Project Name	Robot Visual Perception
Online team meeting	https://fau.zoom-x.de/j/62836031488
Production system (if any)	Service Provided (later in the Project)
Test system (if any)	Local Webcam
GitHub repository	https://github.com/amosproj/amos2025ws04-robot-visual-perception
GitHub feature board	https://github.com/orgs/amosproj/projects/92
GitHub imp-squared backlog	https://github.com/orgs/amosproj/projects/95
Team T-shirt (white)	https://www.shirtinator.de/s/E_eIf-EIT4iidRVnqM8J5w
Team T-shirt (black)	https://www.shirtinator.de/s/R0CANDdWTRCOvCohq_encA
Additional materials	https://discord.gg/VDqD2Zfb
Team maling list	oss-amos-proj4@lists.fau.de
	https://happy-amos.appspot.com/
	Please use our mailing list for written communication. Only CC teachers (university) and coaches on specific topics, as the mailing lists are very long.

Last Name	First Name	GitHub User Name	Email Address
Hilgers	Felix	fhilgers	felix.hilgers@fau.de
Samdani	Sarib	saribx	saribstudent@gmail.com
Chinbat	Anuun	anuunchin	anuun.ch@gmail.com
Goldschmidt	Georgina	bu31punu	dzsini.lost@gmail.com
Zinn	Benedikt	BenediktZinn	benedikt.wh.zinn@gmail.com
Badura	Emil	Tenebrae311	badura@tu-berlin.de
Assenbaum	Paul	Paul2607	paul.assenbaum@fau.de
Mantsch	Christoph	Christoph-Mantsch	christoph.cm.mantsch@fau.de
Asadi	Zohreh	zohrehasadi00	z.asadi@campus.tu-berlin.de

		Product Owner					
#	Meeting Day	Review	Planning	Software Developer	Release Manager	Scrum Master	Comment
1	2025-10-22	Felix Hilgers	Benedikt Zinn	Everyone else	Paul Assenbaum	Georgina Goldschmidt	(Tuesday 21. Discord for Release process example)
2	2025-10-29	Benedikt Zinn	Felix Hilgers	Everyone else	Anuun Chinbat	Georgina Goldschmidt	
3	2025-11-05	Benedikt Zinn	Felix Hilgers	Everyone else	Sarib Samdani	Georgina Goldschmidt	
4	2025-11-12	Felix Hilgers	Benedikt Zinn	Everyone else	Sarib Samdani	Georgina Goldschmidt	
5	2025-11-19	Felix Hilgers	Benedikt Zinn	Everyone else	Anuun Chinbat	Georgina Goldschmidt	
6	2025-11-26	Benedikt Zinn	Felix Hilgers	Everyone else	Emil Badura	Georgina Goldschmidt	
7	2025-12-03	Felix Hilgers	Benedikt Zinn	Everyone else	Emil Badura	Georgina Goldschmidt	
8	2025-12-10	Benedikt Zinn	Felix Hilgers	Everyone else	Paul Assenbaum	Georgina Goldschmidt	
9	2025-12-17	Felix Hilgers	Benedikt Zinn	Everyone else	Christoph Mantsch	Georgina Goldschmidt	
10	2026-01-07	Benedikt Zinn	Felix Hilgers	Everyone else	Christoph Mantsch	Georgina Goldschmidt	
11	2026-01-14	Felix Hilgers	Benedikt Zinn	Everyone else	Zohreh Asadi	Georgina Goldschmidt	
12	2026-01-21	Benedikt Zinn	Felix Hilgers	Everyone else	Zohreh Asadi	Georgina Goldschmidt	
13	2026-01-28	Felix Hilgers	Benedikt Zinn	Everyone else		Georgina Goldschmidt	
14	2026-02-04	Benedikt Zinn	Felix Hilgers	Everyone else		Georgina Goldschmidt	Demo day!
15	2026-02-11	Felix Hilgers	Benedikt Zinn	Everyone else	Anuun Chinbat	Georgina Goldschmidt	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	Development of a visual perception system for robots with an accuracy of >= 90%
Meeting norms	Zoom-Meetings in the meeting room of the university. Joining the meeting is mandatory, with exceptions (like sickness, etc.). As an exception, showing up to 10 minutes late is fine if the team is informed via discord. Happiness Index: has to be completed at the end of the meeting (amos happy). Stand-up-mails: containing -> things done, problems encountered, things up next. -> For SDs: min 2x per week -> For POs: min 1x per week
Working norms	Issues on Github in the “Feature board” project of the repository. Definition of Done: If specified in the issue that the feature has to be tested, tests have to be written for that feature, documented (in code or later in the wiki) Git-branches: The latest commit on the main-branch has to be tagged. Only 2 “in progress” tickets per person at one time.
Coordination norms	Developers can create issues but not put them onto the “Feature board”. If blocked by a different task, discuss it as soon as possible. Communication should happen via Discord & mail. Criticism should be constructive. Language should be respectful at all times, both verbal and written communication.
Communication norms	Developers can create issues but not put them onto the “Feature board”. If blocked by a different task, discuss it as soon as possible. Communication should happen via Discord & mail. Criticism should be constructive. Language should be respectful at all times, both verbal and written communication.
Consideration norms	
Cont. improvement norms	
Rewards	Online coffee or lunch meet at some point.
Sanctions	Document absence. Repeated absence is addressed with the missing person directly but will later be escalated to professors.
Signatures	
Scrum Master	Georgina Goldschmidt
Product owner	Felix Hilgers

