

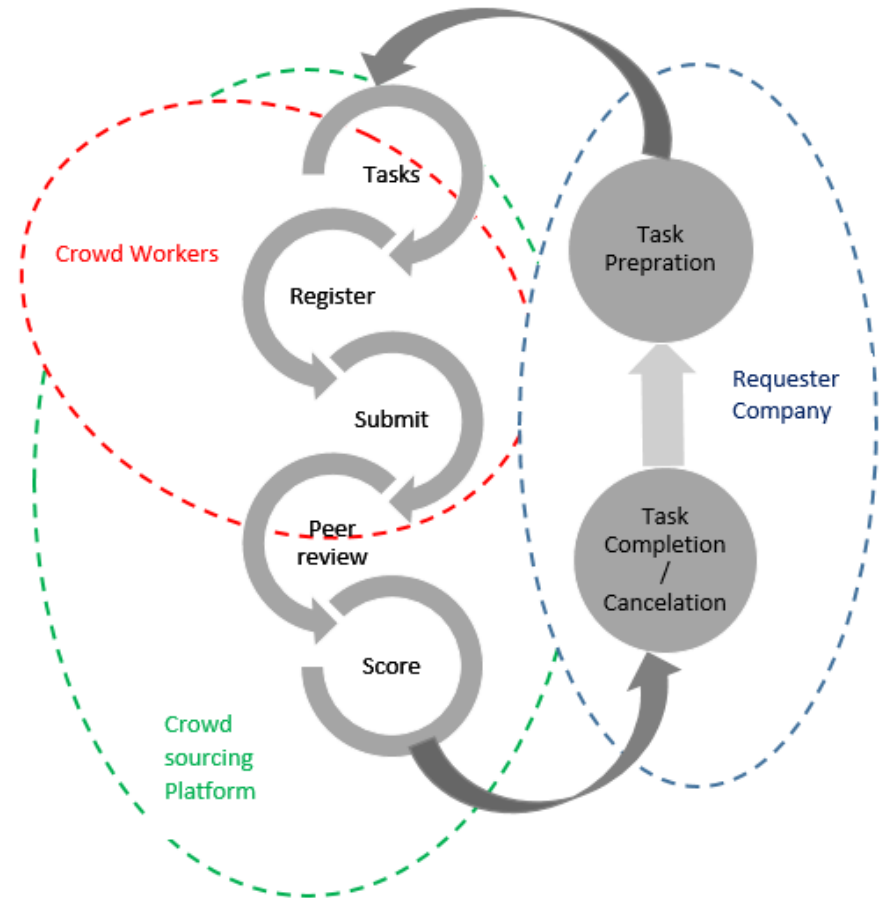
# Empirical Analysis on Parallel Tasks in Crowdsourcing Software Development

Razieh Saremi, Ye Yang

Software Engineering Program  
School of Systems and Enterprises  
Stevens Institute of Technology

