Università degli Studi di Milano - Bicocca
Scuola di Scienze
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di laurea magistrale in Informatica

Relazione di Evolution of Software Systems and Reverse Engineering
Analisi di correlazione tra Architectural Smells

Relazione di:
Refolli Francesco 865955

Anno Accademico 2024-2025

Indice

1 Introduzione **1**

2 Analisi semplice **5**

3 Analisi sequenziale **6**
3.1 Un pò di definizioni . . . 6
3.2 Co-occorrenza istantanea . . . 6
3.3 Co-occorrenza incrementale . . . 7
3.4 Dipendenza in transizione . . . 9
3.4.1 coordinamento . . . 9
3.4.2 alternativa . . . 9
3.4.3 precedenza . . . 10
3.4.4 successione . . . 10

4 Validazione **11**
4.1 Rappresentanza del campione . . . 11
4.2 Il fattore Arcan . . . 12
4.3 Strumenti logico-matematici . . . 12

5 Conclusione **13**

1 Introduzione

Lo scopo del progetto è analizzare diversi progetti con Arcan [4] per estrarre gli Architectural Smells e analizzare i dati di co-occorrenza nella stessa versione e nel tempo per rilevare correlazioni tra gli smells. Per fare questo mi sono servito di una Pipeline [2] (Figura 1) creata ad hoc in grado di scaricare ed analizzare i progetti e i loro sottoprodotti in modo completamente automatico. Volendo testare la pipeline anche su progetti non Java, per verificare l'attendibilità dell'analisi, il Job di importazione repository, oltre a scaricarli, ne analizza il contenuto per identificare il linguaggio di programmazione con cui impostare l'analisi di Arcan (che lo richiede strettamente). Con tre richieste in sequenza si provoca l'importazione dei progetti designati nel documento YAML in ingresso, l'analisi con Arcan e la successiva analisi dei dati degli smell per rilevare le correlazioni. I prodotti delle analisi sono quindi scaricabili dall'interfaccia grafica che dispone anche di un sistema di code per notificare all'utente quando l'analisi è terminana. Si mette in evidenza che è stato possibile analizzare solo un sottoinsieme (hubLikeDep, cyclicDep, unstableDep, godComponent, deepHierarchy) degli smell riconosciuti da Arcan in quanto l'analisi degli altri smells (wideHierarchy, cyclicHierarchy) impiega troppo tempo, memoria o il processo non termina [1] .

[1] Da qui in poi gli smells verranno chiamati per acronimo per brevità. hubLikeDep = HD; cyclicDep = CD; unstableDep = UD; godComponent = GC; deepHierarchy = DH.

Tabella 1: Progetti nel Gruppo di Controllo

Owner	Project	Branch	Language	LoC
frefolli	lartc	master	CPP	6 K
frefolli	stardock	master	CPP	552
frefolli	packer	master	CPP	1 K
frefolli	arcan-pipeline	master	CPP	14 K
ilpincy	argos3	master	CPP	148 K

