Project Name	Building Information Extractor
Online team meeting	https://tu-berlin.zoom.us/j/67365570181?pwd=RXpnY2xleEYvU3JpR3JzSDZFMk01dz09
Online Team Meeting (Backup)	https://discord.gg/X4QMDpgtUR
Production system (if any)	http://prod.amos.b-ci.de/
Test system (if any)	http://test.amos.b-ci.de/
GitHub repository	GitHub - amosproj/amos2024ss04-building-information-enhancer
GitHub feature board	https://github.com/orgs/amosproj/projects/42
GitHub impediments backlog	https://github.com/orgs/amosproj/projects/50
Team T-shirt (white)	https://www.shirtinator.de/s/Qc61I_GoQwObnqsmHY2MpA
Team T-shirt (black)	https://www.shirtinator.de/s/AHGxY1zzT2m-AUhx2lc7Lw
Team T-shirt (black) (women)	https://www.shirtinator.de/s/sjwwt0GtTzGzfjSxn424ig
Additional materials	
Google Drive (notes, files, etc.)	https://drive.google.com/drive/folders/1DAyzaqwj5ID_YVzNBUgNE0JOCAuorzO_?usp=drive_link
Team maling list	oss-amos-proj4@lists.fau.de
Quick links	
Happiness Index Tool	Happiness Index Tool Link (Project specific)
Capabilities Timeline	Capabilities Timeline (by Week)
Capabilities Timeline Explained	Capabilities Timeline Explained
Main AMOS Document	AMOS #22 - Organisation [Public]

Last Name	First Name	GitHub User Name	Email Address
Balitzki	Emil	Corgam	emil.balitzki@gmail.com
Bandel	Nicolas	nicolasbandel	nicolas.bandel@fau.de
Fischer	Erik	battlemech	erik.fischer@campus.tu-berlin.de
Holtmeier	Leon	Superschnizel	l.holtmeier@campus.tu-berlin.de
Nandico	Lucas	Lucas-Nan	lucas.nandico@fau.de
Pfeil	Oliver	op-hub	oli.pfeil@fau.de
Pöhl	Celine	CelineMP	celine.poehl@fau.de
Yakovenko	Tetiana	dancingsushii	tetiana.yakovenko@campus.tu-berlin.de
Khan	Muhammad Ahsan	Ahsankkhan	ahsan.m.khan@fau.de
Dropped Out			
Sivaci	Bartu	-	-