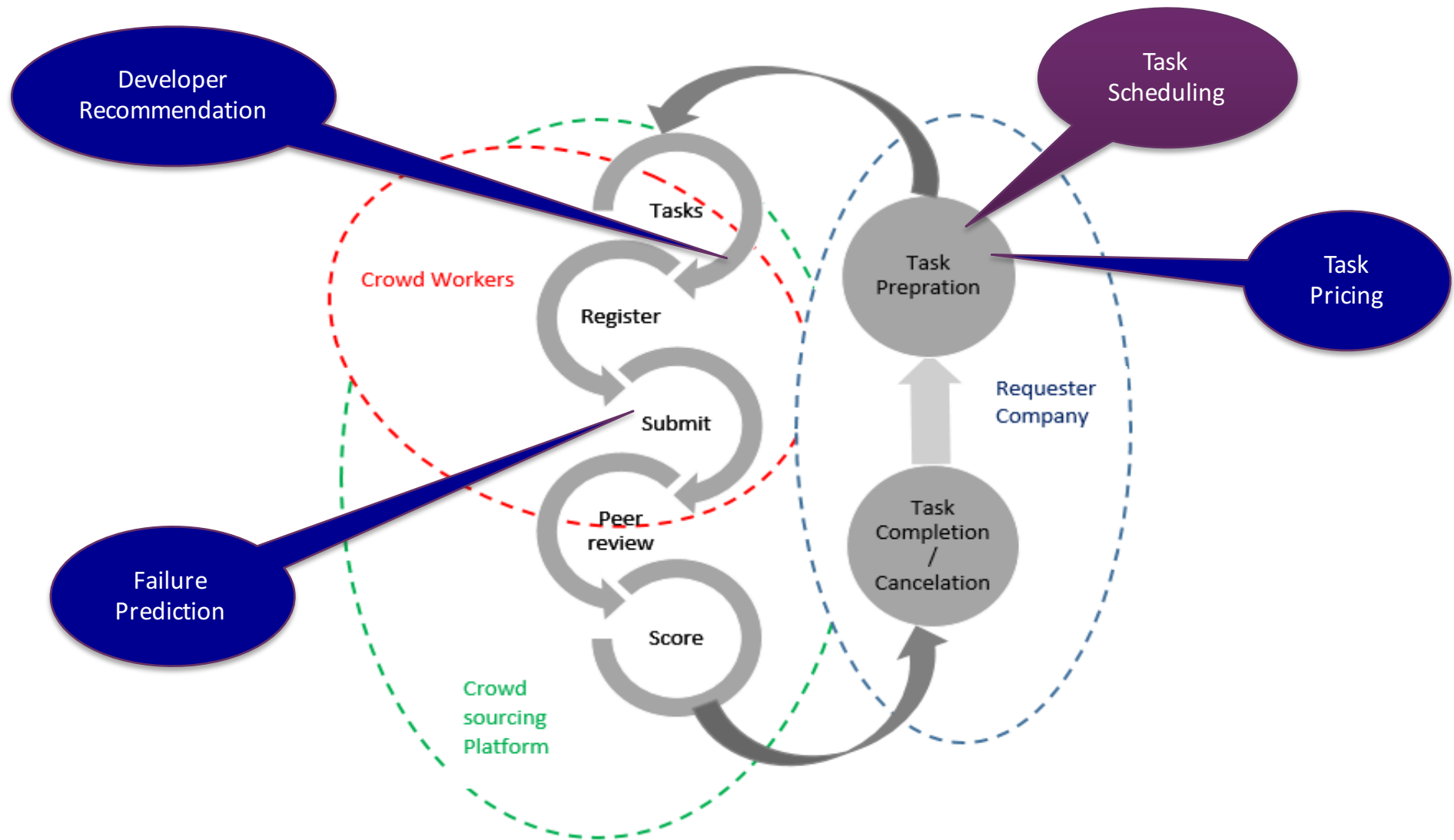
ASE'15 NOV 9, 2015

- Crowdsourced Software Development (CSD)
  - Reported benefits
    - shortened schedule due to mass parallel development
    - innovative solutions based on the “wisdom of crowds”
    - reduced cost due to the pre-pricing and bidding effects
  - Option for rapid development?
  - Lack of project level analysis