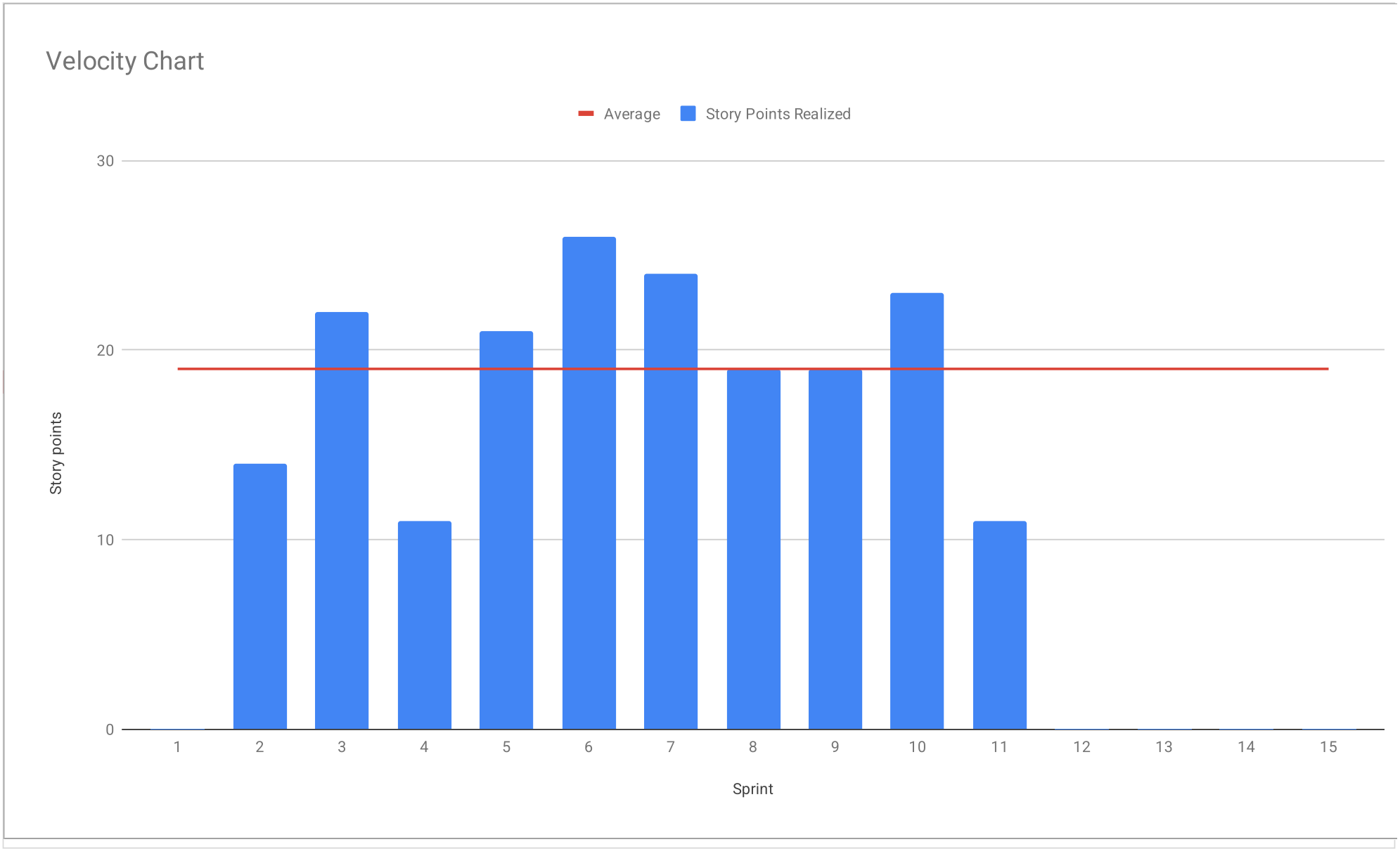
Product owner	Benedikt Zinn
Software developer	Sarib Samdani
Software developer	Emil Badura
Software developer	Anuun Chinbat
Software developer	Paul Assenbaum
Software developer	Christoph Mantsch
Software developer	Zohreh Asadi

