1 PROJECT SCRIPTS

The scripts, queries, and raw data used to generate the statistics in the paper are available, with documentation, in this GitHub repository. Please see the repository for usage instructions and additional implementation details. See this link for the final WorkflowsExplorer tool.

2 GITHUB PROJECTS

See Table 1 for the list of GitHub Projects studied. This table also contains information about the number of PRs and Issues in each project.

3 FURTHER PROJECT STATISTICS

Figure 1 shows the number of components within each project and the project network’s density across the number of nodes in the project. In general, it is possible to observe that we studied a wide variety of projects, i.e., good coverage of projects of various sizes and component counts. As expected, the larger the number of nodes, the larger the number of components. As also expected, the larger the number of nodes, the smaller the density since more edges are necessary to keep the density constant. Furthermore, it is important to notice the density values (y-axis) that are very small, which means that the resulting PR-Issue graphs from each project are mostly disconnected, although with variation among them.

4 CYPER QUERIES

The following are the Cypher queries used to perform our topological analysis. These can also be found in our source code here.

As explained in section 6.10 of the submission, the process of querying workflow types was affected by the order in which these types were matched. In particular, because some of the topological definition of workflows overlapped, even though these workflow practices semantically exclude each other. To address this problem, we first queried for the Dependent PRs and hubs and removed those instances from matches of the other workflow types. This is the
Table 1: GitHub projects studied

