TIPMerge: Recommending Experts for Integrating Changes across Branches

Catarina Costa <catarina@ufac.br>,
Jair Figueirêdo, Leonardo Murta,
Anita Sarma <anita.sarma@oregonstate.edu>

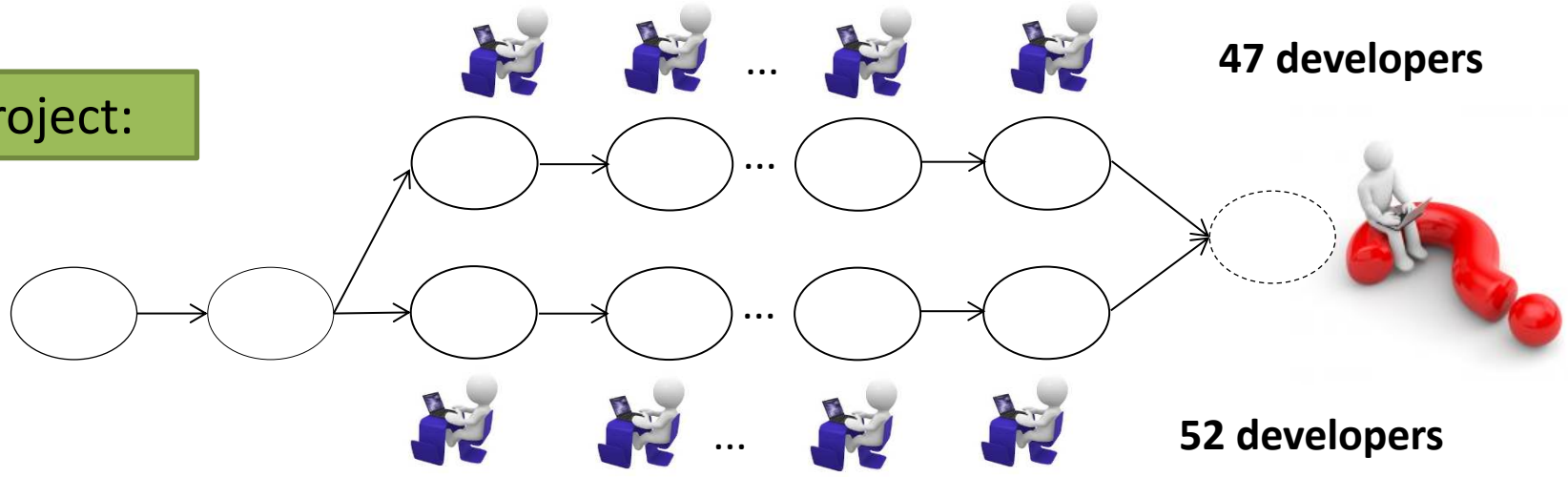
Instituto de Computação

Oregon State University

Parallel Development in Branches

- A common practice to manage parallel development is the use of branches
- At some point, the merge of these branches may be necessary

Rails Project:



hash: bf2b9d2de3f8…

Survey of 164 developers*

62% made decisions with which they were uncomfortable

81% contacted other developers to resolve the conflict together

...

* Costa, C., Figueiredo, J.J.C., Ghiotto, G. and Murta, L. 2014. Characterizing the Problem of Developers' Assignment for Merging Branches. *International Journal of Software Engineering and Knowledge Engineering (IJSEKE)*. 24, 10 (2014), 1489–1508

Problem

- *Who are the most appropriate developers to integrate a pair of branches?*

- Related work:

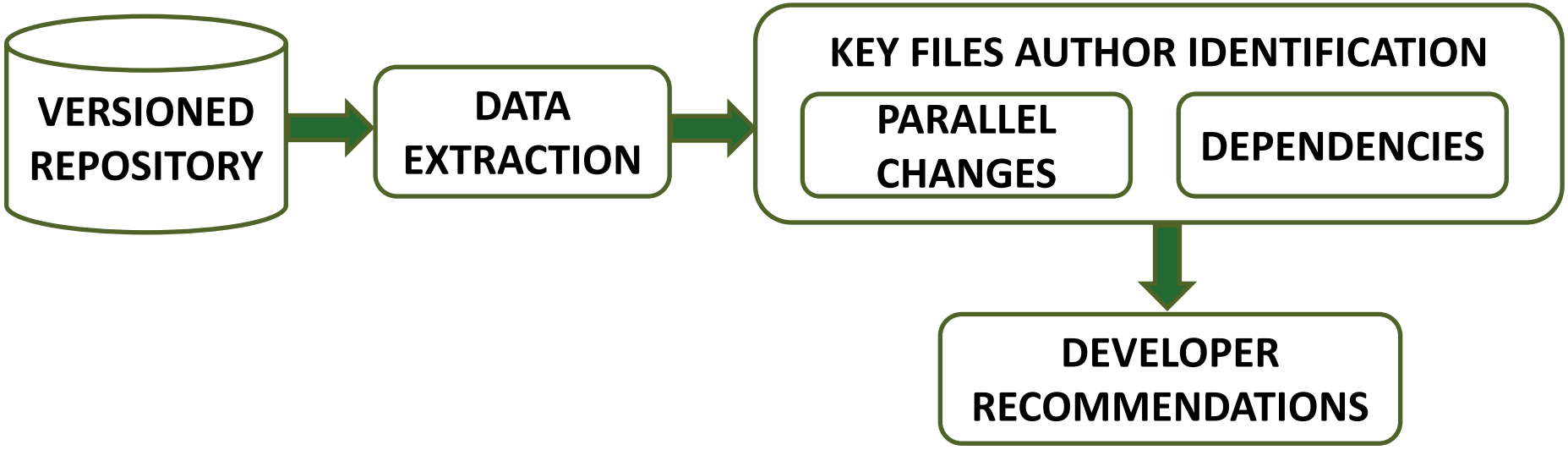
Pull request

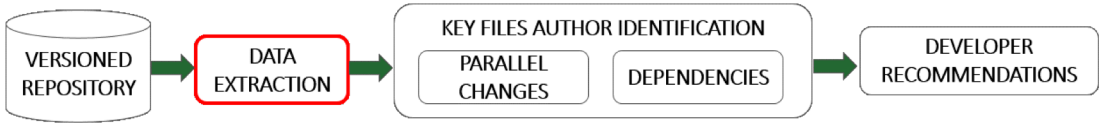
Recommend developers for contribution evaluation using information retrieval [Yu@ICSME2015, Yu@ASPEC2014], vector machines [Jiang@JCST2015], data mining [DeLimaJunior@SAC2015]