Product Vision	Project Mission
Optibot helps people when operating robots equipped with a single camera. It gives real-time distance estimates for known objects inside the field of view of said camera without relying on other sensors. The software stack is easy to integrate with already existing WebRTC based systems, making it easy to enhance existing robot networks.	The mission of this project is to create a containerized system that processes a WebRTC stream as an input and outputs a stream of metadata. This metadata will contain the objects detected in a certain frame of the video stream, their bounding boxes in the image & the estimated distance. This information can be overlayed over the existing video stream in a React component. All components of the project have well definied interfaces and can easily be integrated into existing architectures.

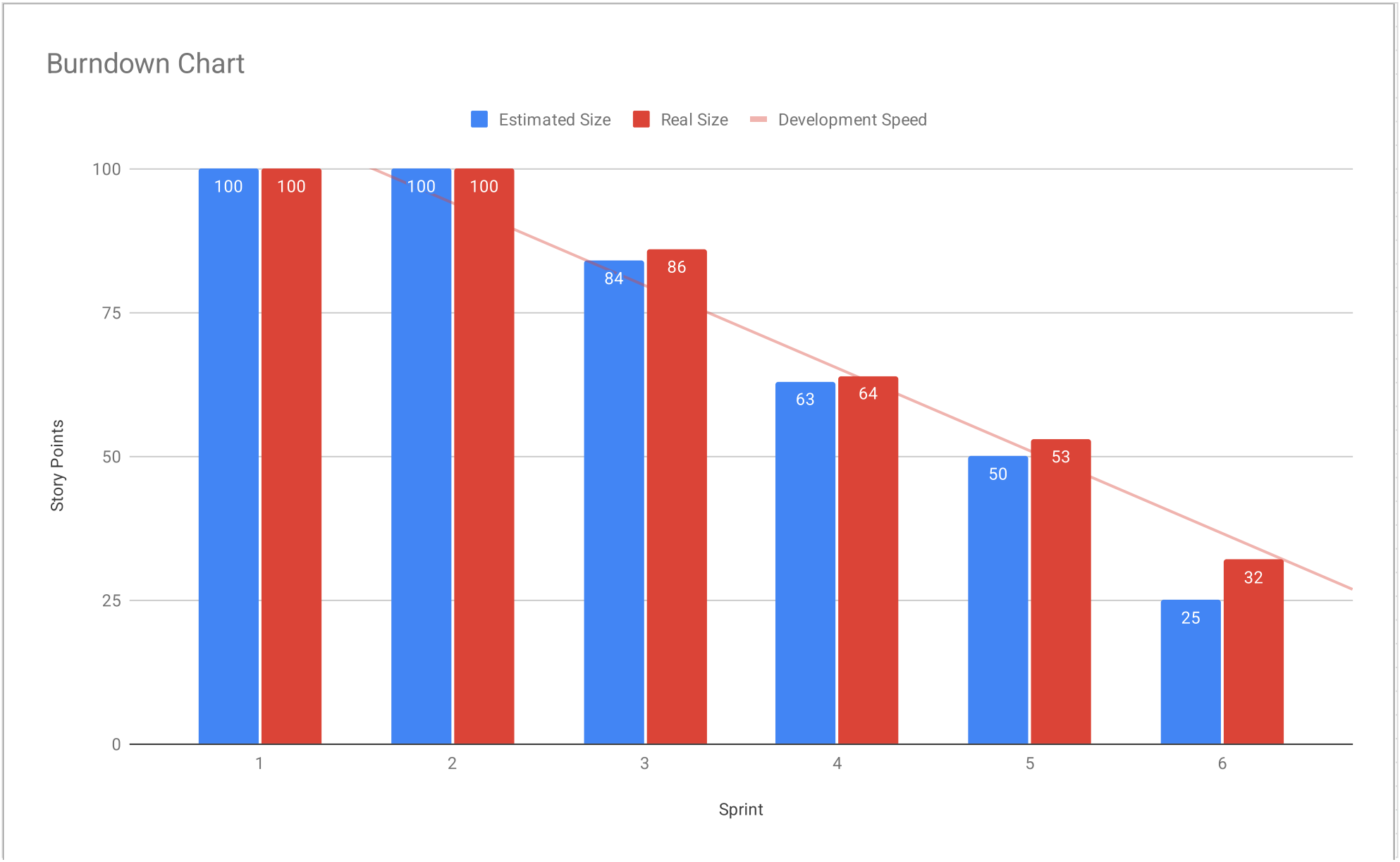
Term	Definition
ONNX	Open Neural Network Exchange (A runtime to run machine learning models on a variety of hardware)
Monocular Depth Estimation	Estimating the Depth in a Scene with only one eye (or camera in our case)
YOLO	You Only Look Once, family of state of the art object detection models (and more)
MiDaS	A ML model for Monocular Depth Estimation
ML	Machine Learning
ORT	ONNX Runtime
CUDA	NVIDIA GPU runtime for machine learning and more
ROCm	AMD GPU runtime for machine learning and more
IoU	Intersection over Union
NMS	Non Maximum Suppression
FPS	Frames per Second
COCO	Common Objects in Context, a commonly used dataset for training and testing Object detection models
WebRTC	Web Real-Time Communication, video streaming between the different services
SBOM	Software Bill of Materials

Sprint #	Sprint goal
1	Meeting with industry partner and getting things started (and the T-shirt of course)
2	Start work on individual components and research unclear topics
3	Combining components into a first working prototype
4	Refining prototype and splitting out components
5	Finishing the Implementation of the Initial Architecture
6	Refine component parts
7	Improve Extensibility and Documentation
8	Refining the overall look and feel
9	Project integration with real SFU
10	Technical Improvements and Experiments
11	Refactoring and Testing
12	Refining the Frontend
13	Performance Improvements
14	Finalizing the Documentation and Preparing for Demo Day
15	Reflecting on the Project