#	Meeting Day	Product Owners	Software Developer	Release Manager	Scrum Master	Comment
1	2024-04-17	Pfeil, Oliver & Yakovenko, Tetiana	Emil, Nicolas, Leon, Muhammad Ahsan, Lucas & Celine	Emil Balitzki	Erik Fischer	
2	2024-04-24	Pfeil, Oliver & Yakovenko, Tetiana	Emil, Nicolas, Leon, Muhammad Ahsan, Lucas & Celine	Emil Balitzki	Erik Fischer	
3	2024-05-01	Pfeil, Oliver & Yakovenko, Tetiana	Emil, Nicolas, Leon, Muhammad Ahsan, Lucas & Celine	Emil Balitzki	Erik Fischer	
4	2024-05-08	Pfeil, Oliver [Notes] & Yakovenko, Tetiana [Orga]	Emil, Nicolas, Leon, Muhammad Ahsan, Lucas & Celine	Emil Balitzki	Erik Fischer	
5	2024-05-15	Pfeil, Oliver [Notes] & Yakovenko, Tetiana [Orga]	Emil, Nicolas, Leon, Muhammad Ahsan, Lucas & Celine	Emil Balitzki	Erik Fischer	
6	2024-05-22	Pfeil, Oliver [Orga] & Yakovenko, Tetiana [Notes]	Emil, Nicolas, Leon, Muhammad Ahsan, Lucas & Celine	Emil Balitzki	Erik Fischer	
7	2024-05-29	Pfeil, Oliver [Orga] & Yakovenko, Tetiana [Notes]	Emil, Nicolas, Leon, Muhammad Ahsan, Lucas & Celine	Emil Balitzki	Erik Fischer	Mid-term due
8	2024-06-05	Pfeil, Oliver [Notes] & Yakovenko, Tetiana [Orga]	Emil, Nicolas, Leon, Muhammad Ahsan, Lucas & Celine	Emil Balitzki	Erik Fischer	
9	2024-06-12	Pfeil, Oliver [Orga] & Yakovenko, Tetiana [Notes]	Emil, Nicolas, Leon, Muhammad Ahsan, Lucas & Celine	Emil Balitzki	Erik Fischer	
10	2024-06-19	Pfeil, Oliver [Notes] & Yakovenko, Tetiana [Orga]	Emil, Nicolas, Leon, Muhammad Ahsan, Lucas & Celine	Emil Balitzki	Erik Fischer	
11	2024-06-26	Pfeil, Oliver [Orga] & Yakovenko, Tetiana [Notes]	Emil, Nicolas, Leon, Muhammad Ahsan, Lucas & Celine	Emil Balitzki	Erik Fischer	
12	2024-07-03	Pfeil, Oliver [Notes] & Yakovenko, Tetiana [Orga]	Emil, Nicolas, Leon, Muhammad Ahsan, Lucas & Celine	Emil Balitzki	Erik Fischer	
13	2024-07-10	Pfeil, Oliver [Notes] & Yakovenko, Tetiana [Orga]	Emil, Nicolas, Leon, Muhammad Ahsan, Lucas & Celine	Emil Balitzki	Erik Fischer	
14	2024-07-17	Pfeil, Oliver [Orga] & Yakovenko, Tetiana [Notes]	Emil, Nicolas, Leon, Muhammad Ahsan, Lucas & Celine	Emil Balitzki	Erik Fischer	Demo day!
15	2024-07-24	Pfeil, Oliver [Notes] & Yakovenko, Tetiana [Orga]	Emil, Nicolas, Leon, Muhammad Ahsan, Lucas & Celine	Emil Balitzki	Erik Fischer	Retrospective
Product owners, software developers, and Scrum Master are set and ideally don't change over time; the critical part is the Release Manager role you need to define here						

Goals	- Collect relevant experiences! - Meet core requirements of the industry partner! - Produce something viable und usable we can be proud of! - Have fun!
Meeting norms	- Be on time. - Notify early if it's not possible to join. - Camera on and feedback will be given "loud" (no quiet "lecture"). - All questions are okay (there are no stupid questions). - Communicate clearly, try to avoid ambiguities. - Don't be rude. - Weekly team meetings are mandatory for each team member.
Working norms	- Stick to your (your co-coder) task, do not interfere with others tasks. - Tell as early as possible when encountering problems. - No late night work expected, focus on consistency. - Asking for help is fine. - Code should be readable and clear. - Code style should be uniform.
Coordination norms	- Roles in the Team Structure should be fixed and only change if really necessary (not randomly). - All team meetings should follow agreed meeting structure and timing. - Task Responsibilities should be assigned clearly for every week with feedback when it is done.
Communication norms	- Communication attempts should be answered within 2 days (eg. "Let's schedule a meeting on x"). - General, weekly comunication via Discord, Critical Communication via Phone (WhatsApp, SMS). - First name basis is default. - When ill, notify as early as possible, other team members should replace missing's person roles for a specific meeting.
Consideration norms	- General approach with problems is to talk directly, then in more general team meetings. If they are still not solvable, they will be escalated to the professor. - Side-conversations are appropiate if they are not necessary for others. General information should be communicated via Discord and/or in the general team meetings. - Disagreements which are not solvable by discussions will be decided by majority vote.
Cont. improvement norms	- Pull requests require review from another person. Keep the main branch clean. - Tracking individual and team progress via boards and weekly sprints, - Feedback should be considered necessary, relevant and as a way to improve for everyone - not as an insult.
Rewards	- Team party at the end of the project. - Small celebrations during online meetings.