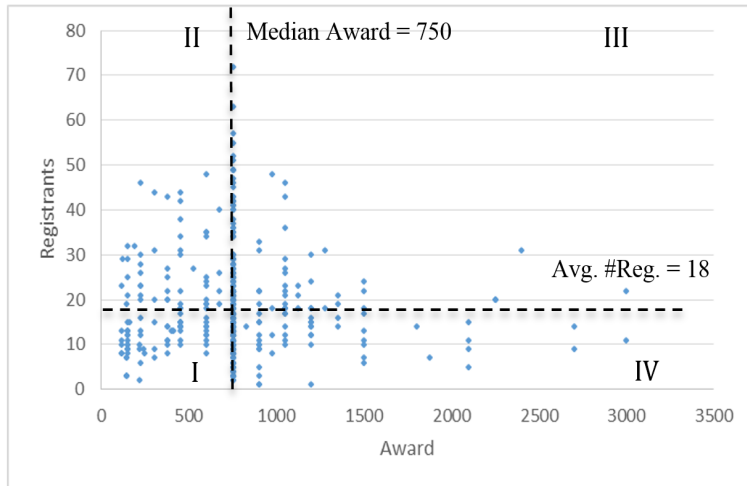


General Competitive CSD Processes

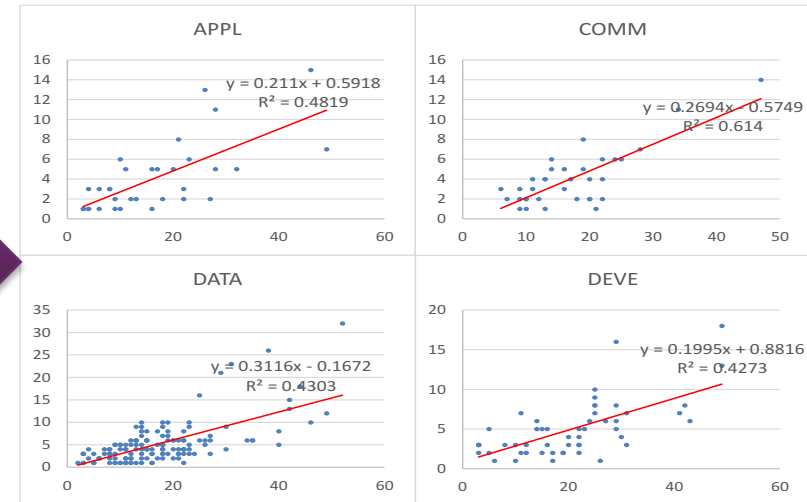
# Needs for Decision Support in CSD



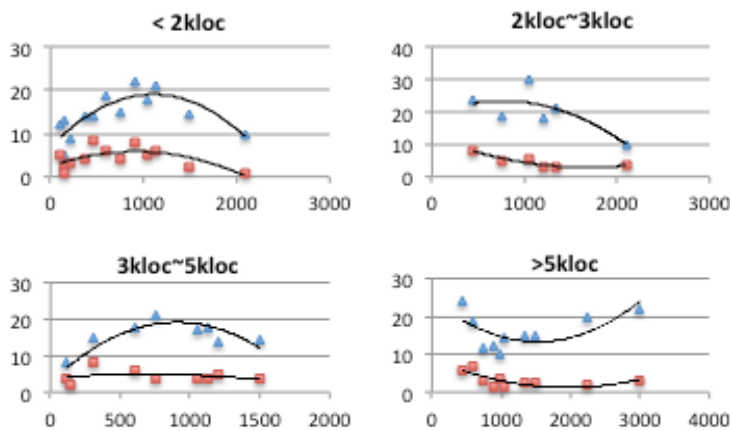
# Relationship between Award and Worker Behavior



