

Students' perceptions of integrating a contribution measurement tool in software engineering projects

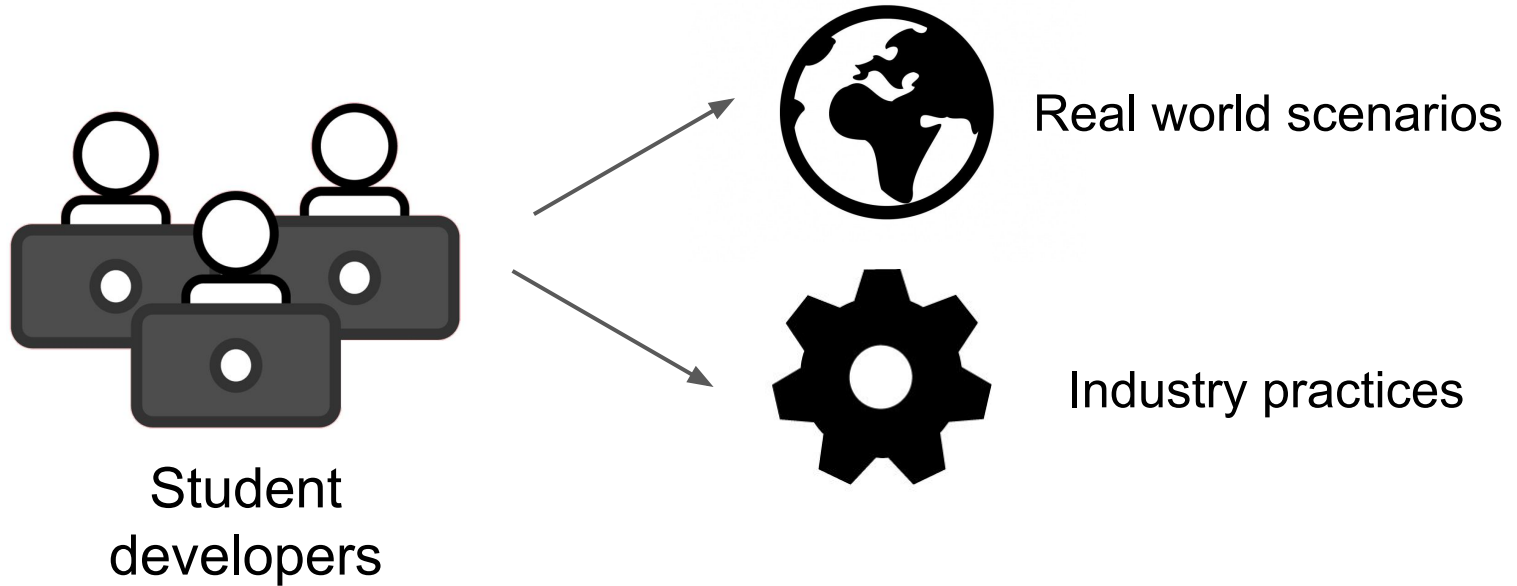
Sivana Hamer, Christian Quesada-López, Marcelo Jenkins
sivana.hamer@ucr.ac.cr, cristian.quesadalopez@ucr.ac.cr, marcelo.jenkins@ucr.ac.cr



UNIVERSIDAD DE
COSTA RICA

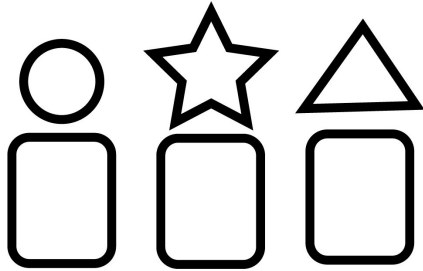


Project-based learning has been integrated into software engineering education

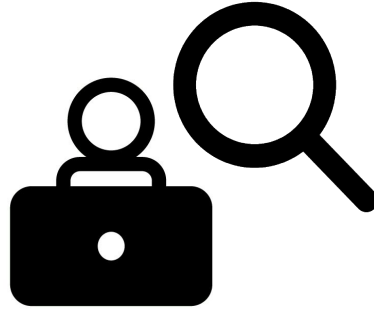


Bastarrica et al. (2017), Quesada-Lopez et al. (2019), Souza et al. (2019).

