AMOS-Project 3 – QAchat

Planning Documents

Last Name	First Name	GitHub User Name	Email Address
Alkadour	Abdelkader	Kadi-7	a.alkadour@campus.tu-berlin.de, basickadour@gmail.com
Arifin	Hafidz	zenzeii	h.arifin@campus.tu-berlin.de, hafidz.harifin@gmail.com
El Brak	Sara	SaraElBrak	sara.el@fau.de
Erben	Emanuel	emuguy1	emanuel.erben@fau.de, emanuel.erben@gmail.com
Konheiser	Tobias	tkonheiser	tobias.konheiser@fau.de
Stojkovic	Vukica	vukica1	vukica.stojkovic@yahoo.de / vukica.stojkovic@campus.tu-berlin.de
Nützel	Felix	Felix-012	felix.nuetzel@fau.de
Palarus	Jesse	jtshark	j.palarus@campus.tu-berlin.de, jtsharkjtshark@gmail.com
Pucic	Amela	amela16	a.pucic@campus.tu-berlin.de, amela1999@hotmail.de

#	Meeting Day	Product Owner	Software Developer	Release Manager	Scrum Master	Comment
1	2022-10-19	Tobias Konheiser, Sara El Brak	Everyone else	Emanuel Erben	Vukica Stojkovic	
2	2022-10-26	Sara El Brak, Tobias Konheiser	Everyone else	Emanuel Erben	Vukica Stojkovic	
3	2022-11-02	Tobias Konheiser, Sara El Brak	Everyone else	Emanuel Erben	Vukica Stojkovic	
4	2022-11-09	Sara El Brak, Tobias Konheiser	Everyone else	Emanuel Erben	Vukica Stojkovic	
5	2022-11-16	Tobias Konheiser, Sara El Brak	Everyone else	Emanuel Erben	Vukica Stojkovic	
6	2022-11-23	Sara El Brak, Tobias Konheiser	Everyone else	Emanuel Erben	Vukica Stojkovic	
7	2022-11-30	Tobias Konheiser, Sara El Brak	Everyone else	Emanuel Erben	Vukica Stojkovic	Mid-term due
8	2022-12-07	Sara El Brak, Tobias Konheiser	Everyone else	Emanuel Erben	Vukica Stojkovic	
9	2022-12-14	Tobias Konheiser, Sara El Brak	Everyone else	Emanuel Erben	Vukica Stojkovic	
10	2023-01-11	Sara El Brak, Tobias Konheiser	Everyone else	Emanuel Erben	Vukica Stojkovic	
11	2023-01-18	Tobias Konheiser, Sara El Brak	Everyone else	Emanuel Erben	Vukica Stojkovic	
12	2023-01-25	Sara El Brak, Tobias Konheiser	Everyone else	Emanuel Erben	Vukica Stojkovic	
13	2023-02-01	Tobias Konheiser, Sara El Brak	Everyone else	Emanuel Erben	Vukica Stojkovic	
14	2023-02-08	Sara El Brak, Tobias Konheiser	Everyone else	Emanuel Erben	Vukica Stojkovic	Demo day!
15	2023-02-15	Tobias Konheiser, Sara El Brak	Everyone else	Emanuel Erben	Vukica Stojkovic	Retrospective

Goals	Develop a good quality and working Chatbot based on defined requirements
	Achieve the technical target in good atmosphere and clear communication
Meeting norms	Meeting topics are inserted in the agenda before the meeting starts
	Everybody aims to be on time, but being late is communicated beforehand and handled in an agile way
	Meeting topics need to be sharp and precise
	Meeting time must no be exceeded more than 30 min, otherwise schedule a new meeting
Working norms	We value quality over quantity
	Everyone contributes regularly and communicates openly
Coordination norms	Everyone sticks to their roles and in case of problems communicates
Communication norms	We check our communication channels at least once a day
	Important messages are send in our WhatsApp group
Consideration norms	We discuss disagreement openly
	We vote for a final resolution
	We help in case someone needs it
Cont. improvement norms	Happiness index and stand up emails are reviewed in team meeting
	If problmes are recognized escalate them to the team
Rewards	Online team event
	Everyone celebrates via a reaction in the zoom chat after each sprint
Sanctions	Assign unwanted jobs to person (rework a file, ...)
Signatures	Tobias Konheiser
	Hafidz Arifin
	Amela Pucic
	Emanuel Erben
	Sara El Brak
	Jesse Palarus
	Felix Nützel
	Abdelkader Alkadour
	Vukica Stojkovic

