

Project Name	RTDIP Timeseries Forecasting
Online team meeting	<a href="https://fau.zoom-x.de/j/65502405196?pwd=8H73lyixQfqKrnO8Eb47kawnuKCnGp.1">https://fau.zoom-x.de/j/65502405196?pwd=8H73lyixQfqKrnO8Eb47kawnuKCnGp.1</a>
Production system (if any)	
Test system (if any)	
GitHub repository	<a href="https://github.com/amosproj/amos2025ws03-rtdip-timeseries-forecasting">https://github.com/amosproj/amos2025ws03-rtdip-timeseries-forecasting</a>
GitHub feature board	<a href="https://github.com/orgs/amosproj/projects/91/">https://github.com/orgs/amosproj/projects/91/</a>
GitHub imp-squared backlog	<a href="https://github.com/orgs/amosproj/projects/96">https://github.com/orgs/amosproj/projects/96</a>
Team T-shirt (white)	<a href="https://www.shirtinator.de/s/QWcDXTKGS72ISnrRNZq1hg">https://www.shirtinator.de/s/QWcDXTKGS72ISnrRNZq1hg</a>
Team T-shirt (black)	<a href="https://www.shirtinator.de/s/0yx3duRjSzW-5hP32KmZ1w">https://www.shirtinator.de/s/0yx3duRjSzW-5hP32KmZ1w</a>
Additional materials	<a href="https://discord.gg/KJXGmjcs">https://discord.gg/KJXGmjcs</a>
Team mailing list	oss-amos-projX@lists.fau.de
Happines Index App	<a href="https://happy-amos.appspot.com/Courses">https://happy-amos.appspot.com/Courses</a>
Planning Poker	<a href="https://planningpokeronline.com/DMhnF5cOAK9ffF5jsRf2/">https://planningpokeronline.com/DMhnF5cOAK9ffF5jsRf2/</a>

Last Name	First Name	GitHub User Name	Email Address
Böhm	Luca	qw3rat	luca.boehm@fau.de
Meusling	Patrick	SirGankalot	meusling@campus.tu-berlin.de
Arifin	Hafidz	zenzeii	h.arifin@campus.tu-berlin.de
Huy	Christoph	Perimora	christoph.huy@campus.tu-berlin.de
Selbig	Simon	simonselbig	simon.selbig@gmx.de
Haseeb	Abdul	abdulhaseeb-se	abdul.ah.haseeb@fau.de
Pohnke	Hannes	hpfref	hannes.j.pohnke@gmail.com
Jokiel	Rene	BelmontR	rene.jokiel@fau.de
Khabouze	Mehdi	Mehdi-kbz	khabouze@campus.tu-berlin.de

#	Meeting Day	Product Owner		Software Developer	Release Manager	Scrum Master	Comment
		Review	Planning				
1	2025-10-15	Patrick Meusling	Hafidz Arifin	Everyone else	N/A	Luca	
2	2025-10-22	Hafidz Arifin	Patrick Meusling	Everyone else	Christoph	Luca	
3	2025-10-29	Patrick Meusling	Hafidz Arifin	Everyone else	Mehdi Khabouze	Luca	
4	2025-11-05	Hafidz Arifin	Patrick Meusling	Everyone else	Rene Jokiel	Luca	
5	2025-11-12	Patrick Meusling	Hafidz Arifin	Everyone else	Abdul Haseeb	Luca	
6	2025-11-19	Hafidz Arifin	Patrick Meusling	Everyone else	Hannes Pohnke	Luca	
7	2025-11-26	Patrick Meusling	Hafidz Arifin	Everyone else	Simon Selbig	Luca	Mid-term due
8	2025-12-03	Hafidz Arifin	Patrick Meusling	Everyone else	Rene Jokiel	Luca	
9	2025-12-10	Patrick Meusling	Hafidz Arifin	Everyone else	Mehdi Khabouze	Luca	
10	2026-01-07	Hafidz Arifin	Patrick Meusling	Everyone else		Luca	
11	2026-01-14	Patrick Meusling	Hafidz Arifin	Everyone else		Luca	
12	2026-01-21	Hafidz Arifin	Patrick Meusling	Everyone else	Hannes Pohnke	Luca	
13	2026-01-28	Patrick Meusling	Hafidz Arifin	Everyone else	Simon Selbig	Luca	
14	2026-02-04	Hafidz Arifin	Patrick Meusling	Everyone else	Mehdi Khabouze	Luca	
15	2026-02-11	Patrick Meusling	Hafidz Arifin	Everyone else	Christoph	Luca	Demo day!
	2026-02-18	Hafidz Arifin	Patrick Meusling	Everyone else	Hannes Pohnke	Luca	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							