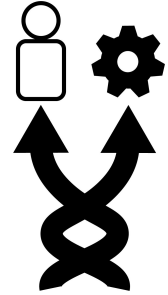
Evaluating the contributions in software engineering education projects is complex



Multi-faceted
nature

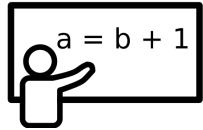


Discern individual
contribution

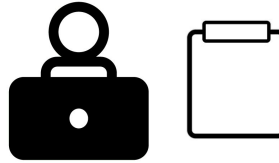


Efficiency
needs

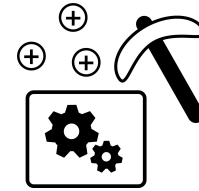
Several student assessment approaches have been applied in software engineering projects



Instructor
observation
Hayes et al. (2003)

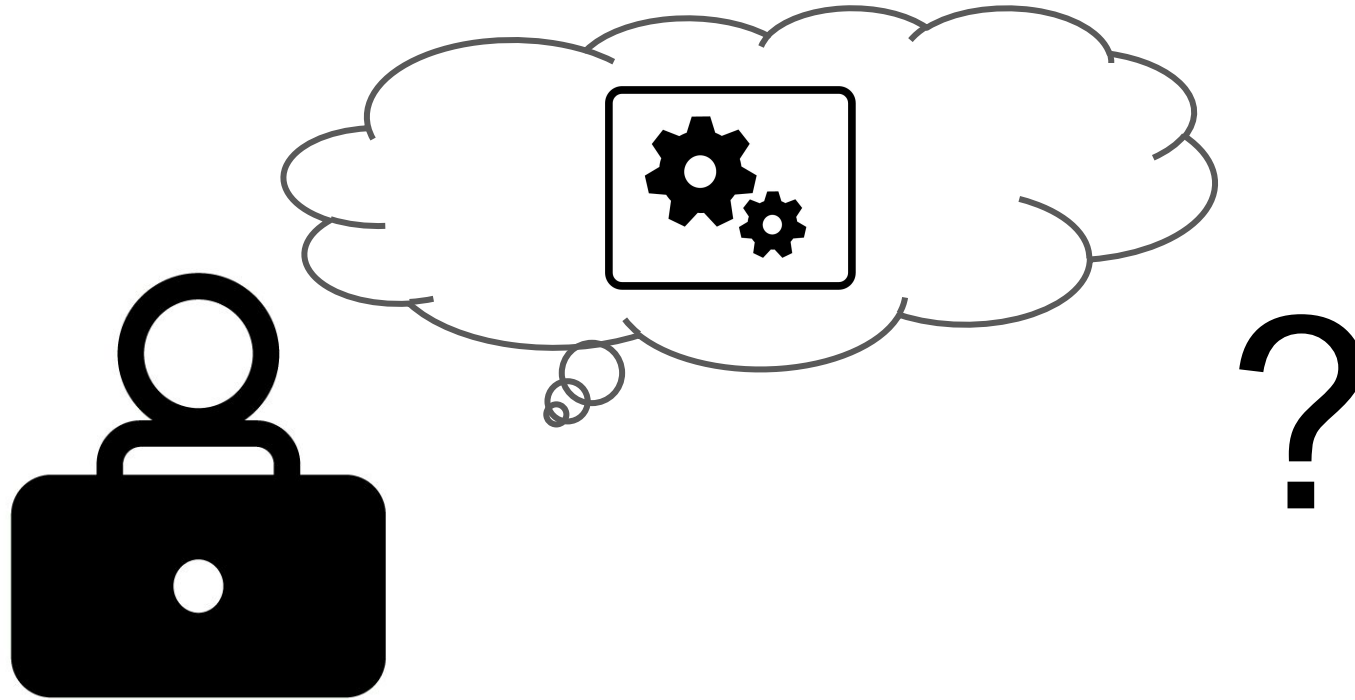


Self and peer
assessment
Yujia et al. (2009)
Mamei (2011)



Mining repositories
for data
Matthies et al. (2016)
Guerrero et al. (2019)
Bai et al. (2018)

We have a limited understanding of what students think of such data-driven measurement tools



Goal:

Analyze students' perceptions of using a data-driven contribution measurement approach within agile software engineering projects.