Sanctions	- No in-team sanctions, but persistent problems may be escalated to the professor if not solvable.
Signatures	
Scrum Master	Erik Fischer
Product owner	Tetiana Yakovenko
Product owner	Oliver Pfeil
Software developer	Lucas Nandico
Software developer	Emil Balitzki
Software developer	Muhammad Ahsan Khan
Software developer	Nicolas Bandel
Software developer	Celine Pöhl
Software developer	Leon Holtmeier

Product Vision	Project Mission
The BCI Building Information Enhancer is a platform for personal building owners or professionals to access information about a specific address (or region). This information can be used for a variety of applications, from sustainability certifications for buildings over calculating the solar power potential up to aiding in district planning. The BCI building information enhancer offers significant benefits for various stakeholders in the property market.	**The team agreed to create an MVP for the BCI Building Information Enhancer, the core functionality will be displaying data from a fixed number of sources, including satellite images, charging stations and data needed for sustainability certification. Our goal is to build a practical tool that can grow with our users' needs.**

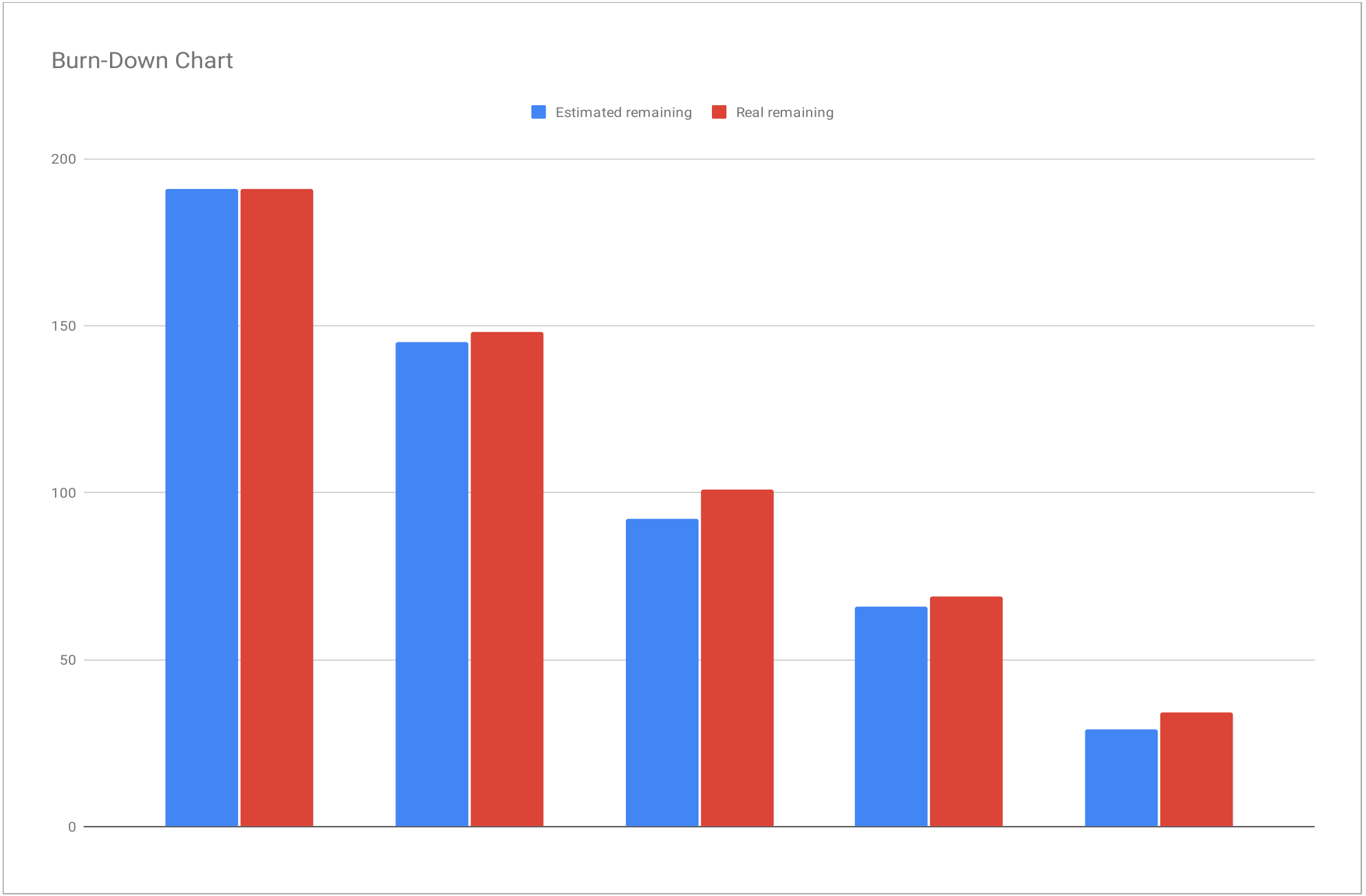
Term	Definition
Data Lake	A centralized repository/database that allows you to store and query all of your structured and unstructured data at any scale. Therefore, the data lake supports a common interface for accessing the heterogenous data.
COPERNICUS	Earth Observation component of the European Union’s space programme, looking at our planet and its environment for the benefit of Europe’s citizens.
ESPON	An EU funded programme that delivers quality expertise to public authorities responsible for designing territorial policies.

