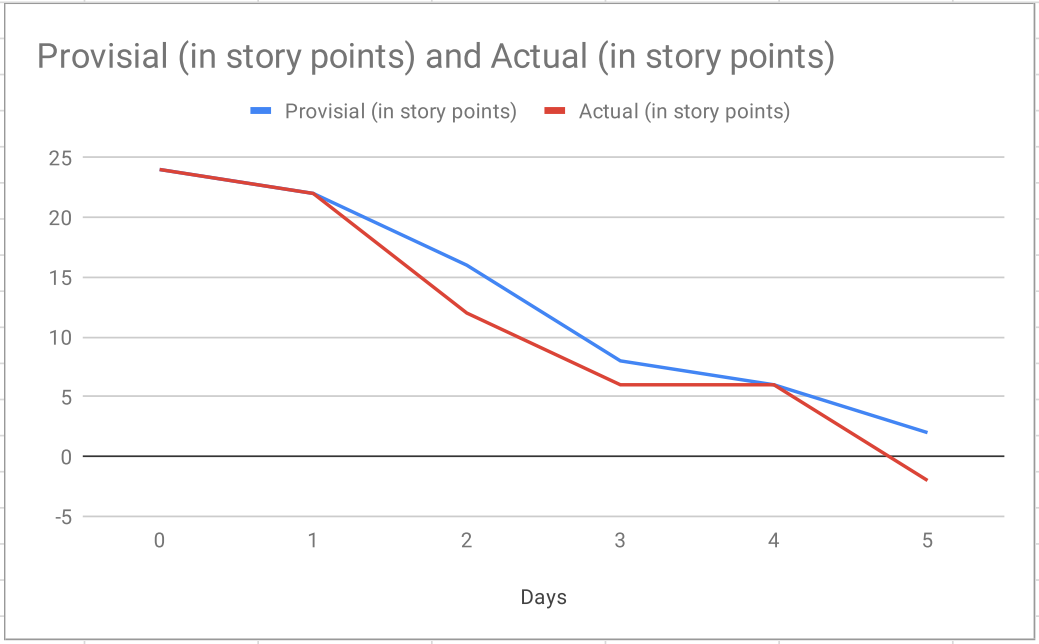
	Tasks	Description	Dependency	Story Points	1	2	3	4	5			
3	3.6	Check the unique identifier for each entry and make sure it is not already in the database before inserting it.		4	D:2	D:2						
3	3.7	Alert the user if the template is already uploaded		2			D:2					
3	3.8	Format JSON template data in the database in the way that was discussed during the sprint planning.		2			R:2					
4	4.1	Plan the technical process of generating a report from frontend to backend.		2	R:2							
4	4.2	Create an HTML page for generating the reports with a dropdown menu to choose the template type that the client wants to generate the report for.		4		C:2	C:2					
4	4.3	Send a post request to the backend when the generate report button is clicked.	4.2	2				C:2				
4	4.4	Access the information in the database needed to generate the report for a certain template type.	4.1, 3.8	2		R:2						
4	4.5	Calculate all the values to be inserted into table.	4.4	8			B:2	B:2	B:2			
4	4.6	Generate a table (spreadsheet or HTML) based on the values from the JSON object.	4.1	8								
4	4.7	Display the table on an HTML page.	4.7	2								
4	4.8	Format the report to contain all the required logos, titles, etc., in a way that can be physically printed on 8.5 x 11 letter sized pages.	4.8	4								
5	5.1	Insert a sample of all 8 templates into the database so we have validation data to compare to.		2								
5	5.2	Check that the headers of each uploaded template match the expected headers of the template type that they choose using the dropdown menu	5.1	2								
		PLAN SPRINT #3										