Research questions



*RQ1. What is the **students' perceived acceptance** (usefulness, ease of use, and intention of use) of the data-driven contribution measurement approach?*



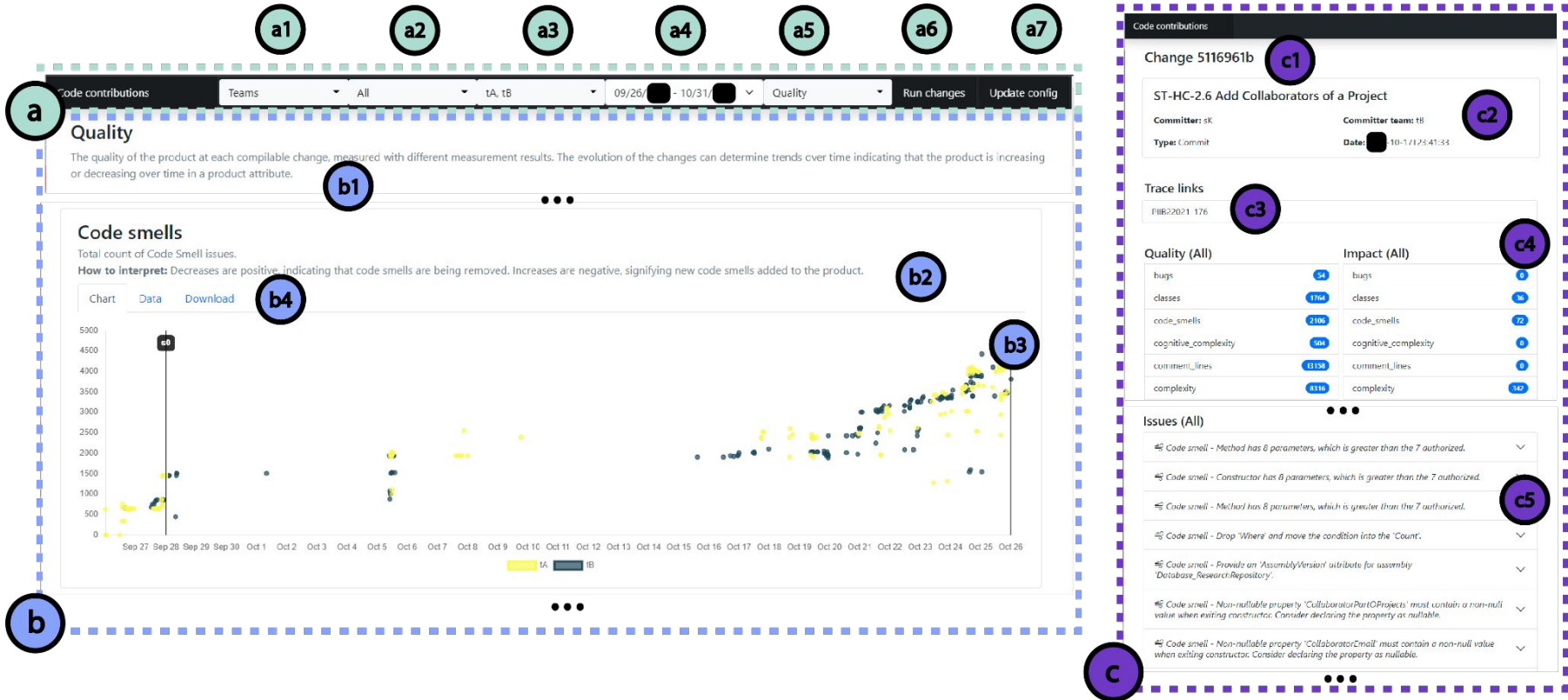
*RQ2. What **student-reported benefits** does the contribution measurement approach offer for software development?*