Sprint #	Sprint goal
1	Getting familiar with the requirements
2	Setting up the infrastructure and first steps
3	Agreed upon backend infrastructure and ingest one dataset for one UI view
4	Finalising the PoC defined in the previous sprint
5	Getting closer to specific cases: ecological calculator and solar potential of a building
6	Fixing bugs and polishing before mid-project release
7	Getting feasible backend and develop further API endpoints
8	Redesign data lake
9	Improve database performance
10	Mitigation of performace issues and LOD2 data
11	Implement functional use-cases and polish frontend
12	Last dataset and final outro

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release						
Total			191	191	189	189
Sprints						
1	Getting familiar with the requirements		0	191	0	191
2	Setting up the infrastructure and first steps		46	191	43	191
3	Agree upon backend infrastructure and ingest one dataset for one UI view		53	145	47	148
4	Finalising the PoC defined in the previous sprint		26	92	32	101
5	Getting closer to specific cases: ecological calculator and solar potential of a building		37	66	35	69
6	Fixing bugs and polishing before mid-project release		29	29	32	34
Features						
1	**Getting familiar with the requirements**	No features/commits	0		0	
2	**Setting up the infrastructure and first steps**	Request Deutsche Bahn dataset	1		1	
		Ingest Data [1]	3		2	
		Ingest Data [2]	3		3	
		Ingest data [3]	3		3	
		Documentation - BE technology	1		1	
		Research on how should data pipeline work	1		1	
		Create FE Concept	3		5	
		Documentation - CI/CD technology	1		1	
		Get Backend container running	2		2	
		Get FE container running	2		2	
		Initialize Github Wiki	1		1	
		Setup deployment pipeline/branches	3		3	
		Research on FE RestAPI requirements	3		2	
		Documentation - FE technology	1		1	
		Research Github Actions constraints	2		2	
		Setup basic React + NodeJS frontend	2		1	
		Automate workflow with github action	3		3	
		Technology Research (Map APIs)	3		3	
		API project docker file	3		3	
		Create boilerplate API project	5		3	
3	**Agree upon backend infrastructure and ingest one dataset for one UI view**	Create multimap view component	3		3	
		Create basic layout for main UI interface	2		1	
		Create basic data view component	3		3	
		Tag sprint candidate	2		2	
		Simplify .env file	1		0	