<b>Goals</b>	Aquire new skills
	Produce a functioning and valuable product (a 1.0)
<b>Meeting norms</b>	We respect the opinions of others and we show up on time
	Summary of Partner Meeting required
<b>Working norms</b>	Clean, testable code with clear commit messages, code comments
	We respect other people's work
<b>Coordination norms</b>	Seperate branches for issues, second person review
	We balance workload among the team
<b>Communication norms</b>	Daily check the corresponding channels (discord)
	We communicate constructively, make sure to communicate possible problems (absence, etc.)
<b>Consideration norms</b>	We discuss issues openly
	We vote in case we can't reach a consensus
<b>Cont. improvement norms</b>	We encourage critique and improvement efforts
	Encourage reaching out for help (use the strength/skills of others)
<b>Rewards</b>	We treat ourselves to a sweet of choice for good work
<b>Sanctions</b>	Push Ups (amount decided on the case)
<b>Signatures</b>	
Scrum Master	Luca Böhm
Product owner	Patrick Meusling
Product owner	Hafidz Arifin
Software developer	Abdul Haseeb
Software developer	Rene Jokiel
Software developer	Simon Selbig
Software developer	Hannes Pohnke
Software developer	Christoph Huy
Software developer	Mehdi Khabouze

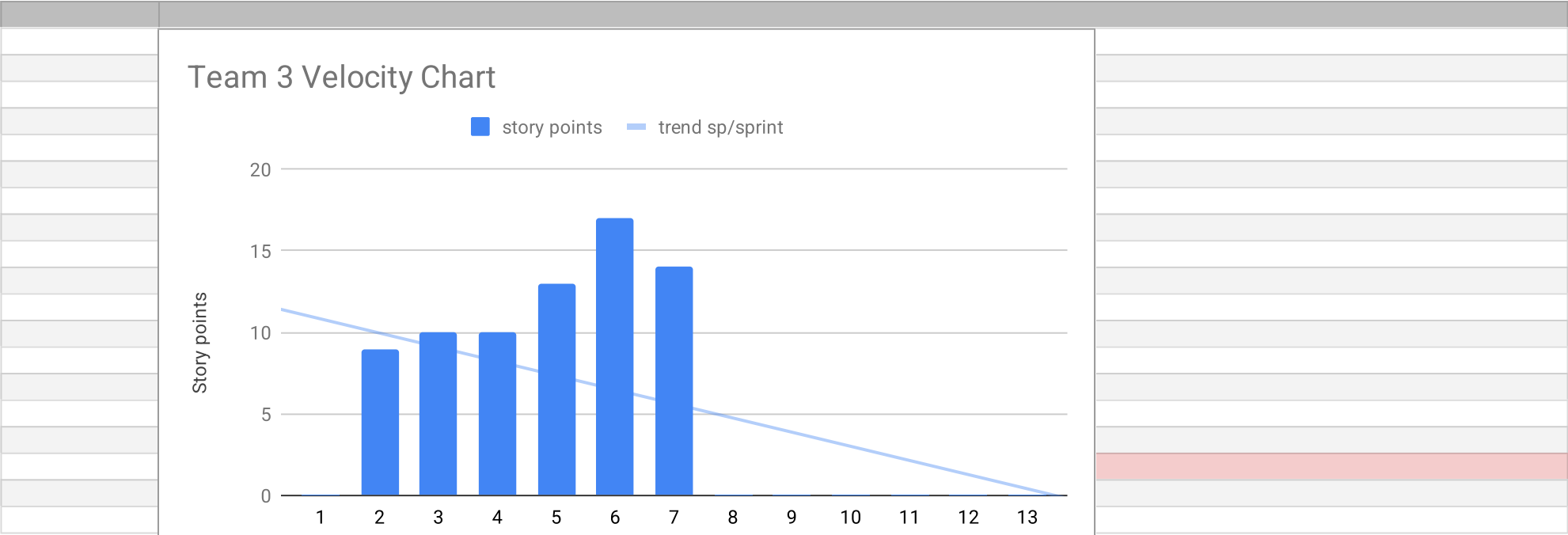
Product Vision	Project Mission
<p>To empower organizations with open-source, scalable, and transparent forecasting capabilities that enable data-driven decision-making across industries.</p> <p>By extending RTDIP with advanced forecasting and anomaly detection components as well as sector specific datasets, we envision a future where time series insights such as trends, seasonality, and predictive patterns are seamlessly integrated into real-time data pipelines.</p> <p>This will allow businesses to understand historical data, anticipate future behavior, optimize operations, and improve efficiency using accessible, community-driven, and production-ready forecasting tools.</p>	<p>To design, develop, and contribute open-source forecasting components for the RTDIP platform that enable trend analysis, anomaly detection, and predictive modeling on time series data.</p> <p>Our mission within this project is to research and implement forecasting techniques using Python and Apache Spark, validate and enrich them with real-world datasets, and ensure they meet RTDIP's modular, tested, and well-documented standards.</p> <p>By doing so, we will enhance RTDIP's functionality and provide the open-source community and industry users such as Shell with reliable, scalable forecasting capabilities that integrate seamlessly into existing data pipelines.</p>