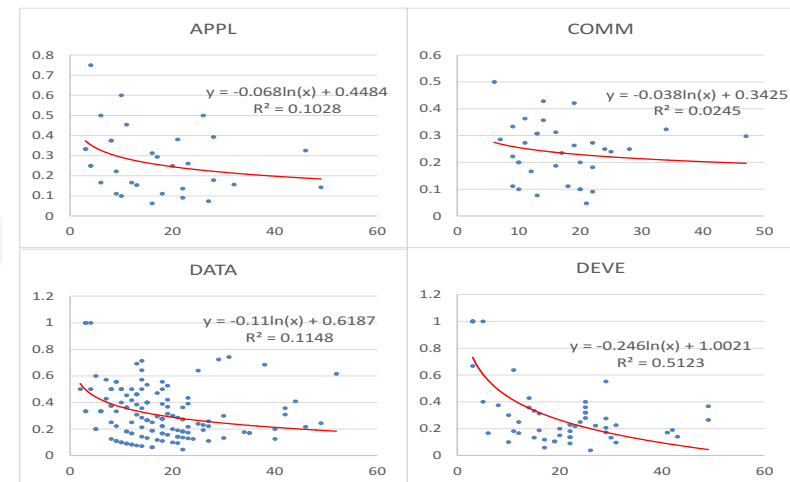
#Registrants vs. Award



#Registrants vs. #Submissions

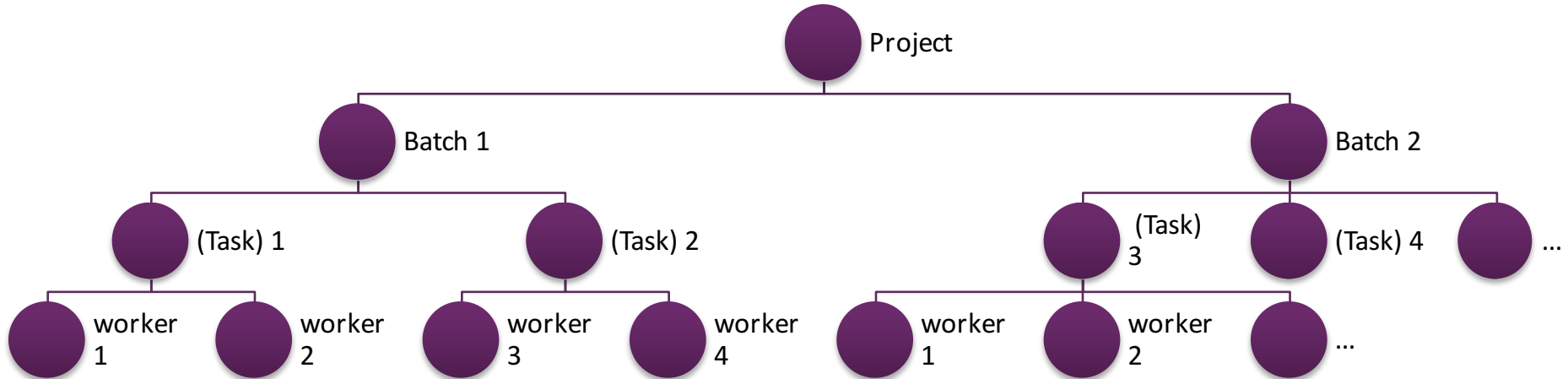


Optimal Award



#Registration vs. Submission Ratio

- Example



- Research questions

- To what extent, CSD is employed to increase task parallelism?
- Is there any patterns between task parallelism and award?
- To what extent, CSD projects can achieve both task parallelism and work behaviors.

Category	Metric	Measurement
<b>Task Setting</b>	Award	Dollars in task description
	# of parallel Task (n)	Total number of tasks uploaded in a limited period of time ( month or week)
<b>Worker Performance</b>	# Registrants	Number of registrants that are willing to compete on total number of tasks in specific period of time
	# Submissions	Number of submissions that a task receives by its submission deadline in specific period of time
<b>Task outcome</b>	# Completed tasks	Number of Acceptable submissions that a task receives by its submission deadline in specific period of time
	# Cancelled tasks	Number of none-acceptable submissions that a task receives by its submission deadline in specific period of time
	Failure rate	# Cancel tasks/ n
	Stability	Average(#submissions/#registration)

- I: General task parallelism trend in CSD
- II: Correlation between task parallelism and award
- III: Correlation between task parallelism and task completion

