Project Name	International Dataspace Station
Online team meeting	https://fau.zoom-x.de/j/64245120479
Production system	https://github.com/amosproj/amos2024ss02-international-dataspace-station
Cloud deployment	https://github.com/projectamoscd/flux
GitHub repository	https://github.com/amosproj/amos2024ss02-international-dataspace-station
GitHub feature board	https://github.com/orgs/amosproj/projects/60
GitHub impediments backlog	https://github.com/orgs/amosproj/projects/59
Team T-shirt (black) (women)	https://www.shirtinator.co.uk/s/3Wt7FEo7RM23NZZT4qwLfw
Team T-shirt (black) (men)	https://www.shirtinator.co.uk/s/3WAORhs4QQ63NQPrUpa0tQ
Additional materials	https://github.com/projectamoscd
Team maling list	oss-amos-proj2@lists.fau.de
AMOS Happy	https://happy-amos.appspot.com/Project?project=5875167674761216&course=6219429234868224

Last Name	First Name	GitHub User Name	Email Address
Zhang	Jin	jinzhangfau	jin.zhang@fau.de
Kurtz	Daniel	daku-de	daniel.kurtz@fau.de
Kanatova	Sezim	skanatova	kanatova.sezim@fau.de
Sanyoto	Matthew Jason	msanyoto	sanyoto@campus.tu-berlin.de
Cosgun	Esra	esracosgun	esra.cosgun@campus.tu-berlin.de
Kröcker	Timo	timoKroecker	t.kroecker@web.de
Wysokinska	Xemena	xenia1w	x.wysokinska@campus.tu-berlin.de
Ivanishcheva	Ekaterina	EkaterinaIvanishcheva	ekaterii39@zedat.fu-berlin.de
~~Hirschpeck~~	~~Leon~~	~~leones18~~	~~leon.hirschpeck@fau.de~~
~~Rameshkumar~~	~~Rathujan~~	~~rathu2712~~	~~rathujan.rameshkumar@fau.de~~
~~Frieß~~	~~Tobias~~	~~Freeze-FF~~	~~tobias.friess@fau.de~~

	Meeting Day	Product Owners	Software Developer	Release Manager	Scrum Master	Comment
1	2024-04-17	Timo Kröcker & Matthew Jason Sanyoto	Everyone else	N/A	Tobias Frieß	
2	2024-04-24	Timo Kröcker & Matthew Jason Sanyoto	Everyone else	Xemena Wysokinska	Jin Zhang	
3	2024-05-01	N/A	N/A	N/A	N/A	
4	2024-05-08	Timo Kröcker & Matthew Jason Sanyoto	Everyone else	Sezim Kanatova	Jin Zhang	
5	2024-05-15	Timo Kröcker & Matthew Jason Sanyoto	Everyone else	Ekaterina Ivanishcheva	Jin Zhang	
6	2024-05-22	Timo Kröcker & Matthew Jason Sanyoto	Everyone else	Daniel Kurtz	Jin Zhang	
7	2024-05-29	Timo Kröcker & Matthew Jason Sanyoto	Everyone else	Esra Cosgun	Jin Zhang	Mid-term due
8	2024-06-05	Timo Kröcker & Matthew Jason Sanyoto	Everyone else	Xemena Wysokinska	Jin Zhang	
9	2024-06-12	Timo Kröcker & Matthew Jason Sanyoto	Everyone else	Sezim Kanatova	Jin Zhang	
10	2024-06-19	Timo Kröcker & Matthew Jason Sanyoto	Everyone else	Ekaterina Ivanishcheva	Jin Zhang	
11	2024-06-26	Timo Kröcker & Matthew Jason Sanyoto	Everyone else	Daniel Kurtz	Jin Zhang	
12	2024-07-03	Timo Kröcker & Matthew Jason Sanyoto	Everyone else	Esra Cosgun	Jin Zhang	
13	2024-07-10	Timo Kröcker & Matthew Jason Sanyoto	Everyone else	Xemena Wysokinska	Jin Zhang	
14	2024-07-17	Timo Kröcker & Matthew Jason Sanyoto	Everyone else	Sezim Kanatova	Jin Zhang	Demo day!
15	2024-07-24	Timo Kröcker & Matthew Jason Sanyoto	Everyone else	Ekaterina Ivanishcheva	Jin Zhang	Retrospective