<table>
<thead>
<tr>
<th>Project Name</th>
<th># of Issues</th>
<th># of PRs</th>
</tr>
</thead>
<tbody>
<tr>
<td>MithrilJS/mithril.js</td>
<td>1259</td>
<td>1218</td>
</tr>
<tr>
<td>tristanhimmelman/ObjectMapper</td>
<td>799</td>
<td>328</td>
</tr>
<tr>
<td>archriss/react-native-snap-carousel</td>
<td>762</td>
<td>167</td>
</tr>
<tr>
<td>roboguice/roboguice</td>
<td>279</td>
<td>72</td>
</tr>
<tr>
<td>mflow/mflow</td>
<td>2338</td>
<td>3878</td>
</tr>
<tr>
<td>Project-OSRM/osrm-backend</td>
<td>4146</td>
<td>2131</td>
</tr>
<tr>
<td>tiny-dnn/tiny-dnn</td>
<td>576</td>
<td>475</td>
</tr>
<tr>
<td>volatilityfoundation/volatility</td>
<td>687</td>
<td>137</td>
</tr>
<tr>
<td>amphp/amp</td>
<td>217</td>
<td>171</td>
</tr>
<tr>
<td>App-vNext/Polly</td>
<td>557</td>
<td>392</td>
</tr>
<tr>
<td>BurntSushi/toml</td>
<td>197</td>
<td>162</td>
</tr>
<tr>
<td>chaijs/chai</td>
<td>877</td>
<td>583</td>
</tr>
<tr>
<td>Flipboard/bottomsheet</td>
<td>143</td>
<td>70</td>
</tr>
<tr>
<td>grpc/grpc-web</td>
<td>657</td>
<td>594</td>
</tr>
<tr>
<td>jhen0409/react-native-debugger</td>
<td>449</td>
<td>260</td>
</tr>
<tr>
<td>John-Lluch/SWRevealViewController</td>
<td>753</td>
<td>70</td>
</tr>
<tr>
<td>rematch/rematch</td>
<td>504</td>
<td>447</td>
</tr>
<tr>
<td>stacktracejs/stacktrace.js</td>
<td>178</td>
<td>51</td>
</tr>
<tr>
<td>jupyterhub/jupyterhub</td>
<td>2117</td>
<td>1801</td>
</tr>
<tr>
<td>metafizzy/flickity</td>
<td>1142</td>
<td>96</td>
</tr>
<tr>
<td>mongo/restangular</td>
<td>1223</td>
<td>269</td>
</tr>
<tr>
<td>nytimes/NYPhotoViewer</td>
<td>156</td>
<td>194</td>
</tr>
<tr>
<td>Rapptz/discord.py</td>
<td>2599</td>
<td>2297</td>
</tr>
<tr>
<td>SwipeCellKit/SwipeCellKit</td>
<td>333</td>
<td>111</td>
</tr>
<tr>
<td>transitive-bullshit/create-react-library</td>
<td>283</td>
<td>86</td>
</tr>
<tr>
<td>iron/iron</td>
<td>259</td>
<td>381</td>
</tr>
<tr>
<td>jonschlinkert/remarkable</td>
<td>297</td>
<td>118</td>
</tr>
<tr>
<td>redis/redis-rb</td>
<td>586</td>
<td>501</td>
</tr>
<tr>
<td>MagicStack/uvloop</td>
<td>282</td>
<td>195</td>
</tr>
<tr>
<td>summernote/summernote</td>
<td>3232</td>
<td>1084</td>
</tr>
<tr>
<td>tsayen/dom-to-image</td>
<td>345</td>
<td>78</td>
</tr>
<tr>
<td>varvet/pundit</td>
<td>396</td>
<td>338</td>
</tr>
<tr>
<td>go-chi/chi</td>
<td>421</td>
<td>307</td>
</tr>
<tr>
<td>kubernetes-sigs/kustomize</td>
<td>1916</td>
<td>2792</td>
</tr>
<tr>
<td>TypeStrong/ts-node</td>
<td>1104</td>
<td>594</td>
</tr>
<tr>
<td>casesandberg/react-color</td>
<td>514</td>
<td>346</td>
</tr>
<tr>
<td>cruffenach/CRToast</td>
<td>113</td>
<td>106</td>
</tr>
<tr>
<td>duo-labs/cloudmapper</td>
<td>531</td>
<td>398</td>
</tr>
<tr>
<td>sosedoff/pgweb</td>
<td>313</td>
<td>254</td>
</tr>
<tr>
<td>zaach/jison</td>
<td>282</td>
<td>119</td>
</tr>
<tr>
<td>Activiti/Activiti</td>
<td>2048</td>
<td>1962</td>
</tr>
<tr>
<td>ag-grid/ag-grid</td>
<td>4602</td>
<td>723</td>
</tr>
<tr>
<td>cglib/cglib</td>
<td>112</td>
<td>92</td>
</tr>
<tr>
<td>deployphp/deployer</td>
<td>1694</td>
<td>1153</td>
</tr>
<tr>
<td>fantasyland/fantasy-land</td>
<td>168</td>
<td>164</td>
</tr>
<tr>
<td>rauchl/slackin</td>
<td>220</td>
<td>173</td>
</tr>
<tr>
<td>tensorpack/tensorpack</td>
<td>1347</td>
<td>205</td>
</tr>
<tr>
<td>ptomasroos/react-native-scrollable-tab-view</td>
<td>870</td>
<td>308</td>
</tr>
<tr>
<td>marko/js/marko</td>
<td>1011</td>
<td>792</td>
</tr>
<tr>
<td>rldwka/sinopia</td>
<td>391</td>
<td>101</td>
</tr>
<tr>
<td>imbrn/vsn</td>
<td>62</td>
<td>171</td>
</tr>
<tr>
<td>apache/dubbo</td>
<td>5310</td>
<td>4912</td>
</tr>
<tr>
<td>pagekit/pagekit</td>
<td>765</td>
<td>198</td>
</tr>
<tr>
<td>deepinsight/insightface</td>
<td>1931</td>
<td>119</td>
</tr>
<tr>
<td>shadowsocks/shadowssocks-manager</td>
<td>537</td>
<td>136</td>
</tr>
<tr>
<td>sdc-alibaba/SUI-Mobile</td>
<td>1001</td>
<td>65</td>
</tr>
</tbody>
</table>

Figure 1: PR-Issue component, edge, and density information per project.

