Bachelor’s Thesis

Complexity Measures in Software Engineering

A Systematic Comparison and Evaluation on Software-Component-Level

Simon Lund

Overview

1. Motivation
2. Measures
3. Theoretical Evaluation
   (with Weyuker's Properties)
4. Practical Evaluation
   (on the example of CPAchecker)
5. Conclusion

Motivation

Objective: *Assess the complexity of the dependencies of a software system as accurately as possible*

Why? *Complexity of large software systems emerges from its dependencies*

Motivation (continued)

Definition:

$$C_P := \text{"the set of all classes of package P"}$$

Definition:

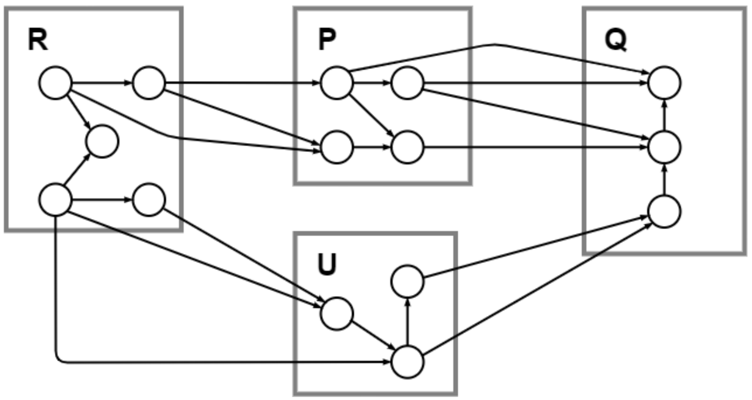
$$NOC(P) = |C_P|$$

(Number of classes of a package)

Motivation (continued)

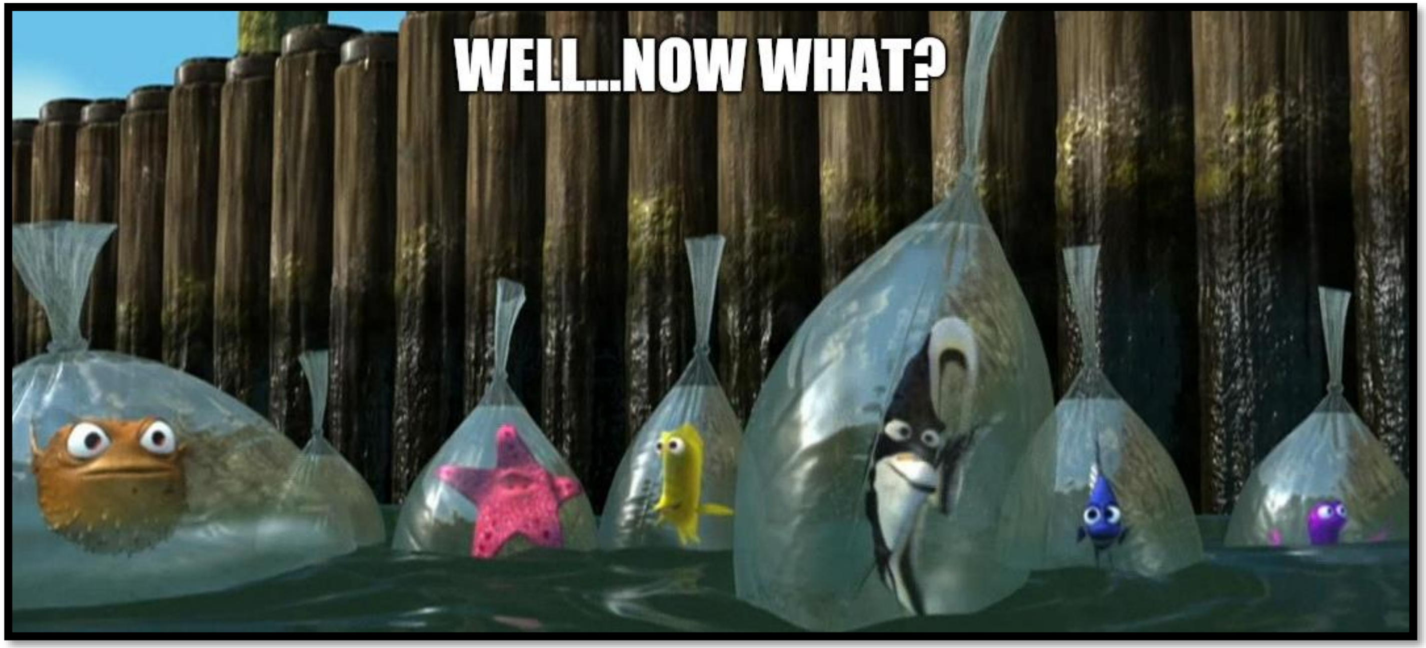
Example:

- $NOC(R) \ = 5$
- $NOC(P) \ = 4$
- $NOC(Q) \ = 3$
- $NOC(U) \ = 3$



$NOC(P) = |C_P|$

$\Rightarrow NOC$ does **not consider** the dependencies of a package



Problem: *How can we measure the complexity?*

Measures

Measures

Proposed Measures: *5 package-level measures that focus on the dependencies of a package*

Existing Measures: *4 established measures*

$\Rightarrow$ **Today:** 2 proposed measures (DCM_{CC}, $P\text{-}DepDegree$)

Further Definitions

Def. $\boldsymbol{C_S}$:

$$C_S := \text{"the set of all classes of system } S\text{"}$$

Def. $\boldsymbol{D_c}$:

$$D_c := \text{"the set of all dependencies of class } c\text{"}$$

Def. $\boldsymbol{D_P}$:

$$D_P := \bigcup_{c \in C_\mathrm{P}} D_c \;\; \text{(set of dependencies of P)}$$

DCM_{CC}

Def. Dependency Cohesion: *The degree to which classes of a given package have the same dependencies*

Def. Count Function:

$$cnt_P(d) = |\{\mathrm{c} \mid \mathrm{c} \in C_P : \mathrm{d} \in D_c\}|$$

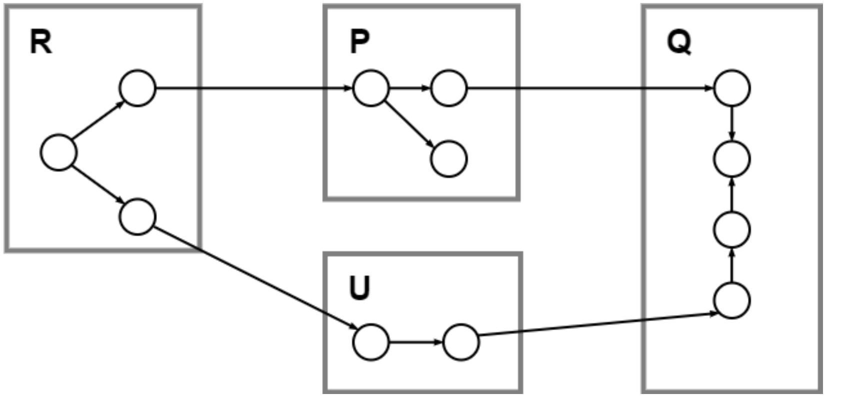
DCM_{CC} (continued)

Def. $\boldsymbol{DCM_{CC}}$**:**

$$DCM_{CC}(P) = \frac{\sum_{d \in \mathrm{D}_P} cnt_P(d)}{|C_P| * |D_P|}$$

Example:

- $DCM_{CC}(R) = \frac{1+1+1+1}{3*4}$
- $DCM_{CC}(P) = \frac{1+1+1}{3*3}$
- $DCM_{CC}(Q) = \frac{2+1}{4*2}$
- $DCM_{CC}(U) = \frac{1+1}{2*2}$