Term	Definition
RTDIP (Real-Time Data Ingestion Platform)	An open-source project under the Linux Foundation designed to simplify the ingestion, transformation, and storage of data from various sources using scalable cloud technologies.
Forecasting	The process of predicting future values or trends based on historical time series data.
Time Series Data	Data collected over time at regular intervals, often used to analyze patterns such as trends and seasonality.
Apache Spark	An open-source distributed computing system that provides fast data processing for large-scale datasets.
PySpark	The Python API for Apache Spark, allowing developers to write Spark applications using Python.
Delta Lake	A storage layer that brings reliability, consistency, and performance to data lakes by supporting ACID transactions and schema enforcement.
Anomaly Detection	The identification of unusual patterns or outliers in data that do not conform to expected behavior.
Predictive Modeling	A statistical or machine learning approach that uses historical data to predict future outcomes.
Time Series Decomposition	The process of breaking down a time series into its core components such as trend, seasonality, and residuals.
ETL (Extract, Transform, Load)	A data pipeline process that extracts data from sources, transforms it into a suitable format, and loads it into a storage system.
Open Source	Software that is freely available to use, modify, and distribute, typically developed collaboratively by a community.
Linux Foundation	A nonprofit organization that supports open-source software projects and fosters collaboration across industries.
EasyCLA (Easy Contributor License Agreement)	A system used by the Linux Foundation to manage contributor license agreements, ensuring legal compliance for open-source contributions.
Modular Architecture	A software design principle that divides a system into separate, interchangeable components that can be developed and maintained independently.
Unit Testing	The practice of testing individual units or components of software to ensure they work as intended.
Documentation	Written descriptions and guides that explain the design, usage, and functionality of a software system.
Data Pipeline	A set of processes that move, transform, and store data from one system to another in a structured and automated way.
Scalability	The ability of a system to handle increasing amounts of work or data by adding resources.
Contribution Guidelines	A set of rules and best practices that contributors must follow to ensure consistency and quality in open-source projects.

Sprint #	Sprint goal
1	None
2	None
3	None
4	Optional
5	Establish a working foundation for the forecasting pipeline
6	Integrate the pipeline into the existing software
7	Advance the pipeline
8	Add more Datasets to the pipeline
9	Testing and Analysing the new Datasets
10	Adding new and different models to the pipeline
11	Testing and Analysing the new Models
12	Finalize the Product
13	Finalize for Demo day
14	Merging the product with the partner, coordinating the merge (they said this might take a bit of time)
15	Backup Sprint

