

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



		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	2023-01-11	Benedikt Zinn	Felix Hilgers	Everyone else	Christoph Mantsch	Georgina Goldschmidt		
11	2023-01-18	Felix Hilgers	Benedikt Zinn	Everyone else	Zohreh Asadi	Georgina Goldschmidt		
12	2023-01-25	Benedikt Zinn	Felix Hilgers	Everyone else	Zohreh Asadi	Georgina Goldschmidt		
13	2023-02-01	Felix Hilgers	Benedikt Zinn	Everyone else		Georgina Goldschmidt		
14	2023-02-08	Benedikt Zinn	Felix Hilgers	Everyone else		Georgina Goldschmidt	Demo day!	
15	2023-02-15	Felix Hilgers	Benedikt Zinn	Everyone else		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								

<b>Goals</b>	Development of a visual perception system for robots with an accuracy of >= 90%
<b>Meeting norms</b>	<p>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 -&gt; things done, problems encountered, things up next.          -&gt; For SDs: min 2x per week          -&gt; For POs: min 1x per week</p>
<b>Working norms</b>	<p>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.</p>
<b>Coordination norms</b>	<p>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 &amp; mail.          Criticism should be constructive. Language should be respectful at all times, both verbal and written communication.</p>
<b>Communication norms</b>	<p>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 &amp; mail.          Criticism should be constructive. Language should be respectful at all times, both verbal and written communication.</p>
<b>Consideration norms</b>	
<b>Cont. improvement norms</b>	
<b>Rewards</b>	Online coffee or lunch meet at some point.
<b>Sanctions</b>	Document absence. Repeated absence is addressed with the missing person directly but will later be escalated to professors.
<b>Signatures</b>	
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

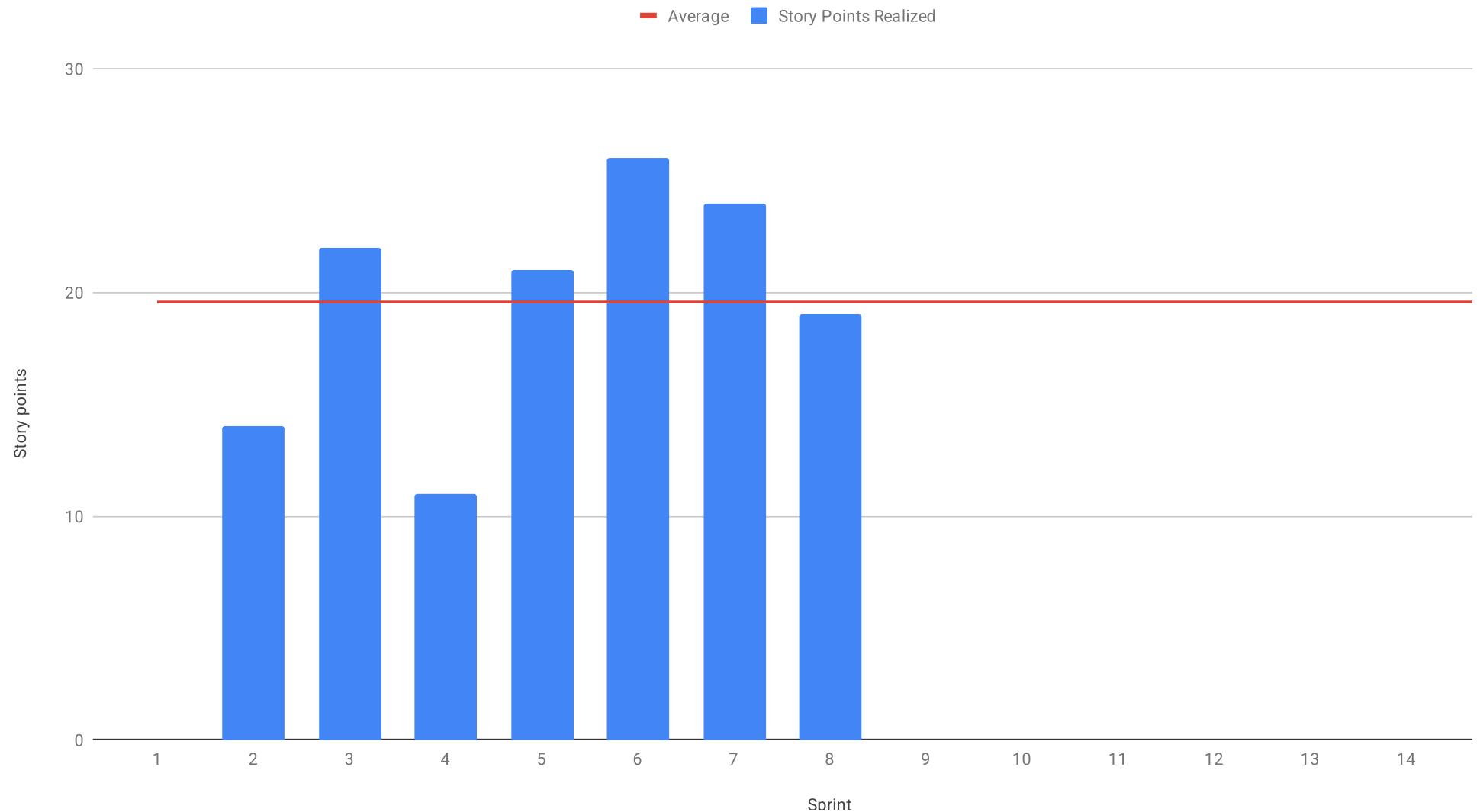
Product Vision	Project Mission
<p>Optibot helps people when operating robots equipped with a single camera. It gives real-time distance estimates for known objects inside the field of view of said camera without relying on other sensors.</p> <p>The software stack is easy to integrate with already existing WebRTC based systems, making it easy to enhance existing robot networks.</p>	<p>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 &amp; the estimated distance.</p> <p>This information can be overlayed over the existing video stream in a React component. All components of the project have well defined interfaces and can easily be integrated into existing architectures.</p>