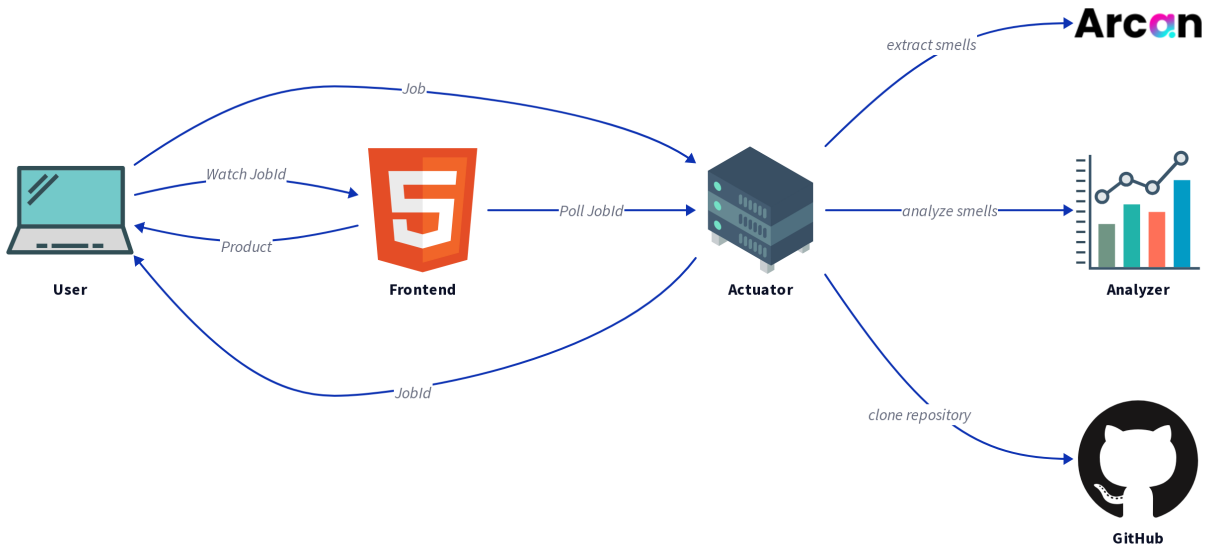

Figura 1: La Pipeline per l'Analisi

I progetti analizzati si dividono nel gruppo di controllo (Tabella 1) e nel gruppo di analisi (Tabelle 2 e 3). Conosco molto bene i progetti del gruppo di controllo e di conseguenza è facile verificare la qualità dell'analisi di Arcan, e di conseguenza verificare l'attendibilità di una conseguente analisi dei sottoprodotti. È inoltre utile per verificare il funzionamento della pipeline senza ricorrere a progetti eccessivamente complessi o presanti. Siccome l'analisi di Arcan è molto pesante, l'analisi *sequenziale* sulle versioni storiche (cadenzate con intervalli di 6 mesi) è stata effettuata su un sottoinsieme dei progetti (Tabella 4), mentre l'analisi *semplice* è effettuata sull'ultima versione di tutti i progetti.

Tabella 2: Progetti nel Gruppo di Analisi Semplice (1)

Owner	**Project**	**Branch**	**Language**	**LoC**
alibaba	druid	master	JAVA	413 K
alibaba	fastjson2	main	JAVA	419 K
alibaba	jetcache	master	JAVA	19 K
alibaba	canal	master	JAVA	91 K
alibaba	nacos	develop	JAVA	200 K
alibaba	QLExpress	master	JAVA	14 K
apache	activemq	main	JAVA	438 K
apache	jena	main	JAVA	676 K
apache	jackrabbit	trunk	JAVA	332 K
apache	archiva	master	JAVA	145 K
apache	geode	develop	JAVA	1.404 M
apache	karaf	main	JAVA	129 K
apache	phoenix	master	JAVA	463 K
apache	solr	main	JAVA	729 K
apache	rocketmq	develop	JAVA	247 K
apache	jmeter	master	JAVA	163 K
apache	shenyu	master	JAVA	179 K
apache	lucene	main	JAVA	875 K
apache	groovy	master	JAVA	202 K
apache	maven	master	JAVA	175 K
apache	ant	master	JAVA	146 K
apache	kafka	trunk	JAVA	822 K
apache	paimon	master	JAVA	320 K
apache	ranger	master	JAVA	351 K
apache	calcite	main	JAVA	443 K
apache	commons-bcel	master	JAVA	46 K
apache	commons-beanutils	master	JAVA	25 K
apache	commons-bsf	master	JAVA	6 K
apache	commons-chain	master	JAVA	9 K
apache	commons-cli	master	JAVA	11 K
apache	commons-codec	master	JAVA	24 K
apache	commons-collections	master	JAVA	72 K
apache	commons-compress	master	JAVA	73 K
apache	commons-configuration	master	JAVA	51 K
apache	commons-crypto	master	JAVA	7 K
apache	commons-csv	master	JAVA	9 K

Tabella 3: Progetti nel Gruppo di Analisi Semplice (2)

Owner	**Project**	**Branch**	**Language**	**LoC**
apache	commons-dbcp	master	JAVA	31 K
apache	commons-dbutils	master	JAVA	7 K
apache	commons-digester	master	JAVA	20 K
apache	commons-email	master	JAVA	10 K
apache	commons-exec	master	JAVA	3 K
apache	commons-fileupload	master	JAVA	6 K
apache	commons-functor	master	JAVA	21 K
apache	commons-geometry	master	JAVA	74 K
apache	commons-graph	master	JAVA	10 K
apache	commons-imaging	master	JAVA	42 K
apache	commons-io	master	JAVA	55 K
apache	commons-jci	master	JAVA	4 K
apache	commons-jcs	master	JAVA	53 K
apache	commons-jelly	master	JAVA	31 K
apache	commons-jexl	master	JAVA	39 K
apache	commons-jxpath	master	JAVA	26 K
apache	commons-lang	master	JAVA	96 K
apache	commons-logging	master	JAVA	6 K
apache	commons-math	master	JAVA	148 K
apache	commons-net	master	JAVA	26 K
apache	commons-numbers	master	JAVA	59 K
apache	commons-ognl	master	JAVA	20 K
apache	commons-pool	master	JAVA	16 K
apache	commons-proxy	master	JAVA	6 K
apache	commons-rdf	master	JAVA	8 K
apache	commons-release-plugin	master	JAVA	1 K
apache	commons-rng	master	JAVA	49 K
apache	commons-scxml	master	JAVA	14 K
apache	commons-signing	master	JAVA	302
apache	commons-statistics	master	JAVA	40 K
apache	commons-testing	master	JAVA	584
apache	commons-text	master	JAVA	27 K
apache	commons-validator	master	JAVA	16 K
apache	commons-vfs	master	JAVA	37 K
apache	commons-weaver	master	JAVA	5 K