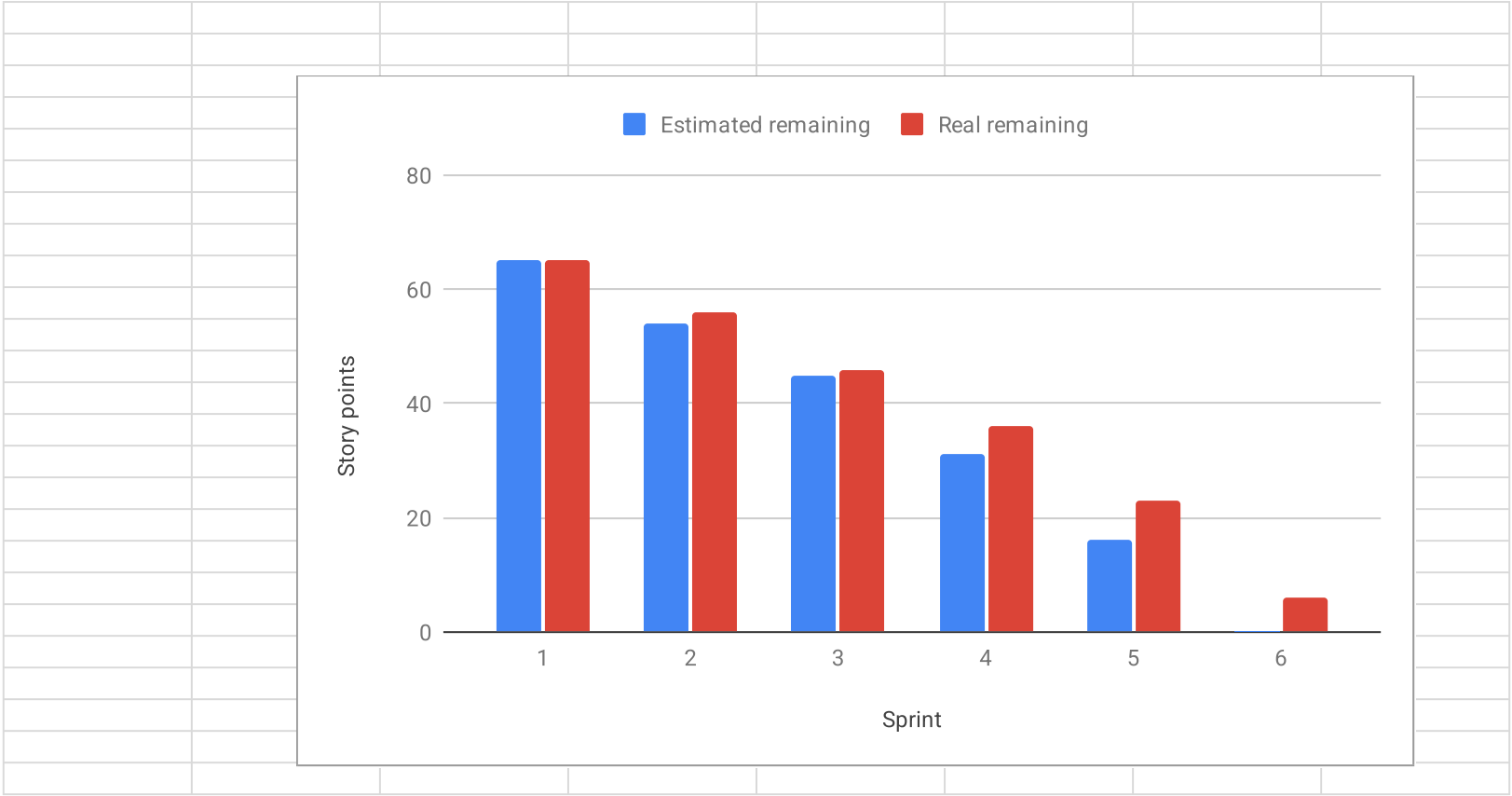
Sprint #	Story Points Realized
1	0
2	9
3	10
4	10
5	13
6	17
7	14
8	0
9	0
10	0
11	0
12	0
13	0
14	0
	PLEASE CREATE THE VELOCITY CHART ON A NEW TAB USING THE DATA FROM THIS TAB