- Just one developer in one of the branches
- Usually contain few commits by a single developer
- Reintegration of their forked branch

TIPMerge

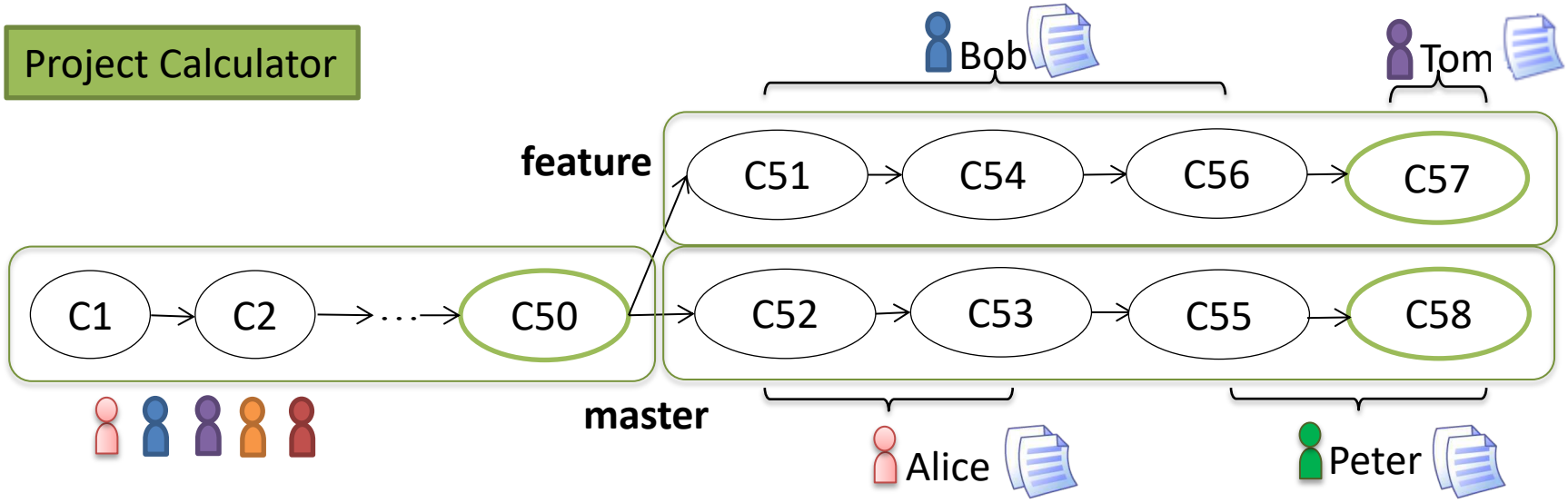
We propose TIPMerge to identify the most appropriate developers to merge branches.

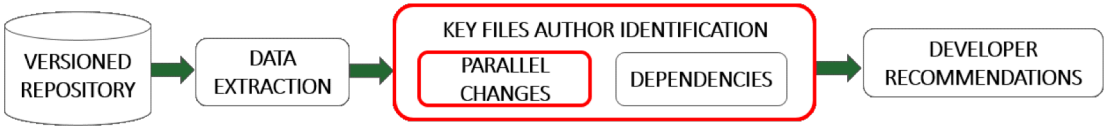




Data Extraction

- The information extraction from the branches and the common history - **changed files and committers in branches and previous history**





Key Files: Parallel Changes

- We identify who altered **key files**: (1) files changed in both branches (potential direct conflicts) and (2) files changed in one branch with dependencies in the other branch (potential indirect conflicts)

Master Branch	Alice	Peter	Feature Branch	Bob	Tom
QuadraticEquation	2 (C52, C53)	0	*QuadraticEquation*	2 (C51, C56)	0
Subtraction	1 (C53)	0	*Subtraction*	3 (C51, C54, C56)	0
Multiplication	0	2 (C55, C58)	*IEquation*	0	1 (C57)
Division	0	2 (C55, C58)			

Key files
QuadraticEquation
Subtraction

Key Files: Dependencies

- We identify who altered **key files**: (1) files changed in both branches (potential direct conflicts) and (2) files changed in one branch with dependencies in the other branch (potential indirect conflicts)

Previous History	Alice	Bob	Tom	Alex	Anna
QuadraticEquation	3	0	0	11	4
Subtraction	0	2	0	3	0
Multiplication	0	0	0	4	2
Division	0	0	0	1	3
IEquation	0	0	4	6	2