Goals	Successful Product, High customer satisfication, Good Teamwork
Meeting norms	Be on time on agreed meetings
Working norms	Transparency, Clear Documentation, Openess, Trust
Coordination norms	Tell the POs if something comes up (organisational/meetings)
Communication norms	Check communication channels daily (WhatsApp, Discord)
Consideration norms	POs: are responsible for the order and relevance of the to be implemented features, SDs: are responsible for the implementation and all technical aspects of the project, SM: is responsible for the organization of the SCRUM
Cont. improvement norms	SDs: we will be having a main branch where the working code from the previous sprints will be located. For the current sprint of branch dev will be created. SDs working on features will create new branches from the dev one: at the end of the sprint these branches will be merged into dev again and after reviewing it finally into main
Rewards	-
Sanctions	-
Signatures	
Scrum Master	Jin Zhang
Product owner	Timo Kröcker
Product owner	Matthew Jason Sanyoto
Software developer	Daniel Kurtz
Software developer	Xemena Wysokinska
Software developer	Sezim Kanatova
Software developer	Esra Cosgun
Software developer	Ekaterina Ivanishcheva
~~Scrum Master~~	~~Tobias Frieß~~
~~Software developer~~	~~Leon Hirschpeck~~
~~Software developer~~	~~Rameshkumar Rathujan~~

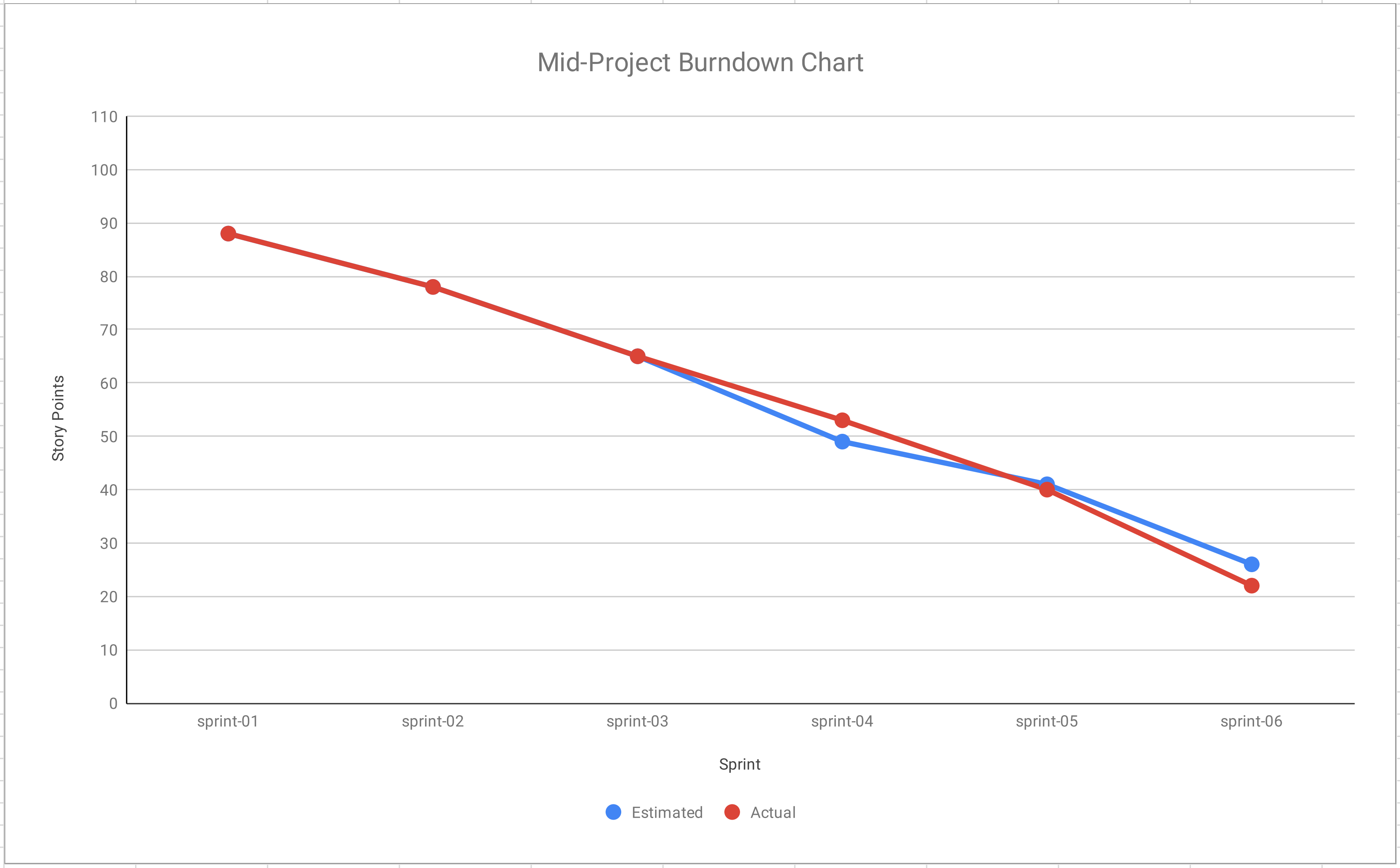
Product Vision	Project Mission
With the increase of data exchange between organizations and other independent instances such as finance, legal, healthcare, government, etc the need of ease of data interoperability while still adhering to data usage, policies, and compliance to local legalizations is becoming significant. Dataspace is the envisioned integrated solutions to tackle these challenges.	Explore the feasibiltiy of dataspace usage with regards to data sovereignty. This includes the testing the maturity of dataspace, which components are important and ease of deployment