Sprint #	Story Points Realized
1	0
2	14
3	22
4	11
5	21
6	26
7	24
8	19
9	19
10	23
11	11
12	0
13	0
14	0
15	0

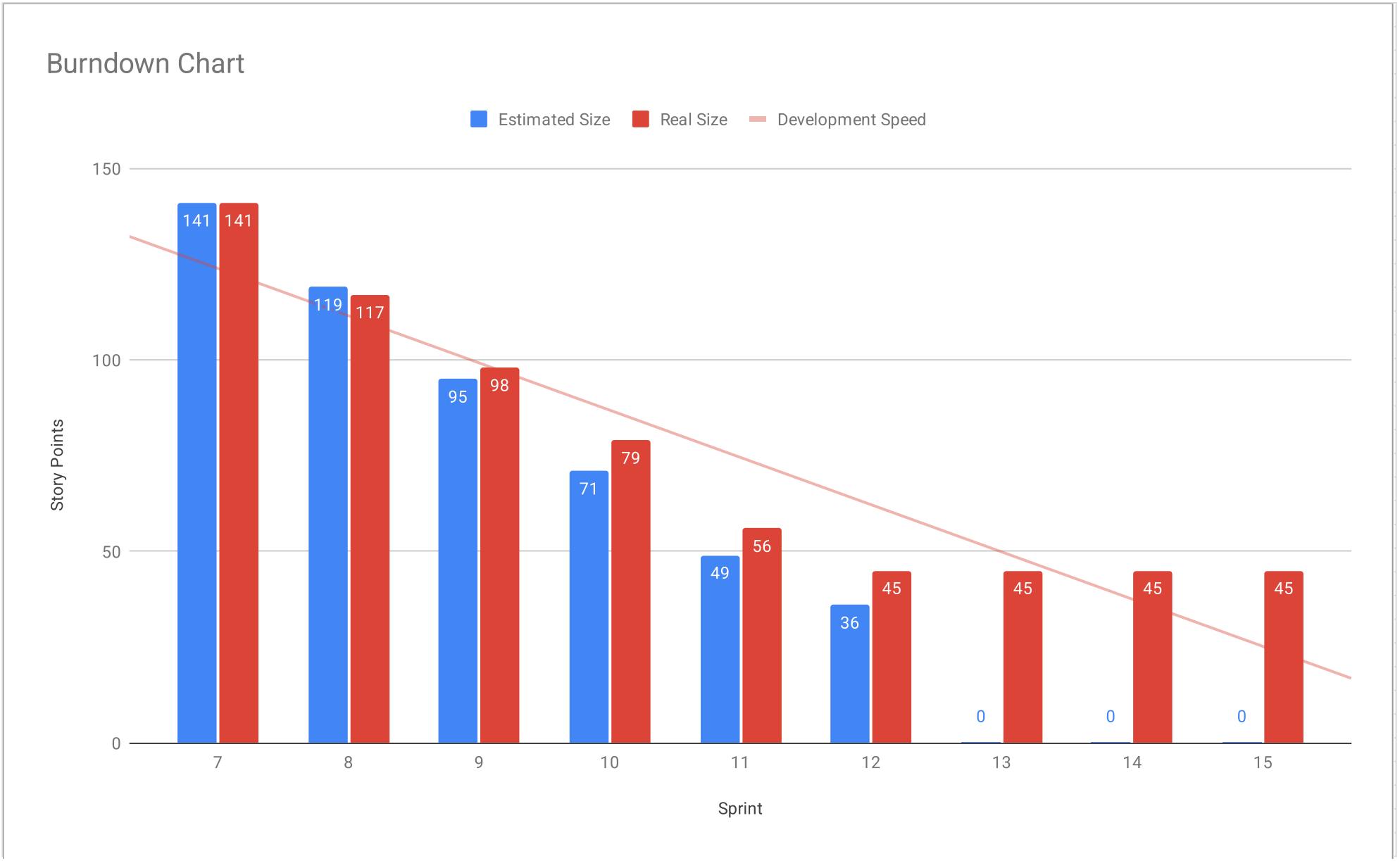

Velocity Chart
Average
Story Points Realized
30
20
10
0
Story points
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Sprint