When IE was changed (12)
QE was changed together (8)
8/12 = 0.66

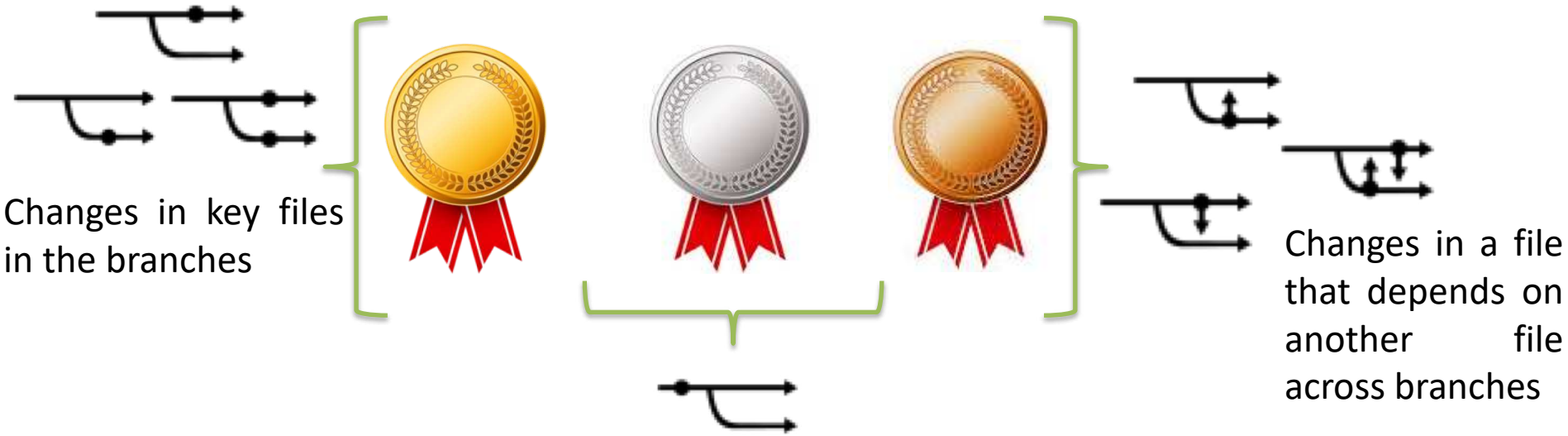
Considering a confidence threshold of 0.6

QuadraticEquation depends on ***IEquation***

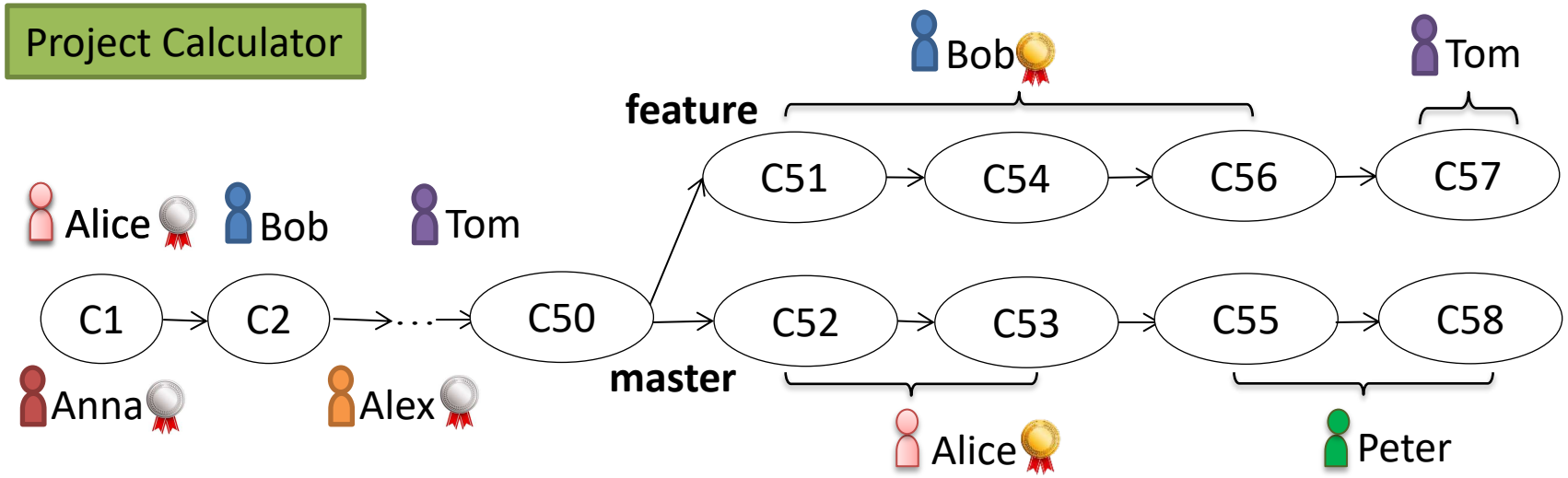
Key files
QuadraticEquation
Subtraction
IEquation

Developer Recommendations

- Our approach applies an algorithm that counts each contribution and creates a ranking of suitable candidates to perform the merge