Package DepDegree

Def. Dependency Graph:

$$DG := \left(C_S, \bigcup_{c \in C_S} \{(c,d) | d \in D_c\}\right)$$

Def. Transitive Dependency Graph:

$$TDG_P := (V_{TDG}, E_{TDG})$$

$$V_{TDG} := C_P \cup \{d \in C_S \mid \exists c \in C_P : c \to^* d\}$$

$$E_{TDG} := \bigcup_{c \in V_{TDG}} \{(c,d) | d \in D_c\}$$

$(c \to^* d \; := \; \text{"path between c and d"})$

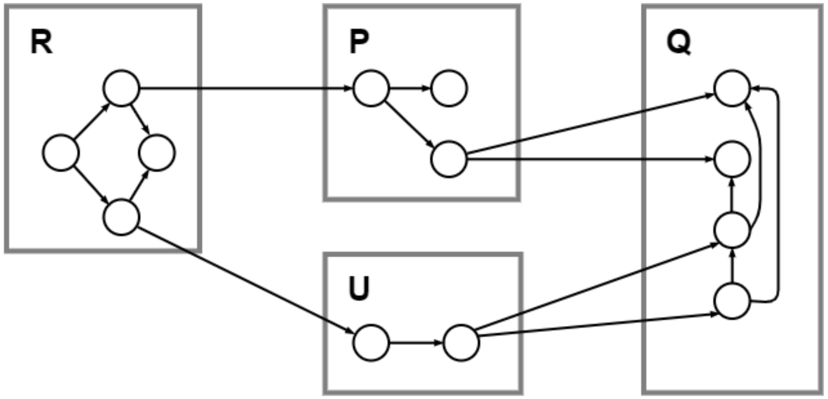
Package DepDegree (continued)

Def. Package DepDegree:

$$P\text{-}DepDegree(P) = \frac{E_{TDG}}{E_{DG}}$$

Example:

- $P\text{-}DepDegree(R) = \frac{17}{17}$
- $P\text{-}DepDegree(P) = \frac{4}{17}$
- $P\text{-}DepDegree(Q) = \frac{4}{17}$
- $P\text{-}DepDegree(U) = \frac{7}{17}$



Other Measures*

- Existing Measures:
  - Afferent Coupling (Ca)
  - Instability (I)
- Proposed Variants of DCM:
  - Based on LCOM3
  - Based on similarity measure
- Dependency Locality Measure

(*not considered in this presentation, but used/proposed in the related thesis)

Theoretical Evaluation

with Weyuker's Properties

Weyuker‘s Properties

- Redefined for package-level
  (Scope is a package with its classes)
- Set of 9 Properties:
  - Properties 1,2,7 and 8 are not relevant for package-level
    (Either always true or not applicable)
  - Properties 3,4,6 and 9 are existential
    (-> Give Witness for each property)
  - Property 5 uses a universal quantor
    (-> Show for any arbitrary packages)
- Operators:
  - $\mu(X) -$ Measurement value of package X for measure μ
  - $P \equiv Q -$ Packages P and Q are functionally equivalent
  - $P + Q -$ Composition of P and Q

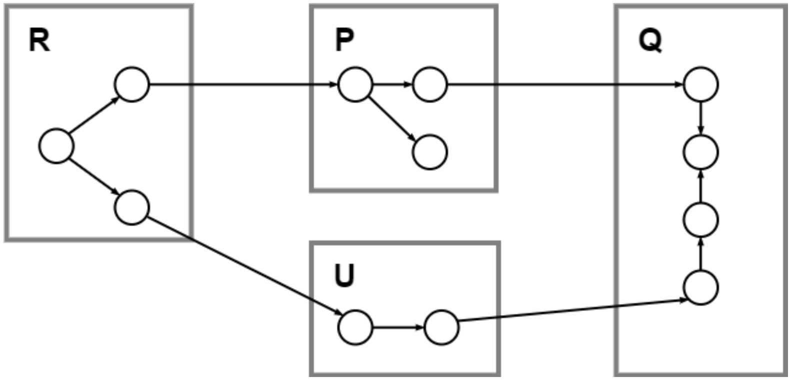
Weyuker's Properties (continued)