Sprint	Goal	Feature Name	Est. size	Est. remaining	Real size	Real remaining
Release						
Total			100	100		
Sprints						
1	**Meeting with industry partner and getting things started (and the T-shirt of course)**		0	100	0	100
2	**Start work on individual components and research unclear topics**		16	100	14	100
3	**Combining components into a first working prototype**		21	84	22	86
4	**Refining prototype and splitting out components**		13	63	11	64
5	**Finishing the Implementation of the Initial Architecture**		25	50	21	53
6	**Refine component parts**		25	25	26	32
Features						
1	**Meeting with industry partner and getting things started (and the T-shirt of course)**					
2	**Start work on individual components and research unclear topics**					
		Setup Architecture	5		5	
		Serve Webcam as WebRTC stream	5		5	
		Setup Initial CI Pipeline	3		3	
		Discuss Team Distribution Preferences	3		1	
3	**Combining components into a first working prototype**					
		Keep BOM up to date	2		2	
		Setup REUSE licensing	2		3	
		Update Architecture document	2		2	
		Create Object Detection Example	3		3	
		Prepare Build Process Review	2		2	
		Prepare Depth Estimation	5		5	
		Research Object and Depth Estimation	5		5	
4	**Refining prototype and splitting out components**					
		Metadata Transport	5		3	
		Split Backend Responsibilities	3		3	
		Improve Code Quality	5		5	
5	**Finishing the Implementation of the Initial Architecture**					
		Remove Redundant Calculations	3		3	
		Utilize Library for detecting Object dimensions	5		5	
		Create a React Video Component	3		3	
		Investigate and Fix startup TIme	5		3	
		Fix Windows Development setup	3		3	
		Create a React Video Overlay	3		3	
		Build Process Video	1		1	
		Update Container Images	2		0	
6	**Refine component parts**					
		Initialize Build/Deployment Documentation	3		1	
		Initialize Technical Documentation	3		2	
		Initialize User Facing Documentation	3		3	
		Setup and Test analyzing with Cuda and Rocm	3		3	
		Properly use CSS	2		2	
		Reduce Initial Connection Time	2		2	
		Automate SBOM Generation	3		5	
		Setup Docker Compose	3		3	
		Separate Model Downloading from Running	3		5	


Burndown Chart
Estimated Size
Real Size
Development Speed
Story Points
100
75
50
25
0
100
100
100
100
84
86
63
64
50
53
25
32
1
2
3
4
5
6
Sprint