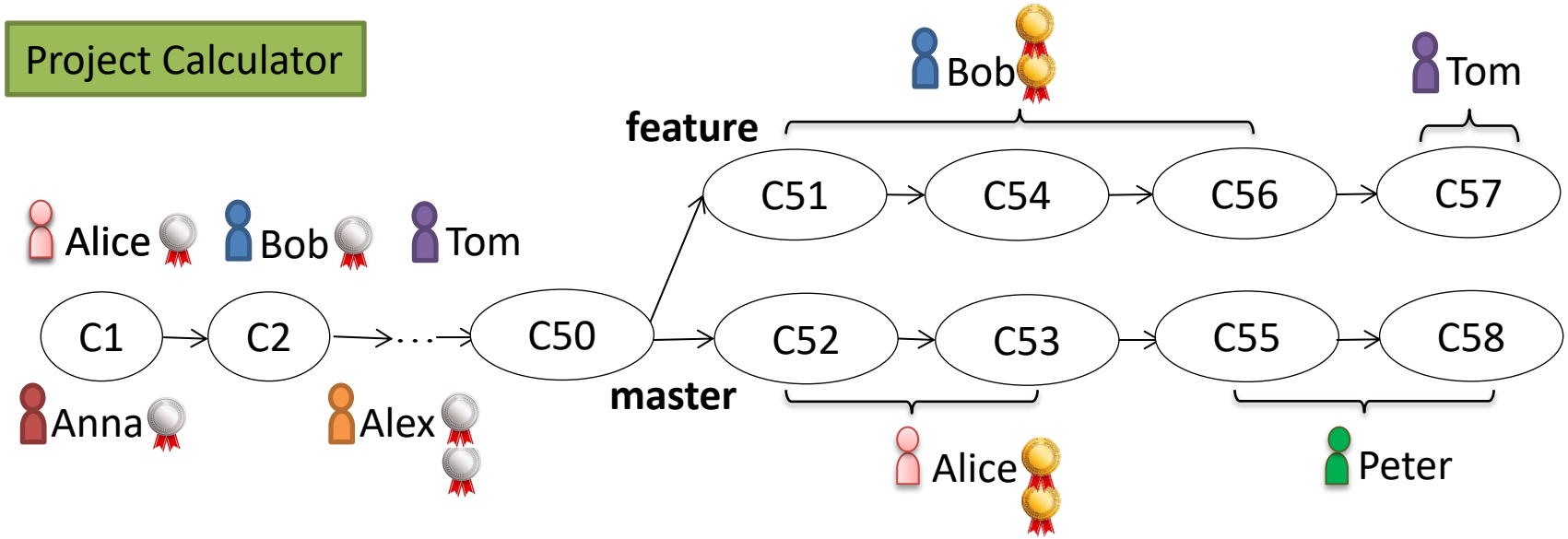


Developer Recommendations



QuadraticEquation

Developer Recommendations

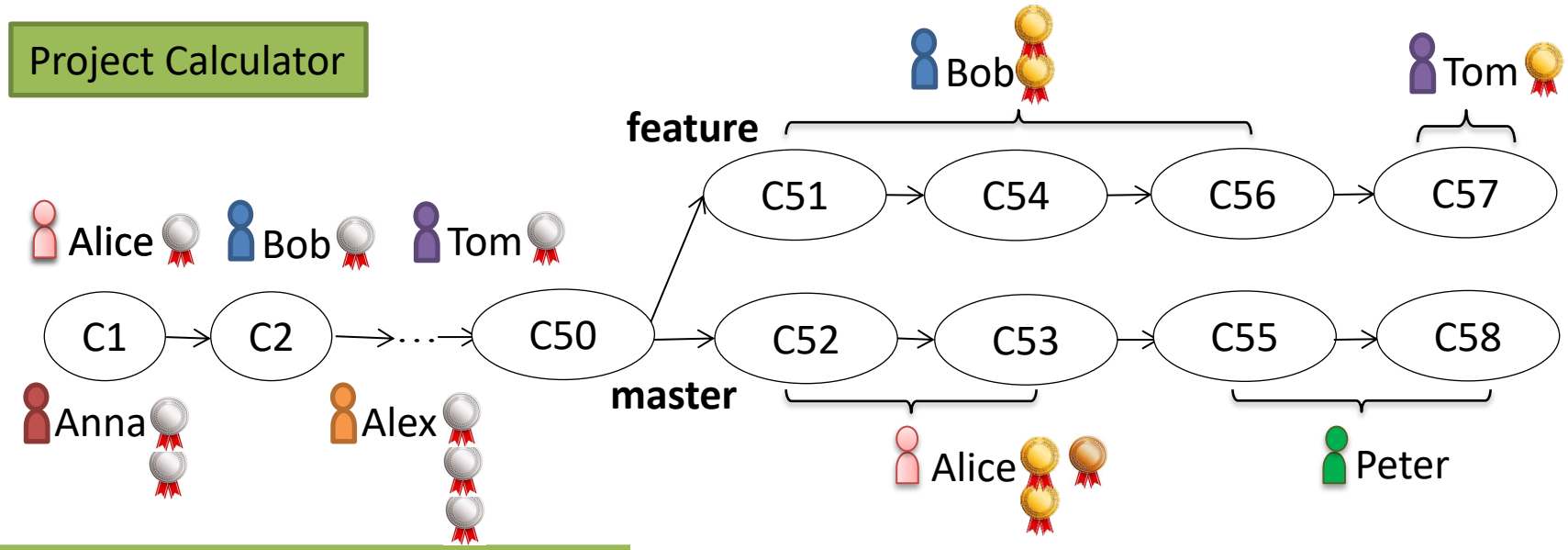


QuadraticEquation

Subtraction

Developer Recommendations

Project Calculator

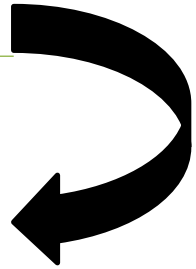


Developer	Gold	Silver	Bronze
1st Alice	2	1	1
2nd Bob	2	1	0
3rd Tom	1	1	0
4th Alex	0	3	0
5th Anna	0	2	0