Tabella 4: Progetti nel Gruppo di Analisi Sequenziale

Owner	Project	Branch	Language	LoC
alibaba	nacos	develop	JAVA	200 K
apache	ant	master	JAVA	146 K
apache	calcite	main	JAVA	443 K
apache	maven	master	JAVA	175 K
apache	jmeter	master	JAVA	163 K

2 Analisi semplice

Dopo aver analizzato con la pipeline i progetti del gruppo di analisi semplice (Tabelle 2 e 3), ci si trova con una matrice di co-occorrenza per ogni progetto in formato JSON, come nel Listato 1. Per ogni tipo di smell si contano le occorrenze (nei componenti) e si controlla nel luogo di occorrenza (nel componente) quante volte appare insieme ad un altro tipo di smell. Quindi i dati sono stati aggregati per co-occorrenza (*smellA-smellB*) e tramite una media delle percentuali di co-occorrenza nei vari progetti (media pesata sul numero di occorrenze in modo da premiare le coppie di smell che co-occorrono molte volte nello stesso progetto e per non dare eccessivo peso alle potenziali fluttuazioni statistiche). La Figura 2 contiene una heatmap che riassume i risultati dell'analisi di correlazione.

Listing 1: Esempio di JSON di analisi semplice del progetto apache/commons-validator

```
{"cyclicDep" :
  {"cooccurrence" :
    {"hublikeDep" :
      {"absolute" : 1,
       "percentage" : 0.045454545454545456}},
   "count" : 22},
 "hublikeDep" :
  {"cooccurrence" :
    {"cyclicDep" :
      {"absolute" : 1,
       "percentage" : 0.5}},
   "count" : 2}}
```

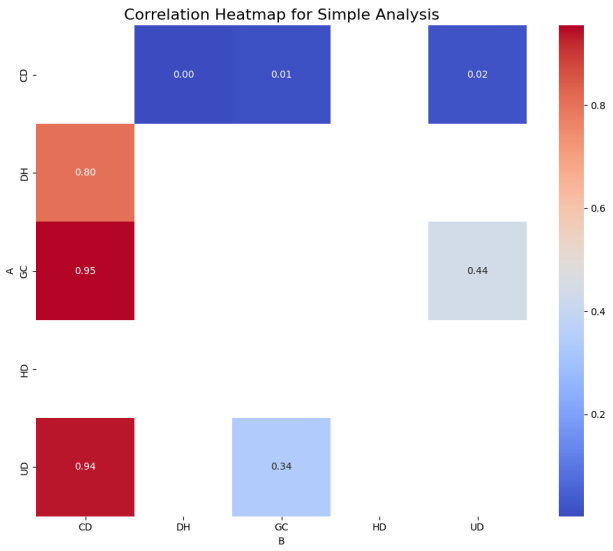

Figura 2: Heatmap di correlazione riassuntiva dei progetti del gruppo di analisi semplice

Osservazioni Si osserva che l'apparizione degli smell UD, GC, DH è fortemente correlata alla presenza di uno smell di tipo CD. Inoltre si nota anche una molto debole correlazione tra GC e UD.

Conferma con test d'ipotesi Siccome sono emerse delle correlazioni empiriche, sono stati eseguiti dei test d'ipotesi per confermare questi risultati. Un t-test [3] sulla media ($H_0 : \mu >= 0$, $H_1 : \mu < 0$) ha confermato e validato le informazioni emerse con p-value molto basso [5]. Si riporta per ridondanza anche questo dato.

- $UD \rightarrow CD$, con p-value 5.343893978079183e-18
- $GC \rightarrow CD$, con p-value 2.46097200582896e-19
- $DH \rightarrow CD$, con p-value 0.0002714398470104095

3 Analisi sequenziale

I progetti del gruppo di analisi sequenziale (Tabella 4) sono stati analizzati una volta per ogni loro versione, il cui risultato sono gli smell associati a ciascun componente in ogni versione. Sono stati eseguiti a questo punto diversi tipi di analisi processando questi dati sequenziali.

3.1 Un pò di definizioni

$I_{c,v}$ Per ogni componente C il suo stato istantaneo $I_{c,v}$ è espresso dal vettore $< CD, HD, GC, DH, UD >$ dove ciascun elemento $x \in AS$ assume il valore $x = 1$ sse quello smell è presente nel documento nella versione V. Ovvero lo stato esprime la presenza istantanea.