<b>Term</b>	<b>Definition</b>
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

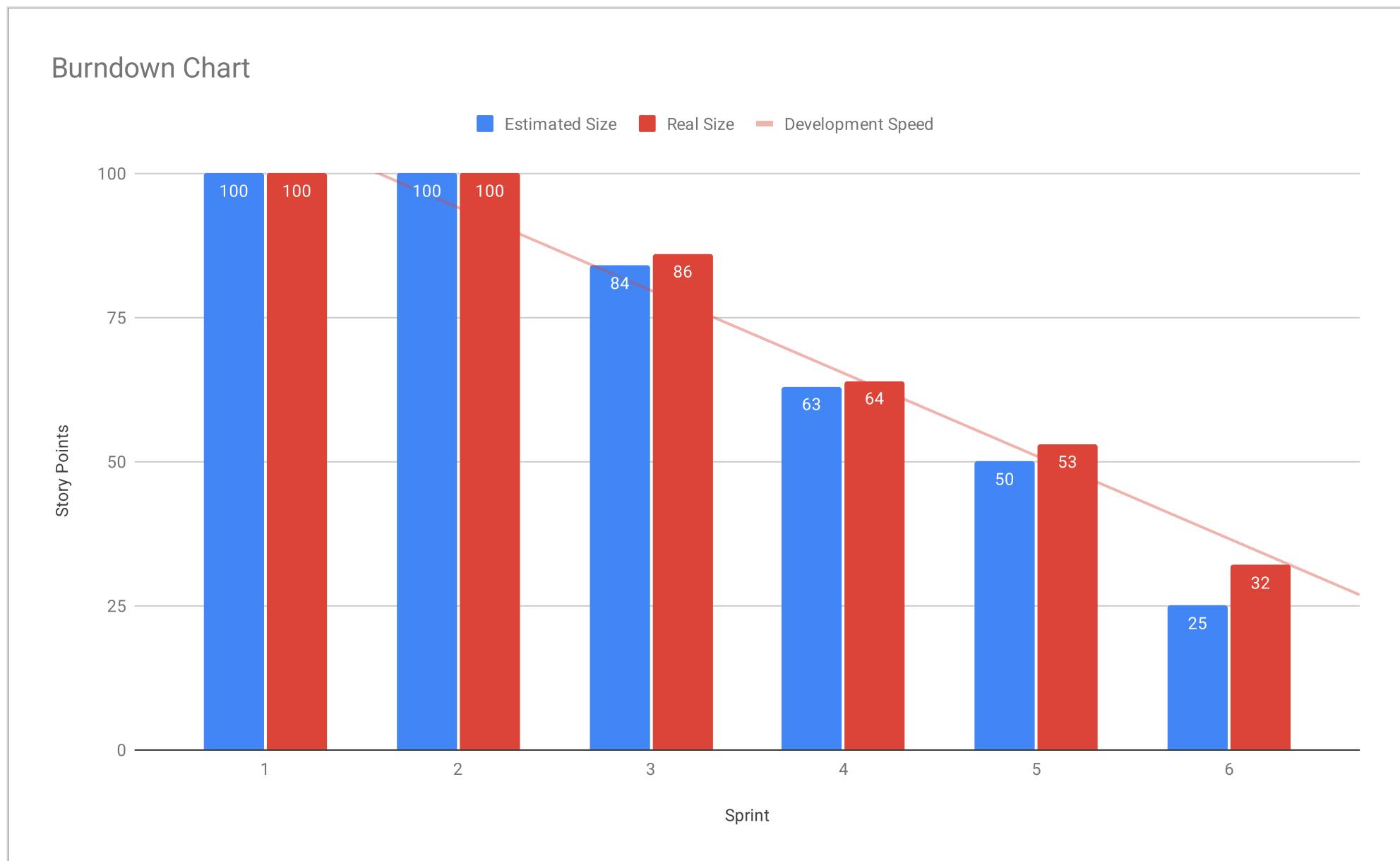
Sprint #	Sprint goal
1	Meeting with industry partner and getting things started (and the T-shirt of course)
2	Start work on individual components and research unclear topics
3	Combining components into a first working prototype
4	Refining prototype and splitting out components
5	Finishing the Implementation 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	0
10	0
11	0
12	0
13	0
14	0
15	0