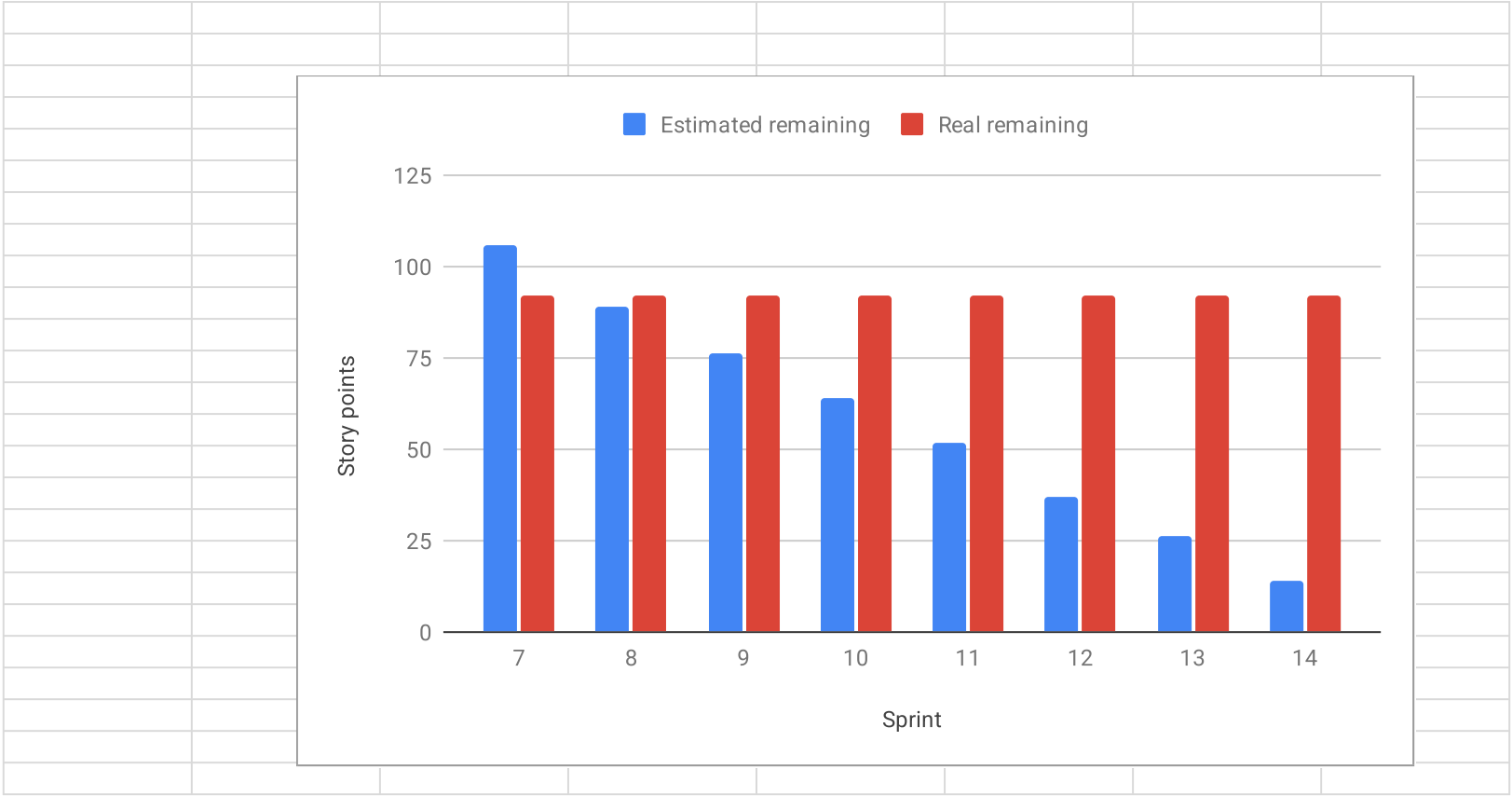
Sprint	Goal	Feature Name	Est. size	Est. remaining	Real size	Real remaining
<b>Release</b>						
<b>Total</b>			65	65		
<b>Sprints</b>						
1	None		0	65	0	65
2	None		11	54	9	56
3	None		9	45	10	46
4	Optional		14	31	10	36
5	Establish a working foundation for the forecasting pipeline		15	16	13	23
6	Integrate the pipeline into the existing software		16	0	17	6
<b>Features</b>						
<b>1</b>	<b>None</b>					
<b>2</b>	<b>None</b>					
		Run the software on your own machine	1		1	
		Understanding the rtdip software	2		2	
		Software Architecture	2		2	
		Find open source time series datasets	2		2	
		Initialize software bill of materials	2		2	
		Research about time series forecasting fundamentals	2		NotCompleted	
<b>3</b>	<b>None</b>					
		Research Possible Models Useful for Shell Dataset	2		2	
		Update our Forked Repository with Latest RTDIP Release	1		1	
		Prepare and Demo Project Build Process	1		2	
		Perform Data Exploration on Shell Dataset	3		3	
		Research about time series forecasting fundamentals	2		2	
<b>4</b>	<b>Optional</b>					
		Getting to Know HPC of AMOS	2		NotCompleted	
		Perform Data Exploration on Opensource Dataset 1	2		3	
		Preprocess Shell Dataset	3		3	
		Implementing a basic Model pipeline	2		2	
		Research Possible Models Useful fo Opensource Dataset	2		NotCompleted	
		CI/CD Pipeline fail bug	3		2	
<b>5</b>	<b>Establish a working foundation for the forecasting pipeline</b>					
		Choosing Models, Datasets, Error Measure Metrics	2		2	
		Create Build Process Video	2		2	
		Model Training and Evaluation with Shell Dataset	3		3	
		Visualization of Results	2		1	
		Preprocess Opensource Dataset 1	3		3	
		PO Homework	3		2	