$D_{c,v}$ Per ogni componente C il suo stato differenziale $D_{c,v}$ è espresso dal vettore $< CD, HD, GC, DH, UD >$ dove ciascun elemento $x \in AS$ assume il valore $x = 1$ sse quello smell è presente nel documento nella versione V in una quantità superiore rispetto alla versione $V - 1$, vice versa $x = -1$ sse è presente in una quantità inferiore. Ovvero lo stato esprime la direzione della differenza.

3.2 Co-occorrenza istantanea

Utilizzando gli stati istantanei $I_{c,v}$ si effettua un sondaggio statistico della proposizione $AS_a \rightarrow AS_b$, ovvero lo smell A implica lo smell B. Ovvero risponde al quesito: è vero che quando lo smell A è presente allora anche lo smell B è presente? Gli stati in cui nessuno dei due smell appare sono state escluse dal conteggio perchè viziano i risultati facendo figurare gli smell che appaiono molto raramente come correlati.

Nella Figura 3 sono riportate le probabilità che $AS_a \rightarrow AS_b$ per ogni coppia di smell diversi (Nota: AS_a nelle colonne, AS_b nelle righe).

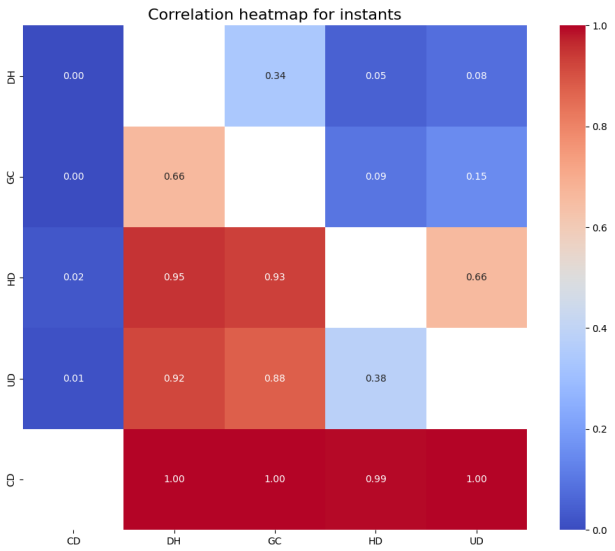

Figura 3

Osservazioni Oltre alla fortissima correlazione istantanea tra CD e gli altri smells (ovvero CD è molto spesso presente quando gli altri smells sono presenti), si nota anche che altri gruppi di smells risultano cooccorrenti. In particolare:

- Quando c'e' GC molto spesso ci sono HD e UD
- Quando c'e' DH molto spesso ci sono HD e UD

Conferma con test d'ipotesi Siccome sono emerse delle correlazioni empiriche, sono stati eseguiti dei test d'ipotesi per confermare questi risultati. Un t-test [3] sulla media ($H_0 : \mu >= 0$, $H_1 : \mu < 0$) ha confermato e validato le informazioni emerse con p-value molto basso [5]. Si riporta per ridondanza anche questo dato.

- $CD \to DH$, con p-value 0.0
- $CD \to GC$, con p-value 0.0
- $CD \to HD$, con p-value 0.0
- $CD \to UD$, con p-value 0.0
- $GC \to DH$, con p-value 1.435256286842854e-06
- $HD \to DH$, con p-value 4.734253100288627e-255
- $HD \to GC$, con p-value 4.256981210065098e-230
- $HD \to UD$, con p-value 5.489034077245229e-37
- $UD \to DH$, con p-value 3.901546614793626e-133

3.3 Co-occorrenza incrementale

Utilizzando gli stati differenziali $D_{c,v}$ si effettua un sondaggio statistico della proposizione $AS_a \to AS_b$, ovvero lo smell A implica lo smell B. Ovvero risponde al quesito: è vero che quando lo smell A incrementa allora anche lo smell B incrementa? (e vice versa: è vero che quando lo smell A decrementa allora anche lo smell B decrementa?). La transizioni in cui nessuno dei due smell cambia di stato (ex: $00 \to 00$) sono state escluse dal conteggio perchè viziano i risultati facendo figurare gli smell che appaiono molto raramente come correlati.

Nella Figure 4 e 5 sono riportate le probabilità che $AS_a \to AS_b$ per ogni coppia di smell diversi (Nota: AS_a nelle colonne, AS_b nelle righe).