QuadraticEquation

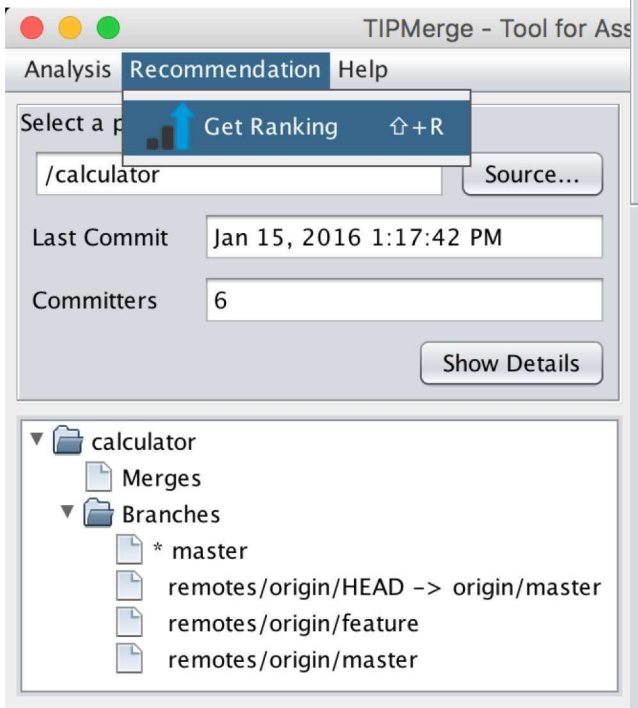
Subtraction

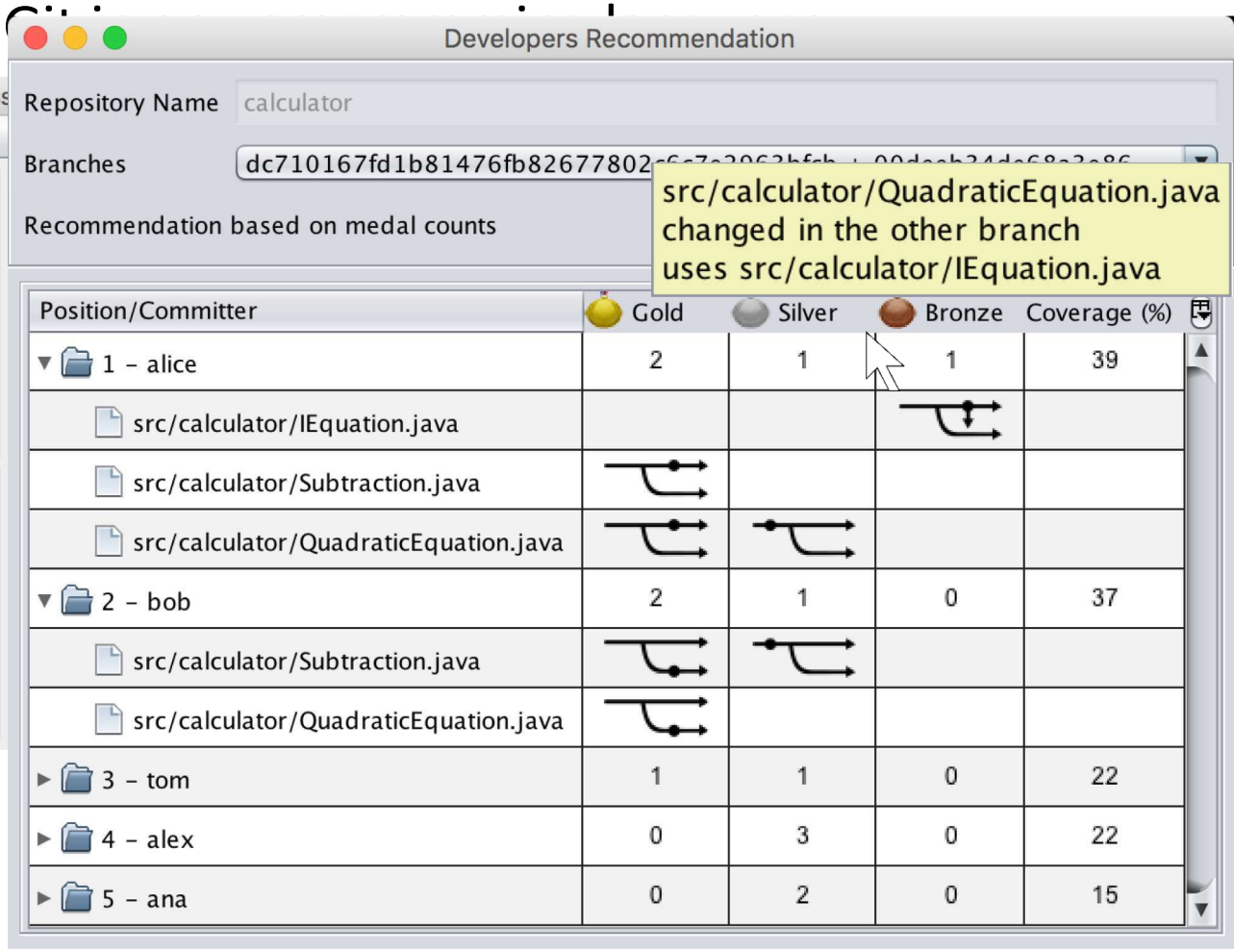
IEquation



TIPMerge Interface

- TIPMerge was implemented in Java and is able to analyze any project versioned on Git





Evaluation

- Quantitative: 27 OSS projects and 1 proprietary project to check accuracy of results
- Qualitative: interviewed 3 developers from 2 projects; a proprietary (Diploma) and an OSS (Sapos) to understand why TIPMerge recommendations diverged

Quantitative Evaluation

Projects

28 randomly selected projects with a total of 34,916 merges

Complex Merges

Selected 15,584 merges with potential for conflicts (direct and indirect)

Majority Classes (MC) Accuracy

Top-3 developers who performed the most merges

Category I – No clear integrator (MC<50%)

Category II – Integration team (MC>50% and Top-1<50%)

Category III – Integrator (Top-1>50%)

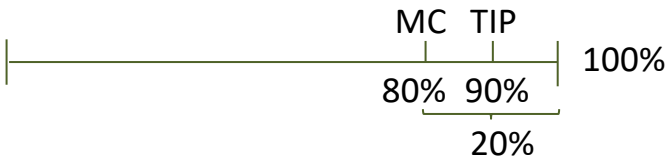
TIPMerge Accuracy

Whether who actually performed the merge is in the first, second or third position of the ranking

Normalized Improvement of Accuracy

Proportion of improvement of TIPMerge accuracy over Majority class accuracy

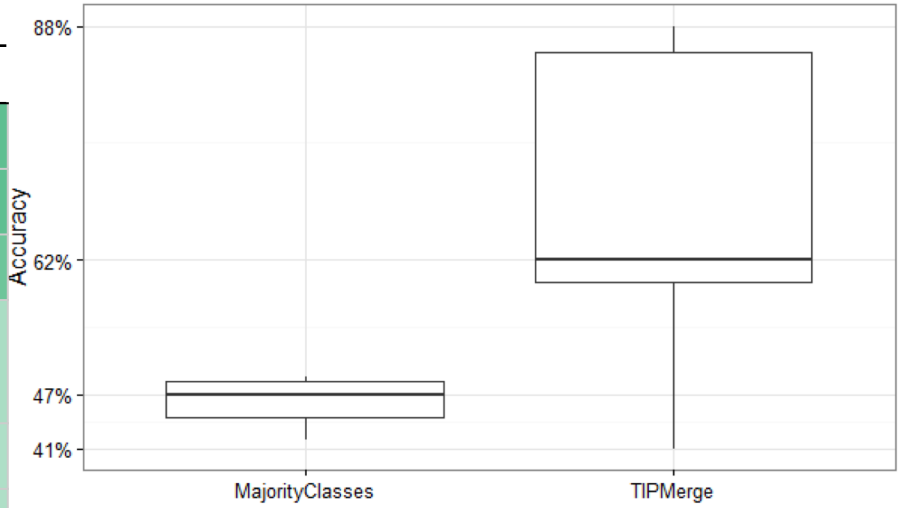
Ex.:



TIP achieved 50% (10% ÷ 20%) of this gain

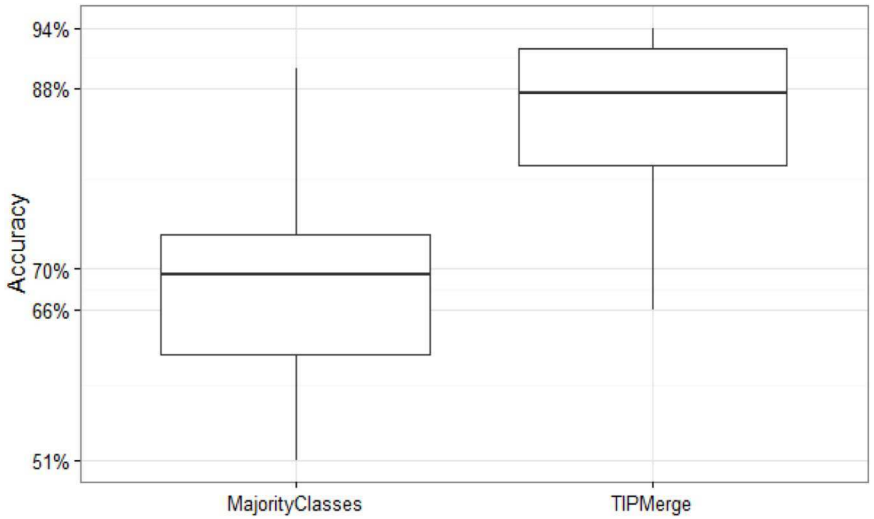
Category I – No clear integrator

Project	Majority Classes	TIPMerge	Normalized Improv. Accuracy
Category I			**TOP3**
Lantern	47.89%	87.79%	76.57%
Katello	41.52%	86.28%	76.54%
Voldemort	49.35%	83.55%	67.52%
Active Merchant	45.45%	60.61%	27.79%
Symply	49.23%	62.11%	25.37%
TYPO3	44.00%	58.00%	25.00%
Angular	47.06%	41.18%	-12.49%



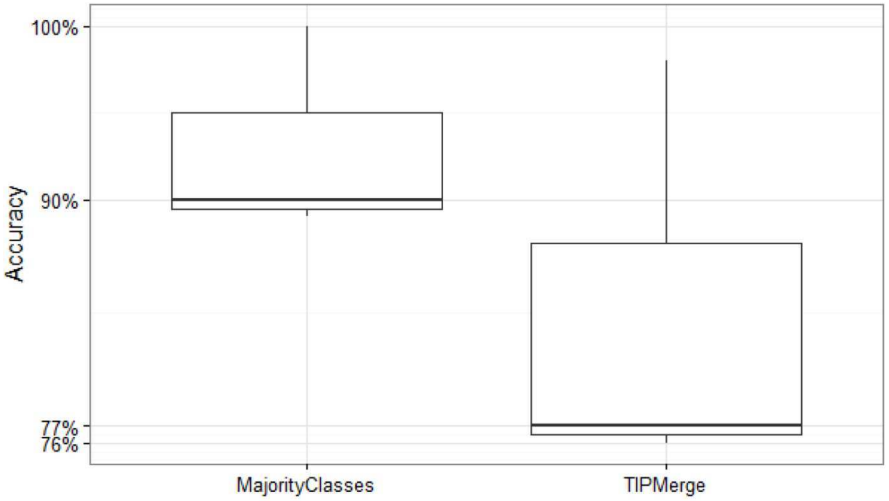
Category II – Integration team

Project	Majority Classes	TIPMerge	Normalized Improv. Accuracy
Category II			**TOP3**
Eureka	72.22%	94.44%	79.99%
Khmer	64.90%	92.81%	79.52%
MCT	73.53%	94.12%	77.79%
Sapos	68.24%	91.76%	74.06%
Akka	55.55%	87.94%	72.87%
Priam	70.10%	91.75%	72.41%
Comm-central	70.00%	90.00%	66.67%
Nomad	75.00%	90.63%	62.52%
Perl5	69.30%	87.45%	59.12%
Cassandra	52.20%	79.98%	58.12%
FirefoxiOS	70.73%	85.85%	51.66%
Spree	61.39%	80.53%	49.57%
Amarok	51.01%	69.70%	38.15%
Falcor	90.00%	93.00%	30.00%
Errbit	60.00%	70.40%	26.00%
Astropy	63.27%	65.85%	7.02%
Phoenix	87.10%	87.10%	0.00%
jQuery	74.24%	69.70%	-6.12%



Category III – Integrator

Project	Majority Classes	TIPMerge	Normalized Improv. Accuracy
Category III			**TOP3**
Diploma	89.10%	98.08%	82.39%
PIConGPU	89.59%	75.57%	-15.65%
Maven	100.00%	76.92%	-23.08%



Qualitative Evaluation

- Project Selection - Developer selection
  - Diploma: proprietary (5 developers); interviewed 1 (D1)
  - Sapos: open-source (10 developers); interviewed 2 (D1, D3)

- Merge selection

TIPMerge Position	Diploma	Sapos
3rd	D1, D5, D4	D4, D5, D3
	D4, D2, D1	D4, D5, D3
2rd	D3, D2	D1, D2
	D1, D3	D1, D2
	-	D4, D1
	-	D4, D1
1st	D1	

Results

Collaborative practices:

D3 (sapos): *"we did these merges together in my office"*

D1 (diploma): *"I would still have helped him in this merge. While D3 could have perform [the merge], I would have followed it closely"*