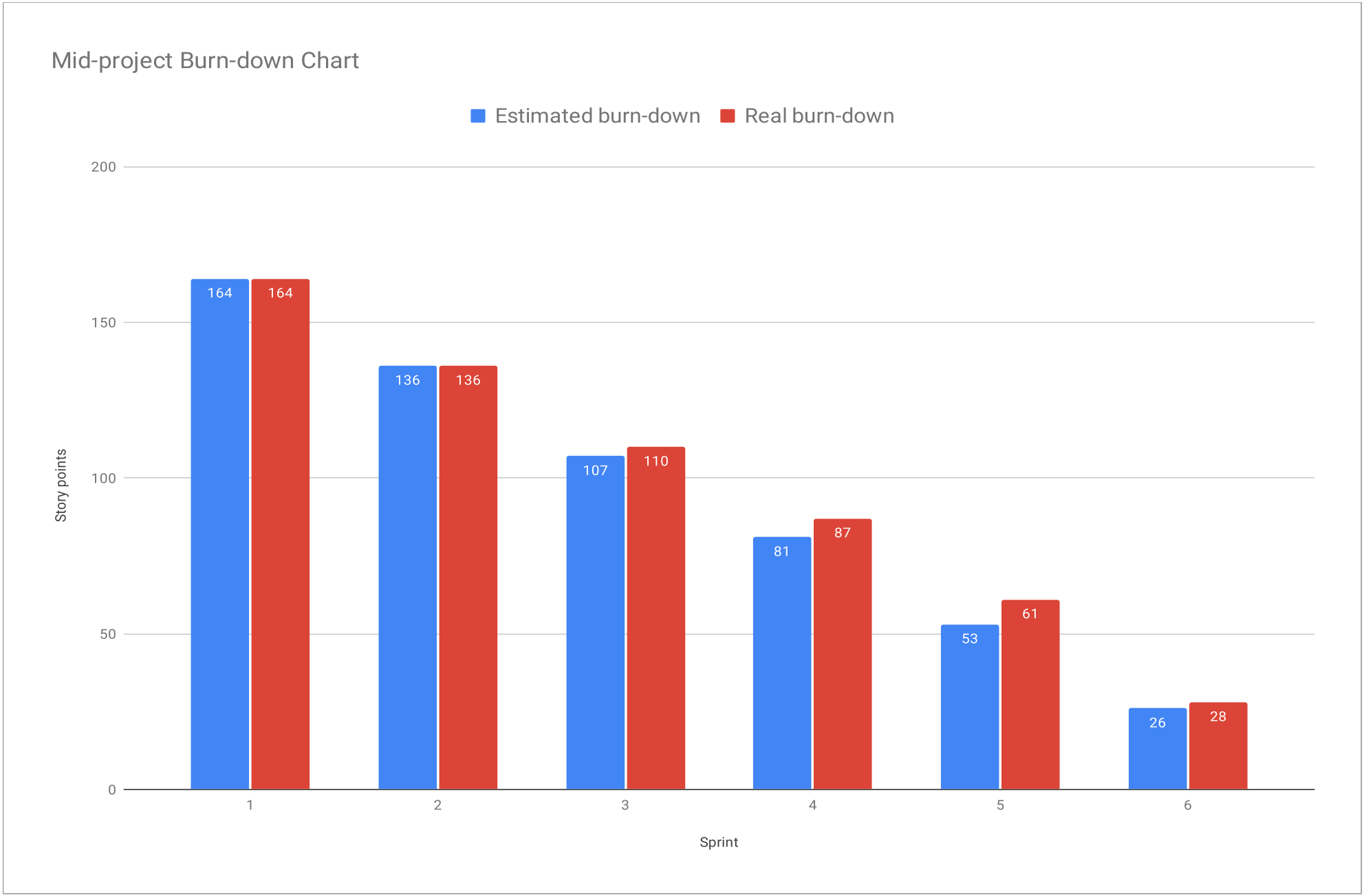
Product Vision	Project Mission
QAchat envisions an environment in which access to knowledge is just a message away. We aim to leverage the rapid advancement in language model technologies to create a seamless interface that enables employees to get their questions answered accurately, quickly, efficiently, and with ease - by a general language model that is trained on specific knowledge. Our goal is to provide a simple and convenient point of contact, with an easy-to-use interface that is integrated into widely used communication tools, and to make knowledge accessible to everyone - irrespective of their geographical location, language or technical ability.	QAchat evaluates newly developed LLMs to create a chatbot that provides users with accurate, reliable and context-specific answers to their questions - with a focus on accessibility and ease of use. The best suited network is trained on provided data that is collected from existing communication and documentation sources. The model is made available to users through a Slackbot integration, where questions can be asked and answers are provided.