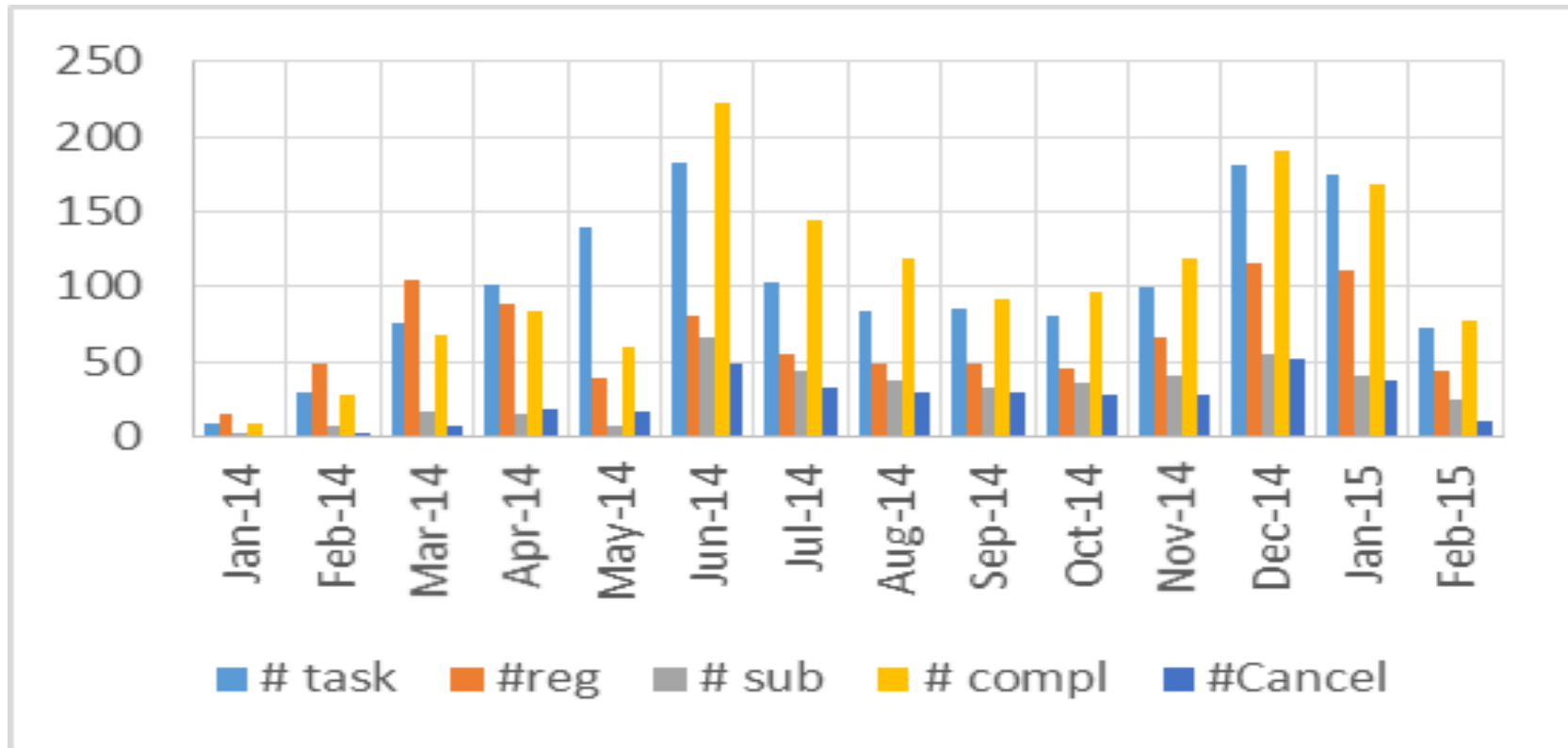
- Dataset: 403 individual projects including 4908 software development tasks from Jan 2014 to Feb2015
- Mostly one-month-duration
- Submissions reviewed by experts to check the final results and grant the scores
- Task failure:
  - No submission or
  - Score < 75
- Task Types
  - First2Finish: The first person to submit a passing entry wins
  - Assembly Competition: Assemble previous tasks
  - Bug Hunt: Find and fix available Bugs
  - Code: Programing specific task
  - UI Prototype



