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 [Notes] & Yakovenko, Tetiana [Orga]	Emil, Nicolas, Leon, Muhammad Ahsan, Lucas & Celine	Emil Balitzki	Erik Fischer	
10	2024-06-19	Pfeil, Oliver [Orga] & Yakovenko, Tetiana [Notes]	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.**

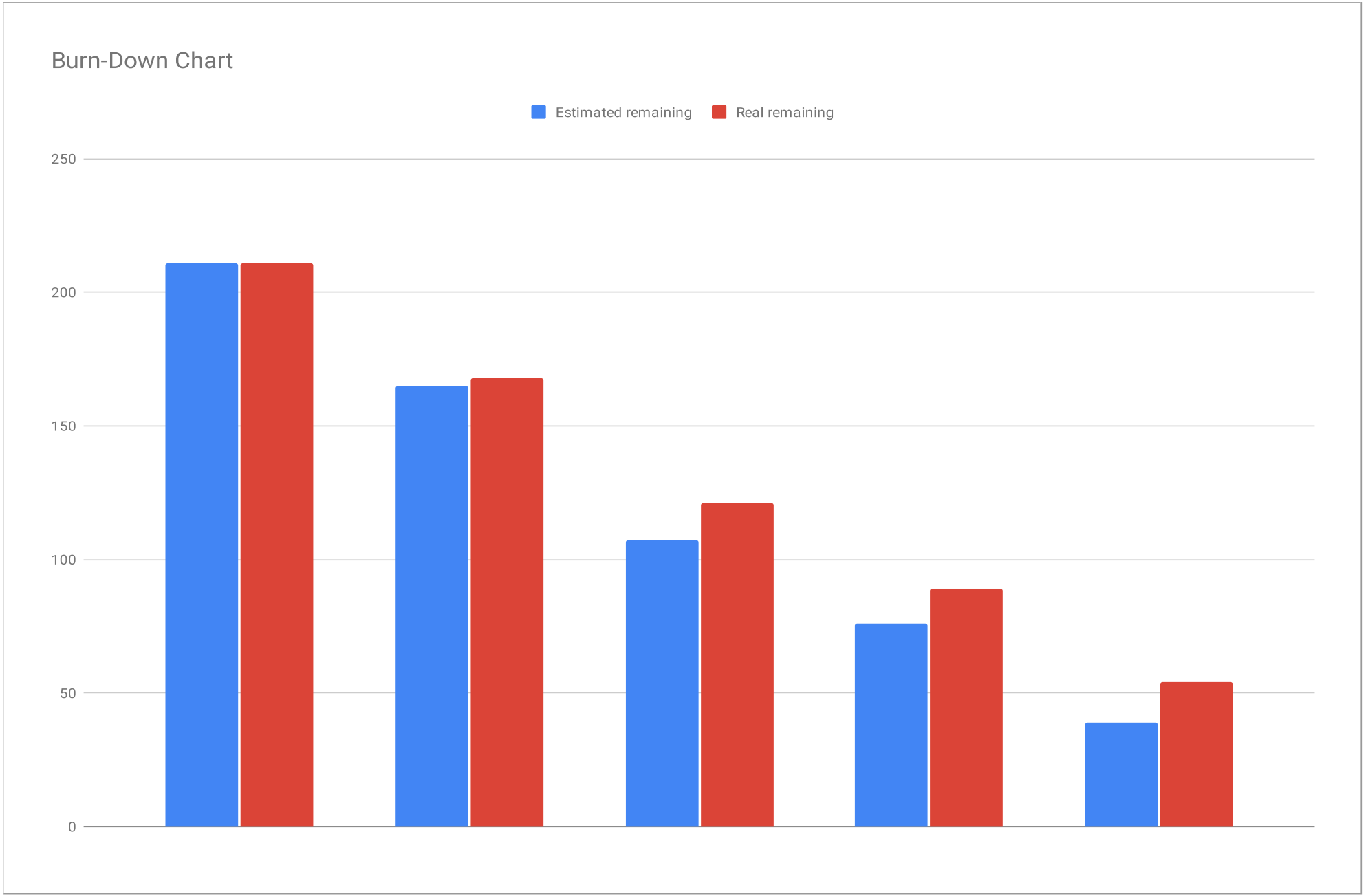
Term	Definition
Data Lake	A centralized repository/database that allows you to store and query all of your structured and unstructured data at any scale. Therefore, the data lake supports a common interface for accessing the heterogenous data.
COPERNICUS	Earth Observation component of the European Union’s 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	
9	
10	
11	
12	
13	
14	
15	