[illegible]



Sprint	Goal	Feature Name	Est. size	Est. remaining	Real size	Real remaining
<b>Release</b>						
<b>Total</b>			106	106		
<b>Sprints</b>						
7	Advance the pipeline		17	106	14	92
8	Add more Datasets to the pipeline		13	89	0	92
9	Different Methods to Improve Forecast Accuracy		12	76	0	92
10	Customer Feedback and Adjustments		12	64	0	92
11	Improving Forecast Accuracy		15	52	0	92
12	Smaller features for more/different preprocessing etc.		11	37	0	92
13	Demo Day Preparation		12	26	0	92
14	Finish the Project		14	14	0	92
<b>Features</b>						
7	<b>Advance the pipeline</b>					
		Collect and Document Customer Feedback on Project Progress	1		Not completed	
		Adjust and Fix CI/CD Pipeline	3		Not completed	
		Model Optimization and Improvement	3		3	
		Implementation of Time Series Decomposition	3		3	
		Implementation of Anomaly Detection	5		5	
		Model Training and Evaluation of Dataset SCADA	2		3	
8	<b>Add more Datasets to the pipeline</b>					
		Perform Data Exploration additional Opensource Dataset	3			
		Implement RTDIP Ingestion	2			
		Restructure and Refactor Wiki	2			
		Standardize Visualization	3			
		Model Optimization and Improvement for Timeseries Forecasting	3			
9	<b>Different Methods to Improve Forecast Accuracy</b>					
		Perform Data Preprocessing of additional Opensource Dataset	2			
		Integrate Visualization into the Pipeline	2			
		Timeseries Forecasting Improvement by Combining FEs from Best Models	3			
		Timeseries Forecasting Improvement with Exogenous Variables	3			
		Timeseries Forecasting Improvement with Other Models	2			
10	<b>Customer Feedback and Adjustments</b>					
		Based on Customer Feedback add changes to preprocessing	3			
		Based on Customer Feedback add changes to feature engineering	3			
		Based on Customer Feedback add changes to Visualization	2			
		Based on Customer Feedback add suggested models	2			
		Based on Customer Feedback change library versions	2			

Sprint	Goal	Feature Name	Est. size	Est. remaining	Real size	Real remaining
11	Improving Forecast Accuracy					
		Testing our own and new model approaches	3			
		Timeseries Forecasting Improvement by Combining FEs from Best Models	3			
		Timeseries Forecasting Improvement with Exogenous Variables	3			
		Timeseries Forecasting Improvement with Other Models	3			
		Timeseries Forecasting Improvement by Trying Different Versions of Models	3			
12	Finish the Product					
		Adjusting our new approaches based on Customer Feedback	3			
		Final Refactoring	2			
		Testing our own and new preprocessing approaches	3			
		Testing our own and new postprocessing approaches	3			
13	Demo Day Preparation					
		Create demo day slide	3			
		Create demo video	3			
		Finalize user, (technical) design, and build/deploy documentation	3			
		Final Refactoring and Cleaning Up Repository	3			
14	Finish the Project					
		Create Project Report	3			
		Tasks related to Industry Partner	5			
		Merging the code with main repo of partner	3			
		Final Bug Fixes	3			
		PLEASE CREATE THE BURNDOWN CHART ON A NEW TAB USING THE DATA FROM THIS TAB				