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
		Create video recording and documentation about build process	2		1	
		Automate workflow with GitHub Actions	3		3	
		Design Data Pipeline CLI Application	3		3	
		Develop YAML Parser	5		2	
		Develop CSV parser	5		5	
		Dockefile for data pipeline	3		3	
		Configure database connection	3		3	
		Dockerfile for database	3		5	
		Create generic pop-up container	1		1	
		Technology Research (Map APIs)	3		1	
		Create pop-up with favourites	2		2	
		Create map component from OSM	3		3	
		Create 3d view component				
		Row mapping/filtering	3		3	
		Design Data Pipeline CLI Application	3		3	
4	**Finalising the PoC defined in the previous sprint**	Row mapping/filtering	3		3	
		Create 3d view component				
		Implement search by coordinate	3		3	
		FE filtering changes data entries	1		1	
		Compose and finish the UI of the FE	3		5	
		Fix pinning of the tabs going crazy after deleting some tabs	1		1	
		Decide on API endpoints - to have one hour meeting	5		5	
		Add discard_if_empty attribute to yaml.	2		2	
		Crash on special character	1		1	
		Allow building of BE projects with command line	2		3	
		Create endpoint to request datapoints for am area	5		8	
5	**Getting closer to specific cases: ecological calculator and solar potential of a building**	Research and protoypr on Geospatial Database	5		5	
		Implement shapefile data importer for database integration	5		5	
		Implementation of a unified search interface	2		3	
		Irrelevant search results for query "1" in DataView	2		1	
		Make padding displaying correctly	1		1	
		Reset input fields when switching search modes	1		1	
		Persistent input in search popup	1		1	
		Enhance search suggestion relevancy	2		1	

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
		Rework the datasets tab, add metadata for datasets and main menu	2		2	
		Fix connection between FE and BE	5		5	
		Create build proces video and upload to Deliverables folder	2		2	
		Add support for satellite image in map	2		2	
		Allow default values in data description yaml	2		1	
		Data pipelin crashes while used in docker container	5		5	
6	**Fixing bugs and polishing before mid-project release**	Extend API Endpoints for Hausumringe	3		3	
		Create 3d view component				
		Map interaction from search	2		2	
		Change map controll button visibility	1		1	
		Clean Up data view	3		5	
		Trigger Data view only on button press	1		1	
		Display Hausumringe in FE				
		FE boundary for Germany				
		FE centering map on Germany	1		1	
		Improve display of markers for zoomed out maps	3		3	
		Improve Satellite view perfomance and visualization	2		2	
		Add linting and testing to backend	5		5	
		Add option to drop existing table in data pipeline	2		2	
		Extend data pipeline to handle zip files	3		5	
		Extend YAML handling to shapefiles	3		2	


Burn-Down Chart
Estimated remaining
Real remaining
200
150
100
50
0

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining	
Release							
Total			422	422	415	415	
Sprints							
1	Getting familiar with the requirements		0	422	0	422	
2	Setting up the infrastructure and first steps		46	422	43	422	
3	Agree upon backend infrastructure and ingest one dataset for one UI view		53	376	47	379	
4	Finalising the PoC defined in the previous sprint		26	323	32	332	
5	Getting closer to specific cases: ecological calculator and solar potential of a building		37	297	35	300	
6	Fixing bugs and polishing before mid-project release		34	260	32	265	
7	Getting feasible backend and develop further API endpoints		45	226	44	233	
8	Redesign data lake		32	181	34	189	
9	Ingesting further dataset: natural hazards		21	149	21	155	
10	Energy consumption is displayed in the platform		51	128	54	134	
11	Polishing big picture		23	77	20	80	
12	Last dataset and final outro		54	54	53	60	
Features							
1	**Getting familiar with the requirements**	No features/commits	0		0		
2	**Setting up the infrastructure and first steps**	Request Deutsche Bahn dataset	1		1		
		Ingest Data [1]	3		2		
		Ingest Data [2]	3		3		
		Ingest data [3]	3		3		
		Documentation - BE technology	1		1		
		Research on how should data pipeline work	1		1		
		Create FE Concept	3		5		
		Documentation - CI/CD technology	1		1		
		Get Backend container running	2		2		
		Get FE container running	2		2		
		Initialize Github Wiki	1		1		
		Setup deployment pipeline/branches	3		3		
		Research on FE RestAPI requirements	3		2		
		Documentation - FE technology	1		1		
		Research Github Actions constraints	2		2		
		Setup basic React + NodeJS frontend	2		1		
		Automate workflow with github action	3		3		
		Technology Research (Map APIs)	3		3		
		API project docker file	3		3		
		Create boilerplate API project	5		3		
3	**Agree upon backend infrastructure and ingest one dataset for one UI view**	Create multimap view component	3		3		
		Create basic layout for main UI interface	2		1		
		Create basic data view component	3		3		