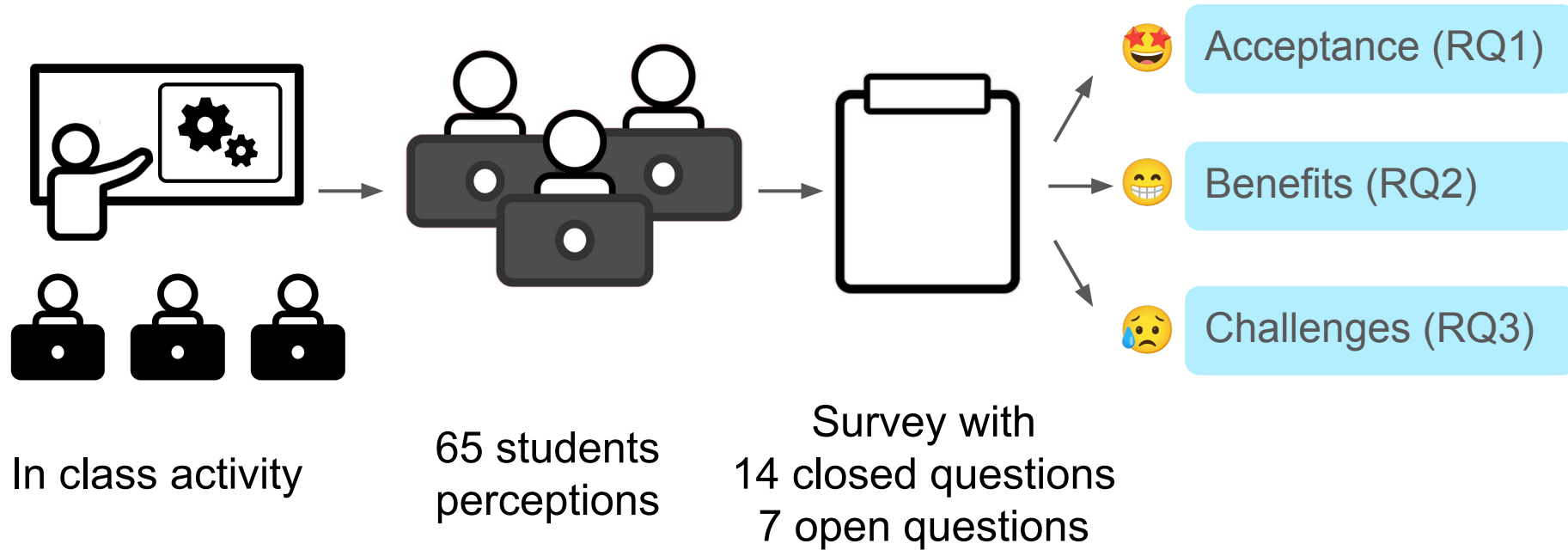
*RQ3. What are the **challenges reported by students** in using the data-driven contribution measurement approach?*

Our measurement approach to characterize contributions

Example visualization screen

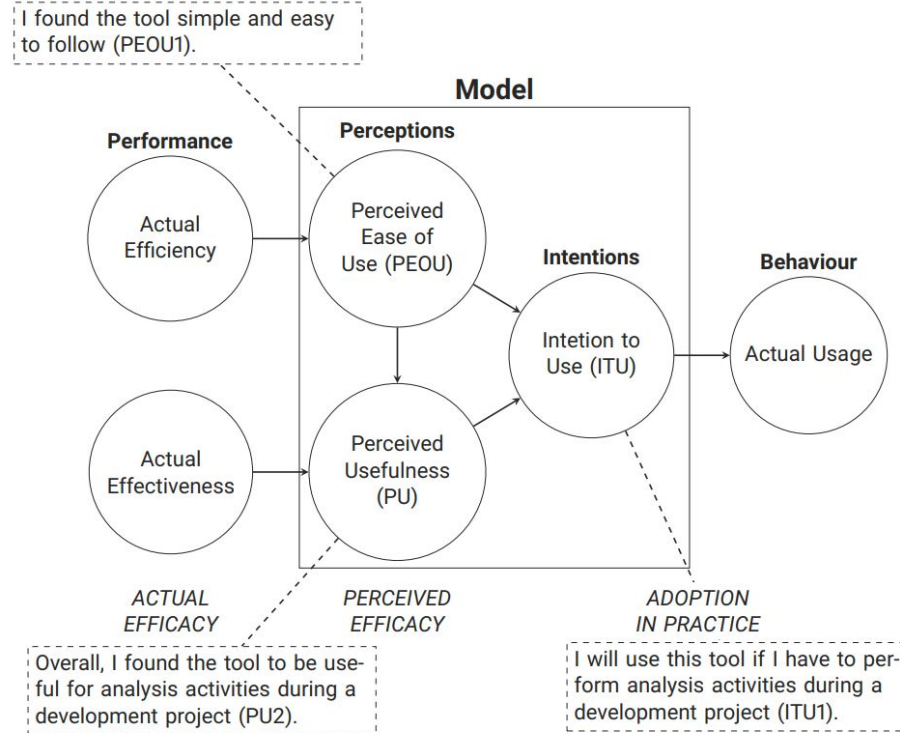


Research method



Data was gathered across several semesters

Technology acceptance model (TAM)