#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done
	Acceptance criteria are met.		
	Work are pushed to the Github repository.		
	Create a branch for each backlog items (coding)		
	A pull request is created for each related branch.		
	The work products in the pull requests are reviewed.		
	The corresponding branches are merged and closed.		
	The bill of materials section of the planning documents is updated.		
	Work needs to be documented in the corresponding wiki section		
	For new features unit test have to be written.		
	Update our forked repository with the latest RTDIP release.		
	If the task involves coding, the implementation is integrated into the RTDIP framework and verified to function correctly within it.		
	All unit tests must pass successfully in the continuous integration (CI) pipeline.		
	All assignees must communicate and coordinate with each other to complete the task.		
		Release candidates with a working and meaningful update to the previous sprint is tagged.	
		Previously established features must continue to work.	
			The RTDIP forecasting components can be successfully built and deployed within the RTDIP environment.
			All automated tests and validation checks for the developed components pass without errors.
			The implemented forecasting features have been tested end-to-end and verified through a basic user workflow.
			Developer documentation is complete, clearly describing the architecture, setup, and contribution process for RTDIP.
			User documentation (usage examples, configuration steps, expected outputs) is finalized and up to date.
			The final release has been reviewed and approved by all project team members and the industry partner (Shell).



Type	Link / reference
User Documentation	<a href="https://github.com/amosproj/amos2025ws03-rtdip-timeseries-forecasting/wiki/Documentation#user-documentation">https://github.com/amosproj/amos2025ws03-rtdip-timeseries-forecasting/wiki/Documentation#user-documentation</a>
Design Documentation	<a href="https://github.com/amosproj/amos2025ws03-rtdip-timeseries-forecasting/wiki/Documentation#design-documentation">https://github.com/amosproj/amos2025ws03-rtdip-timeseries-forecasting/wiki/Documentation#design-documentation</a>
Build & Deploy Documentation	<a href="https://github.com/amosproj/amos2025ws03-rtdip-timeseries-forecasting/wiki/Documentation#build-and-deploy-documentation">https://github.com/amosproj/amos2025ws03-rtdip-timeseries-forecasting/wiki/Documentation#build-and-deploy-documentation</a>