The reason why a set of component IDs is presented in the beginning of the Cypher specification for some queries.

call {
  match (i:issue {status: "closed"})-[r {labels: "fixes"}]-(:project {name: pr.pull_request})
  with i, collect(distinct pr) as pull_requests, collect (distinct pr.user) as users, collect(r) as
  match_relationships, max(pr.creation_date) as max_date, min(pr.creation_date) as min_date
}
with [43529, 92173, 56349, 56350, 56355, 53854, 53855, 53860, 53861, 61038, 57447, 61029, 61035, 68741, 68742, 92057, 68768, 15619, 6833, 3252, 3253, 57529, 6842, 57530, 98308, 98299, 81991, 33998, 33999, 33991, 57552, 87767, 87768, 87784, 6892, 6893, 34303, 40361, 69361, 69362, 69363, 69366, 54816, 54817, 54018, 34058, 3347, 6950, 98932, 98933, 90959, 81229, 81238, 86349, 81234, 43353, 36698, 36699, 43354, 86361, 36703, 60257, 60258, 60259, 69990, 69991, 70008, 39816, 39817, 39818, 83356, 83357, 83358, 28581, 28583, 88088, 88088, 88084, 86446, 29104, 29105, 29106, 15806, 87487, 87496, 87497, 29132, 48428, 48421, 48422, 9199, 92150, 92151, 88089, 57342, 57343] as all_ids // see cypher_scripts/fetch_all_pr_stack_ids

match (hub)-[r:pull_request { status: "merged"}]->(pr2:pull_request { status: "merged"}),(i2:issue { status:"closed"})-[r2]-[pr:pull_request { status:"merged"}]
where ((i2.creation_date > pr.creation_date or i2.creation_date > i.creation_date) and i.number <> i2.number and p.number <> p2.number and not id(p) in exclude_ids and not id(p2) in exclude_ids
return collect(distinct id(i))+collect(distinct id(p))+collect(distinct id(p2)) as known_consq

Listing 1: Competing PRs Workflow Type Query

call {
  with known_hubs
  with known_hubs=[88088, 92885] as exclude_ids
  match (i:issue { status:"closed"})-[r {labels:"fixes"}]-[ pr:pull_request]
  with i, collect(distinct pr) as pull_requests, collect ( distinct pr.user) as users, collect(r) as
  match_relationships, max(pr.creation_date) as max_date, min(pr.creation_date) as min_date ,
  known_consq

  where size([p.r in pull_requests where p.r.status="merged "] ) = 1 and size([p.r in pull_requests where p.r.
  status="closed"] ) >= 1 and size(pull_requests) >= 2
  and size(users) > 1 and max_date - min_date <=
  604800 // dates are in Unix timestamps so difference is in seconds
  return collect(distinct id(i)) as known_competition
}

Listing 2: Consequent Issue Workflow Type Query

call {
  with known_hub
  with known_hubs=[88088, 92885] as exclude_ids
  match (i:issue { status:"closed"})-[r {labels:"fixes"}]-[ p:pull_request { status: "merged"}]), (i2:issue { status:"closed"})-[r2]-[ pr: pull_request { status:"merged"}]
  where i2.creation_date > p.creation_date or i2.
  creation_date > i.creation_date and i.number <> i2.
  number and p.number <> p2.number and not id(p) in
  exclude_ids and not id(p2) in exclude_ids
  return collect(distinct id(i))+collect(distinct id(p))+
  collect(distinct id(i2))+collect(distinct id(p2)) as
  known_consq_2
}

Figure 2: Component duration vs. component size across all projects.
Listing 4: Decomposed Issue Workflow Type Query

```cypher
with [43529, 92173, 56349, 56358, 56355, 53854, 53855, 53860, 53861, 61038, 57447, 61029, 61035, 68741, 68742, 98057, 68768, 15019, 6833, 3252, 3253, 57529, 6842, 57530, 98038, 98299, 81001, 33985, 33989, 33990, 33991, 57552, 87767, 87768, 87784, 6892, 6893, 34830, 34031, 69361, 69362, 69363, 69366, 54016, 54017, 54018, 34858, 3347, 6950, 69932, 69933, 90953, 81229, 81238, 86349, 81234, 43353, 36698, 36699, 43354, 86361, 36703, 66257, 66258, 68259, 66990, 69991, 70008, 39816, 39817, 39818, 83356, 83357, 83358, 28581, 28583, 80888, 80889, 86446, 29184, 29105, 29106, 15882, 15886, 87487, 87496, 87497, 29132, 40428, 40424, 40422, 9199, 92150, 92151, 88890, 57342, 57343] as all_ids // see cypher_scripts/fetch_all_pr_stack_ids
match (hub: repository {id: (pr: pull_request). repository_id})<-[:owns]-(:user) -[:contributes_to]->(:repository)
return unique (hub), all_ids

Listing 3: Consequent Issue-PR Workflow Type Query