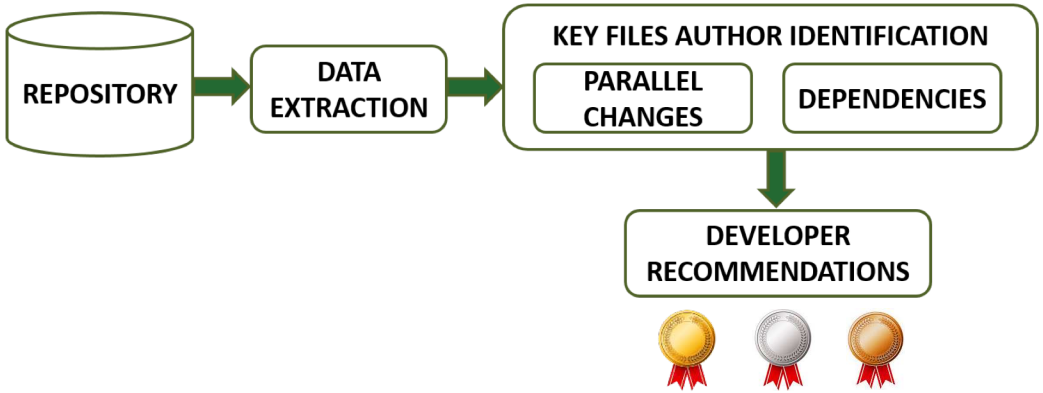
Person performing the most recent change:

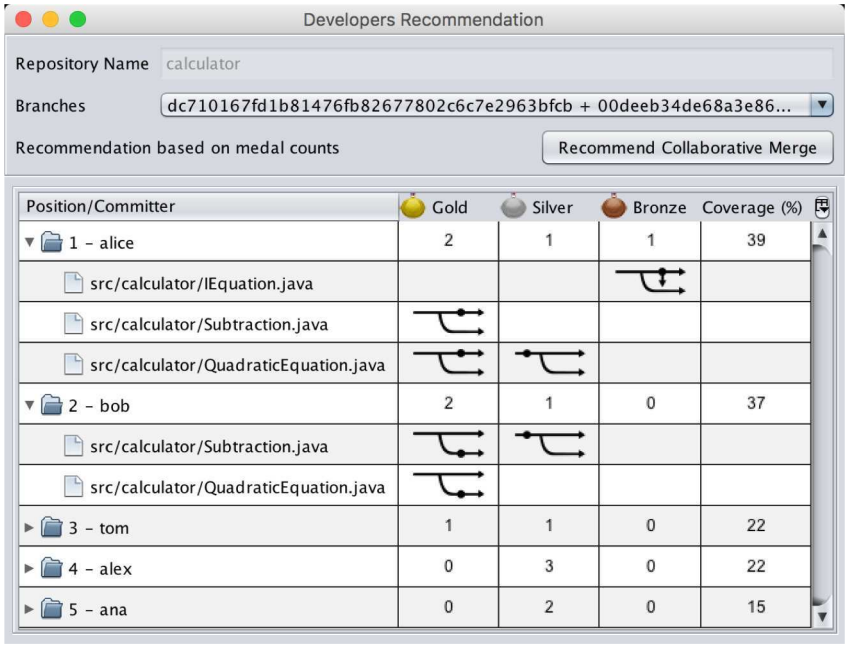
D1 (diploma): *"It makes sense... I help him in the merge... D4 must have been the one to do the merge ultimately because he was the last to commit"*

Knowledge about specific parts of the code base:

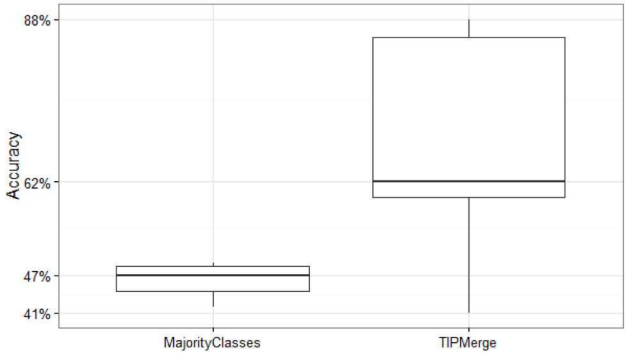
D1 (diploma): *"as there were some parts of a legacy system, and only I know this part, I should indeed have done this merge"*