## Velocity Chart

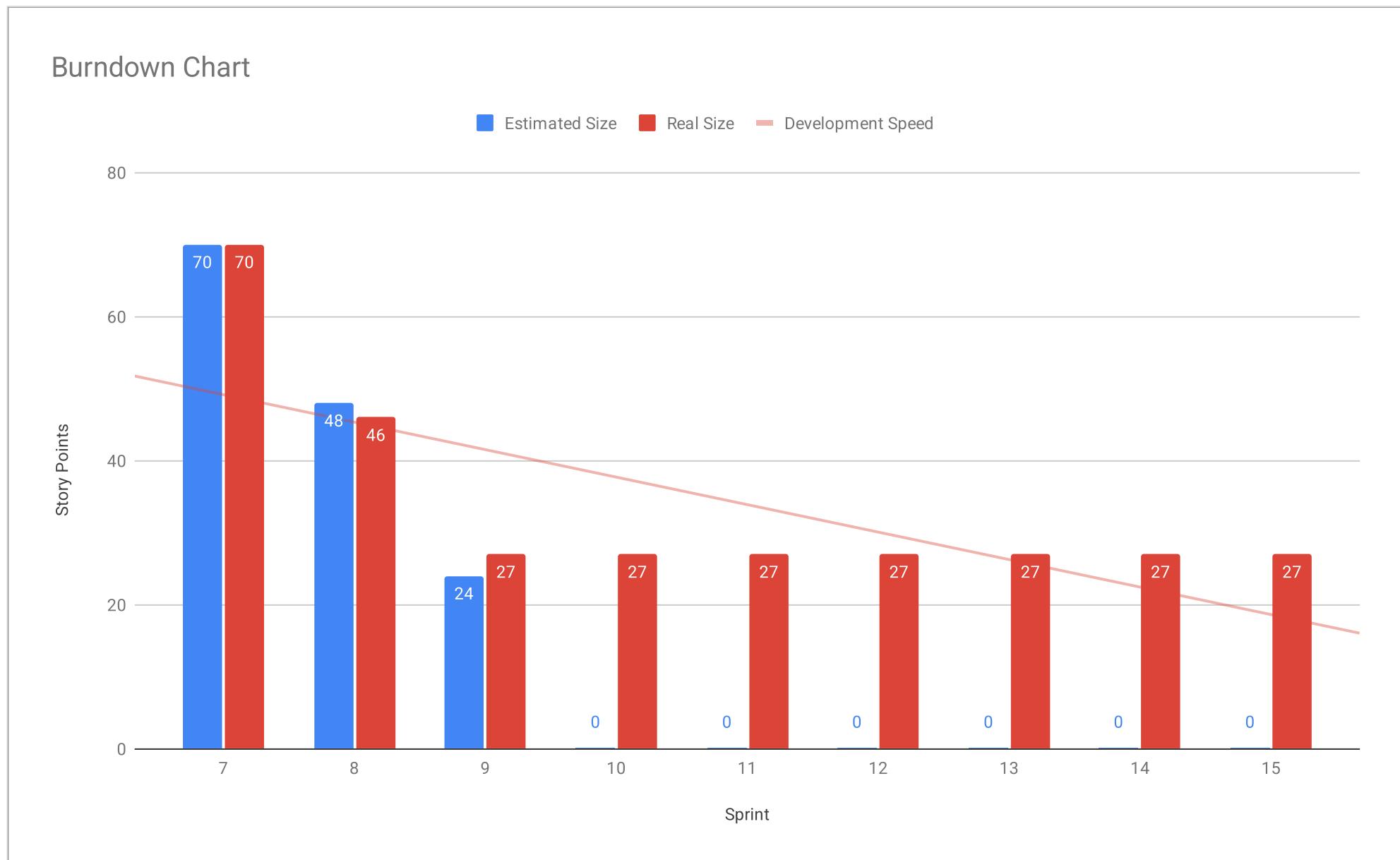


Sprint	Goal	Feature Name	Est. size	Est. remaining	Real size	Real remaining
	<b>Release</b>					
	<b>Total</b>		100	100		
	<b>Sprints</b>					
1	Meeting with industry partner and getting things started (and the T-shirt of course)		0	100	0	100
2	Start work on individual components and research unclear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