Project Name	...
Online team meeting	https://fau.zoom.us/j/68283073150
Production system (if any)	...
Test system (if any)	...
GitHub repository	amosproj/amos2023ss03-qachat (github.com)
GitHub feature board	amos2023ss03-feature-board (github.com)
GitHub impediments backlog	amos2023ss03-impediments-backlog (github.com)
Team T-shirt (white)	...
Team T-shirt (black)	woman design: https://www.shirtinator.de/t-shirts/gestalten/t-shirt-bedrucken#/load/share/931c832c-67cc-46ca-bca7-e49019a052f2
	man design: https://www.shirtinator.de/t-shirts/gestalten/t-shirt-bedrucken#/load/share/d45e26d4-77f0-42cf-a412-f67b2071facf
Additional materials	...
Course information	https://amos.uni1.de
Happiness index tool	https://happy-amos.appspot.com/
Planning Poker	https://planningpokeronline.com/

Term	Definition
Administrator (Admin)	An Administrator is a person who has access to all parts of the project.
Application Programming Interface (API)	An API is a defined interface that applications can use to exchange data and information.
Artificial Intelligence (AI)	Artificial Intelligence is a field of research that aims to make computers think and act like humans.
BERT	BERT is an open source LLM that has been developed by Google.
Chatbot	A chatbot is an application that can communicate with a user through short text messages and answer questions using artificial intelligence.
Company-Internal Information	Company-Internal Information is information about the company and its projects and processes that is publicly available or stored in Confluence, Slack, and Google Drive.
Confluence	Confluence is a software used to document various types of data.
Google Drive	Google Drive is a cloud storage solution provided by Google.
Large Language Model (LLM)	A Large Language Model is an AI model specialized for text and sentence generation.
LLaMA	LLaMA is an open source LLM that has been developed by Meta and Stanford.
Slack	Slack is a software that is used for text messaging between groups or individuals.
Slackbot	A Slackbot is a chatbot that is integrated into Slack.
T5	T5 is an open source LLM that has been developed by Google.
User	A user is a person who interacts with the system by chatting with the Slackbot.

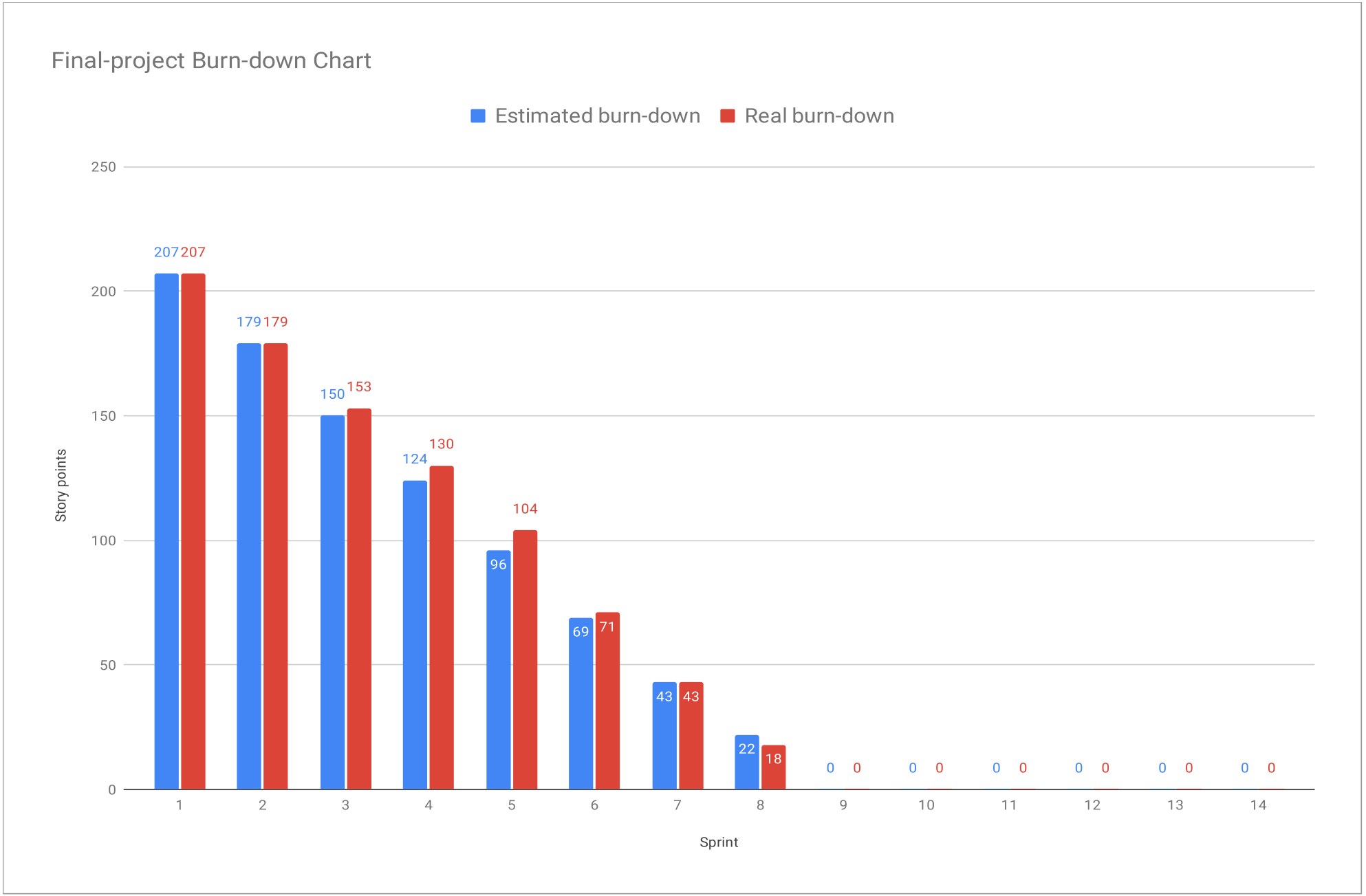
Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release							
	Total			164	164		
Sprints							
1	**Large Language Model methods**			28	164	28	164
2	**Software architecture**			29	136	26	136
3	**Large Language Model prototypes**			26	107	23	110
4	**Code frameworks**			28	81	26	87
5	**Setup & Documentation**			27	53	33	61
6	**Data Integration & Documentation Enhancement**			26	26	28	28
Features							
1	**Large Language Model methods**						
		Identify the existing capabilities of LLM methods and their underlying algorithms					
			Research Slack bot requirement	5		5	
			Research LLM models	5		5	
			Research LLM method 1 (search API) properties	5		5	
			Research LLM method 2 (semantic search) properties	5		5	
			Research LLM method 3 (fine tuning) properties	5		5	
			Team logo design	3		3	
2	**Software architecture**						
		Determine the software architecture and the used components					
			Define diagram of runtime components	5		5	
			Define diagram of code components	8		8	
			A summary of the underlying technology stack	5		3	
			Textual explanation of the diagrams and choices	5		5	
			Initialize code repository	3		3	
			Initialize the software bill of materials	3		2	
3	**Large Language Model prototypes**						
		Further evaluate the functionality of each LLM method					
			Research semantic search vector storage	5		3	
			Research Slack web server hosting	5		3	
			Implement Alpaca/LLaMA LLM prototype	3		5	
			Implement BERT LLM prototype	5		3	
			Implement T5 LLM prototype	3		3	
			Create the LLM-server code framework	3		5	
			Create coding guidelines	2		1	
4	**Code frameworks**						
		Provide a structured foundation for building the chatbot					
			Research LLM server hosting	5		5	
			Determine the communication protocols used	3		3	
			Create Slack bot code framework	5		3	
			Create the semantic search code framework	3		3	
			Create the data processing code framework	3		5	
			Update product vision and product mission	3		2	
			Test Slack	3		2	
			Test DeepL API	3		3	
5	**Setup & Documentation**						

Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
		Prepare the necessary setups and extend the documentation					
			Create a build process video	5		8	
			Create a secure and private file exchange channel	3		3	
			Create testing setup	5		8	
			Set up vector database	2		2	
			Set up LLM for embedding generation	2		2	
			Move existing documentation to GitHub Wiki	5		5	
			Document Slackbot setup process	2		2	
			Set up LLM for chat message generation	3		3	
6	**Data Integration & Documentation Enhancement**						
		Enhance data integration capabilities and improve project documentation					
			Setup LLM in the Google could	8		8	
			Implement a blacklist for Confluence pages and other data sources	3		3	
			Read data from Confluence into vector database	5		5	
			Read data from PDF into vector database	3		5	
			Initialize user, (technical) design, and build/deploy documentation	5		5	
			Clean-up mid-project release plan & create final project release plan	2		2	


Mid-project Burn-down Chart
Estimated burn-down
Real burn-down
200
150
100
50
0
Story points
164
164
136
136
107
110
81
87
53
61
26
28
1
2
3
4
5
6
Sprint

Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release							
	Total			207	207		
Sprints							
1	**Large Language Model methods**			28	207	28	207
2	**Software architecture**			29	179	26	179
3	**Large Language Model prototypes**			26	150	23	153
4	**Code frameworks**			28	124	26	130
5	**Setup & Documentation**			27	96	33	104
6	**Data Integration & Documentation Enhancement**			26	69	28	71
7	**Cloud Hosting & Data Integration**			21	43	25	43
8	**Cloud Documentation & Database Management**			22	22		18
9					0		0
10					0		0
11					0		0
12					0		0
13					0		0
14					0		0
Features							
1	**Large Language Model methods**						
		Identify the existing capabilities of LLM methods and their underlying algorithms					
			Research Slack bot requirement	5		5	
			Research LLM models	5		5	
			Research LLM method 1 (search API) properties	5		5	
			Research LLM method 2 (semantic search) properties	5		5	
			Research LLM method 3 (fine tuning) properties	5		5	
			Team logo design	3		3	
2	**Software architecture**						
		Determine the software architecture and the used components					
			Define diagram of runtime components	5		5	
			Define diagram of code components	8		8	
			A summary of the underlying technology stack	5		3	
			Textual explanation of the diagrams and choices	5		5	
			Initialize code repository	3		3	
			Initialize the software bill of materials	3		2	
3	**Large Language Model prototypes**						
		Further evaluate the functionality of each LLM method					
			Research semantic search vector storage	5		3	
			Research Slack web server hosting	5		3	
			Implement Alpaca/LLaMA LLM prototype	3		5	
			Implement BERT LLM prototype	5		3	
			Implement T5 LLM prototype	3		3	
			Create the LLM-server code framework	3		5	
			Create coding guidelines	2		1	

Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
4	**Code frameworks**						
		Provide a structured foundation for building the chatbot					
			Research LLM server hosting	5		5	
			Determine the communication protocols used	3		3	
			Create Slack bot code framework	5		3	
			Create the semantic search code framework	3		3	
			Create the data processing code framework	3		5	
			Update product vision and product mission	3		2	
			Test Slack	3		2	
			Test DeepL API	3		3	
5	**Setup & Documentation**						
		Prepare the necessary setups and extend the documentation					
			Create a build process video	5		8	
			Create a secure and private file exchange channel	3		3	
			Create testing setup	5		8	
			Set up vector database	2		2	
			Set up LLM for embedding generation	2		2	
			Move existing documentation to GitHub Wiki	5		5	
			Document Slackbot setup process	2		2	
			Set up LLM for chat message generation	3		3	
6	**Data Integration & Documentation Enhancement**						
		Enhance data integration capabilities and improve project documentation					
			Setup LLM in the Google could	8		8	
			Implement a blacklist for Confluence pages and other data sources	3		3	
			Read data from Confluence into vector database	5		5	
			Read data from PDF into vector database	3		5	
			Initialize user, (technical) design, and build/deploy documentation	5		5	
			Clean-up mid-project release plan & create final project release plan	2		2	
7	**Cloud Hosting & Data Integration**						
		Establish cloud hosting of the software and expand data integration abilities					
			Setup LLM in the Google could	5		8	
			Read data from Confluence into vector database	3		5	
			Document and summarise all services available to the project team	2		1	
			Read data from Slack conversation into vector database	5		5	
			Detect Slackbot language	3		3	
			Create integration tests for database reading and writing	3		3	
8	**Cloud Documentation & Database Management**						
		Document cloud hosting setup and further enhance database management					
			Setup LLM in the Google could (documentation)	3			
			Check conformity with NDA	1			
			Cleanup the repository	5			
			Split up long text block for database	5			
			Delete outdated database enteries for all data sources	5			
			Read data from Confluence into vector database (finalization)	3			


Final-project Burn-down Chart
Estimated burn-down
Real burn-down
Story points
Sprint
250
200
150
100
50
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
207 207
179 179
150 153
124 130
96 104
69 71
43 43
22 18
0 0
0 0
0 0
0 0
0 0
0 0

#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done
10	Acceptance criteria are met.		
11	Work products are uploaded to the Github repository.		
12	A pull request is created for each related branch.		
13	The work products in the pull requests are reviewed.		
14	The corresponding branches are merged and closed.		
15	The bill of materials section of the planning documents is updated.		
16	All defined conventions are complied with.		
21		A release candidate with a working and meaningful increment to the previous sprint is tagged.	
22		Previously established features and security mechanisms must continue to work.	
23			
31			The project can be successfully built and deployed.
32			All created tests are passed.
34			The implemented features pass a simple user test.
35			Developer documentation is created.
36			User documentation is created and updated
37			The release has been approved by all team members

Type	Link / reference
Team Meeting Agenda	Team Meeting Agenda
Checklists	https://docs.google.com/spreadsheets/d/1K46ImoocSKWYXWQgVVGndU6QzNazhF-i7bsbbovnpMk/edit?usp=sharing