- $\boldsymbol{P3}$: $\exists P, Q : P \not\equiv Q \land \mu(P) = \mu(Q)$
- $\boldsymbol{P4}$: $\exists P, Q : P \equiv Q \land \mu(P) \neq \mu(Q)$

Proof for $\boldsymbol{DCM_{CC}}$:

- $P, R \Longrightarrow \boldsymbol{P3}$
- $P, U \Longrightarrow \boldsymbol{P4}$

- $DCM_{CC}(R) = \frac{1}{3}$
- $DCM_{CC}(P) = \frac{1}{3}$
- $DCM_{CC}(Q) = \frac{3}{8}$
- $DCM_{CC}(U) = \frac{1}{2}$



$P \equiv U$ and $P \not\equiv R$

Weyuker's Properties (continued)

- $\boldsymbol{P5}$: $\forall P, Q: \mu(P) \geq \mu(P + Q) \wedge \mu(Q) \geq \mu(P + Q)$

Proof for $\boldsymbol{DCM_{CC}}$: We consider the composition $V + W$ of any two packages V, W. We know that the composition does not yield new dependencies such that the number of dependencies in $V + W$ is equal to the sum of the dependencies of V, W. Furthermore, the denominator of the formula of DCM_{CC} increases for $V + W$ as the number of classes of $V + W$ is the sum of the number of classes of V, W. Thus, it follows that $DCM_{CC}(V) \geq DCM_{CC}(V + W)$ *and* $DCM_{CC}(W) \geq DCM_{CC}(V + W)$ holds for V, W such that DCM_{CC} satisfies this property.

Weyuker's Properties (continued)

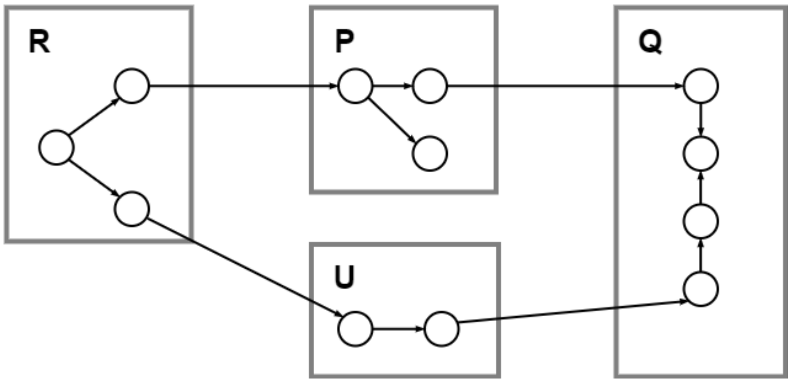
- $\boldsymbol{P6}: \exists P, Q, R: \mu(P) = \mu(Q) \wedge \mu(P + R) \neq \mu(Q + R)$

Proof for $\boldsymbol{DCM_{CC}}$:

- $DCM_{CC}(P + Q) = \frac{5}{28}$
- $DCM_{CC}(R + Q) = \frac{1}{6}$

$P, R, Q \implies \boldsymbol{P6}$

- $DCM_{CC}(R) = \frac{1}{3}$
- $DCM_{CC}(P) = \frac{1}{3}$
- $DCM_{CC}(Q) = \frac{3}{8}$
- $DCM_{CC}(U) = \frac{1}{2}$



$P \equiv U$ and $P \not\equiv R$

Weyuker‘s Properties (continued)

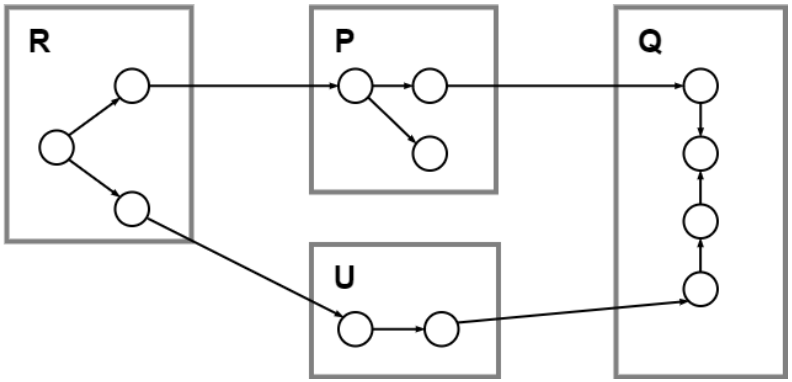
- $\boldsymbol{P9}: \exists P, Q: \mu(P) + \mu(Q) > \mu(P+Q)$

