
RFC	12	30	–	–
LOC	84	199	2435	38
TNCL	–	–	8	2
Επαναχρησιμοποιησιμότητα	95.78%	10.8%	90.59%	16.75%

- Πιο συνεκτικές κλάσεις → υψηλότερη επαναχρησιμοποιησιμότητα
- Καλή τεκμηρίωση → υψηλή επαναχρησιμοποιησιμότητα

Συμπεράσματα

# Συμπεράσματα (1/2)

- Εφαρμογή τεχνικών εξόρυξης δεδομένων
  - Υποστήριξη επαναχρησιμοποίησης απαιτήσεων/πηγαίου κώδικα, και αξιολόγηση επαναχρησιμοποιησιμότητας
- Ενοποιημένη λύση για την υποστήριξη της διαδικασίας ανάπτυξης λογισμικού μέσω της επαναχρησιμοποίησης



# Συμπεράσματα (2/2)

- Εφαρμογή τεχνικών εξόρυξης δεδομένων
  - Υποστήριξη επαναχρησιμοποίησης απαιτήσεων/πηγαίου κώδικα, και αξιολόγηση επαναχρησιμοποιησιμότητας
- Εξόρυξη Απαιτήσεων
  - Μεθοδολογία για μοντελοποίηση απαιτήσεων διαφορετικών τύπων
  - Εξόρυξη για επαναχρησιμοποίηση απαιτήσεων
  - Εξαγωγή προδιαγραφών
- Εξόρυξη πηγαίου κώδικα
  - Αποτελεσματική ευρετηριοποίηση
  - Εξόρυξη σε επίπεδο τμημάτων κώδικα και σε επίπεδο API snippets
  - Αξιολόγηση αν πληρείται η επιθυμητή λειτουργικότητα
  - Περαιτέρω βελτίωση κώδικα μέσω κοινότητας ερωταπαντήσεων
- Αξιολόγηση επαναχρησιμοποιησιμότητας
  - Κατασκευή πρωτότυπου συστήματος αξιολόγησης λειτουργικότητας και επαναχρησιμοποιησιμότητας τμημάτων κώδικα
  - Μεθοδολογία για την αξιολόγηση επαναχρησιμοποιησιμότητας όπως γίνεται αντιληπτή από τον προγραμματιστή

# Μελλοντική Έρευνα

- Εξόρυξη Απαιτήσεων
  - Εισαγωγή πληροφορίας προτίμησης απαιτήσεων από stakeholders
  - Υποστήριξη επιπλέον τύπων απαιτήσεων
  - Περαιτέρω διερεύνηση σύνδεσης απαιτήσεων/προδιαγραφών με υλοποίηση
- Εξόρυξη πηγαίου κώδικα
  - Αυτοματοποιημένη εξαγωγή και ικανοποίηση εξαρτήσεων
  - Ενσωμάτωση σημασιολογικής πληροφορίας
- Αξιολόγηση επαναχρησιμοποιησιμότητας
  - Ενσωμάτωση επιπλέον μετρικών ποιότητας
  - Περαιτέρω αξιολόγηση μοντέλων επαναχρησιμοποιησιμότητας
  - Περαιτέρω διερεύνηση της επαναχρησιμοποιησιμότητας
    - Υπολογισμός πραγματικού reuse rate μέσω εξόρυξης κώδικα από αποθετήρια
- Συνολικά: παροχή μιας ολοκληρωμένης δομής για την ευρετηριοποίηση συνδεδεμένων αντικειμένων λογισμικού (απαιτήσεις, κώδικας, κ.λπ.) και χρήση της με σκοπό την επαναχρησιμοποίηση



# Λίστα Δημοσιεύσεων (1/2)

## Άρθρα σε Περιοδικά

1. **Themistoklis Diamantopoulos** and Andreas Symeonidis. “Enhancing Requirements Reusability through Semantic Modeling and Data Mining Techniques”. In: *Enterprise Information Systems* (2017), pp. 1–22.
2. **Themistoklis Diamantopoulos**, Michael Roth, Andreas Symeonidis, and Ewan Klein. “Software Requirements as an Application Domain for Natural Language Processing”. In: *Language Resources and Evaluation* 51.2 (2017), pp. 495–524.
3. Christoforos Zolotas, **Themistoklis Diamantopoulos**, Kyriakos Chatzidimitriou, and Andreas Symeonidis. “From requirements to source code: a Model-Driven Engineering approach for RESTful web services”. In: *Automated Software Engineering* (2016), pp. 1–48.
4. **Themistoklis Diamantopoulos** and Andreas Symeonidis. “Localizing Software Bugs using the Edit Distance of Call Traces”. In: *International Journal On Advances in Software* 7.1 (2014), pp. 277–288.
5. **Themistoklis Diamantopoulos** and Andreas L. Symeonidis. “AGORA: A Search Engine for Source Code Reuse”. In: *SoftwareX* (under review).