Davis (1985), Abrahao (2004)



Student had a positive perceived acceptance of the approach

Neutral or higher Statistical significance*

Perceived usefulness	98.7%	
Perceived ease of use	93.5%	
Perceived interest in use	93.8%	

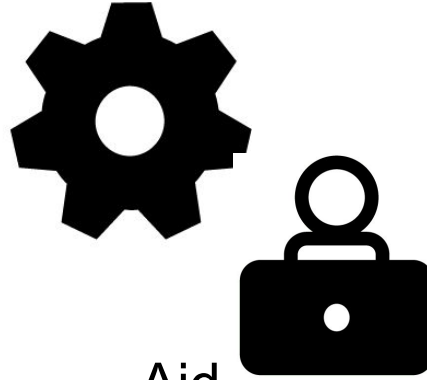
* Answers were compared to a neutral score



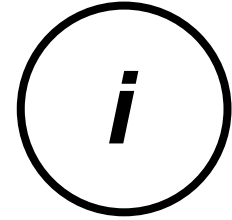
The most mentioned benefits included



Get valuable
feedback



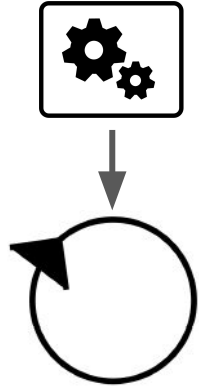
Aid
development
practices



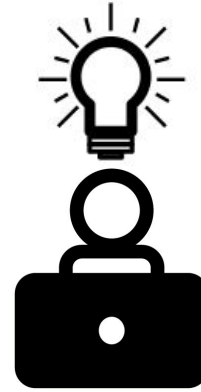
Facilitate
information



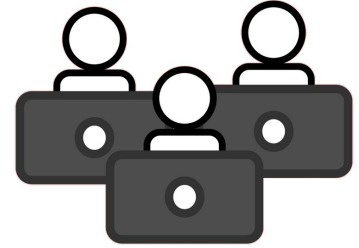
Challenges mentioned included



Tool
adoption



Learning
curve



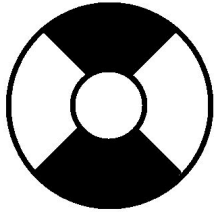
Team
dynamics



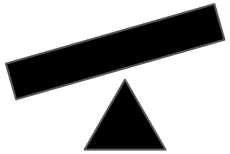
Main takeaways



Data-driven tools are welcome addition in software engineering projects

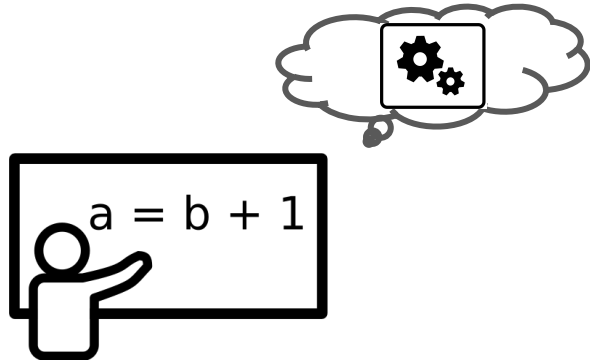


Students require aid in learning and taking advantage of such tools

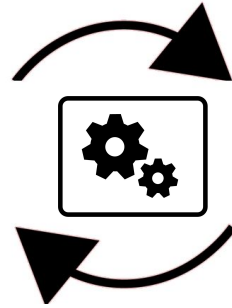


Caution while using is required as there are several usage challenges

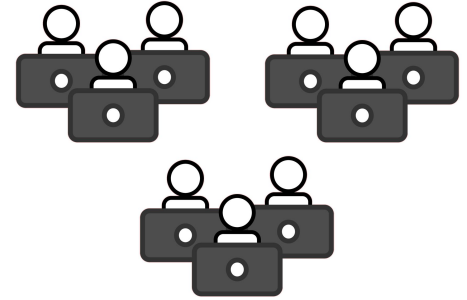
Future work



Instructors
perceptions



Continuous
use

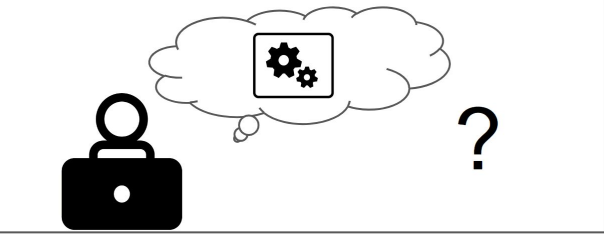


More
validations

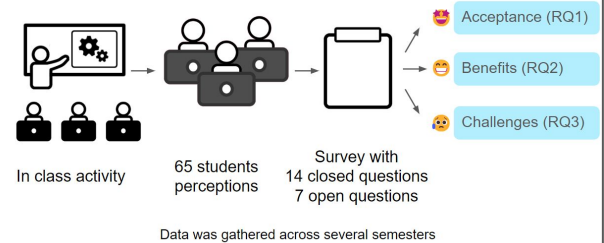
Several student assessment approaches have been applied in software engineering projects



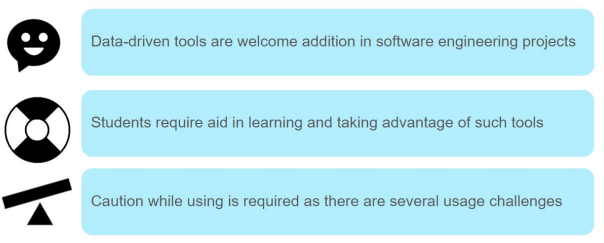
We have a limited understanding of what students think of such data-driven measurement tools



Research method



Main takeaways



Thank you!