# Sample projects characteristics

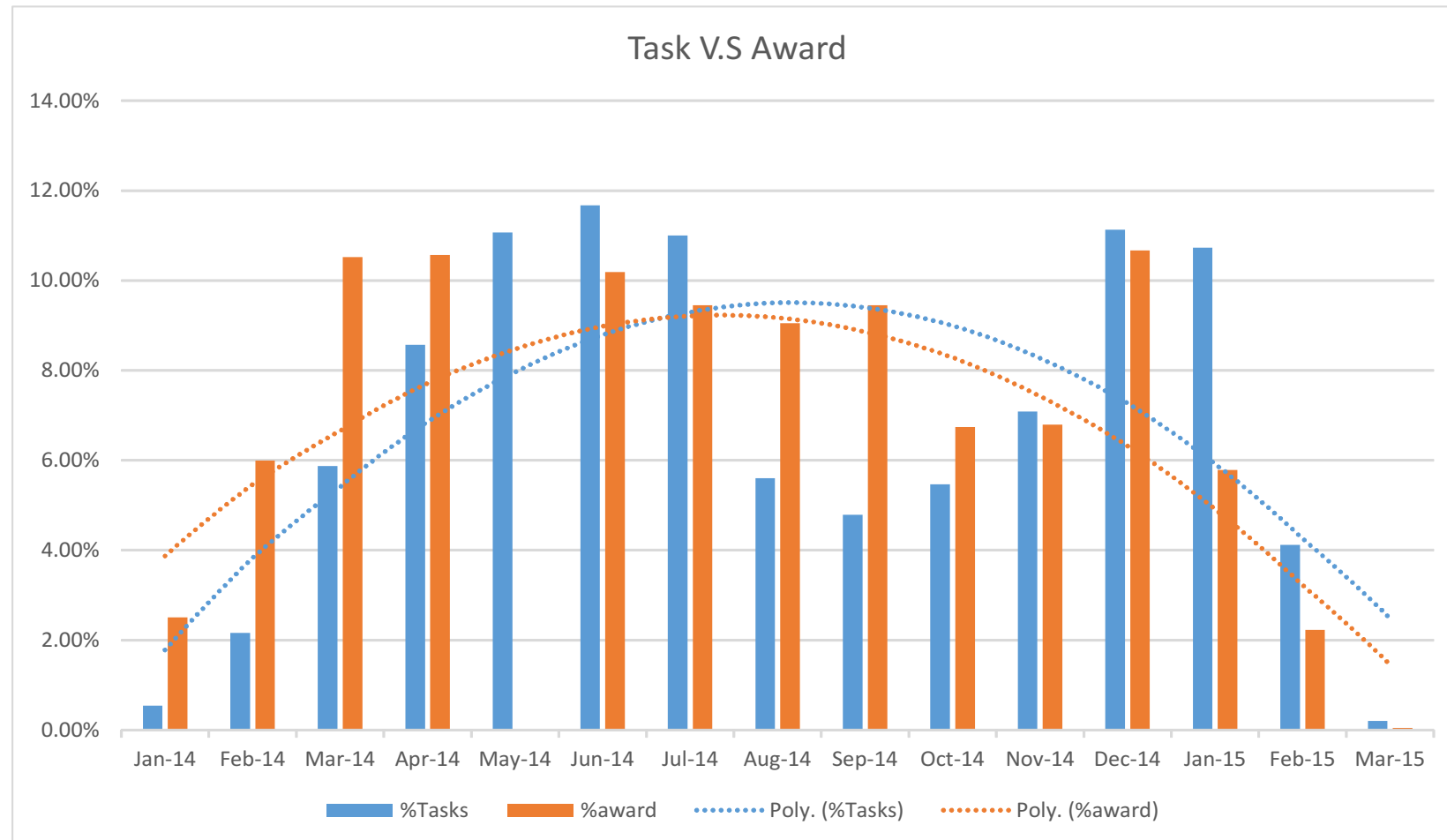
	# tasks	Application	Top 3 Technology Involved	Top 3 Challenge type	Top Platform Type
Project I	156	Topcoder API	Java (21),Node.js (123) SQL(4), REST (4)	First2Finish (85) ,Assembly Competition (67), Code(2) , Bug Hunt (2)	HTML, DocuSign, Heroku, NodeJS (142), Other
Project II	306	Topcoder, Web Arena IDE	Node.js(45),Angular.js (53), CSS (53), JavaScript (53), Bootstrap (53), HTML5 (53), CSS (51)	First2Finish (114),Assembly Competition(36), Bug Hunt(120)	Facebook, HTML(158), Other, NodeJSEC2,Heroku,un known
Project III	177	Topcoder Community	Angular.js (39), PHP (33, CSS (35)	Assembly Competition(51),First2Finis h (93),Code(16)	NodeJS, Wordpress (151),Heroku,EC2,Bea nstalk,AWS, HTML ,Unknown, Other
Project IV	277	Hercules Android APP	HTML5 (16), iOS (77), Android (177)	First2Finish(241),Bug Hunt (23), Assembly Competition (7)	iOS, Android(161), Mobile,HTML,EC2,Oth er,Unknown

# Trend of overall uploading tasks and associated award



- Dataset: 403 individual projects includes 4908 component development/testing/assembly tasks from Jan 2014 to Feb2015
- 4 projects with more than 100 tasks selected for further analysis