## Κεφάλαια σε Βιβλία

1. Valasia Dimaridou, Alexandros-Charalampos Kyprianidis, Michail Papamichail, **Themistoklis Diamantopoulos**, and Andreas Symeonidis. “Assessing the User-Perceived Quality of Source Code Components using Static Analysis Metrics”. In: *Communications in Computer and Information Science*. Springer, 2018, in press.

## Δημοσιεύσεις σε Συνέδρια

1. **Themistoklis Diamantopoulos**, Georgios Karagiannopoulos, and Andreas L. Symeonidis. “CodeCatch: Extracting Source Code Snippets from Online Sources”. In: *Proceedings of the IEEE/ACM 6th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering. RAISE '18*. Gothenburg, Sweden, 2018, pp. 21–27.
2. Michail Papamichail, **Themistoklis Diamantopoulos**, Ilias Chrysovergis, Philippos Samlidis, and Andreas Symeonidis. “User-Perceived Reusability Estimation based on Analysis of Software Repositories”. In: *Proceedings of the 2018 IEEE International Workshop on Machine Learning Techniques for Software Quality Evaluation. MaLTeSQuE*. Campobasso, Italy, 2018, pp. 49–54.

# Λίστα Δημοσιεύσεων (2/2)

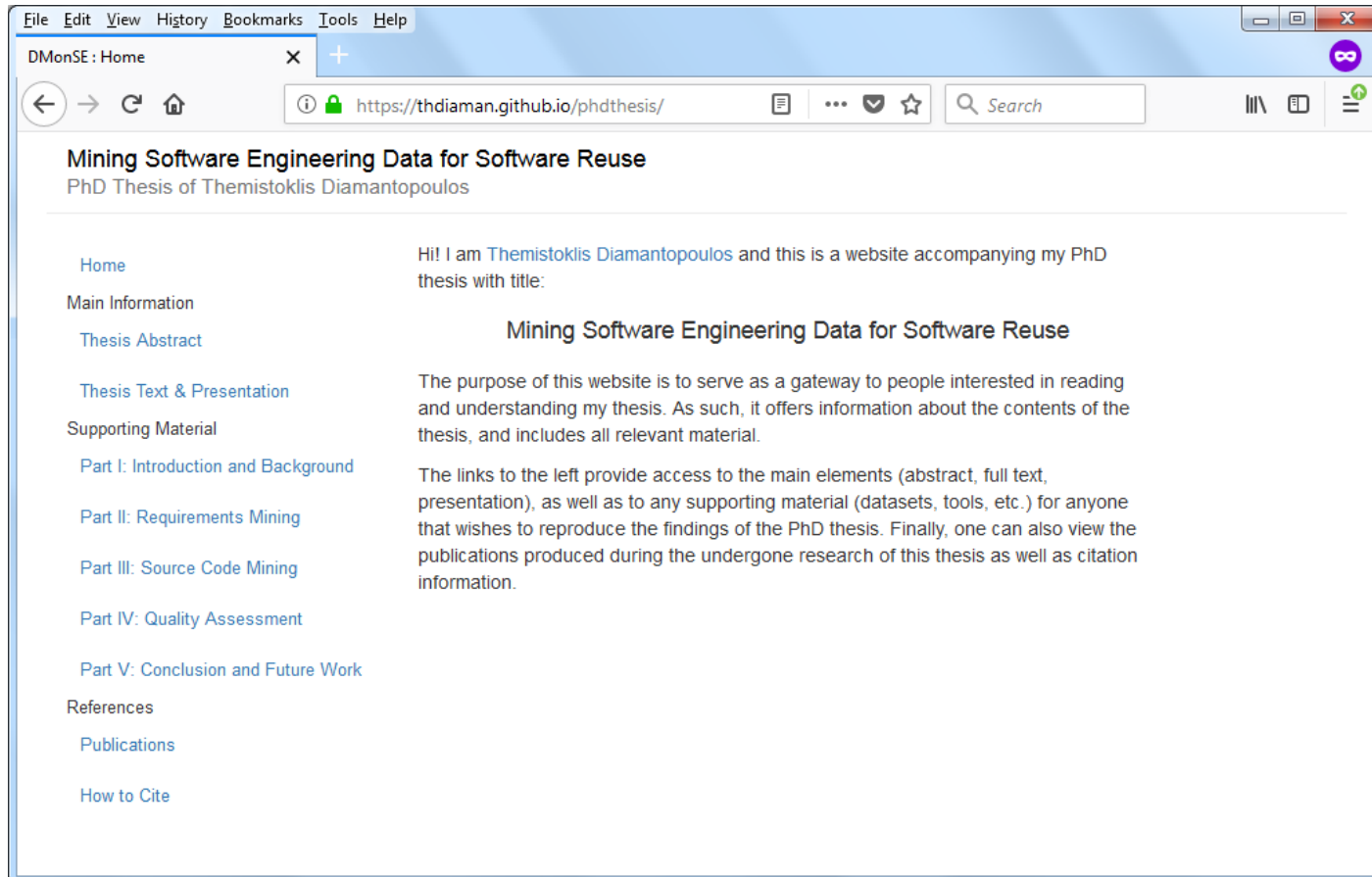
## Δημοσιεύσεις σε Συνέδρια (συνέχεια)

