

Project 3: Automating Database Question Generation and Marking - Team A

Aug 4 - Aug 9

Task Summary

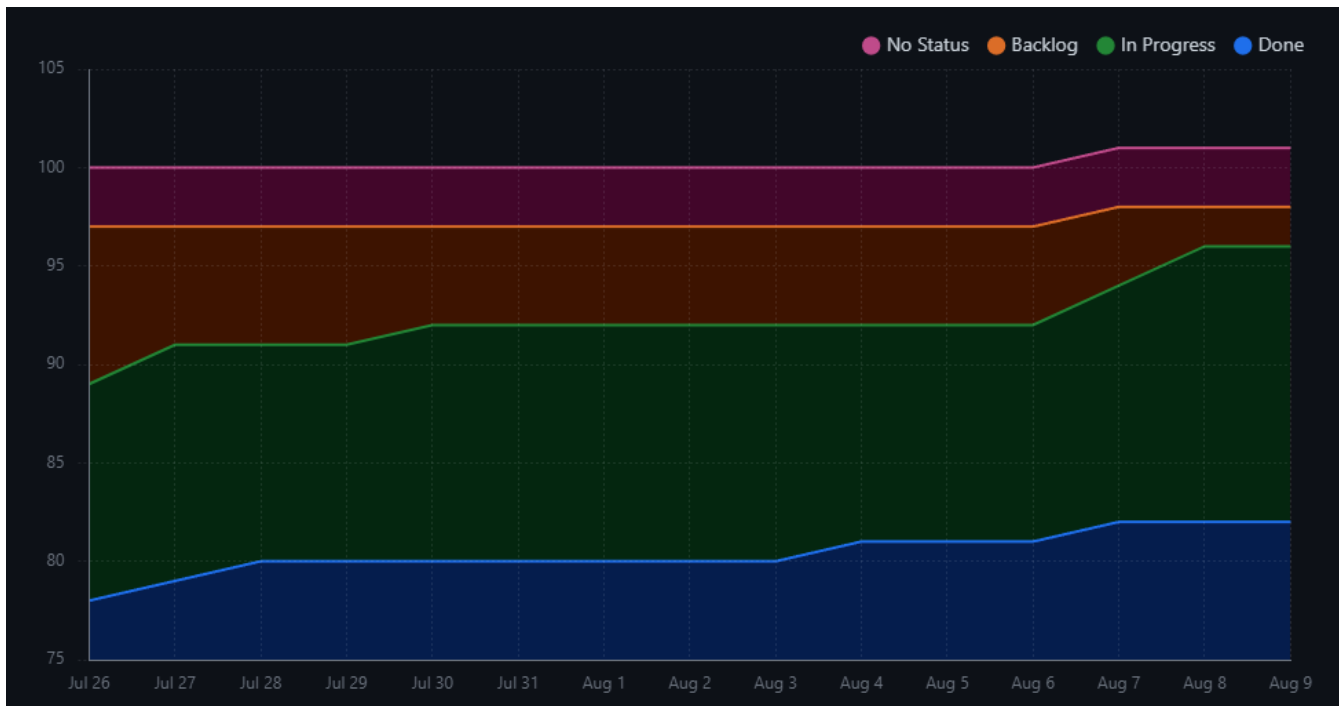
Completed since last meeting:

None - Development Finished

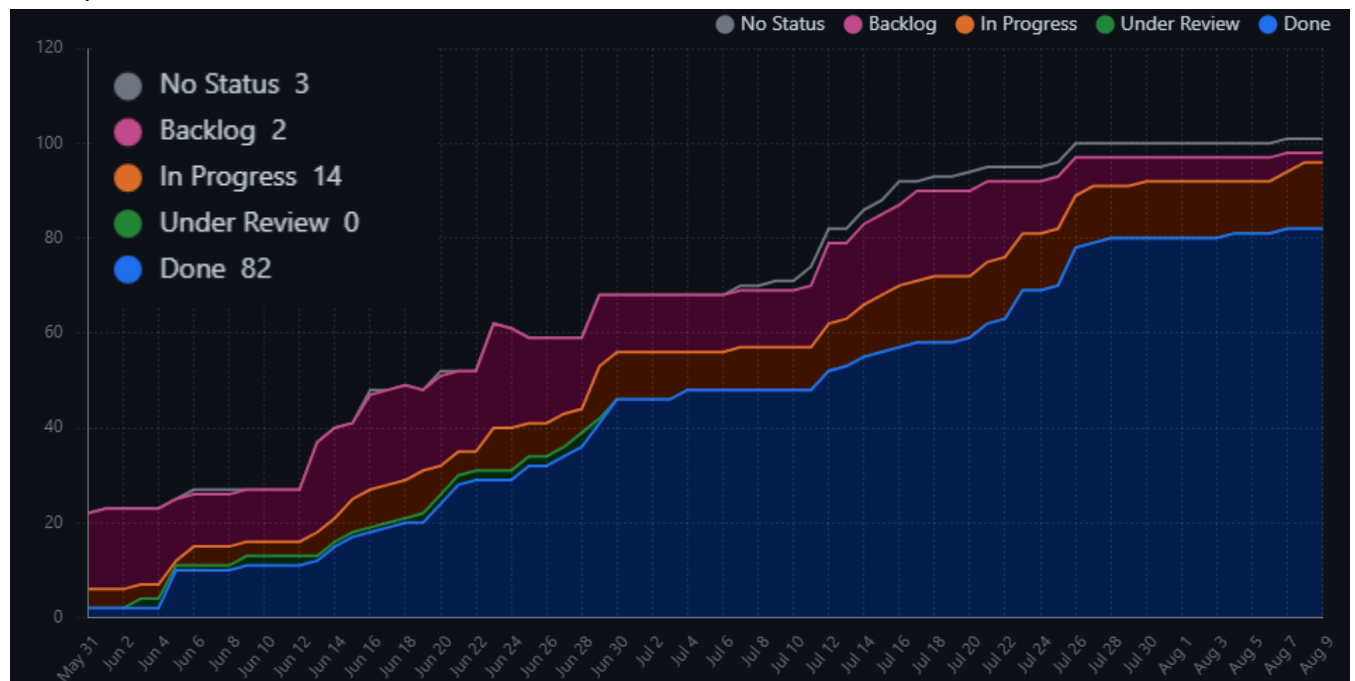
In progress:

	Title	Assignees	Status	T-Shirt Size
1	RelaX - Write Autogenerator Tests #61	MatthewObirek an...	In Progress	Small
2	Professor can set whether a question shows correct answer after student submits their ans... #44	azipis, Finkch, Mat...	In Progress	Medium
3	Update docker documentation for current repository. #62	azipis and Matthew...	In Progress	Small
4	Video Walkthrough #197	azipis and nishant-sr	In Progress	Small
5	Final Document #198	nishant-sr	In Progress	Small
6	Final Presentation #43	azipis, Finkch, Mat...	In Progress	Medium

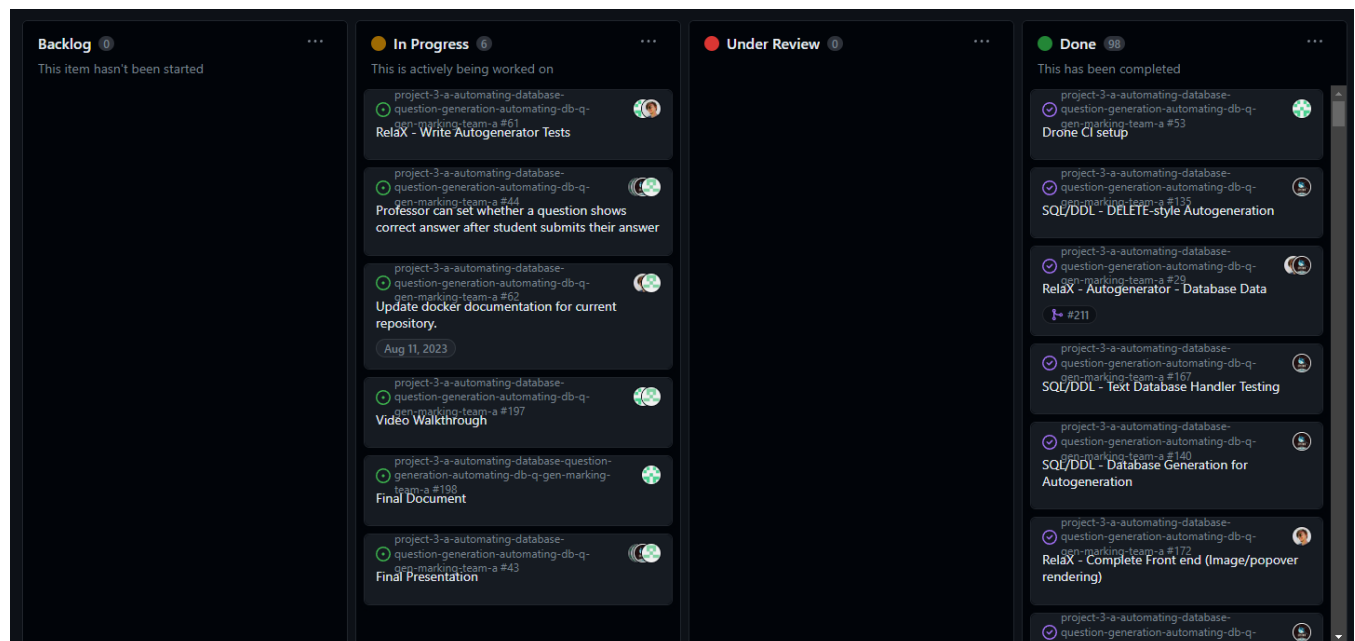
Burnup since last meeting:



Burnup to date:



Kanban Board:

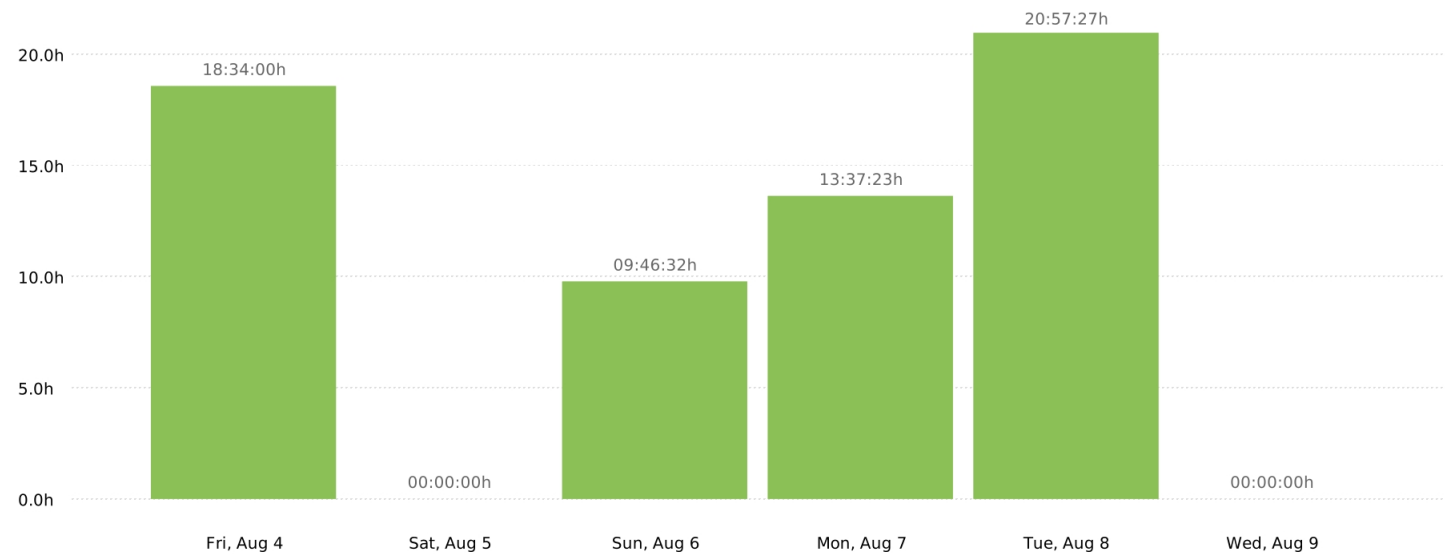


Summary report

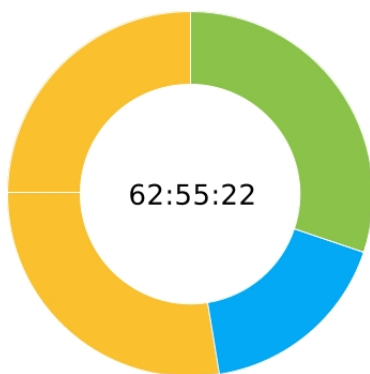
08/04/2023 - 08/09/2023



Total: 62:55:22

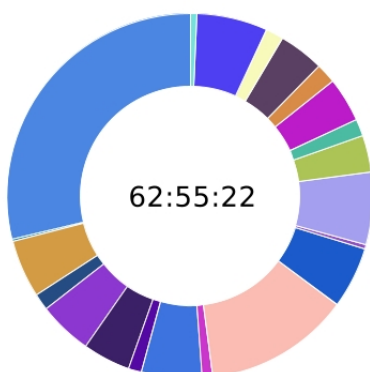


User



Andrei Zipis	15:36:51	24.81%
Matthew Obirek	17:26:37	27.72%
Nishant Srinivasan	10:54:41	17.34%
Skyler A.	18:57:13	30.12%