Proof for $\boldsymbol{DCM_{CC}}$:

- $DCM_{CC}(R+Q) = \frac{1}{6}$
- $\frac{1}{3} + \frac{3}{8} > \frac{1}{6}$

$R, Q \Longrightarrow \boldsymbol{P9}$

- $DCM_{CC}(R) = \frac{1}{3}$
- $DCM_{CC}(P) = \frac{1}{3}$
- $DCM_{CC}(Q) = \frac{3}{8}$
- $DCM_{CC}(U) = \frac{1}{2}$



$P \equiv U$ and $P \not\equiv R$

Summary

Measures	1	2	3	4	5	6	8	9
NOC	✓	✓	✓	✓	✓	✗	✓	✗
Ca	✓	✓	✓	✓	✗	✓	✓	✗
Ce	✓	✓	✓	✓	✗	✓	✓	✗
I	✓	✓	✓	✓	✗	✓	✓	?
DCM_{LCOM3}	✓	✓	✓	✓	✗	✓	✓	✗
DCM_{SIM}	✓	✓	✓	✓	✗	✓	✓	✗
DCM_{CC}	✓	✓	✓	✓	✓	✓	✓	✓
$P\text{-}DepDegree$	✓	✓	✓	✓	✓	✓	✓	✗
DLM	✓	✓	✓	✓	✗	✓	✓	✓

Practical Evaluation

on the example of CPAchecker

(Data Repository: https://github.com/simon-lund/cpachecker-data)

Implementation of **Jade**

- Developed in Python
- Uses dependency graph generated by Jdeps
- Code Repository: https://github.com/simon-lund/jade

$$[\overbrace{\{\text{“e”}, \text{“f”}, \text{“g”}\}}^{D_a}, \overbrace{\{\text{“e”}, \text{“h”}\}}^{D_b}, \overbrace{\{\text{“g”}\}}^{D_c}]$$

```python


def dcm_cc(pkg: list):
    """
    Variant of dcm based on cohesion count
    """

    # Count function
    count = lambda d, package: len([c for c in pkg if d in c])

    deps = set.union(*pkg)
    counts = [count(d, pkg) for d in deps]

    return sum(counts) / (len(pkg) * len(deps)) if len(pkg) > 0 and len(deps) > 0 else 0