3. Valasia Dimaridou, Alexandros-Charalampos Kyprianidis, Michail Papamichail, **Themistoklis Diamantopoulos**, and Andreas Symeonidis. “Towards Modeling the Userperceived Quality of Source Code using Static Analysis Metrics”. In: Proceedings of the 12th International Conference on Software Technologies - Volume 1. ICISOFT. INSTICC. Setubal, Portugal: SciTePress, 2017, pp. 73–84.
4. Michail Papamichail, **Themistoklis Diamantopoulos**, and Andreas L. Symeonidis. “User-Perceived Source Code Quality Estimation based on Static Analysis Metrics”. In: Proceedings of the 2016 IEEE International Conference on Software Quality, Reliability and Security. QRS. Vienna, Austria, 2016, pp. 100–107.
5. **Themistoklis Diamantopoulos**, Antonis Noutsos, and Andreas L. Symeonidis. “DPCORE: A Design Pattern Detection Tool for Code Reuse”. In: Proceedings of the 6th International Symposium on Business Modeling and Software Design. BMSD. Rhodes, Greece, June 2016, pp. 160–169.
6. **Themistoklis Diamantopoulos**, Klearchos Thomopoulos, and Andreas L. Symeonidis. “QualBoa: Reusability-aware Recommendations of Source Code Components”. In: Proceedings of the IEEE/ACM 13th Working Conference on Mining Software Repositories. MSR ’16. 2016, pp. 488–491.
7. **Themistoklis Diamantopoulos** and Andreas Symeonidis. “Towards Interpretable Defect-Prone Component Analysis using Genetic Fuzzy Systems”. In: Proceedings of the IEEE/ACM 4th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering. RAISE ’15. Florence, Italy, 2015, pp. 32–38.
8. **Themistoklis Diamantopoulos** and Andreas L. Symeonidis. “Employing Source Code Information to Improve Question-answering in Stack Overflow”. In: Proceedings of the 12th Working Conference on Mining Software Repositories. MSR ’15. Florence, Italy: IEEE Press, 2015, pp. 454–457.
9. Michael Roth, **Themistoklis Diamantopoulos**, Ewan Klein, and Andreas L. Symeonidis. “Software Requirements: A new Domain for Semantic Parsers”. In: Proceedings of the ACL 2014 Workshop on Semantic Parsing. SP14. Baltimore, Maryland, USA, 2014, pp. 50–54.
10. **Themistoklis Diamantopoulos** and Andreas L. Symeonidis. “Towards Scalable Bug Localization using the Edit Distance of Call Traces”. In: Proceedings of the Eighth International Conference on Software Engineering Advances. ICSEA 2013. Venice, Italy, 2013, pp. 45–50.

# Συμπληρωματικό Υλικό

Ιστοσελίδα με σύνολα δεδομένων, μοντέλα, εργαλεία, κ.λπ.

<https://thdiaman.github.io/phdthesis/>



Ευχαριστώ για την προσοχή σας!