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining	
		Tag sprint candidate	2		2		
		Simplify .env file	1		0		
		Create video recording and documentation about build process	2		1		
		Automate workflow with GitHub Actions	3		3		
		Design Data Pipeline CLI Application	3		3		
		Develop YAML Parser	5		2		
		Develop CSV parser	5		5		
		Dockefile for data pipeline	3		3		
		Configure database connection	3		3		
		Dockerfile for database	3		5		
		Create generic pop-up container	1		1		
		Technology Research (Map APIs)	3		1		
		Create pop-up with favourites	2		2		
		Create map component from OSM	3		3		
		Create 3d view component					
		Row mapping/filtering	3		3		
		Design Data Pipeline CLI Application	3		3		
4	**Finalising the PoC defined in the previous sprint**	Row mapping/filtering	3		3		
		Create 3d view component					
		Implement search by coordinate	3		3		
		FE filtering changes data entries	1		1		
		Compose and finish the UI of the FE	3		5		
		Fix pinning of the tabs going crazy after deleting some tabs	1		1		
		Decide on API endpoints - to have one hour meeting	5		5		
		Add discard_if_empty attribute to yaml	2		2		
		Crash on special character	1		1		
		Allow building of BE projects with command line	2		3		
		Create endpoint to request datapoints for am area	5		8		
5	**Getting closer to specific cases: ecological calculator and solar potential of a building**	Research and protoypr on Geospatial Database	5		5		
		Implement shapefile data importer for database integration	5		5		
		Implementation of a unified search interface	2		3		
		Irrelevant search results for query "1" in DataView	2		1		
		Make padding displaying correctly	1		1		
		Reset input fields when switching search modes	1		1		
		Persistent input in search popup	1		1		
		Enhance search suggestion relevancy	2		1		
		Rework the datasets tab, add metadata for datasets and main menu	2		2		
		Fix connection between FE and BE	5		5		
		Create build proces video and upload to Deliverables folder	2		2		
		Add support for satellite image in map	2		2		
		Allow default values in data description yaml	2		1		
		Data pipelin crashes while used in docker container	5		5		
6	**Fixing bugs and polishing before mid-project release**	Extend API Endpoints for Hausumringe	3		3		
		Create 3d view component					
		Map interaction from search	2		2		

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining	
		Change map controll button visibility	1		1		
		Clean Up data view	3		5		
		Trigger Data view only on button press	1		1		
		Display Hausumringe in FE	3				
		FE boundary for Germany	2				
		FE centering map on Germany	1		1		
		Improve display of markers for zoomed out maps	3		3		
		Improve Satellite view perfomance and visualization	2		2		
		Add linting and testing to backend	5		5		
		Add option to drop existing table in data pipeline	2		2		
		Extend data pipeline to handle zip files	3		5		
		Extend YAML handling to shapefiles	3		2		
7	**Getting feasible backend and develop further API endpoints**	Implement clustering mechanism and define endpoint for clustered data	5		5		
		Add argument to data pipeline to overwrite if_table_exists option	1		2		
		Data loading by location	2		1		
		Refactor FE for new API endpoints	2		2		
		Extend layer select control	2		3		
		Create 3d view component					
		Implement clustering mechanism and define endpoint for clustered data	5		5		
		Code documentation					
		Code cleanup data pipeline	2		2		
		Fetch Hausumringe API	2		2		
		Storing additional docker images for local deployment	2		0		
		Create script to run datapipeline for each dataset upon deployment	1		2		
		Clean up data pipeline	2		2		
		Display Hausumringe in FE	3		2		
		FE boundary for Germany					
		Extend API Endpoints for Tatsächliche Nutzung (actual use)	3		3		
		Fix the error while building data pipeline	3		1		
		Refactor FE for new API endpoints	2		2		
		Add argument to data pipeline to overwrite if_table_exists option	1		2		
		Ensure pin doesn't switch tabs	1		2		
		FE search and display polygon for specific region	3		2		
		Functionality to hide data view panel	2		3		
		The titles in the data view sections should be bold text	1		1		
8	**Redesign data lake**	Extend API Endpoints for Tatsaechliche Nutzung (actual use)	3		3		
		Create indexes for location column in SQL server	2		2		
		Utilize metadata DB to store metadata	2		3		
		Test environment connection bugs from mid-project review	3		5		
		Implement MongoDB metadata DB	3		3		
		Discussion about: API Composer: merge the viewportData endpoints into a single endpoint, using the datasetID field	2		2		
		Extend the APIs	2		2		