```

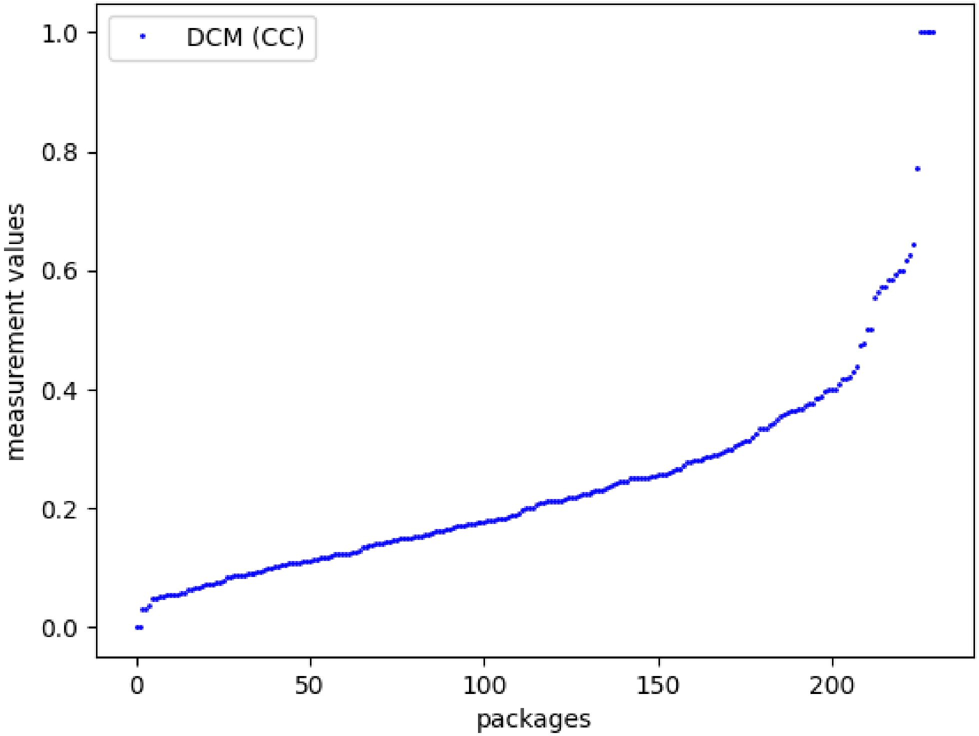
Example: Python Code for DCM_{CC}

CPAchecker

- Used Version: 1.9.1
- Domain “org.sosy_lab.cpachecker”:
  - 230 packages
  - 3596 classes
    - including interfaces, abstract and static classes
    - 1440 of which are nested classes
- In addition:
  - 115 test classes
  - References to 1015 external classes

Approach

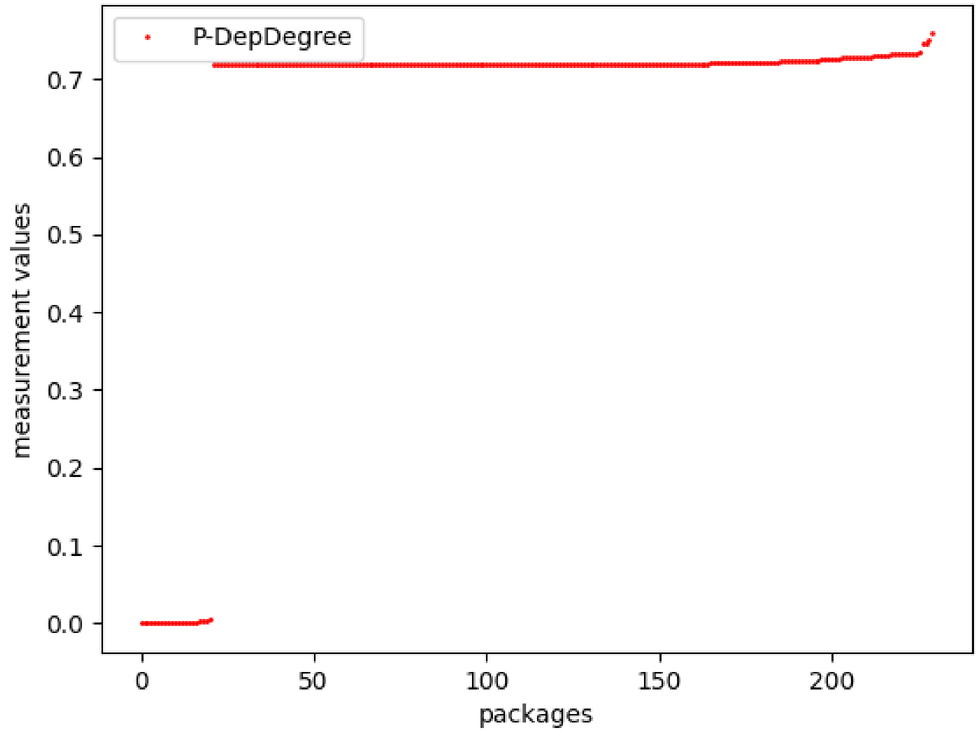
1. Analyze distribution of measurement values
2. Compare packages with highest values
3. Identify outliers
4. Evaluate correlation matrix