ありがとうございました!

Sivana Hamer

sivana.hamer@ucr.ac.cr



UNIVERSIDAD DE
COSTA RICA



Question	MD	AV	SD	Percentage of answers					Distribution				
				1	2	3	4	5	1	2	3	4	5
PU	5	4.48	0.70	1%	1%	6%	35%	58%					
PU1	5	4.46	0.77	2%	0%	8%	32%	58%					
PU2	5	4.62	0.68	2%	0%	2%	29%	68%					
PU3	5	4.46	0.66	0%	2%	5%	40%	54%					
PU4	5	4.42	0.66	0%	0%	9%	40%	51%					
PU5	4	4.32	0.81	2%	2%	8%	42%	48%					
PU6	5	4.63	0.57	0%	0%	5%	28%	68%					
PEOU	4	3.85	0.85	0%	6%	24%	47%	22%					
PEOU1	4	3.98	0.74	0%	2%	23%	51%	25%					
PEOU2	4	3.82	0.86	0%	6%	29%	42%	23%					
PEOU3	4	3.98	0.87	2%	6%	11%	55%	26%					
PEOU4	4	3.71	0.88	0%	9%	29%	43%	18%					
PEOU5	4	3.74	0.85	0%	8%	29%	45%	18%					
ITU	4	3.98	0.94	2%	5%	22%	38%	34%					
ITU1	4	4.25	0.87	2%	3%	9%	42%	45%					
ITU2	3	3.52	0.89	2%	8%	42%	35%	14%					
ITU3	4	4.19	0.91	2%	3%	14%	35%	43%					

	Statistical significance
PU	$p < 0.001$
PEOU	$p < 0.001$
ITU	$p < 0.001$

Benefits

Get valuable feedback. Provides objective data useful for the project, people, product, and process.



Aid development practices. Assists with ensuring and improving agile software engineering practices.



Facilitate information. Helps by automating the collection and visualization of data.



Intention to use tools. Having an interest in using the tool or similar tools in academic and professional software projects.



Challenges

Tool adoption. Requires commitment from the team to take advantage of the insights.



Learning curve. Needs experience to understand measures, analyze visualizations, and determine improvement actions.