System Components					Third-Party Dependencies and Libraries					Tools				
Name	Category	Status	Description		Name	Category	Version	License	Description	Name	Category	Version	Description	
Frontend	UI Component	Active	ReactJS - Frontend with TypeScript		ReactJS	Library	18.2.0	MIT	Open-source JavaScript library for building user interfaces.	VS Code	Development	1.85.0	Open-source code editor with integrated support for Git, Docker, and remote Explorer.	
Backend	API Service	Active	NodeJS - Backend with ExpressJS		ExpressJS	Library	4.18.2	MIT	Minimalist web framework for NodeJS.	Git	Development	2.43.0	Distributed version control system for tracking file changes and managing project history and dependencies.	
Database	Data Storage	Active	MongoDB - Database with MongooseJS		MongooseJS	Library	6.11.1	MIT	Object modeling tool for MongoDB.	Docker	Development	24.0.6	Open-source container management software for running applications in isolated environments.	
Authentication	Security	Active	JWT - Authentication with PassportJS		PassportJS	Library	0.6.0	MIT	Authentication framework for NodeJS.	Postman	Development	10.18.3	Open-source API client for testing and documenting APIs.	
Deployment	Infrastructure	Active	AWS - Cloud Services		AWS SDK	Library	2.1245.0	Apache License 2.0	Amazon Web Services SDK for JavaScript.	Webpack	Development	5.75.0	Module bundler for JavaScript applications.	
Monitoring	Observability	Active	Prometheus - Monitoring with Grafana		Prometheus	Library	2.47.0	Apache License 2.0	Open-source monitoring system and time series database.	Babel	Development	7.23.3	JavaScript compiler that translates modern JavaScript into older versions.	
Logging	Logging	Active	Winston - Logging with MochaJS		Winston	Library	3.11.0	MIT	Logging library for NodeJS.	ESLint	Development	8.56.0	JavaScript linting tool for identifying and preventing code errors.	
Testing	Testing	Active	Jest - Testing Framework		Jest	Library	29.7.0	MIT	JavaScript testing framework.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
CI/CD	CI/CD	Active	GitHub Actions - CI/CD Pipeline		GitHub Actions	Library	1.154.0	MIT	Open-source CI/CD pipeline as a service.	OpenAPI	Development	3.1.0	Open-source specification for REST APIs.	
Security	Security	Active	OWASP - Security Framework		OWASP	Library	1.0.0	MIT	Open-source security framework.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
Documentation	Documentation	Active	Docusaurus - Documentation Framework		Docusaurus	Library	2.4.0	MIT	Open-source documentation framework.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
Deployment	Deployment	Active	Vercel - Deployment Platform		Vercel	Library	1.0.0	MIT	Open-source deployment platform.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
Monitoring	Monitoring	Active	Datadog - Monitoring Platform		Datadog	Library	1.0.0	MIT	Open-source monitoring platform.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
Logging	Logging	Active	Elasticsearch - Logging Platform		Elasticsearch	Library	8.10.0	Apache License 2.0	Open-source search and analytics engine.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
Testing	Testing	Active	Cypress - Testing Framework		Cypress	Library	12.17.0	MIT	Open-source end-to-end testing framework.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
CI/CD	CI/CD	Active	CircleCI - CI/CD Pipeline		CircleCI	Library	0.147.0	MIT	Open-source CI/CD pipeline.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
Security	Security	Active	Snyk - Security Framework		Snyk	Library	1.1000.0	MIT	Open-source security framework.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
Documentation	Documentation	Active	ReadTheDocs - Documentation Platform		ReadTheDocs	Library	1.0.0	MIT	Open-source documentation platform.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
Deployment	Deployment	Active	Netlify - Deployment Platform		Netlify	Library	1.0.0	MIT	Open-source deployment platform.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
Monitoring	Monitoring	Active	New Relic - Monitoring Platform		New Relic	Library	1.0.0	MIT	Open-source monitoring platform.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
Logging	Logging	Active	Logstash - Logging Platform		Logstash	Library	8.11.0	Apache License 2.0	Open-source logging platform.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
Testing	Testing	Active	Protractor - Testing Framework		Protractor	Library	5.4.2	MIT	Open-source end-to-end testing framework.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
CI/CD	CI/CD	Active	TravisCI - CI/CD Pipeline		TravisCI	Library	1.0.0	MIT	Open-source CI/CD pipeline.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
Security	Security	Active	SonarQube - Security Framework		SonarQube	Library	9.9.0	MIT	Open-source security framework.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
Documentation	Documentation	Active	Sphinx - Documentation Framework		Sphinx	Library	7.2.6	MIT	Open-source documentation framework.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
Deployment	Deployment	Active	Heroku - Deployment Platform		Heroku	Library	1.0.0	MIT	Open-source deployment platform.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
Monitoring	Monitoring	Active	Grafana - Monitoring Platform		Grafana	Library	9.5.0	Apache License 2.0	Open-source monitoring platform.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
Logging	Logging	Active	Kibana - Logging Platform		Kibana	Library	8.11.0	Apache License 2.0	Open-source logging platform.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
Testing	Testing	Active	MochaJS - Testing Framework		MochaJS	Library	9.2.1	MIT	Open-source testing framework.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
CI/CD	CI/CD	Active	Jenkins - CI/CD Pipeline		Jenkins	Library	2.402.1	MIT	Open-source CI/CD pipeline.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	
Security	Security	Active	Checkmar		Checkmar	Library	1.0.0	MIT	Open-source security framework.	Swagger	Development	2.20.13	Open-source framework for describing REST APIs.	

Last Name	First Name	Value		#DIV/	#DIV/		
Böhm	Luca			0!	0!		
Meusling	Patrick						
Arifin	Hafidz						
Huy	Christoph						
Selbig	Simon			0	No size		
Haseeb	Abdul			1	Trivial size		
Pohnke	Hannes			2	Small size		
Jokiel	Rene			3	Medium size		
Khabouze	Mehdi			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							