Conclusion

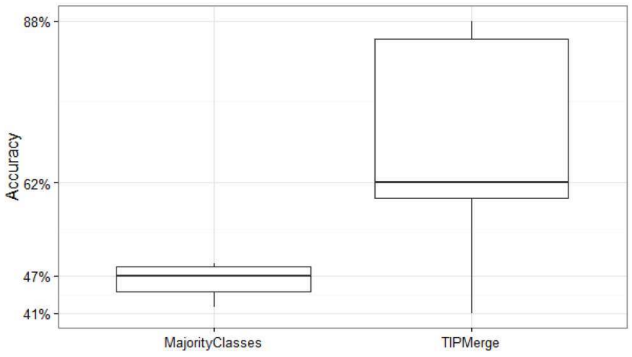




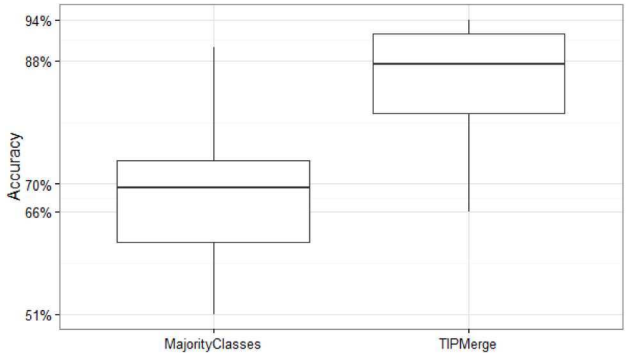
TOP3 – Cat1



TOP3 – Cat2



TOP3 – Cat3



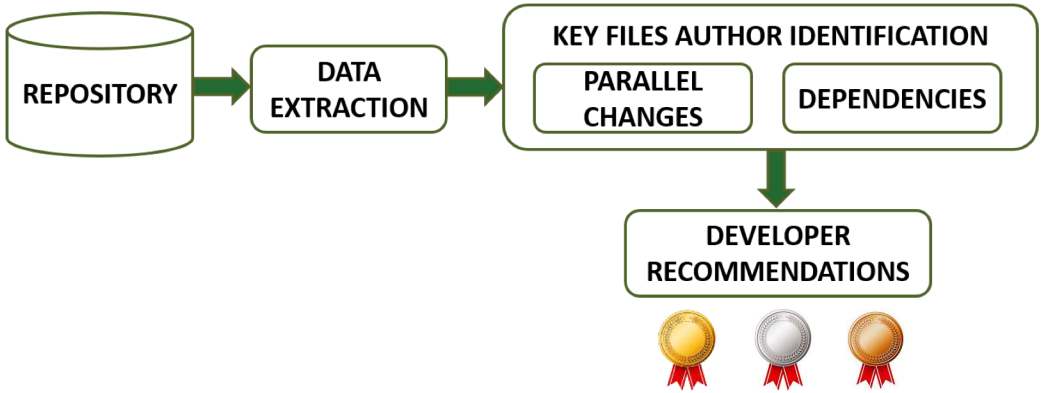
Future Work

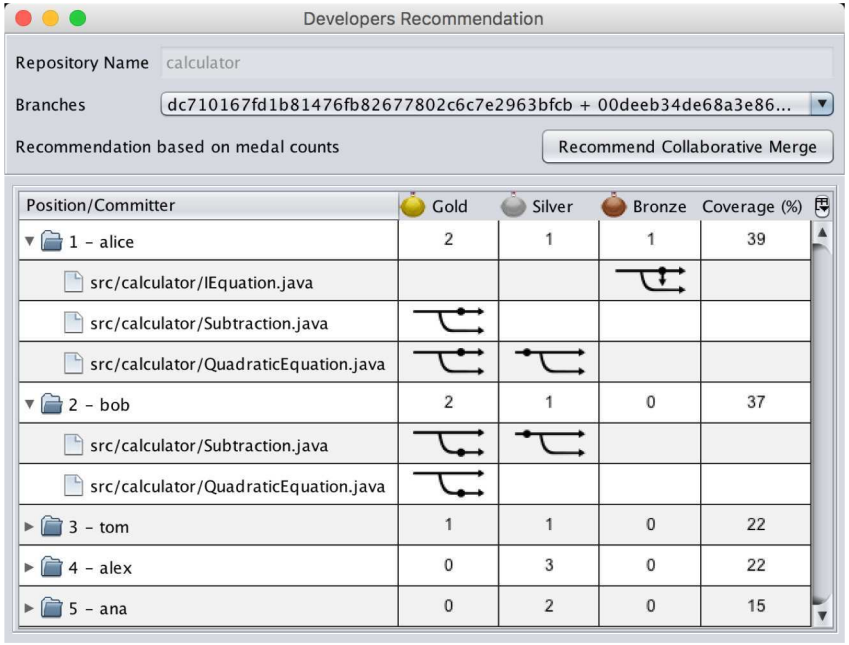
- Run analysis at a finer grain (method level)
- Concept of joint expertise coverage
- Encode development policy as heuristics

Acknowledgment

- STI/UFF – Diploma Team Members
- GEMS/UFF - Sapos Team Members

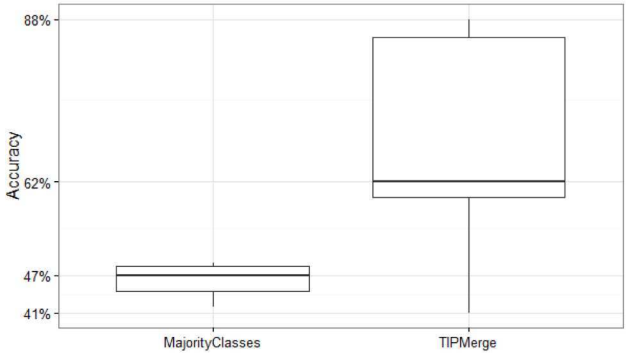
Questions



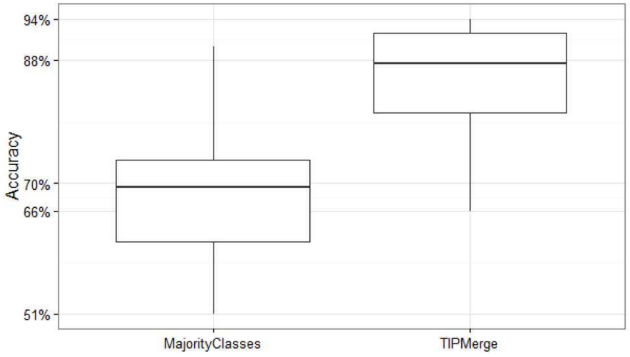


Position/Committer	Gold	Silver	Bronze	Coverage (%)
1 - alice	2	1	1	39
src/calculator/IEquation.java				
src/calculator/Subtraction.java				
src/calculator/QuadraticEquation.java				
2 - bob	2	1	0	37
src/calculator/Subtraction.java				
src/calculator/QuadraticEquation.java				
3 - tom	1	1	0	22
4 - alex	0	3	0	22
5 - ana	0	2	0	15

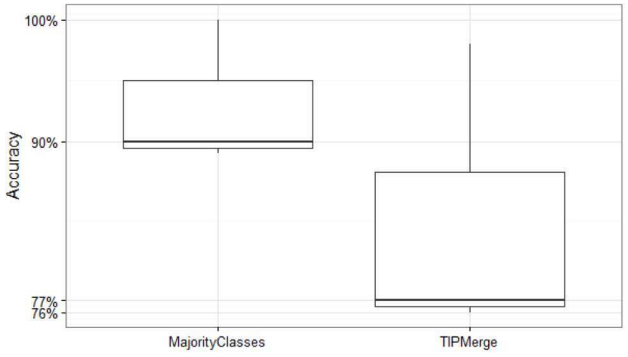
TOP3 – Cat1



TOP3 – Cat2



TOP3 – Cat3



TIPMerge: Recommending Experts for Integrating Changes across Branches

Catarina Costa <catarina@ufac.br>,
Jair Figueirêdo, Leonardo Murta,
Anita Sarma <anita.sarma@oregonstate.edu>

Instituto de Computação

OSU Oregon State University