# Parallelism and Maximum uploaded tasks



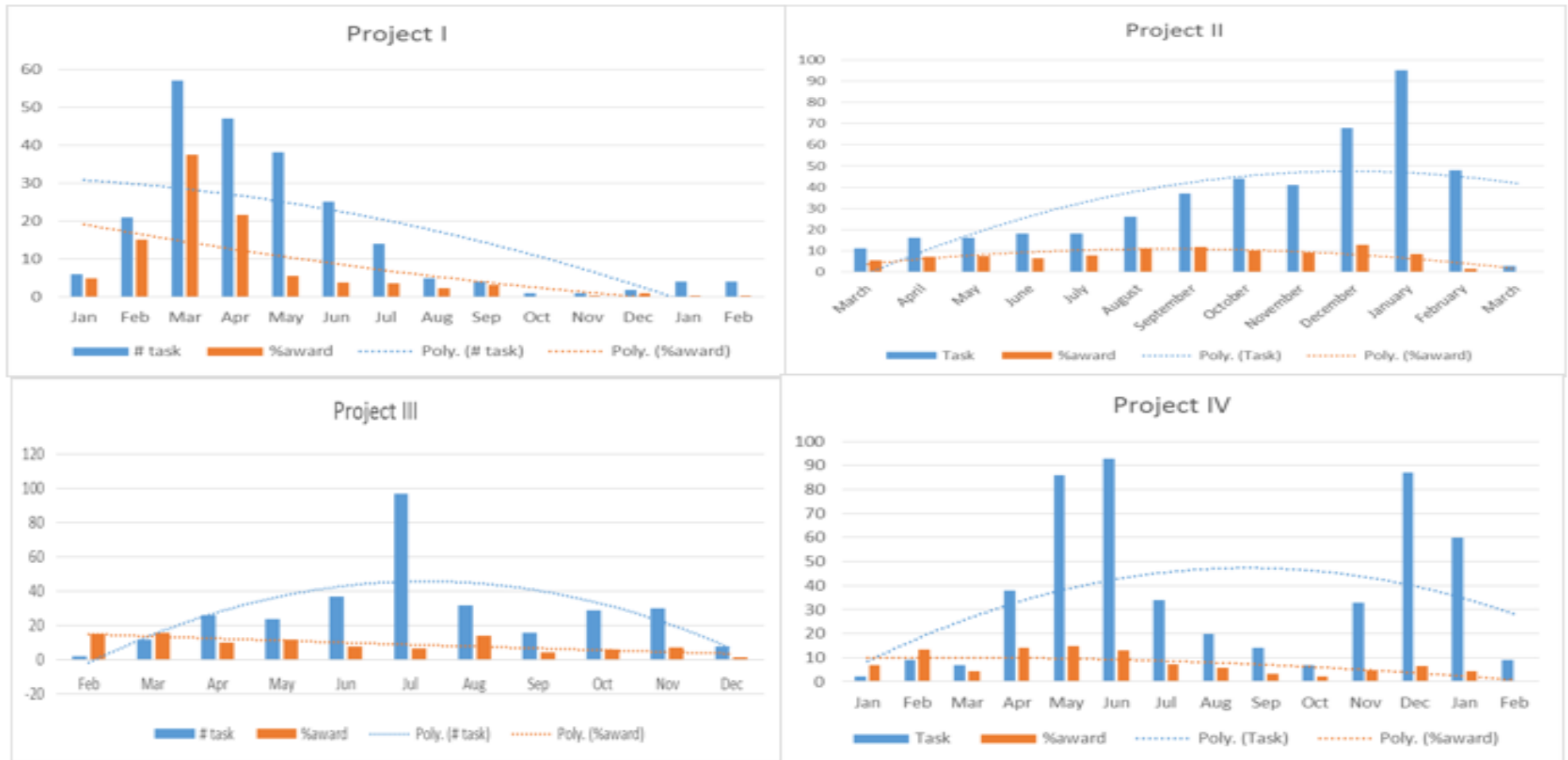
- Maximum #tasks don't correspond to max award%
- Individual task size negatively influences on associated award

# Statistics of Task Parallelism by Month

		# Task	Award	# Reg	# Sub	# comp task	#Can tasks
Sample Project 1	Min	1	30	6	0	0	0
	Median	5.5	3850	69	12	5.5	0
	Max	57	43225	763	117	51	6
	Average	16.35	8233.21	155.28	30.57	15.21	1.14
Sample Project 2	Min	3	280	18	4	3	0
	Median	26	20910	274	37	23	3
	Max	95	34671	752	159	89	10
	Average	33.92	20853.84	318.15	62.07	30.84	3.076
Sample Project 3	Min	2	2300	30	5	2	0
	Median	25	10632	235	49	16	8
	Max	47	22426	410	97	38	10
	Average	23.81	12842.09	219.72	49.18	17.45	6.36
Sample Project 4	Min	2	355	43	5	2	0
	Median	26.5	8667.5	147	43.5	22	3
	Max	93	21045	464	173	82	25
	Average	35.64	10180.92	172.28	61.92	29.21	6.57