Term	Definition
Data Sovereignity	The collecting and processing of data should be subjected to the laws of the country of which the data are being generated.
Dataspace	Data ecosystem that is built upon commonly agreed policies
(EDC) Connector	Entry port to the dataspace as well as means to exchange data
Catalog	A database storing all the metadata. It is part of the metadata broker
Policy	Preset rules of communication
Contract	Custom terms of agreements between two connectors regarding the use of data
Contract Negotiation	The process of validating the contract offered by the Data consumer
Policy Enforcement	The process to ensure that the data that is shared is strictly adhered to the agreed contract
Clearing House	a service for logging data exchange transactions within the International Data Space (i.e. contract negotiation)
Metadata	Information about the sovereign data (e.g. title, content, owner)
Metadata Broker	independent entity within the dataspace, responsible for metadata management
Identity Provider	system that manage identity information and provides authentication within the dataspace

Sprint #	Sprint goal
1	Understanding the concept of Dataspace and it's components
2	Run samples of EDC Connector (Team 1) and Gaia-X framework samples (Team 2)
3	Build the Kubernetes pipeline for automatic deployment in collaboration with DATEV
4	Start establising frontend framework early
5	Continue with the frontend framework and make the application user friendly
6	Prepare for mid-project release
7	Documentation & refactoring
8	Decentralising the connector
9	Create web frontend for each connector
10	Connect the connector with a database
11	Transfer data between 2 connectors
12	UI improvement, bug fixes, deployment to cloud

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release						
Total			88	88		
Sprints						
1	Understanding the concept of Dataspace and it's components		10	88	10	88
2	Run samples of EDC Connector (Team 1) and Gaia-X framework samples (Team 2)		13	78	13	78
3	Build the Kubernetes pipeline for automatic deployment in collaboration with DATEV		16	65	12	65
4	Start establising frontend framework early		8	49	13	53
5	Continue with the frontend framework and make the application user friendly		15	41	18	40
6	Prepare for mid-project release		26	26	26	22
	Sum					
Features						
1	**Understanding the concept of Dataspace and it's components**					
	Gain understanding of Gaia-X		5		5	
	Research connector concepts and the EDC connector repository		5		5	
2	**Run samples of EDC Connector (Team 1) and Gaia-X framework samples (Team 2)**					
	Eun EDC connector samples		8		8	
	Gain more understanding of Gaia-X		5		5	
3	**Build the Kubernetes pipeline for automatic deployment in collaboration with DATEV**					
	Create two connector instances		3		3	
	Transfer data on localhost		5		3	
	Create docker image output		5		3	
	Create documentation for the localhost connection process		3		3	
4	**Start establising frontend framework early**					
	Establish a frontend framework		8		13	
5	**Continue with the frontend framework and make the application user friendly**					
	Establish a frontend framework		8		13	
	Create build process video		2		2	
	Create a button in the frontend which runs the CLI automatically		5		3	
6	**Prepare for mid-project release**					
	Create a fuctioning button to establish a two-connector-connection		5		3	
	Open three ports for three connectors		2		2	
	Create UI design for login page		3		3	
	Create UI design for connector		5		5	
	Create UI frontend of the connector page		5		5	
	Create / Update the Dockerfile		3		5	