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining	
		Extend the BE docker architecture and CI/CD pipeline	5		3		
		Create missing descriptions for the actual use dataset	1		2		
		Ingest all Hausumringe datasets for whole of bavaria	1		1		
		Decide on a storage strategy for split datasets	2		1		
		Check and update architecture documentation	1		1		
		Different icons in dataset selection dialog	2		3		
		Integrate charging stations as geospatial data	3		3		
		Execute data pipeline on test environment via SSH					
9	**Improve database performance**	Open as map functionality	3		3		
		Research for performance enhancement	3		5		
		Discussion about limiting of ingested data	2		2		
		Insert data about amount of entries in dataset to metadata db	3		3		
		Create API endpoint for single location	3		2		
		Metadata Improvements and small fixes	2		3		
		Yaml files split + DB health checks	3		2		
		Csv importer bug	2		1		
10	**Mitigation of performace issues and LOD2 data**	Combine meta data objects into class library	2		3		
		Create CityGML importer	5		5		
		Extracting additional data from the CityGML file	2		3		
		Comprehensive map marker enhancements	3		3		
		Create 3d view component					
		Implement the getLocationData endpoint on the FE	2		3		
		Improve database indexing logic					
		Extend API loadLocationData endpoints for LoD2 data in getting volume	5		3		
		Discussion needed - Implement area calculation for dataset 'House Footprints'	3		2		
		Ingest air pollution data set	2		3		
		Search bar improvements	2		3		
		Extend layer select control	2		3		
		Display ALKIS-Parzellarkarte					
		Implement an ability to hover on the map icon to get info about a specific marker					
		FE boundary for Germany	2				
		Change API endpoint to not require a district name	3		5		
		Fix data download issue in production envrironment	1		1		
		BE problems when retrieving Hausumringe from database	3		3		
		FE displaying metadata for map in second tab	2		3		
		Storing zoom and marker threshold values for specific datasets as metadata	2		2		
		Integrate peformance research results	5		3		
		Calculate distance to nearest EV charging station	1		2		
		Fix GeoJSON formatting for 'House Footprints'	2		2		
		Correct latitude and longitude order in 'House Footprints'	2		2		
11	**Implement functional use-cases and polish frontend**	Documentation: Metadata pipeline interactions	1		1		
		Improve database indexing logic					

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining	
		Extend API loadLocationData endpoints for LoD2 data in getting volume	5		3		
		Discussion needed - Implement area calculation for dataset 'House Footprints'	3		2		
		Calculation of building volume for each dataset	3		2		
		Ingest air pollution data set	2		3		
		Automate CityGML import process for each data set in the meta4 file	2		2		
		Further extraction of GML data and saving to DB	5		5		
		FE Borders for searched region dont dissappear	1		1		
		FE display address next to Coordinates	1		1		
12	**Energy consumption is displayed in the platform**	Finish Documentation	1		1		
		Comprehensive map marker enhancements	3		0		
		Polygon select not displaying the correct results	1		1		
		Add missing info about citygml dataset	3		2		
		Extend database attributes for "Actual Use"	3		3		
		Reimplemented and polish the 3D view component	5		5		
		Adjust loadLocationData Endpoint json for subsections in dataview	2		2		
		Fix data insertion limitation for small datasets	1		1		
		Update GitHub README.md	1		1		
		Ingest air pollution data set	2		3		
		Append more chemical components to air quality CSV data	1		1		
		Polish air quality data source	1		1		
		Code cleanup	2		2		
		Extend API loadLocationData endpoints for LoD2 data in getting volume	5		3		
		Color Actual use based on value	3		5		
		Add fly to [datapoint] functionality	2		3		
		Polish up the product for the final release	5		5		
		Calculation of building metrics via 3d data	2		1		
		Calculation of building volume for each dataset	3		2		
		Implement an ability to click on the map icon to get info about a specific marker	2		5		
		FE: custom polygon color	1		1		
		FE: Page padding is not uniform	1		1		
		FE: Switching map resets layer and removes polygon	2		1		
		FE: Searchbar area bug	1		1		
		Extend dataview entries	1		2		