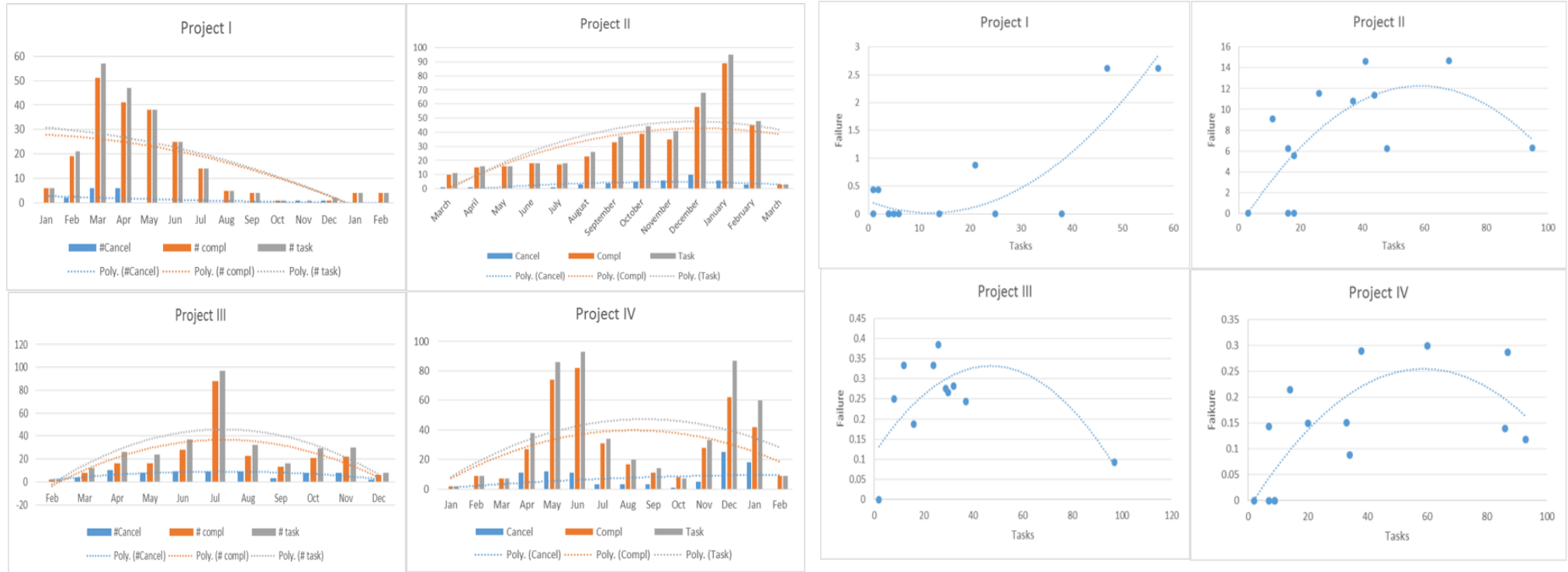
# Task Parallelism vs. Task Award%

## Award and Tasks and Registrants:



- Positive correlation of 0.44 between total number of uploaded tasks and total prize associated

# Task Parallelism vs. Stability



$$\text{Failure rate} = \# \text{ cancel tasks} / \# \text{ total tasks}$$

- (1) On average, 23% of the registrants submit their files by submissions deadline;
- (2) Stability of the projects is changing based on the number of uploading tasks

# Schedule Acceleration

Project	Project I	project II	Project III	Project IV
Effort (Worker-day)	6005.7	11037.2	8552.7	13841
Effort (Worker-month)	273	501.7	388.8	629.1
Duration (months)	9	13	11	14
McConnell	19.1	23.3	21.5	25.2
COCOMO II	17.7	20.9	19.5	22.3
QSAM	16.5	22.4	19.7	25.1
SAR (McConnell)	2.1	1.8	2	1.8
SAR (COCOMO II)	2	1.6	1.8	1.6
SAR (QSAM)	1.8	1.7	1.8	1.8

- Future Work
  - Mining task descriptions to derive better task parallel metrics
  - Mining patterns in task decomposition/scheduling
    - Consider more attributes such as worker motivation patterns, worker tenure and achievement level, etc.
    - Analyze impact of different task decomposition methods
      - Vertical decomposition vs. horizontal decomposition
  - Develop analytics for task decomposition w.r.t staffability and expected outcomes



# Thank you!