Sprint	Goal	Feature Name	Est. size	Est. remaining	Real size	Real remaining
Release						
Total			141	141		
Sprints						
7	**Improve Extensibility and Documentation**		22	141	24	141
8	**Refining the overall look and feel**		24	119	19	117
9	**Project integration with real SFU**		24	95	19	98
10	**Technical Improvements and Experiments**		22	71	23	79
11	**Refactoring and Testing**		13	49	11	56
12	**Refining the Frontend**		36	36	0	45
13	**Performance Improvements**		0	0	0	45
14	**Finalizing the Documentation and Preparing for Demo Day**		0	0	0	45
15	**Reflecting on the Project**		0	0	0	45
Features						
7	**Improve Extensibility and Documentation**					
		Extensible Object Analysis	3		2	
		Send Object Location (x, y, z) relative to Camera	3		5	
		Move to pyproject.toml	3		2	
		Reduce Container Size	3		1	
		React Metadata Widget	2		3	
		Reduce Inference Time	3		3	
		Transported Metadata Usage	5		8	
8	**Refining the overall look and feel**					
		Ability to Hide Overlay	1		0	
		Debounce Detections in Metadata Widget	3		3	
		Add Support for Yolo11	2		1	
		Create Make Targets for ONNX Export	1		1	
		Run MiDaS via ONNX	2		2	
		Overlay Flickering When Zoomed-in / -out	3		2	
		Clearing Overlay Unreliable	2		2	
		Overlay Drawn Out-of-bounds	2		1	
		Disconnecting Unreliable	3		3	
		Document Analyzer API Endpoints	2		1	
		Create Object Filter	3		3	
9	**Project integration with real SFU**					
		Smoother Overlays	5		5	
		Add Support for Depth Anything V2	3		2	
		Standalone Model Downloading	2		2	
		Run on real Hardware	2		2	
		Video Stream from File	3		3	
		Backend Logging	3		2	
		Frontend Logging	3		1	
		Low Confidence Filter	3		2	
10	**Technical Improvements and Experiments**					
		Prometheus for Backend	3		3	
		Keep Metadata In Sync	5		8	
		Calibration Guide	2		1	
		Add Detection Threshold	3		3	
		Evalutate Apple Depth Pro	2		2	
		Label Mapping in Backend	3		2	
		Support multiple languages	2		2	
		Dark Light mode	2		2	

Sprint	Goal	Feature Name	Est. size	Est. remaining	Real size	Real remaining
11	**Refactoring and Testing**					
		Improve depth estimation	5		3	
		UI Improvements	3		5	
		Overlay Frame Smoothing	5		3	
12	**Refining the Frontend**					
		Restore Firefox support	2			
		Minor UI improvements	2			
		Top Down Radar View	5			
		Video Router & Dispatching Service	8			
		Centralized Configuration	3			
		Refactor & Test: Drawing Logic of frontend	5			
		Refactor & Test: Math functions	3			
		Add Ion-Sfu Support	3			
		Refactor & Test: Object Tracking	5			
13	**Performance Improvements**					
		Model quntization FP16 / INT8				
		Tensor IO Binding				
		Optimize Preprocessing				
		Optimize Postprocessing				
		Align Model Inputs				
14	**Finalizing the Documentation and Preparing for Demo Day**					
		Finalize User Facing Documentation				
		Finalize Design Documentation				
		Finalize Build Documentation				
		Update Architecture Diagrams				
		Create Demo Day Slide				
		Create Demo Day Video				
15	**Reflecting on the Project**					
		Create Project Report				


Burndown Chart
Estimated Size
Real Size
Development Speed
150
100
50
0
Story Points
141
141
119
117
95
98
71
79
49
56
36
45
0
45
0
45
0
45
7
8
9
10
11
12
13
14
15
Sprint

#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done
	Code has been reviewed by at least one SD	All Software Components are functioning	Architecture documentation has been updated
	Feature has been reviewed by at least one PO	Sprint release notes have been generated from all merge-requests since the last release	Tests exist for all mayor features
	Build scripts have been adapted if necessary	All container communication is working	All tests are passing
	CI Pipeline Passes		
	Code has been merged to main		
	BOM has been updated if necessary		