#	Context	Name	Version	License	Comment
1	Programming Language	python	3.8	Python License 2.0.1	
2	Backend-as-a-Service (BaaS)	supabase	1.0.3	MIT License	
3	Python-C++ Integration	llama-cpp-python	0.1.39	MIT License	
4	Natural Language Processing (NLP)	langchain	0.0.154	MIT License	
5	Slack Integration	slack_sdk	3.21.3	MIT License	
6	Artificial Intelligence (AI)	openai	0.27.5	MIT License	
7	Atlassian API Wrapper	atlassian-python-api	3.36.0	Apache License 2.0	
8	Web Automation Testing	selenium	4.9.0	Apache License 2.0	
9	Numerical Computing	numpy	1.24.3	BSD License (BSD-3-Clause)	
10	Data Manipulation	pandas	2.0.1	BSD License (BSD-3-Clause)	
11	Aleph Alpha API Client	aleph-alpha-client	3.1.0	MIT License	
12	Sentence Embeddings	sentence_transformers	2.2.2	Apache License 2.0	
13	Embedding for Instructors	InstructorEmbedding	1.0.0	MIT License	
14	Slack App Framework	slack-bolt	1.18.0	MIT License	
15	Slack App Framework	slack-sdk	3.21.3	MIT License	
16	Machine Translation	deepl	1.14.0	MIT License	
17	Environment Variables	python-dotenv	1.0.0	BSD License (BSD-3-Clause)	
18	Hugging Face Model Hub	huggingface_hub	0.14.1	Apache Software License	
19	Type Inspection	typing-inspect	0.8.0	MIT License	
20	Type Hints Extensions	typing_extensions	4.5.0	Python Software Foundation License	
21	Unit Testing	pytest	7.3.1	MIT License	
22	PDF Parsing	pdfminer.six	20221105	MIT License	
23	Deep Learning	pytorch	1.0.2	BSD-3	
24	Natural Language Processing (NLP)	transormers			
25	Natural Language Toolkit	nltk	3.8.1	Apache Software License	
26	Optical Character Recognition (OCR)	pytesseract	0.3.10	Apache Software License	image analysis. Tesseract needs to be installed an dpath added

Last Name	First Name	Value					
Alkadour	**Abdelkader**	0		**0.00**	**OK**		
Arifin	**Hafidz**	0					
El Brak	**Sara**						
Erben	**Emanuel**	0					
Konheiser	**Tobias**			0	No size		
Stojkovic	**Vukica**			1	Trivial size		
Nützel	**Felix**	0		2	Small size		
Palarus	**Jesse**	0		3	Medium size		
Pucic	**Amela**	0		5	Large size		
				8	Very large size		
				13	Too large (size)		

Additional Documentation

Team Meeting Agenda "AMOS QAchat"

Date: 2023_06_14

ID	Topic	Time	Author	Description	Result	Decision	Responsible
1	Sprint Review	30	PO	- Release Manager creates release candidate build - PO walks through "awaiting review" tickets, probing SDs			
2	Sprint Release	5	PO	- PO decides release - Release Manager creates release			
3	Sprint Retrospective	15	SM	- SM reviews the impediments - SM performs roll calls - Everyone answers happiness index - Review of Happiness index and standup emails			
4	Sprint Planning	30	PO	- PO works through product backlog - SD perform planning poker - Introduce sprint goal			
5	Planning to the End	5	Tobi	What do you want to implement in any case?			
6							
7							
8							
9							
10	Open Points	5	Everybody				
		90					

Team Meeting Agenda "AMOS QAchat"

Date: 2023_06_07

ID	Topic	Time	Author	Description	Result	Decision	Responsible
1	Sprint Review	30	PO	- Release Manager creates release candidate build - PO walks through "awaiting review" tickets, probing SDs			
2	Sprint Release	5	PO	- PO decides release - Release Manager creates release			
3	Sprint Retrospective	15	SM	- SM reviews the impediments - SM performs roll calls - Everyone answers happiness index - Review of Happiness index and standup emails			
4	Sprint Planning	30	PO	- PO works through product backlog - SD perform planning poker - Introduce sprint goal			
5	NDA signing	1	Everyone	Reminder			
6							
7							
8							
9							
10	Open Points	5	Everybody				
		86					

Team Meeting Agenda "AMOS QAchat"

Date: 2023_05_22

ID	Topic	Time	Author	Description	Result	Decision	Responsible
1	Confluence Data Reading Demonstration	10	Hafidz	short demo of current process for Confluence data extraction			
2	Cloud Hosting Demonstration	10	Jesse	demo of Google Cloud hosting structure			
3	questions from Sebastian	10		Google Docs contain a lot of information			
4	questions to Sebastian	5		In which form should Slack channels be scrapped -> add scrapper to channel manually			
5	comments from Sebastian			https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard https://www.terraform.io/			
6	ToDo			sign NBA			
7							
8							
9							
10							
		35					

Team Meeting Agenda "AMOS QAchat"

Date: 2023_05_31

ID	Topic	Time	Author	Description	Result	Decision	Responsible
1	Sprint Review	30	PO	- Release Manager creates release candidate build - PO walks through "awaiting review" tickets, probing SDs			
2	Sprint Release	5	PO	- PO decides release - Release Manager creates release and mid project tag			
3	Sprint Retrospective	15	SM	- SM reviews the impediments - SM performs roll calls - Everyone answers happiness index - Review of Happiness index and standup emails			
4	Sprint Planning	30	PO	- PO works through product backlog - SD perform planning poker			
5	Branch Protection Rule	5	Emanuel				
6							
7							
8							
9							
10	Open Points	5	Everybody				
		90					

Team Meeting Agenda "AMOS QAchat"

