

Project Name	RTDIP Timeseries Forecasting
Online team meeting	https://fau.zoom-x.de/j/65502405196?pwd=8H73lyixQfqKrnO8Eb47kawnuKChGp.1
Production system (if any)	
Test system (if any)	
GitHub repository	https://github.com/amosproj/amos2025ws03-rtdip-timeseries-forecasting
GitHub feature board	https://github.com/orgs/amosproj/projects/91/
GitHub imp-squared backlog	https://github.com/orgs/amosproj/projects/96
Team T-shirt (white)	https://www.shirtinator.de/s/QWcDXTKGS72ISnrRNZq1hg
Team T-shirt (black)	https://www.shirtinator.de/s/0yx3duRJSzW-5hP32KmZ1w
Additional materials	https://discord.gg/KJXGmjcs
Team mailing list	oss-amos-projX@lists.fau.de
Happines Index App	https://happy-amos.appspot.com/Courses
Planning Poker	https://planningpokeronline.com/DMhnF5cOAk9ffF5jsRf2/

#	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 Scurm Master are set and ideally don't change over time; the critical part is the Release Manager role you need to define here

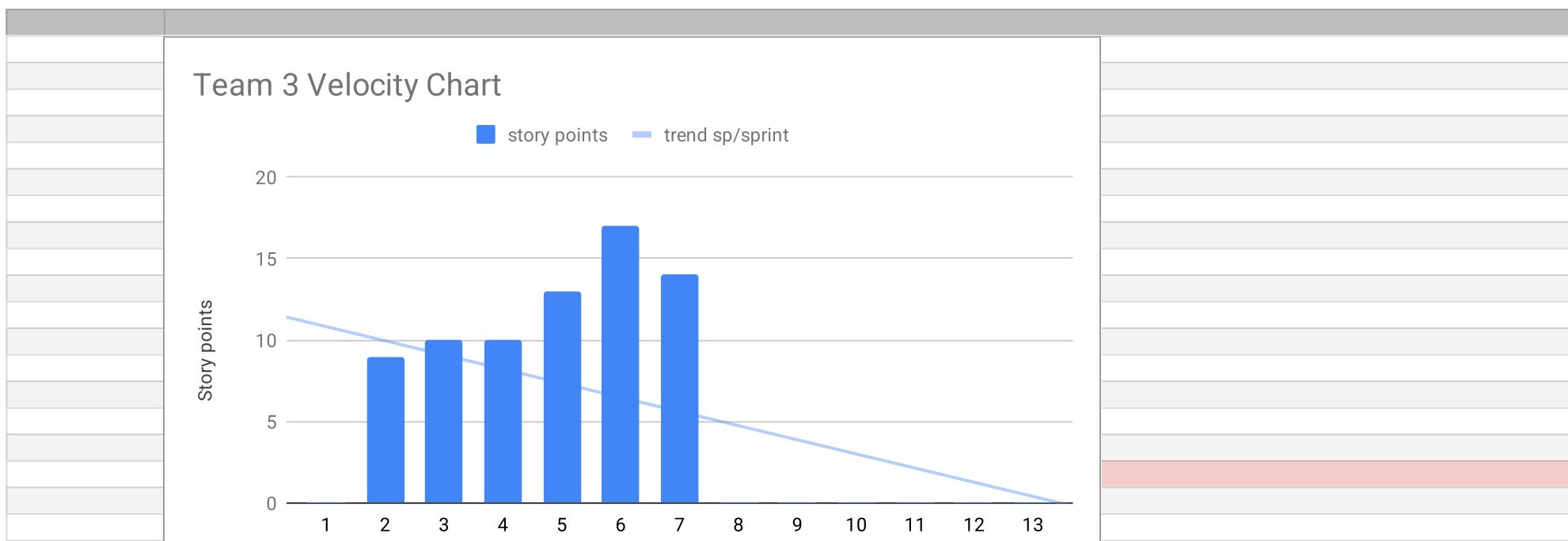
Goals	Aquire new skills Produce a functioning and valuable product (a 1.0)
Meeting norms	We respect the opinions of others and we show up on time Summary of Partner Meeting required
Working norms	Clean, testable code with clear commit messages, code comments We respect other people's work
Coordination norms	Seperate branches for issues, second person review We balance workload among the team
Communication norms	Daily check the corresponding channels (discord) We communicate constructively, make sure to communicate possible problems (absence, etc.)
Consideration norms	We discuss issues openly We vote in case we can't reach a consensus
Cont. improvement norms	We encourage critique and improvement efforts Encourage reaching out for help (use the strength/skills of others)
Rewards	We treat ourselfes to a sweet of choice for good work
Sanctions	Push Ups (amount decided on the case)
Signatures	
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. 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. 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. 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. 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>