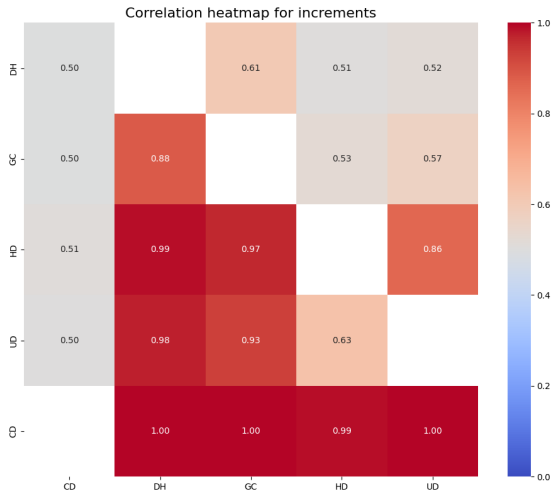

Figura 4

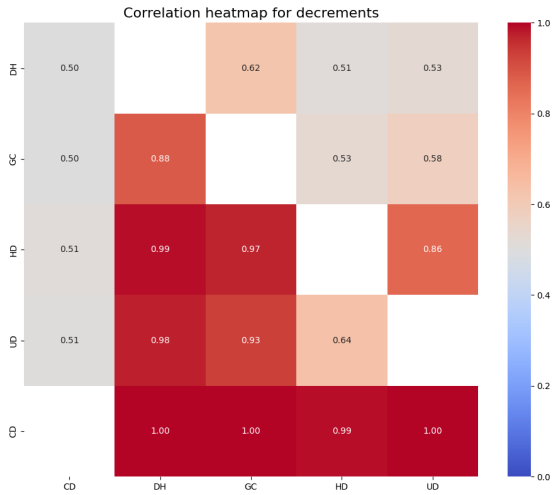

Figura 5

Osservazioni Oltre alla fortissima correlazione differenziale tra CD e gli altri smells (ovvero CD è molto spesso in aumento/diminuzione quando gli altri smells sono in aumento/diminuzione), si nota anche che gli incrementi/decrementi degli altri tipi di smells (DH, GC, HD, UD) sono tra loro fortemente correlati.

Conferma con test d'ipotesi Siccome sono emerse delle correlazioni empiriche, sono stati eseguiti dei test d'ipotesi per confermare questi risultati. Un t-test [3] sulla media ($H_0 : \mu >= 0$, $H_1 : \mu < 0$) ha confermato e validato le informazioni emerse con p-value molto basso [5]. Si riporta per ridondanza anche questo dato.

incrementi

- $CD \rightarrow DH$, con p-value 0.0
- $CD \rightarrow GC$, con p-value 0.0
- $CD \rightarrow HD$, con p-value 0.0
- $CD \rightarrow UD$, con p-value 0.0
- $GC \rightarrow DH$, con p-value 0.0006597639058512969
- $HD \rightarrow DH$, con p-value 3.5275628507291554e-83
- $HD \rightarrow GC$, con p-value 4.611684131720051e-72
- $HD \rightarrow UD$, con p-value 3.193730087158467e-25
- $UD \rightarrow DH$, con p-value 1.8998519322958519e-31
- $UD \rightarrow GC$, con p-value 1.6012390509547418e-22

decrementi

- $CD \rightarrow DH$, con p-value 0.0
- $CD \rightarrow GC$, con p-value 0.0
- $CD \rightarrow HD$, con p-value 0.0
- $CD \rightarrow UD$, con p-value 0.0
- $GC \rightarrow DH$, con p-value 0.0010106968491028834
- $HD \rightarrow DH$, con p-value 4.385944110852566e-82
- $HD \rightarrow GC$, con p-value 7.16476232040879e-72

- $HD \to UD$, con p-value 2.4581552135851697e-25
- $UD \to DH$, con p-value 1.404439800557754e-30
- $UD \to GC$, con p-value 3.83035233087132e-22

3.4 Dipendenza in transizione

Utilizzando gli stati istantanei $I_{c,v}$ si effettua un sondaggio statistico sui tipi di transizioni (nel passaggio di versione) a coppie di smells. La transizioni in cui nessuno dei due smell cambia di stato (ex: $00 \to 00$) sono state escluse dal conteggio perchè viziano i risultati facendo figurare gli smell che appaiono molto raramente come correlati.

3.4.1 coordinamento

La transizione indica che gli smell cambiano di stato allo stesso modo nello stesso istante ($00 \to 11 \vee 11 \to 00$). Riportato in Figura 6.

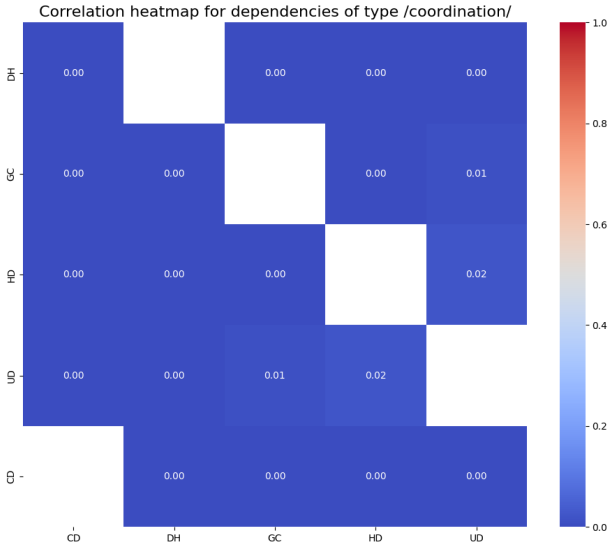

Figura 6: Heatmap sulle relazioni di coordinamento

Osservazioni Non emergono correlazioni significative.