Date: 2023_05_24

ID	Topic	Time	Author	Description	Result	Decision	Responsible
1	Sprint Review	30	PO	- Release Manager creates release candidate build - PO walks through "awaiting review" tickets, probing SDs			
2	Sprint Release	5	PO	- PO decides release - Release Manager creates release			
3	Sprint Retrospective	15	SM	- SM reviews the impediments - SM performs roll calls - Everyone answers happiness index - Review of Happiness index and standup emails			
4	Sprint Planning	30	PO	- PO works through product backlog - SD perform planning poker			
5							
6							
7							
8							
9							
10	Open Points	5	Everybody				
		85					

Team Meeting Agenda "AMOS QAchat"

Date: 2023_05_22

ID	Topic	Time	Author	Description	Result	Decision	Responsible
1	demonstration of current state	10					
2	questions from Sebastian	10		TBD			
3	questions to Sebastian	10	Tobi	- About the NDA: Shoud we plan for real data or create dummy data? - Which datatypes should be supported (PDF, Docx, HTML, Confluence, Slack)? - Who will be allowed to add data to the database? - Which interface (CLI, GUI) should be created for data ingestion?	- NDA for real data - focus on Confluence, Slack General channel would be gread - automatic database update with blacklist		
4	questions to Sebastian	20	Team	- should there be a IAM or sth like this for the data when the user ask a question - in which language should the ChatBot answer (always german???)	- language change would be nice to have		
5							
6							
7							
8							
9							
10							
		50					

Team Meeting Agenda "AMOS QAchat"

Date: 2023_05_17

ID	Topic	Time	Author	Description	Result	Decision	Responsible
1	Sprint Review	30	PO	- Release Manager creates release candidate build - PO walks through "awaiting review" tickets, probing SDs			
2	Sprint Release	5	PO	- PO decides release - Release Manager creates release			
3	Sprint Retrospective	15	SM	- SM reviews the impediments - SM performs roll calls - Everyone answers happiness index - Review of Happiness index and standup emails			
4	Sprint Planning	30	PO	- PO works through product backlog - SD perform planning poker			
5	Definition of Done	5	Tobi + Sara	agree on project specific DoD			
6							
7							
8							
9							
10	Open Points	5	Everybody				
		90					

Team Meeting Agenda "AMOS QAchat"

Date: 2023_05_10

ID	Topic	Time	Author	Description	Result	Decision	Responsible
1	Sprint Review	30	PO	- Release Manager creates release candidate build - PO walks through "awaiting review" tickets, probing SDs			
2	Sprint Release	5	PO	- PO decides release - Release Manager creates release			
3	Sprint Retrospective	15	SM	- SM reviews the impediments - SM performs roll calls - Everyone answers happiness index - Review of Happiness index and standup emails			
4	Sprint Planning	30	PO	- PO works through product backlog - SD perform planning poker			
5	Definition of Done	3	Tobi + Sara	agree on project specific DoD			
6	Sprint Goal	2	Tobi + Sara	agree on sprint goal			
7							
8							
9							
10	Open Points	5	Everybody				
		90					

Team Meeting Agenda "AMOS QAchat"

Date: 2023_05_03

ID	Topic	Time	Author	Description	Result	Decision	Responsible
1	Sprint Review	30	PO	- Release Manager creates release candidate build - PO walks through "awaiting review" tickets, probing SDs			
2	Sprint Release	5	PO	- PO decides release - Release Manager creates release			
3	Sprint Retrospective	15	SM	- SM reviews the impediments - SM performs roll calls - Everyone answers happiness index - Review of Happiness index and standup emails			
4	Sprint Planning	30	PO	- PO works through product backlog - SD perform planning poker			
5	Stand up Emails	2	Tobi + Sara	please write your standup emails regularly		first standup email is sent by sunday evening	
6							
7							
8							
9							
10	Open Points	10	Everybody				
		92					

Team Meeting Agenda "AMOS QAchat"

Date: 2023_04_26

ID	Topic	Time	Author	Description	Result	Decision	Responsible
1	Sprint Review	30	PO	- Release Manager creates release candidate build - PO walks through "awaiting review" tickets, probing SDs		Method 2: semantic search a Google Open Source model	
2	Sprint Release	5	PO	- PO decides release - Release Manager creates release		released	
3	Sprint Retrospective	15	SM	- SM reviews the impediments - SM performs roll calls - Everyone answers happiness index - Review of Happiness index and standup emails		see imp board	
4	Sprint Planning	30	PO	- PO works through product backlog - SD perform planning poker			
5	Get to know each other	5	Vukica				
6	Project Setup	5	Tobi	- programming language and coding guidelines - tools - branching and merging		issue is in progress	
7							
8							
9							
10	Open Points	5	Everybody				
		95					

Team Meeting Agenda "AMOS QAchat"