```cypher
with [43529, 92173, 56349, 56358, 56355, 53854, 53855, 53860, 53861, 61038, 57447, 61029, 61035, 68741, 68742, 98057, 68768, 15019, 6833, 3252, 3253, 57529, 6842, 57530, 98038, 98299, 81001, 33985, 33989, 33990, 33991, 57552, 87767, 87768, 87784, 6892, 6893, 34830, 34031, 69361, 69362, 69363, 69366, 54016, 54017, 54018, 34858, 3347, 6950, 69932, 69933, 90953, 81229, 81238, 86349, 81234, 43353, 36698, 36699, 43354, 86361, 36703, 66257, 66258, 68259, 66990, 69991, 70008, 39816, 39817, 39818, 83356, 83357, 83358, 28581, 28583, 80888, 80889, 86446, 29184, 29105, 29106, 15882, 15886, 87487, 87496, 87497, 29132, 40428, 40424, 40422, 9199, 92150, 92151, 88890, 57342, 57343] as all_ids // see cypher_scripts/fetch_all_pr_stack_ids
match (hub: repository {id: (pr: pull_request). repository_id})<-[:owns]-(:user) -[:contributes_to]->(:repository)
return unique (hub), all_ids
```
with pr, pr2, pr3, number_list, match_relationships, status_list, optional_issue, optional_r, apoc.agg.maxItems(user_list, user_count) as max_users
where size(number_list) = 3 and not (:pull_request) -[:pull_request] size([i in
status_list where i = "merged"] >= 3/2 and
max_users.value = 3/2
)
call apoc.path.subgraphAll(pr, {limit: 50, bfs: true})
yield nodes, relationships
}
with pr, pr2, pr3, [id(pr), id(pr2), id(pr3)] as
all_ids, optional_issue, optional_r, size(collect(
i_node in nodes where i_node.type = "pull_request" and
i_node.number <> pr.number and not id(i_node) in
known_stacks)) as len_nodes, size(nodes) as
len_nodes, nodes, relationships, match_relationships
return pr, pr2, pr3, all_ids, optional_issue,
optional_r, nodes, relationships, toFloat(len_nodes) as proportion
match_relationships

Listing 5: Dependent PRs Workflow Type Query

call {
match (pr:pull_request {status: "merged"})-[r {labels: "fixes"}]->(i:issue)
with pr, collect(distinct i) as issues, collect(distinct
r) as match_relationships
where size([issue in issues where issue.status = "closed"])
> 1
return collect(distinct id(pr)) as known_dependent
}
match (pr:pull_request {status: "merged"})-[r {labels: "fixes"}]->(i:issue)
with pr, collect(distinct i) as issues, collect(distinct
r) as match_relationships, known_dependent
where size([issue in issues where issue.status = "closed"])
> 1
with pr, [issue in issues where issue.status = "closed"] as
closed_issues, match relationships, known_dependent
call apoc.path.subgraphAll(pr, {limit: 50 > size(
closed_issues) + 1 end, bfs: true })
yield nodes, relationships
with pr, closed_issues, nodes, relationships,
match_relationships, size(collect((i_node in nodes
where i_node.type = "pull_request" and i_node.status =
"merged" and i_node.number <> pr.number and not id(i_node)
in known_dependent)) as not_dep, size(nodes)
) as len_nodes
return pr, closed_issues, nodes, relationships,
match_relationships, toFloat(not_dep) / toFloat(
len_nodes) as proportion

Listing 6: Divergent PR Workflow Type Query

call {
match (i:issue)-[r {labels: "duplicate"}]->(i2:issue)
where i.creation_date > i2.creation_date and i2.user <> i.
user
with i, collect(distinct i2) as spoke_issues, collect(
distinct r) as match_relationships, known_dups
where size(spoke_issues) > 1
return apoc.coll.toSet(apoc.coll.flatten(collect(
known_dups))) as known_dups
}
match (i:issue)-[r {labels: "duplicate"}]->(i2:issue)
where i2.creation_date > i.creation_date and i2.user <> i.
user
Listing 8: Extended PR Workflow Type Query