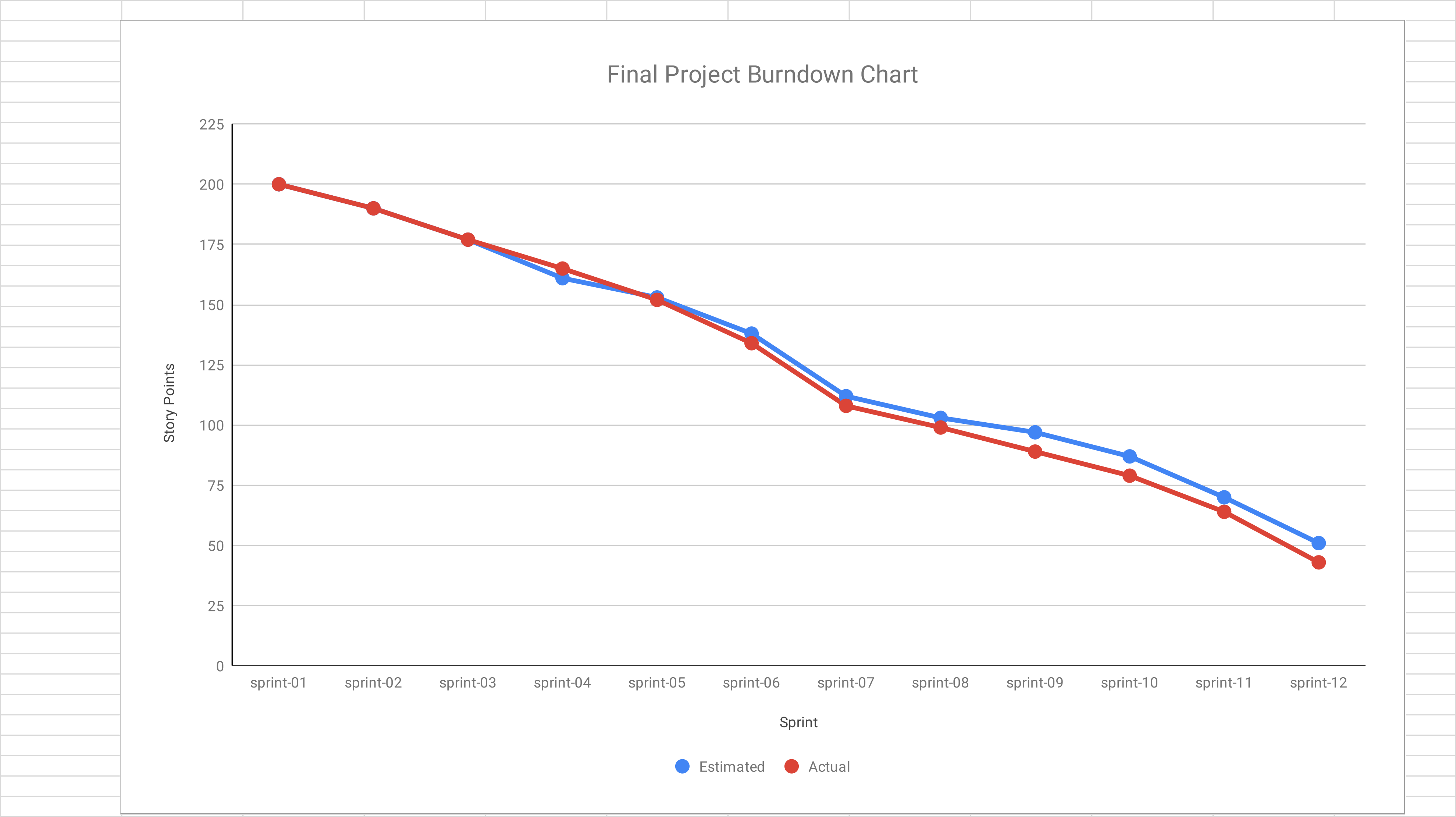
Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
	Test connector configurations		1		1	
	Response - bug fix		2		2	


Mid-Project Burndown Chart
Story Points
110
100
90
80
70
60
50
40
30
20
10
0
sprint-01
sprint-02
sprint-03
sprint-04
sprint-05
sprint-06
Sprint
● Estimated
● Actual