Date: 2023_04_24

ID	Topic	Time	Author	Description	Result	Decision	Responsible
1	getting to know each other	5		introduce new team members			
2	organization	20		- access to Slack - access to Confluence - access to GDrive / Google Cloud	NDA is a problem, work in progress		
3	project topics	20		talk about new advances in LLMs and project requirements	- documents mostly in german - chatbot has no specific language requirement		
4							
5							
6							
7							
8							
9							
10							
		45					

Team Meeting Agenda "AMOS QAchat"

Date: 2023_04_19

ID	Topic	Time	Author	Description	Result	Decision	Responsible
1	Ensure that everybody has access	5	Deliverables	- shared folder with planning documents - Github Repo - Happiness index tool			
2	initialize planning documents	10	Deliverables	insert base data, agree on role assignments			
3	Agree on team contract	10	Deliverables	submit as part of planning documents			
4	Discussion about first project impressions	10	Tobi	What are your first impressions from yesterday? (good, bad, suggestions, concerns, ...) What documents / workspaces do we have (from Sebastian and Prof. Riehle)?			
5	Getting to know each other	10	Tobi	What experiences do you have (regarding topics that might be needed in this project) ? How do you work (Timeslot, Tools, ...) ?			
6	Slack	10	Tobi	Do we want to create our own Slack channel?			
7	Homework	30	Tobi	Go through Homework 1 tasks			
8	Fill in happiness index	5	Deliverables	counts as sprint 0, closes at midnight			
9							
10							
		90					

Team Meeting Agenda "AMOS QAchat"

Date: 2023_04_18

ID	Topic	Time	Author	Description	Result	Decision	Responsible
1	getting to know each other	10		short introduction of each person			
2	project introduction	40		- go through project definition from Sebastian - look at methods Sebastian already collected	Sebastian provides us the presented sildes		
3	organization	20		- discuss access to Slack, Confluence, GDrive and processing resources	in progress, discuss results in next meeting, Sebastian will invite us to the Google Cloud		
4							
5							
6							
7							
8							
9							
10							
		70					

Team Meeting Agenda "AMOS QAchat"

Date: tbd

ID	Topic	Time	Author	Description	Result	Decision	Responsible
1	Sprint Review	30	PO	- Release Manager creates release candidate build - PO walks through "awaiting review" tickets, probing SDs			
2	Sprint Release	5	PO	- PO decides release - Release Manager creates release			
3	Sprint Retrospective	15	SM	- SM reviews the impediments - SM performs roll calls - Everyone answers happiness index - Review of Happiness index and standup emails			
4	Sprint Planning	30	PO	- PO works through product backlog - SD perform planning poker			
5							
6							
7							
8							
9							
10	Open Points	10	Everybody				
		90					

Role	Tasks
Everyone	participate in lecture participate in team meeting write 2 stand up emails
PO	update feature board update planning documents
SD	work on issues update bill of materials
SM	update impediments backlog
Release Manager	ensure that sprint release candidate is tagged

Role	Tasks
PO	create feature board screenshot create planning document PDF
SD	push current work update assigned issues
SM	create impediments backlog screenshot
Release Manager	tag sprint release candidate

Meeting Preparation	ensure product backlog is ready coordinate with Release Manager	
Sprint Review	ask Release Manager to build release candidate walk through "Awaiting review" issues - ask SD to demo item under review - chech fulfillment of acceptance and DoD criteria - move item to feature archive (add label "Real Size = Y") or move issue to product backlog	Product Owner 1
Sprint Release	decide whether release candidate should be released coordinate with Release Manager	
Sprint Retrospective	SM TODOs answer Happiness Index	Scrum Master
Sprint Planning	reprioritize product backlog items start by most important backlog item and ask SDs to estimate the story points, do until SDs have enough work story points = {0, 1, 2, 3, 5, 8 ,13}	Product Owner 2
Meeting After-work	update planning documents update feature board	

Steps to create a Github Issue:

1.	Go here:	Issues · amosproj/amos2023ss03-qachat (github.com)
2.	click "New issue"	
3.	select corret template	
4.	write a title and description that follow the INVEST criteria	Independent, Negotiable, Valuable, Estimatable, Small, Testable
5.	select the correct project	"amos2023ss03-feature-board"
6.	select the correct milestone (optional)	"sprint-{XY}"
7.	add correct labels	"Est. size = X" and issue type
8.	add Assignees (optional)	
9.	click "Submit new issue"	
10.	go here:	amos2023ss03-feature-board (github.com)
11.	move issue to Product Backlog	
12.	open issue and set corresponding priority	

Average Story Points per Sprint:	25
Current Sprint:	8
Available Sprints:	5
Available Story Points:	**125**
Homework	**Rough Estimations**
Sprint 12	20
Sprint 13	10
Story Points available for Product:	**95**
Open Issues	**Rough Estimations**
setup everything in the Cloud and make is usable	15
work with real data (testing, add to database, ...)	20
testing	15
bugfixes	20
documentation	5
beautification	10
buffer task 1	5
buffer task 2	5
Story Point Account at the END:	**0**