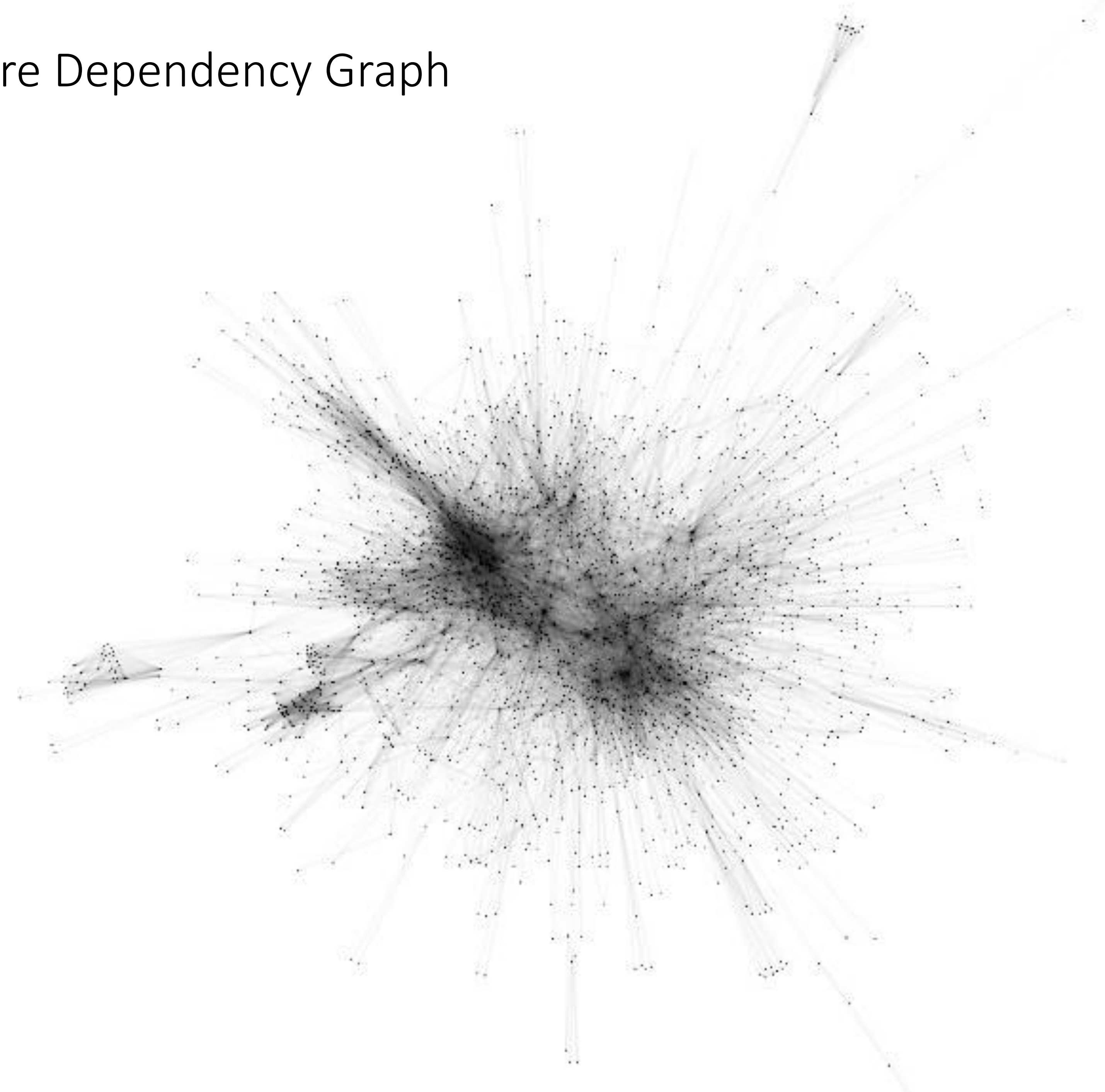
P-DepDegree

- 21 packages with a value close to 0
- 209 packages with a value > 0.71

⇒ Identified subgraph of 2646 classes which all the *TDGs* of packages with a *P-DepDegree > 0.71* share (Core Dependency Graph)