Description



Team Meeting	18:05:48	28.76%
Logs	00:08:27	0.22%
Testorama prep	03:13:19	5.12%
Deployment	00:55:43	1.48%
Test-O-Rama	02:58:20	4.72%

● Documentation	02:40:09	4.24%
● Client Meeting	00:45:00	1.19%
● Testorama	03:20:00	5.30%
● Design and Testing	00:34:57	0.93%
● Question Improvement and Refactoring	08:09:56	12.98%
● Test-o-Rama!	03:24:11	5.41%
● #238 Relax/lab1 integration	00:12:16	0.32%
● RelaX API	04:03:39	6.45%
● Bug Fix	02:01:16	3.21%
● OnClick-afy-ing autoGenerated code	00:57:29	1.52%
● Final Documentation	02:29:11	3.95%
● Test-o-rama prep	01:05:11	1.73%
● Tweeking Join randomization, and Other AUtogen to max output size	02:32:43	4.05%
● Bugfixes and Deployment	01:00:00	1.59%
● Commenting Code	03:55:27	6.24%
● Negative Marks + SQLite Fixes	00:22:20	0.59%

User / Description	Duration
Andrei Zipis	15:36:51
Testorama prep	03:13:19
Deployment	00:55:43
Team Meeting	06:11:54
#238 Relax/lab1 integration	00:12:16
RelaX API	04:03:39
Bugfixes and Deployment	01:00:00
Matthew Obirek	17:26:37
Team Meeting	03:56:11

Test-O-Rama	02:58:20
Bug Fix	02:01:16
OnClick-afy-ing autoGenerated code	00:57:29
Test-o-rama prep	01:05:11
Tweeking Join randomization, and Other AUtogen to max output size	02:32:43
Commenting Code	03:55:27
Nishant Srinivasan	10:54:41
Logs	00:08:27
Client Meeting	00:45:00
Testorama	03:20:00
Team Meeting	03:49:43
Final Documentation	02:29:11
Negative Marks + SQLite Fixes	00:22:20
Skyler A.	18:57:13
Team Meeting	04:08:00
Documentation	02:40:09
Design and Testing	00:34:57
Question Improvement and Refactoring	08:09:56
Test-o-Rama!	03:24:11

REQUIREMENTS		Type of Test	Pass or Fail	Contributor
Functional				N=Nishant, A=Andrei, S=Skyler, M=Matthew
1.1	System will allow for relational algebra statements to be entered.	UI Testing Integration Testing	Pass Pass	N,N
1.2	System will show visualizations of the resulting entered statement prior to submission.	UI Testing Integration Testing	Pass Pass	N, N
1.3	System will automatically mark the relational algebra questions once submitted.	Unit Testing	Pass	A
1.4	System will allow for DDL/SQL code to be entered. System will show resulting tables of queries prior to submission.	UI Testing Integration Testing	Pass Pass	N, N
1.5	System will automatically mark the DDL/SQL questions once submitted.	Unit Testing	Pass	N, N
1.6	Student will be able to see the correct answer if the professor has allowed for the correct answer to be displayed after the question is submitted.	Unit Testing Integration Testing UI Testing	Pass Pass Pass	A, M, N
1.7	Professor will be able to set whether the correct answer will be displayed after the question is submitted.	Unit Testing UI Testing	Fail Fail	
1.8	Professor will be able to see the correct answer.	UI Testing	Fail	
Non-Functional				
2.1	The system will support all COSC 304 users simultaneously – about 200 students.	Performance Testing	Fail	
2.2	The system will ensure data integrity and preservation so that no data is lost upon submission.	Performance Testing	Pass	
2.3	The system will display entered queries within 3 seconds at scale and under optimal conditions.	Performance Testing	Pass	
2.4	The system will return automarked submissions within 5 seconds at scale and under optimal conditions.	Performance Testing	Pass	
2.5	The user interface will match existing software used for COSC 304.	UI Testing	Pass	A, M
Technical Requirements				
3.1	Rebuild RelaX editor and calculator into PrairieLearn	UI Testing Integration Testing	Pass Pass	N, N
3.2	Frontend: JavaScript, HTML, CSS	UI Testing	Pass	N
3.3	Backend: Python, Node.JS	Unit Testing	Pass	N,A,S
3.4	Write JavaScript code that takes in SQL/DDI statements and displays appropriate table results	Integration Testing	Pass	N
3.5	Write Python code that automatically marks submitted data and returns the students grade	Unit Testing	Pass	A, N