	Tasks	Description	Dependency	Story Points	1	2	3	4	5			
3	3.6	Check the unique identifier for each entry and make sure it is not already in the database before inserting it.		4	D:2	D:2						
3	3.7	Alert the user if the template is already uploaded		2			D:2					
3	3.8	Format JSON template data in the database in the way that was discussed during the sprint planning.		2			R:2					
4	4.1	Plan the technical process of generating a report from frontend to backend.		2	R:2							
4	4.2	Create an HTML page for generating the reports with a dropdown menu to choose the template type that the client wants to generate the report for.		4		C:4						
4	4.3	Send a post request to the backend when the generate report button is clicked.	4.2	2			C:2					
4	4.4	Access the information in the database needed to generate the report for a certain template type.	4.1, 3.8	2		R:2						
4	4.5	Calculate all the values to be inserted into table.	4.4	8			B:2	B:2	B:4			
4	4.6	Generate a table (spreadsheet or HTML) based on the values from the JSON object.	4.1	8								
4	4.7	Display the table on an HTML page.	4.7	2								
4	4.8	Format the report to contain all the required logos, titles, etc., in a way that can be physically printed on 8.5 x 11 letter sized pages.	4.8	4								
5	5.1	Insert a sample of all 8 templates into the database so we have validation data to compare to.		2								
5	5.2	Check that the headers of each uploaded template match the expected headers of the template type that they choose using the dropdown menu	5.1	2								
		EXECUTION SPRINT #3										

Days	0	1	2	3	4	5
Provisial (in story points)	24	22	16	8	6	2
Actual (in story points)	**24**	22	12	6	6	-2

BURNDOWN SPRINT # 3



	Tasks	Description	Dependency	Story Points	1	2	3	4	5
4	4.6	Generate a table (spreadsheet or HTML) based on the values from the JSON object.	4.1	6	D:2	D:2	D:2		
4	4.7	Display the table on an HTML page.	4.7	2					
4	4.8	Format the report to contain all the required logos, titles, etc., in a way that can be physically printed on 8.5 x 11 letter sized pages.	4.8	4					
5	5.1	Insert a sample of all 8 templates into the database so we have validation data to compare to.		2		B:2			
5	5.2	Check that the headers of each uploaded template match the expected headers of the template type that they choose using the dropdown menu	5.1	4			B:2	B:2	
5	5.3	Unit Test the Database Endpoints and make sure they cover cases where a new entry is being inserted, getting an existing entry, creating an entry that is already there, and getting a entry that does not exist.		6	R:2			R:2	R:2
5	5.4	Work on the Acceptance Test document and make sure to include photos as well as descriptions on what each test is and how to test it. The tests should cover last sprints and also the ones in this deliverable		6	C:2	C:2	C:2		
		PLAN SPRINT #4							

	Tasks	Description	Dependency	Story Points	1	2	3	4	5
4	4.6	Generate a table (spreadsheet or HTML) based on the values from the JSON object.	4.1	6	D:2	D:2			
4	4.7	Display the table on an HTML page.	4.7	2			D:2		
4	4.8	Format the report to contain all the required logos, titles, etc., in a way that can be physically printed on 8.5 x 11 letter sized pages.	4.8	4					
5	5.1	Insert a sample of all 8 templates into the database so we have validation data to compare to.		2		B:2			
5	5.2	Check that the headers of each uploaded template match the expected headers of the template type that they choose using the dropdown menu	5.1	4			B:2	B:2	
5	5.3	Unit Test the Database Endpoints and make sure they cover cases where a new entry is being inserted, getting an existing entry, creating an entry that is already there, and getting a entry that does not exist.		6	R:2			R:2	R:2
5	5.4	Work on the Acceptance Test document and make sure to include photos as well as descriptions on what each test is and how to test it. The tests should cover last sprints and also the ones in this deliverable		6			C:2	C:2	C:2
		EXECUTIONSPRINT #4							

Days	0	1	2	3	4	5
Provisial (in story points)	24	24	22	10	6	0
Actual (in story points)	**24**	24	16	14	10	-2

BURNDOWN SPRINT # 4