#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done
	Component tests have been written and pass	Project builds, deploys, and tests successfully	User interaction tests pass on all major browsers
	All CI tests passed for the pull request	Database update scripts succeed, consistency tests pass	Component test coverage is above 70%
	Code review has been completed and code has been merged into sprint-release branch	Sprint release notes have been written	Design documentation has been updated
	All feature branches have been merged and closed	Change log has been updated	User documentation has been updated
	New feature code has been documented	The value of the product has been improved	

Type	Link / reference	Description
GitHub Wiki	https://github.com/amosproj/amos2024ss04-building-information-enhancer/wiki	System Architecture, Meeting Notes, Software decisions
README .md	https://github.com/amosproj/amos2024ss04-building-information-enhancer/blob/main/README.md	Main redme of the repository, setup and deployment explanation
Google Drive	https://drive.google.com/drive/folders/1DAyzaqwj5ID_YVzNBUgNE0JOCAuorzO_?usp=drive_link	Shared team files associated with the project

#	Context	Name	Version	License	Comment
1	CI/CD	Docker - build scripts	26.0.0	MIT	Docker framework and tools used to build and publish container images
3	CI/CD	Node.js	>= 20.12.2	MIT	Free, open-source, cross-platform JavaScript runtime environment, here used for the npm tool.
2	Frontend	npm:typescript	5.4.5	Apache-2.0	TypeScript is a superset of JavaScript that compiles to clean JavaScript output.
3	Frontend	npm:react	18.3.0	MIT	The library for web and native user interfaces.
4	Frontend	npm:eslint	8.57.0	MIT	Find and fix problems in your JavaScript code.
5	Frontend	npm:react-dom	18.3.0	MIT	The library for web and native user interfaces.
7	Frontend	npm:vite	5.2.10	MIT	Frontend tooling for easier frontend developement
8	Frontend	npm:acorn	8.11.3	MIT	JavaScript-based JavaScript parser
9	Backend	dotnet-docker	6.0	MIT	.NET is a general purpose development platform maintained by Microsoft and the .NET community on GitHub. This also includes all dotnet docker containers used for the Backend.
10	Backend	nuget:MySql.Data	8.0.23	-	Connector/NET is a fully-managed ADO.NET driver for MySQL.
11	Backend	nuget:Swashbuckle.AspNetCore	5.6.3	MIT	Swagger tools for documenting API's built on ASP.NET Core
12	Data processing	nuget:Microsoft.Data.SqlClient	3.0.1	MIT	Microsoft.Data.SqlClient provides database connectivity to SQL Server for .NET applications.
Software bill of materials(generated from Github):		https://drive.google.com/file/d/1J4CUERrZ2o0EwbyKaxO4Be_iZJ0CoUfK/view?usp=sharing			

Last Name	First Name	Value					
Balitzki	**Emil**			#DIV/0!	#DIV/0!		
Bandel	**Nicolas**						
Fischer	**Erik**						
Holtmeier	**Leon**						
Nandico	**Lucas**			0	No size		
Pfeil	**Oliver**			1	Trivial size		
Pöhl	**Celine**			2	Small size		
Khan	**Muhammad Ahsan**			3	Medium size		
Yakovenko	**Tetiana**			5	Large size		
				8	Very large size		
				13	Too large (size)		
How to play planning poker							
1. Everyone type their number into their value field, don't hit return yet							
2. Someone, perhaps a product owner, count down 3.. 2.. 1..							
3. Then, everyone hit return to submit their value							