```
call {
  with [43529, 92173, 56349, 56350, 56355, 53854, 53855, 53860, 53861, 61838, 57447, 61029, 61835, 68741, 68742, 98257, 68768, 15019, 6833, 3252, 3253, 57352, 6842, 57350, 90299, 90299, 81091, 33989, 33990, 33991, 57352, 87767, 87768, 87784, 6892, 6893, 34830, 34081, 69361, 69362, 69363, 69366, 54816, 54817, 54018, 34085, 3347, 6950, 90932, 90933, 90953, 81229, 81238, 68349, 81234, 43353, 36698, 36699, 43354, 68631, 36793, 68257, 68258, 68259, 69990, 69991, 70009, 39816, 39817, 39818, 83356, 83357, 83358, 28581, 28583, 80808, 80814, 86446, 29104, 29105, 29106, 15802, 15806, 87487, 87496, 87497, 29132, 40420, 40421, 40422, 9199, 92150, 92151, 80890, 53742, 53743] as all_ids // see cypher_scripts/fetch_all_pr_stack_ids
  match (hub)-[r]->(pr2:pull_request { status: "merged"})
  where ((hub:pull_request and hub.status = "merged") or (hub:issue and hub.status = "closed")) and pr2.
  creation_date < hub.creation_date and not id(hub) in
  all_ids and not id(pr2) in all_ids
  with hub, collect(different pr2) as prs, collect(different
  pr2.user) as users, collect(different id(pr2)) as
  pr_ids
  where size(prs) >= 3 and size(users) >= 2
  return collect(different id(hub))++apoc.coll.toSet(apoc.
  call.flatten(collect(pr_ids))) as known_hubs
}
```

Listing 9: Integrating PR/Issue Hub Workflow Type Query

```
call {
  with [43529, 92173, 56349, 56350, 56355, 53854, 53855, 53860, 53861, 61838, 57447, 61029, 61835, 68741, 68742, 98257, 68768, 15019, 6833, 3252, 3253, 57352, 6842, 57350, 9080, 90299, 90299, 81091, 33989, 33990, 33991, 57352, 87767, 87768, 87784, 6892, 6893, 34830, 34081, 69361, 69362, 69363, 69366, 54816, 54817, 54018, 34085, 3347, 6950, 90932, 90933, 90953, 81229, 81238, 68349, 81234, 43353, 36698, 36699, 43354, 68631, 36793, 68257, 68258, 68259, 69990, 69991, 70009, 39816, 39817, 39818, 83356, 83357, 83358, 28581, 28583, 80808, 80814, 86446, 29104, 29105, 29106, 15802, 15806, 87487, 87496, 87497, 29132, 40420, 40421, 40422, 9199, 92150, 92151, 80890, 53742, 53743] as all_ids // see cypher_scripts/fetch_all_pr_stack_ids
  match (hub)-[r]->(pr2:pull_request { status: "merged"})
  where ((hub:pull_request and hub.status = "merged") or (hub:issue and hub.status = "closed")) and pr2.
  creation_date < hub.creation_date and not id(hub) in
  all_ids and not id(pr2) in all_ids
  with hub, collect(different pr2) as prs, collect(different
  pr2.user) as users, collect(different id(pr2)) as
  pr_ids
  where size(prs) => 3 and size(users) => 2
  return collect(different id(hub))++apoc.coll.toSet(apoc.
  call.flatten(collect(pr_ids))) as known_hubs
}
```