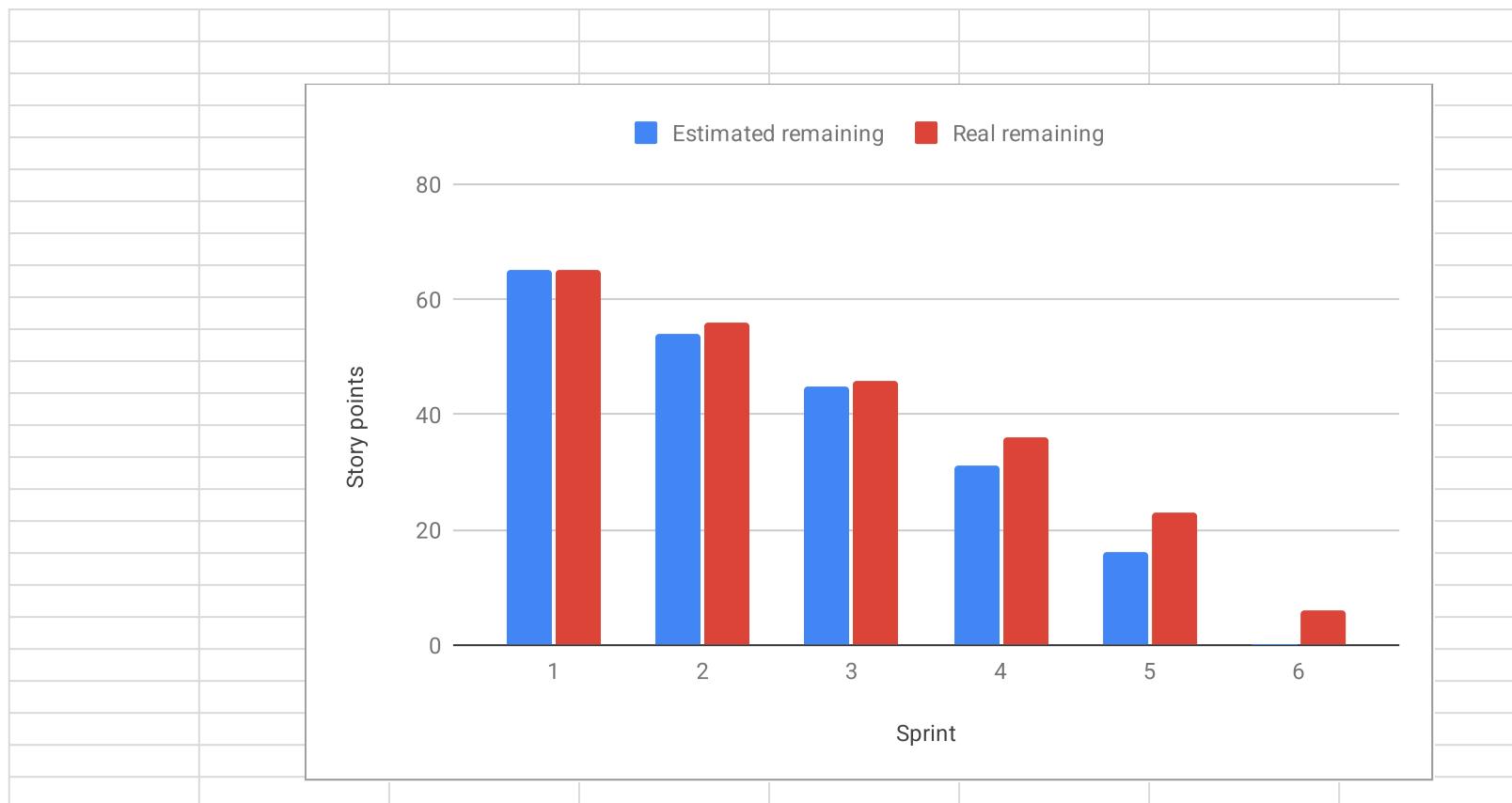
Term	Definition
RTDIP (Real-Time Data Ingestion Platform)	An open-source project under the Linux Foundation designed to simplify the ingestion, transformation, and storage of data from various sources using scalable cloud technologies.
Forecasting	The process of predicting future values or trends based 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	