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release						
Total			200	200		
Sprints						
1	Understanding the concept of Dataspace and it's components		10	200	10	200
2	Run samples of EDC Connector (Team 1) and Gaia-X framework samples (Team 2)		13	190	13	190
3	Build the Kubernetes pipeline for automatic deployment in collaboration with DATEV		16	177	12	177
4	Start establising frontend framework early		8	161	13	165
5	Continue with the frontend framework and make the application user friendly		15	153	18	152
6	Prepare for mid-project release		26	138	26	134
7	Documentation & refactoring		9	112	9	108
8	Decentralising the connector		6	103	10	99
9	Create web frontend for each connector		10	97	10	89
10	Connect the connector with a database		17	87	15	79
11	Transfer data between 2 connectors		19	70	21	64
12	UI improvement, bug fixes, deployment to cloud		51	51	45	43
Features						
1	**Understanding the concept of Dataspace and it's components**					
	Gain understanding of Gaia-X		5		5	
	Research connector concepts and the EDC connector repository		5		5	
2	**Run samples of EDC Connector (Team 1) and Gaia-X framework samples (Team 2)**					
	Eun EDC connector samples		8		8	
	Gain more understanding of Gaia-X		5		5	
3	**Build the Kubernetes pipeline for automatic deployment in collaboration with DATEV**					
	Create two connector instances		3		3	
	Transfer data on localhost		5		3	
	Create docker image output		5		3	
	Create documentation for the localhost connection process		3		3	
4	**Start establising frontend framework early**					
	Establish a frontend framework		8		13	
5	**Continue with the frontend framework and make the application user friendly**					
	Establish a frontend framework		8		13	
	Create build process video		2		2	
	Create a button in the frontend which runs the CLI automatically		5		3	
6	**Prepare for mid-project release**					
	Create a fuctioning button to establish a two-connector-connection		5		3	
	Open three ports for three connectors		2		2	
	Create UI design for login page		3		3	
	Create UI design for connector		5		5	

				Est.		Real
Sprint	Goal	Feature Name	Est. Size	Remaining	Real Size	Remaining
	Create UI frontend of the connector page		5		5	
	Create / Update the Dockerfile		3		5	
	Test connector configurations		1		1	
	Response - bug fix		2		2	
7	**Documentation & refactoring**					
	Create build documentation		1		1	
	Create design documentation		3		2	
	Create user documentation		1		2	
	Bugfix: connector status checking		3		3	
	Refactor & delete unused branches		1		1	
8	**Decentralising the connector, creating web frontend for each connector, and start transfering the data**					
	Bugfix: "Execute command"-button		3		5	
	Uploadable files in the UI		3		5	
9	**Create web frontend for each connector**					
	Create web frontend for the bank connector		3		5	
	Create web frontend for tax advisor and company		2		1	
	Update docker file to accomodate three seperate web frontends		2		2	
	Check feasibility of establishing a connection with object storage		3		2	
10	**Connect the connector with a database**					
	Fix cloud UI accessibiliy		3		2	
	Create JSON files for all scenarios		5		5	
	Link a role-exclusive login page to each port		3		3	
	Transfer a text file between two connectors - frontend		3		3	
	Connector status checking 2.0 Docker		3		2	
11	**Transfer data between 2 connectors**					
	Display real data in the receive section on the dashboard		3		3	
	Create a upload section in the dashboard		3		5	
	Establish database connection with our connector		5		5	
	Error: connector is not running in cloud status		2		2	
	Add a button to connector status checking		2		2	
	Enable backend to be started through the frontend		3		3	
	Bugfix: download button from baseURL		1		1	
12	**UI improvement, bug fixes, deployment to cloud**					
	Bump EDC version from 0.6.0 to 0.7.0+		2		2	
	Seamless communication between frontend and backend		5		3	
	Transfer a text file between two connectors in cloud		3		2	
	Research how to add a new user to the dataplane		5		2	
	Finalize Dockerfile to the latest version		2		1	
	Finalize user documentation		1		1	
	Finalize design documentation		2		2	
	Finalize build documentation		1		1	
	Bugfix: select other roles when sending data		2		2	
	Seed connector with some initial data (a policy and a dataplane)		2		2	
	Create functionality to delete uploaded files from backend and link to frontend		2		3	
	Create a personal database for each connector and adjust backend and deployment pipeline		2		2	
	Add token-based authentication to frontend, connector and database for each connector		3		5	
	Prepare demo day presentation samples		2		2	
	Add new page to create Policies in frontend		3		2	
	Delete duplicate Dockerfiles and adjust docker-compose and github workflows		2		2	
	Add functionality to negotiate contracts and receive files from other connectors		5		3	

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
	Transfer a text file between two connectors - backend		5		5	
	Create a demo day video		2		3	


Final Project Burndown Chart
Story Points
225
200
175
150
125
100
75
50
25
0
sprint-01
sprint-02
sprint-03
sprint-04
sprint-05
sprint-06
sprint-07
sprint-08
sprint-09
sprint-10
sprint-11
sprint-12
Sprint
Estimated
Actual