Sprint	Goal	Feature Name	Est. size	Est. remaining	Real size	Real remaining
	<b>Release</b>					
Total			70	70		
	<b>Sprints</b>					
7	<b>Improve Extensibility and Documentation</b>		22	70	24	70
8	Refining the overall look and feel		24	48	19	46
9	Project integration with real SFU		24	24	0	27
10	Technical Improvements and Experiments		0	0	0	27
11	Refactoring and Testing		0	0	0	27
12	Refining the Frontend		0	0	0	27
13	Performance Improvements		0	0	0	27
14	Finalizing the Documentation and Preparing for Demo Day		0	0	0	27
15	Reflecting on the Project		0	0	0	27
	<b>Features</b>					
7	<b>Improve Extensibility and Documentation</b>	Extensible Object Analysis Send Object Location (x, y, z) relative to Camera Move to pyproject.toml Reduce Container Size React Metadata Widget Reduce Inference Time Transported Metadata Usage	3 3 3 3 2 3 5		2 5 2 1 3 3 8	
8	<b>Refining the overall look and feel</b>	Ability to Hide Overlay Debounce Detections in Metadata Widget Add Support for Yolo11 Create Make Targets for ONNX Export Run MiDaS via ONNX Overlay Flickering When Zoomed-in / -out Clearing Overlay Unreliable Overlay Drawn Out-of-bounds Disconnecting Unreliable Document Analyzer API Endpoints Create Object Filter	1 3 2 1 2 3 2 2 3 2 3		0 3 1 1 2 2 2 1 3 1 3	
9	<b>Project integration with real SFU</b>	Smoother Overlays Add Support for Depth Anything V2 Standalone Model Downloading Run on real Hardware Video Stream from File Backend Logging Frontend Logging Low Confidence Filter	5 3 2 2 3 3 3 3			
10	<b>Technical Improvements and Experiments</b>	Add Ion-Sfu Support Prometheus for Backend Keep Metadata In Sync Calibration Guide Overlay Frame Smoothing Add Detection Threshold Evaluate Apple Depth Pro Label Mapping in Backend				

Sprint	Goal	Feature Name	Est. size	Est. remaining	Real size	Real remaining
11	Refactoring and Testing	Improve depth estimation UI Improvements Refactor & Test: Object Tracking Refactor & Test: Drawing Logic of frontend Refactor & Test: Math functions				
12	Refining the Frontend	Top Down Radar View Video Router & Dispatching Service Centralized Configuration Dark Light mode Support multiple languages				
13	Performance Improvements	Model quantization 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				







#	Context	Name	Version	License	Comment
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:httptx	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:pydantic	2.12.3	MIT	
14	Backend (FastAPI API)	pypi:timm	1.0.22	Apache-2.0	
15	Backend (FastAPI API)	pypi:ultralytics	8.3.58	AGPL-3.0	
16	Backend (FastAPI API)	pypi:uvicorn	0.38.0	BSD-3-Clause	
17	Backend (FastAPI API)	pypi:websockets	15.0.1	BSD-3-Clause	
18	Frontend (React UI)	npm:@types/react	18.3.12	MIT	
19	Frontend (React UI)	npm:@types/react-dom	18.3.1	MIT	
20	Frontend (React UI)	npm:@typescript-eslint/eslint-plugin	8.16.0	MIT	
21	Frontend (React UI)	npm:@typescript-eslint/parser	8.16.0	BSD-2-Clause	
22	Frontend (React UI)	npm:@vitejs/plugin-react	4.3.4	MIT	
23	Frontend (React UI)	npm:autoprefixer	10.4.22	MIT	
24	Frontend (React UI)	npm:eslint	9.17.0	MIT	
25	Backend Dev Dependencies	pypi:mypy	1.13.0	MIT	
26	Backend Dev Dependencies	pypi:onnx	1.19.1	Apache License v2.0	
27	Backend Dev Dependencies	pypi:onnxscript	0.5.6	MIT	
28	Backend Dev Dependencies	pypi:onnxslim	0.1.75	MIT	
29	Frontend (React UI)	npm:postcss	8.5.6	MIT	
30	Frontend (React UI)	npm:prettier	3.4.2	MIT	
31	Backend Dev Dependencies	pypi:pytest	8.3.3	MIT	
32	Backend Dev Dependencies	pypi:pytest-asyncio	1.2.0	Apache-2.0	
33	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	
34	Backend Dev Dependencies	pypi:ruff	0.7.0	MIT	
35	Frontend (React UI)	npm:tailwindcss	3.4.16	MIT	
36	Frontend (React UI)	npm:typescript	5.6.3	Apache-2.0	
37	Frontend (React UI)	npm:vite	6.0.3	MIT	
38	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
Hilgers	Felix		3	Medium size
Goldschmidt	Georgina		5	Large size
Zinn	Benedikt		8	Very large size
			13	Too large (size)
<b>How to play planning poker</b>				
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				
				no cap yet