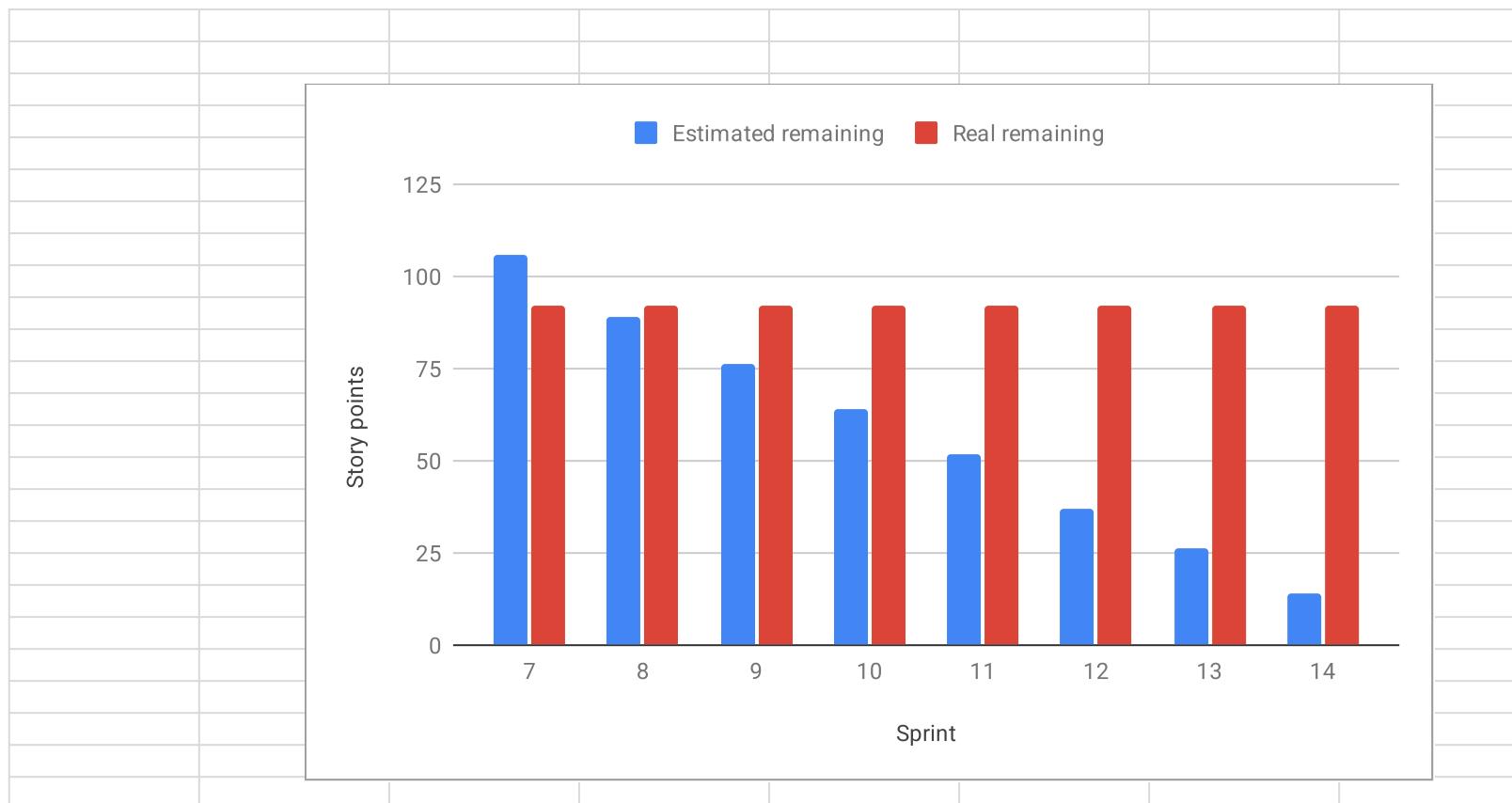


Sprint	Goal	Feature Name	Est. size	Est. remaining	Real size	Real remaining
	Release					
	Total		65	65		
	Sprints					
1	None		0	65	0	65
2	None		11	54	9	56
3	None		9	45	10	46
4	Optional		14	31	10	36
5	Establish a working foundation for the forecasting pipeline		15	16	13	23
6	Integrate the pipeline into the existing software		16	0	17	6
	Features					
1	None					
2	None	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	
3	None	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	
4	Optional	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 Userful fo Opensource Dataset	2		NotCompleted	
		CI/CD Pipeline fail bug	3		2	
5	Establish a working foundation for the forecasting pipeline	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	



Sprint	Goal	Feature Name	Est. size	Est. remaining	Real size	Real remaining
	Release					
	Total		106	106		
	Sprints					
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/diffrnt preprocessing etc.		11	37	0	92
13	Demo Day Preparation		12	26	0	92
14	Finish the Project		14	14	0	92
	Features					
7	Advance the pipeline	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	Add more Datasets to the pipeline	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	Different Methods to Improve Forecast Accuracy	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	Customer Feedback and Adjustments	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).