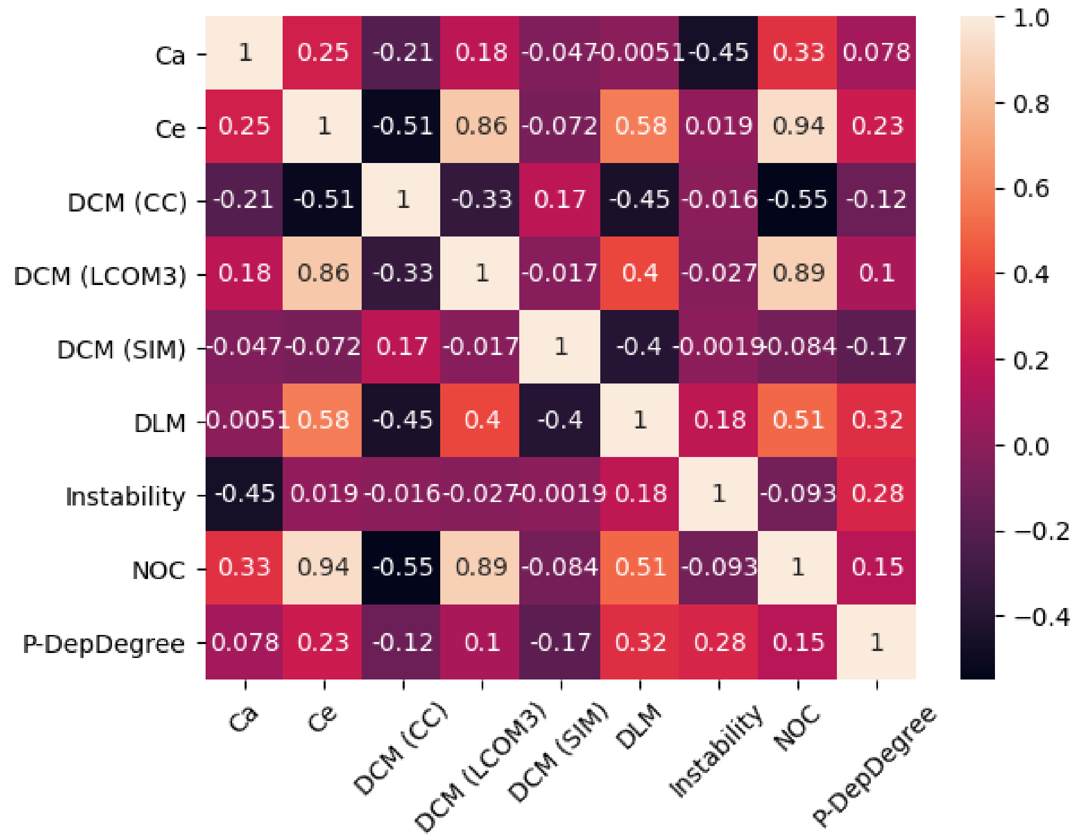


Core Dependency Graph



Correlation Matrix

- $NOC \leftrightarrow Ce = 0.94$
  (strong correlation)
- $NOC \leftrightarrow DCM_{CC} = -0.55$
- $Ce \leftrightarrow DCM_{CC} = -0.51$



$$DCM_{CC}(P) = \frac{\sum_{d \in D_P} cnt_P(d)}{|C_P| * |D_P|}$$

$$NOC(P) = |C_P|$$

Future Work & Conclusion

What's next?

- Further evaluation necessary (to clearly prove usefulness and applicability of the measures)
- Implementation of measures on analysis platform (e.g. SonarQube)
- In-detail analysis of the dependencies of CPAchecker (e.g. based on the core dependency graph)

What's done?

- Proposed 5 package-level dependency measures
- Theoretical Evaluation with Weyuker's Properties
- Practical Evaluation on the example of CPAchecker
- Implementation of Jade

$\Rightarrow$ **3 measures met expectations (*DCM_{CC}, P-DepDegree, DLM*)**


THANK YOU FOR LISTENING
ANY QUESTIONS?