3.4.2 alternativa

La transizione indica che gli smell non possono coesistere, se appare uno scompare l'altro - indica anche successione stretta - ($10 \to 01$). Riportato in Figura 7.

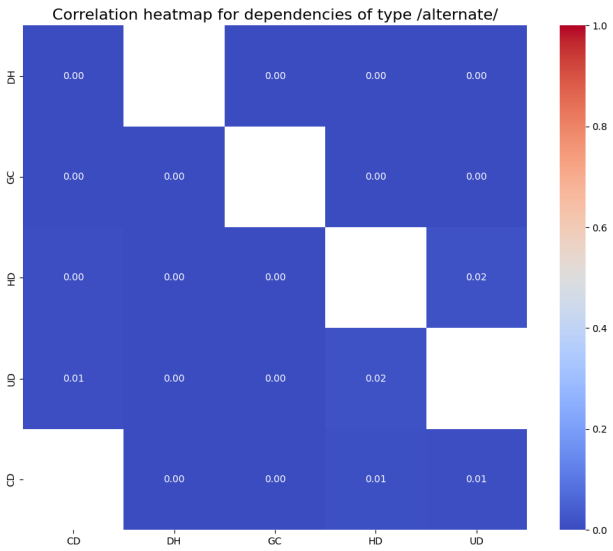

Figura 7: Heatmap sulle relazioni di alternatività

Osservazioni Non emergono correlazioni significative.

3.4.3 precedenza

La transizione indica che la comparsa di uno smell è subordinata temporalmente alla presenza di un'altro (01 → 11). Riportato in Figura 8.

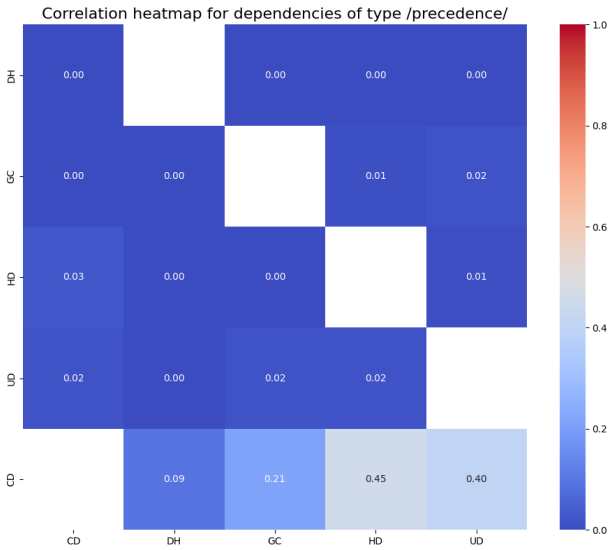

Figura 8: Heatmap sulle relazioni di precedenza

Osservazioni Non emergono correlazioni significative, oltre al fatto che spesso gli smell di tipo CD sono già presenti quando gli altri tipi minori di smell compaiono.

3.4.4 successione

La transizione indica che lo smell sopravvive temporalmente alla scoparsa di un altro smell (11 → 10). Questa transizione può essere vista come il complementare della *precedenza* e può indicare un possibile legame di dipendenza tra smells. Riportato in Figura 9.

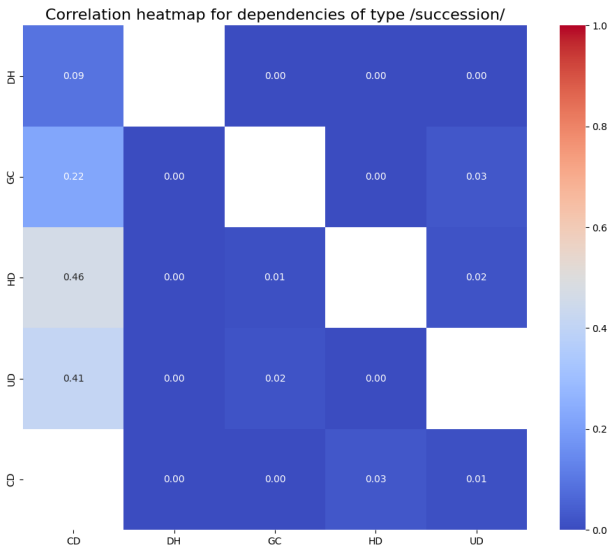

Figura 9: Heatmap sulle relazioni di successione

Osservazioni Non emergono correlazioni significative, oltre al fatto che spesso gli smell di tipo CD rimangono presenti quando gli altri tipi minori di smell scompaiono.