Project: AMOS - Forecasting and Analytics Services			
Name	Type	Description	Version
AMOS	System	AMOS - Forecasting and Analytics Services	0.3.0-2024
ML Pipeline	System	ML Pipeline	0.1.0
Workers	System	Cloud Workers	Cloud

System Components			
Name	Type	Description	Dependencies
Forecasting	ML Component	ML pipeline that performs forecasting.	Data Pipeline with Forecasting Module.
Transformer	ML Pipeline	ML Pipeline that performs transformation and machine learning.	Data: Forecasting for ML processes.
Data Pipeline	Data Pipeline	ML pipeline that performs quality data management.	ML Pipeline to ensure data quality.
Worker	Data Worker	Cloud workers that perform data processing.	Machine specific of other service data.
Source	Data Source	External data sources.	Machine specific of other service data.
Data Storage	Data Storage	External storage source.	Provides raw time-series data for ingestion and processing.

First Level Dependencies and Runtimes					
Name	Category	Version	Licenses	Vendor	Description
Distro	System	1.8 + 1.12.11	Apache	Apache Software Foundation	Base runtime system used for analysis.
OpenJDK	Runtime	11.0.20	GPL-2.0 + Classpath Exception	OpenJDK Community	Java Virtual Machine required for Apache Spark and ML Pipelines.
Apache Spark (Spark)	Runtime	3.5	Apache-2.0	Apache Software Foundation	Flink-based engine for ETL, data quality and lineage pipelines.
Apache Flink (Flink)	Runtime	1.14.1	Apache-2.0	Apache Software Foundation	Apache Flink is a distributed system for real-time data processing.
HDFS (Hadoop)	Runtime	3.3.1	Apache-2.0	Apache Software Foundation	Apache HDFS is a distributed file system.
Apache Beam	Runtime	2.2.2	Apache-2.0	Apache Software Foundation	Apache Beam provides a programming model for distributed data processing.
Hive	Runtime	3.3.0	MIT	Hortonworks	An SQL-like engine written in Java for high-level data processing.
Prometheus	Runtime	2.16.2	Apache-2.0	Apache Software Foundation	Time-series data access and Prometheus API for monitoring.
TensorFlow	Runtime	2.7.0	Apache-2.0	TensorFlow Dev Team	Machine learning application and processing framework.
Amazon S3	Runtime	1.32	Apache-2.0	Amazon Alexa Developers / Amazon Textract	Amazon S3 is a cloud storage service for storing and retrieving data from anywhere.
Amazon Lambda	Runtime	3.14.7	Apache-2.0	Amazon Lambda Developers	Amazon Lambda is a serverless compute service for running code in response to events.
Amazon Kinesis	Runtime	1.9.4	MIT	Amazon Management Group	Amazon Kinesis is a real-time data streaming service.
Amazon SNS	Runtime	1.0.0	Apache-2.0	Amazon Elasticache - Amazon Kinesis	Amazon SNS is a publish-subscribe messaging service.
Amazon SQS	Runtime	1.7.1	Apache-2.0	Amazon Kinesis	Amazon SQS is a message queuing service.
Amazon VPC	Runtime	2.17.0	Apache-2.0	Amazon VPC	Amazon VPC is a virtual private cloud service.
Amazon S3 - Amazon Lambda - Amazon Kinesis	Runtime	1.27	MIT	Amazon AWS SDK	Amazon S3 - Amazon Lambda - Amazon Kinesis - Amazon VPC.
Amazon Redshift	Runtime	1.4.4	MIT	Amazon Community	Amazon Redshift is a fast and integrated data warehousing service.
Amazon Textract	Runtime	2.5.1 + 2.2.27	MIT - BSD-3	Amazon Software Foundation / Textract	Amazon Textract is a document analysis service.
Amazon Textract - Amazon Rekognition - Amazon Translate	Runtime	1.4 + 2.27	MIT	Amazon Community	Amazon Textract - Amazon Rekognition - Amazon Translate.
Amazon Translate	Runtime	3.1.4	Apache-2.0	Amazon Dev - Amazon Transl - Amazon Dev Team	Amazon Translate is a machine translation service.

Tools			
Name	Category	Version	Description
git	Development - Publish	2.35.0	Git-based version control system used for source code management.
IDEs	Development	IDE-1.0	Integrated development environment used for writing and testing code.
Code Review	Development	Code-1.0	Tool for reviewing and discussing code changes.
Build	Development - Build	Build-2.1.1	Continuous integration and dependency tool.

Last Name	First Name	Value	#DIV/0!	#DIV/0!
Böhm	Luca			
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				