Complex configuration. Has a difficult setup due to the amount of information needed to be configured.



Measures completeness. Shortcomings in automated measures necessitate other information to be considered as a contribution.



Team dynamics. Requires care as it may lead to harmful dynamics between team members.



Goodhart's law. Overemphasis on measures can neglect other important aspects of the project.

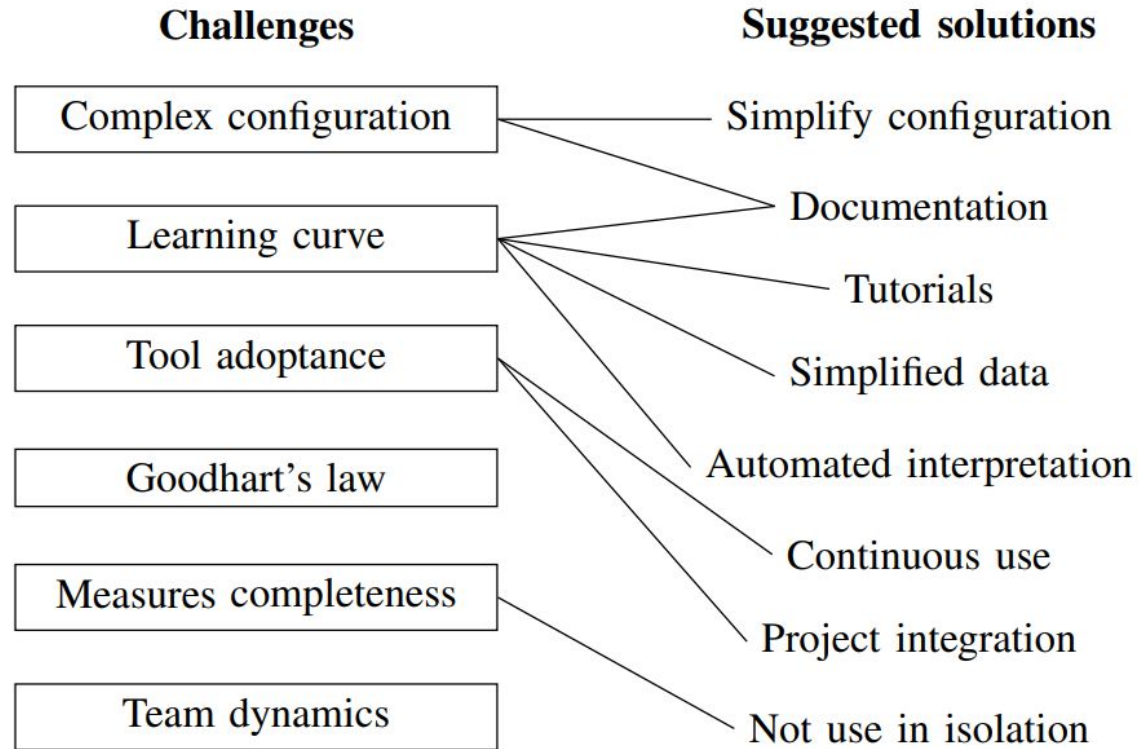


= 1 – 10 mentions, = 11 – 20 mentions, =

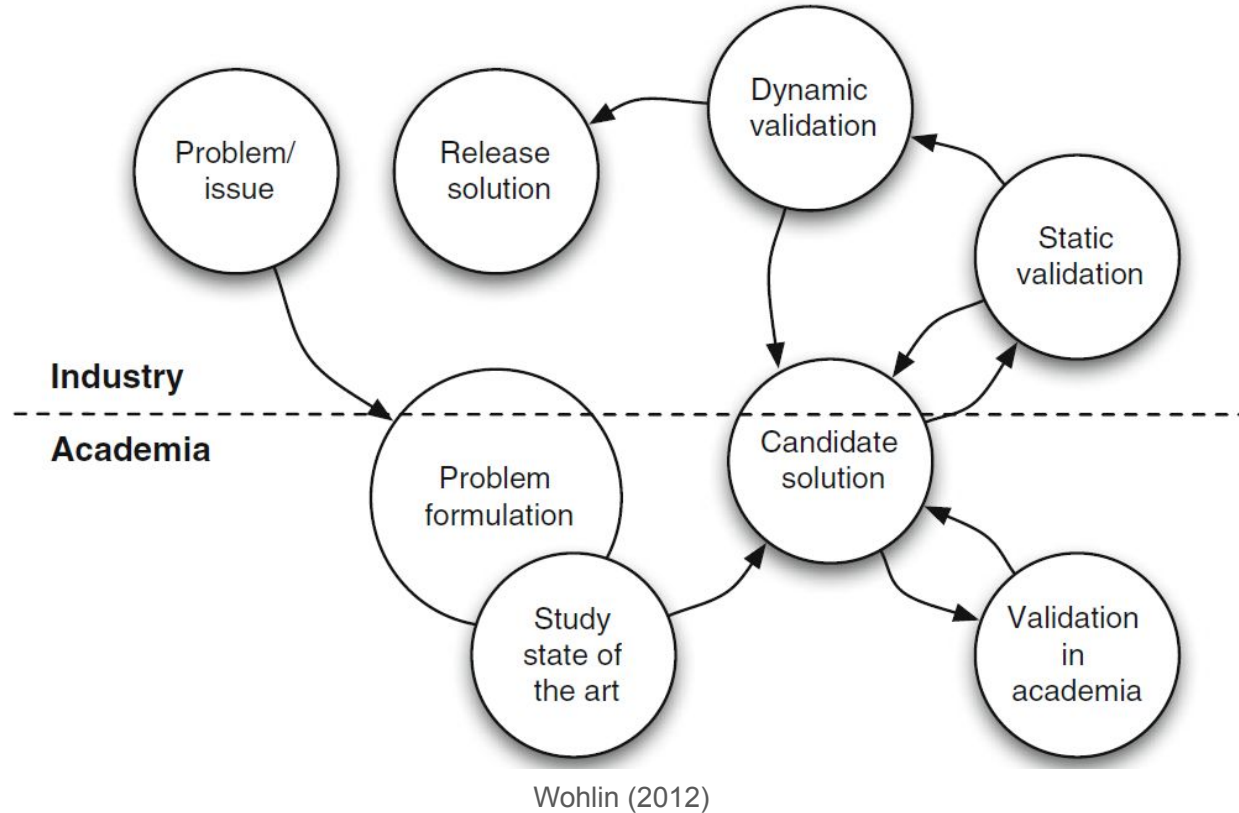
= 21 – 40 mentions, = 41 – 60 mentions, and = 60+ mentions.

Practice	Example excerpts from responses
Coding standards	<i>"The tool can be of great help to ensure better code quality by [...] helping to have a better clean code, and thus a more maintainable, reusable and readable code." [S16c6]</i>
Reviews and inspections	<i>"It also helps to more easily comply with the DoD as it saves time identifying those parts of the code that are not in compliance." [S16c1a]</i>
Planning game	<i>"The main one is scrum, being short iterations, the tool gives information about each iteration, very useful for the retrospective session and to plan better in the future, an extremely useful tool for the scrum master." [S24c6]</i>
Continuous integration	<i>"For agile methodologies, you can use the tool to ensure that continuous integrations are taking place [...]" [S17c6]</i>
Incremental customer deliveries	<i>"[...] it also offers business value by offering the opportunity to identify improvement actions that allow the delivery of functional and quality software." [S44c7]</i>
Sustainable pace	<i>"Especially, the continuous work that is sought to be implemented can be verified when working with methodologies such as XP or Scrum [...]" [S12c6]</i>
Low dependency architecture	<i>"An analysis by layers can be used, in this way it gives us a lot of information to take improvement actions regarding agile development and clean architectures in each layer." [S37c6]</i>
Configuration management	<i>"They would help improve the versioning strategies of a development team." [S32c6]</i>
Refactoring	<i>"[...] refactoring by keeping track of when and how much these practices are done." [S49c6]</i>

Suggestions were also proposed by students



Technology transfer model



Contributions

We provide a measurement approach integrating data driven tools to extract, analyze, quantify, and visualize software contributions in development projects.

We present the theoretical foundation for acceptance of data-driven tools and empirically evaluate students' perceived acceptance (usefulness, ease of use, and intention of use) of the contribution measurement approach

We categorize students' perceived benefits and challenges of the contribution measurement approach.