Type	Link / reference
GitHub DATEV repo	https://github.com/projectamoscd
GitHub Wiki	https://github.com/amosproj/amos2024ss02-international-dataspace-station/wiki
DataSpace Connector Github Wiki	https://international-data-spaces-association.github.io/DataspaceConnector/
InternationalDataSpaces Docs	https://docs.internationaldataspaces.org/ids-knowledgebase

#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done
	Product Owners approved of the features	There is no major regression compared to previous release	Same as sprint release
	Software Developers agree to release the features	All Software Developers agree to release	Relevant features have been documented and put into the wiki
	All acceptance criteria need to be met	Release Manager set the tag to release	Bugs should be kept as minimal as possible
	Pull request has been reviewed, approved, and merged into the main branch	Application should be able to be executed	Documentation (User, Build, and Design) should reflect the latest version as of 17.07.2024)
		If there is no major changes and regressions, it still needs to be documented	

#	Name	Context	Version	License	Comment
1	org.eclipse.edc	EDC Connector	0.6.3	Apache-2.0	The base of our project
2	Docker	Docker	26.1.4	Apache-2.0	Docker framework for easy deployment
3	Gradle	Gradle	8.7.0	Apache-2.0	Build framework
4	@heroicons/react	JSON Package	2.1.3	JSON license	
5	axios	JSON Package	1.7.2	JSON license	
6	clsx	JSON Package	2.1.1	JSON license	
7	express	JSON Package	4.19.2	JSON license	
8	js-cookie	JSON Package	3.0.5	JSON license	
9	net	JSON Package	1.0.2	JSON license	
10	next	JSON Package	14.2.3	JSON license	
11	react	JSON Package	18.0.0	JSON license	Frontend
12	react-dom	JSON Package	4.7.5	JSON license	
13	socket.io	JSON Package	latest	JSON license	
14	node	JSON Package	20.14.1	JSON license	
15	prop-type	JSON Package	15.7.12	JSON license	
16	csstype	JSON Package	3.1.3	JSON license	
17	undici-type	JSON Package	5.26.5	JSON license	
18	org.springframework.boot	SpringBoot Framework	3.0.0	Apache-2.0	We use SpringBoot Framework and File Storage as a miniature database
19	io.spring.dependency-management	SpringBoot Framework	1.0.14	Apache-2.0	
20	jakarta.ws	Jakarta RESTful Web Services	2.0.1	Apache-2.0	
21	assertj	JSON Package	3.25.3	Apache-2.0	
22	awaitility	JSON Package	4.2.1	Apache-2.0	
23	jakarta-json	JSON Package	2.2.0	Apache-2.0	
24	junit-pioneer	JSON Package	5.10.2	Apache-2.0	
25	jupiter	JSON Package	5.0.0-alpha.14	Apache-2.0	
26	okhttp-mockwebserver	JSON Package	4.12.0	Apache-2.0	
27	restAssured	JSON Package	5.4.0	Apache-2.0	
28	rsApi	JSON Package	3.1.0	Apache-2.0	
29	testcontainers	JSON Package	1.19.7	Apache-2.0	
30	kafkaClients	JSON Package	3.7.0	Apache-2.0	
31	java	JDK	17+	Oracle	Programming language
32	curl	client URL library	8.6.0	SPDX	client URL to send command to the connector
33	jq	command-line JSON processor	1.7.1	MIT license	
	eslint	JSON Package	^8	JSON license	
	eslint-config-next: "14.2.3",	JSON Package	14.2.3	JSON license	
	postcss	JSON Package	^8	JSON license	

#	Name	Context	Version	License	Comment
	tailwindcss	JSON Package	3.4.1	JSON license	
	typescript	JSON Package	^5	JSON license	Frontend
	react-loader-spinner	JSON Package	6.1.6	JSON license	
	react-toastify	JSON Package	10.0.5	JSON license	
	next-auth	JSON Package	5.0.0-beta.19	JSON license	
	util-deprecate	JSON Package	1.0.2	JSON license	
	react-tabs	JSON Package	6.0.2	JSON license	

Last Name	First Name	Value					
Kurtz	**Daniel**			**3.00**	**OK**		
Kanatova	**Sezim**						
Cosgun	**Esra**	3					
Wysokinska	**Xemena**	3					
Ivanishcheva	**Ekaterina**			0	No size		
				1	Trivial size		
				2	Small size		
				3	Medium size		
				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