4 Validazione

4.1 Rappresentanza del campione

Nelle Figure 10, 11 e 12 sono riportati i grafici di distribuzione delle grandenze dei progetti analizzati: i progetti selezionati (con particolare riferimento ai due gruppi di analisi) sono un campione che ben rappresenta la popolazione, data la varietà delle lunghezze e dato il fatto che provengono da due collezioni di software libero rilevante nel settore [2] [3].

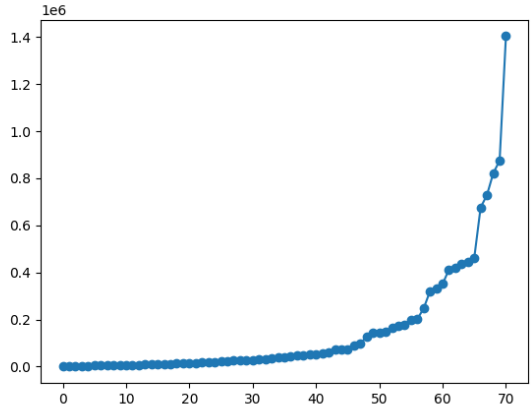

Figura 10: Distribuzione delle lunghezze (in Milioni di LoC) dei progetti del gruppo di semplice

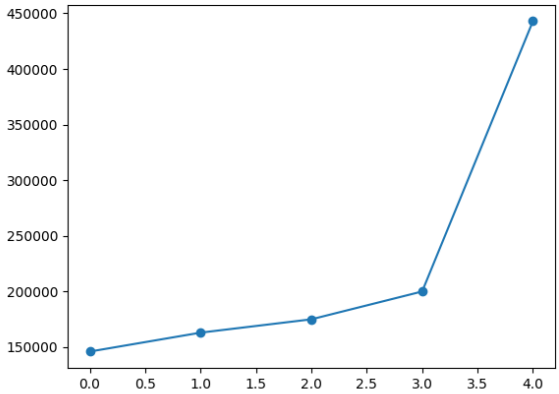

Figura 11: Distribuzione delle lunghezze dei progetti del gruppo di analisi sequenziale

[2] Alibaba
[3] Apache

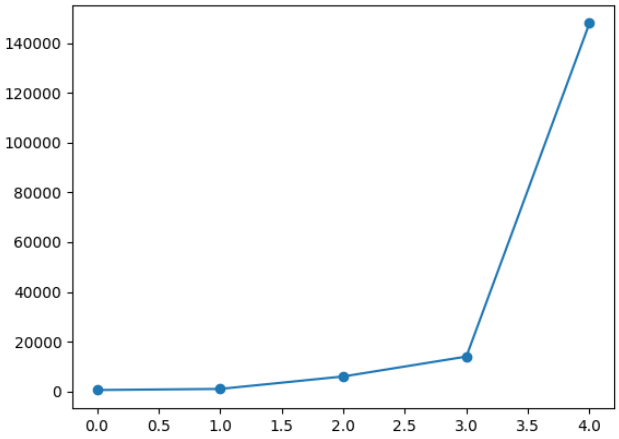

Figura 12: Distribuzione delle lunghezze dei progetti del gruppo di controllo

4.2 Il fattore Arcan

Analizzando i progetti ho identificato un pattern significativo. Si riportano nelle Figure 13 e 14 i grafici a torta rappresentanti le quote di smells riconosciuti nei progetti del gruppo di analisi semplice. Come si può notare, gli smells riconosciuti da Arcan sono estremamente sbilanciati verso le dipendenze cicliche [4], il che fa sorgere il legittimo dubbio che l'analisi di Arcan possa essere inesatta.

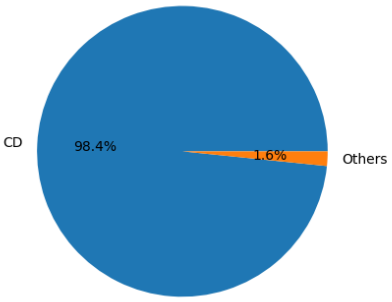

Figura 13: Grafico a torta con le quote di smells riconosciuti nel gruppo di analisi semplice (Vista macroscopica)

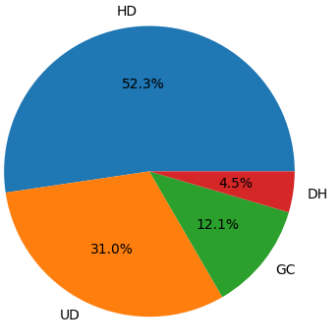

Figura 14: Grafico a torta con le quote di smells riconosciuti nel gruppo di analisi semplice (Vista in dettaglio della quota 'Altri')