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release						
Total			211	211	157	157
Sprints						
1	Getting familiar with the requirements		0	211	0	211
2	Setting up the infrastructure and first steps		46	211	43	211
3	Agree upon backend infrastructure and ingest one dataset for one UI view		58	165	47	168
4	Finalising the PoC defined in the previous sprint		31	107	32	121
5	Getting closer to specific cases: ecological calculator and solar potential of a building		37	76	35	89
6	Fixing bugs and polishing before mid-project release		39	39	0	54
7						
8						
9						
10						
11						
12						
13						
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	

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
		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	
		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	5			
		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	5			
		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	

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
		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			
		Create 3d view component	5			
		Map interaction from search	2			
		Change map controll button visibility	1			
		Clean Up data view	3			
		Trigger Data view only on button press	1			
		Display Hausumringe in FE	3			
		FE boundary for Germany	2			
		FE centering map on Germany	1			
		Improve display of markers for zoomed out maps	3			
		Improve Satellite view perfomance and visualization	2			
		Add linting and testing to backend	5			
		Add option to drop existing table in data pipeline	2			
		Extend data pipeline to handle zip files	3			
		Extend YAML handling to shapefiles	3			


Burn-Down Chart
Estimated remaining
Real remaining
250
200
150
100
50
0

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release						
Total			211	211	157	157
Sprints						
1	Getting familiar with the requirements		0	211	0	211
2	Setting up the infrastructure and first steps		46	211	157	211
3	Agree upon backend infrastructure and ingest one dataset for one UI view		58	165	0	54
4	Finalising the PoC defined in the previous sprint		31	107	0	54
5	Getting closer to specific cases: ecological calculator and solar potential of a building		37	76	0	54
6	Fixing bugs and polishing before mid-project release		39	39	0	54
7	Getting feasible backend and develop further API endpoints					
8	Backend and frontend work coordinated with 3 datasets					
9	Ingesting further dataset: natural hazards					
10	Energy consumption is displayed in the platform					
11	Polishing big picture					
12	Four datasets are ingested					
13	Last dataset and final outro					
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	

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
		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	
		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	5			
		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	5			
		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	

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
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			
		Create 3d view component	5			
		Map interaction from search	2			
		Change map controll button visibility	1			
		Clean Up data view	3			
		Trigger Data view only on button press	1			
		Display Hausumringe in FE	3			
		FE boundary for Germany	2			
		FE centering map on Germany	1			
		Improve display of markers for zoomed out maps	3			
		Improve Satellite view perfomance and visualization	2			
		Add linting and testing to backend	5			
		Add option to drop existing table in data pipeline	2			
		Extend data pipeline to handle zip files	3			
		Extend YAML handling to shapefiles	3			
7	**Getting feasible backend and develop further API endpoints**	Ensure pin doesnt switch tabs				
		Create 3d view component				
		Code cleanup				
		Code documentation				
		Storing additional docker images for local deployment				
		Extend layer select control				
		Consume Hausumringe API				

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
		Implement clustering mechanism and define endpoint for clustered data				
		Clean up data pipeline				
		Create script to run datapipeline for each dataset upon deployment				
		Create API endpoint for single location				
		Extend API Endpoints for Tatsächliche Nutzung (actual use)				
8	**Backend and frontend work coordinated with 3 datasets**	Integrate third dataset with backend services				
		Develop unified data handling interface in backend				
		Optimize frontend to dynamically display multiple datasets				
		Implement responsive filtering mechanisms for data layers				
		Design and implement cross-dataset analysis tools				
9	**Ingesting further dataset: natural hazards**	Ingest natural hazards data into data lake				
		Develop API endpoints for natural hazards information retrieval				
		Create visualization tools for natural hazards on map view				
		Prepare documentation on handling natural hazard data				
10	**Energy consumption is displayed in the platform**	Develop energy consumption visualization interface				
		Integrate real-time energy data feeds				
		Create comparative analysis tools for energy usage				
		Design energy optimization recommendations system				
		Perform load testing on energy data processing				
		Document energy data sourcing and processing methods				
11	**Polishing big picture**	Refine user interface design for clarity and accessibility				
		Enhance data synchronization across all modules				
		Optimize backend for faster data retrieval				
		Prepare comprehensive end-user documentation				
12	**Four datasets are ingested**	Verify integrity and accuracy of all ingested data				
		Enhance data export and reporting features				
		Optimize data storage and retrieval mechanisms				
13	**Last dataset and final outro**	Ingest final dataset and ensure compatibility				
		Finalize all API integrations and endpoint documentations				
		Conduct final performance tuning across the platform				
		Release the final version of the platform				

#	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/1CPA89OGH_Cr0poRdWhUKLSSHlUuKb8Fm/view?usp=sharing			

Last Name	First Name	Value					
Balitzki	**Emil**	3		**3.00**	**OK**		
Bandel	**Nicolas**						
Fischer	**Erik**						
Holtmeier	**Leon**						
Nandico	**Lucas**	3		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)		