Type	Link / reference

#	Context	Name	Version	License
1	Frontend (React UI)	npm:react	18.3.1	MIT
2	Frontend (React UI)	npm:react-dom	18.3.1	MIT
3	Backend (FastAPI API)	pypi:aioice	0.10.1	BSD-3-Clause
4	Backend (FastAPI API)	pypi:aiortc	1.14.0	BSD-3-Clause
5	Backend (FastAPI API)	pypi:av	16.0.1	BSD-3-Clause
6	Backend (FastAPI API)	pypi:fastapi	0.115.10	MIT
7	Backend (FastAPI API)	pypi:httpx	0.27.2	BSD License
8	Backend (FastAPI API)	pypi:numpy	1.26.4	BSD-3-Clause
9	Backend (FastAPI API)	pypi:onnxruntime	1.20.1	MIT License
10	Backend (FastAPI API)	pypi:onnxruntime	1.20.1	MIT License
11	Backend (FastAPI API)	pypi:onnxruntime-gpu	1.23.2	MIT License
12	Backend (FastAPI API)	pypi:opencv-python	4.9.0.80	Apache 2.0
13	Backend (FastAPI API)	pypi:opentelemetry-exporter-otlp	1.27.0	Apache-2.0
14	Backend (FastAPI API)	pypi:opentelemetry-sdk	1.27.0	Apache-2.0
15	Backend (FastAPI API)	pypi:opentelemetry-semantic-conventions	0.48	Apache-2.0
16	Backend (FastAPI API)	pypi:prometheus-client	0.21.0	Apache Software License 2.0
17	Backend (FastAPI API)	pypi:pydantic	2.12.3	MIT License
18	Backend (FastAPI API)	pypi:timm	1.0.22	Apache-2.0
19	Backend (FastAPI API)	pypi:transformers	4.49.0	Apache 2.0 License
20	Backend (FastAPI API)	pypi:ultralytics	8.3.58	AGPL-3.0
21	Backend (FastAPI API)	pypi:uvicorn	0.38.0	BSD-3-Clause
22	Backend (FastAPI API)	pypi:websockets	15.0.1	BSD-3-Clause
23	Frontend (React UI)	npm:@types/react	18.3.12	MIT
24	Frontend (React UI)	npm:@types/react-dom	18.3.1	MIT
25	Frontend (React UI)	npm:@typescript-eslint/eslint-plugin	8.16.0	MIT
26	Frontend (React UI)	npm:@typescript-eslint/parser	8.16.0	BSD-2-Clause
27	Frontend (React UI)	npm:@vitejs/plugin-react	4.3.4	MIT
28	Frontend (React UI)	npm:autoprefixer	10.4.22	MIT
29	Frontend (React UI)	npm:eslint	9.17.0	MIT
30	Backend Dev Dependencies	pypi:mypy	1.13.0	MIT
31	Backend Dev Dependencies	pypi:onnx	1.19.1	Apache License v2.0
32	Backend Dev Dependencies	pypi:onnxscript	0.5.6	MIT
33	Backend Dev Dependencies	pypi:onnxslim	0.1.75	MIT
34	Frontend (React UI)	npm:postcss	8.5.6	MIT
35	Frontend (React UI)	npm:prettier	3.4.2	MIT
36	Backend Dev Dependencies	pypi:pytest	8.3.3	MIT
37	Backend Dev Dependencies	pypi:pytest-asyncio	1.2.0	Apache License v2.0
38	Backend Dev Dependencies	pypi:reuse	4.0.3	Apache-2.0 AND CC0-1.0 AND CC-BY-SA-4.0 AND GPL-3.0-or-later
39	Backend Dev Dependencies	pypi:ruff	0.7.0	MIT
40	Frontend (React UI)	npm:tailwindcss	3.4.16	MIT

#	Context	Name	Version	License
41	Frontend (React UI)	npm:typescript	5.6.3	Apache-2.0
42	Frontend (React UI)	npm:vite	6.0.3	MIT
43	Frontend (React UI)	npm:vitest	2.1.5	MIT

Last Name	First Name	Value		#DIV/0!	#DIV/0!		
Samdani	**Sarib**						
Chinbat	**Anuun**						
Badura	**Emil**						
Assenbaum	**Paul**			0	No size		
Mantsch	**Christoph**			1	Trivial size		
Asadi	**Zohreh**			2	Small size		
				3	Medium size		
Hilgers	**Felix**			5	Large size		
Goldschmidt	**Georgina**			8	Very large size		
Zinn	**Benedikt**			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							