Ebbene il gruppo di controllo è stato pensato propriò per dipanare questo dubbio. Tali progetti sono stati scelti tutti in virtù del fatto che sono stati da me sviluppati o mantenuti di recente, per cui risulta facile verificare la presenza effettiva degli smells rilevati da Arcan. Si prenda come esempio il compilatore LartC [1] (C++). L'analisi di Arcan ha restituito gli smells elencati nella Tabella 5. Con una rapida verifica (a colpo sicuro) si è rilevato che le dipendenze cicliche segnalate non sono in realtà presenti nel codice. Sorge quindi il legittimo dubbio che buona parte degli smells di tipo cyclicDependency siano in realtà inattendibili (il che spiegherebbe la precedentemente discussa implicazione degli altri smell verso CD) [5].

4.3 Strumenti logico-matematici

Nell'analisi degli smells sono stati usati concetti quali *istantanea* o *differenziale* di stato di componente, il concetto di *transizione* e di *implicazione*, sufficientemente semplici da spiegare e giustificare. In particolare le

[4] si noti che questi numeri rispecchiano il numero di smells distinti, non il numero di partecipazioni!
[5] Anche altri smells sono fuorvianti o borderline, ma non è questo il punto che si cerca di trasmettere

Tabella 5: Smells rilevati in LartC

smellType	smellId	component	componentId
cyclicDep	3650	variants.cc	175
cyclicDep	3650	internal_errors.cc	625
cyclicDep	3658	frefolli_lartc.git.src.lartc.ast	731
cyclicDep	3658	frefolli_lartc.git.src.lartc	583
cyclicDep	3658	frefolli_lartc.git.src.lartc.ast.declaration	107
hubLikeDep	3667	expression.cc	1158
hubLikeDep	3690	file_db.cc	1179

implicazioni sugli incrementi sono state equiparate a quelle sulle istantanee tramite la riduzione a forma comune booleana disgiunta in incrementi positivi e negativi. Tutte le correlazioni significative sono state accompagnate da test d'ipotesi (*t-test* [3]) per verificarne l'attendibilità usando un p-value di soglia molto basso [5] per evitare falsi positivi.

5 Conclusione

Riassunto Sono stati analizzati con Arcan 70+ progetti alla ricerca di correlazioni o co-occorrenza tra architectural smells. Delle correlazioni emerse empiricamente, al netto dei problemi di Arcan e dei test d'ipotesi condotti, le uniche considerabili attendibili coinvolgono gli smells hublikeDependency, godComponent, deepHierarchy e unstableDependency.

Interpretazione I risultati dell'analisi sono spiegabili considerando le caratteristiche degli smell coinvolti. Un God Component è probabilmente anche un Hublike Dependency e un Unstable Dependency se concentra molte funzionalità e molti componenti dipendono da esso. Allo stesso modo, una Deep Hierarchy può essere un Hublike Dependency e un Unstable Dependency in molti progetti, specie se la gerarchia di classi è utilizzata in molti punti del sistema.

Sviluppi futuri Per generalizzare meglio all'intero mondo del software libero i risultati del progetto bisognerebbe analizzare più progetti (magari anche quelli proprio considerati universalmente *inattaccabili* dalla community come *sqlite* o *busybox*) ed estendere l'analisi ad altri linguaggi di programmazione quali C, C++, Python, Rust ... cosa al momento difficile visto che Arcan non supporta molti linguaggi, è molto lento, può riscontrare problemi anche con quelli supportati [6] e non è preciso nella rilevazione di alcuni tipi di smells.

Bibliografia

[1] Refolli F. *Lart compiler in C++ and LLVM IR*. https://github.com/frefolli/lartc. 2024.

[2] Refolli F. *Pipeline for running Arcan and analyzing smells*. https://github.com/frefolli/arcan-pipeline. 2024.

[3] J. Sun. *The Microbiome in Health and Disease - Student's t-test*. Progress in Molecular Biology and Translational Science. Academic Press, 2020, p. 397. ISBN: 9780128200018. URL: https://books.google.it/books?id=kiToDwAAQBAJ.

[4] Francesca Arcelli Fontana et al. «Arcan: A tool for architectural smells detection». In: *2017 IEEE International Conference on Software Architecture Workshops (ICSAW)*. IEEE. 2017, pp. 282–285.

[6] Per qualche ragione è impossibile eseguire analisi su progetti Python in quanto non riesce a trovare un runtime anche se presente nell'ambiente

[5] Megan L Head et al. ≪The extent and consequences of p-hacking in science≫. In: *PLoS biology* 13.3 (2015), e1002106.