Project Name	NFT Playbook
Online team meeting	https://fau.zoom.us/j/3837513668
Production system (if any)	...
Test system (if any)	...
Happines Index	https://happy-amos.appspot.com/Project?project=6309985969504256&course=4791918638661632
GitHub repository	https://github.com/amosproj/amos2022ss07-nft-playbook
GitHub kanban board (project)	https://github.com/amosproj/amos2022ss07-nft-playbook/projects
GitHub Slides	https://github.com/dirkriehle/amos-course/tree/main/Generated/Lecture%20slides
Team T-shirt (white)	https://www.shirtinator.de/loadBasket/_3v8qHAfTpV
Team T-shirt (black)	-
Additional materials	...
Google Drive Link	https://drive.google.com/drive/folders/15LVL6MDP8iO_iAu4pZdy2X0DyU4kXmWl
Communication Channel	https://mm.phildree.de/amos-dev
Homework File	https://docs.google.com/document/d/1Lfi9pyeh-xWk1wJCQhiwSFgJIl6zEdHNs6-yTb9BWYw/edit#
Mailverteiler	amos-proj7@group.riehle.org

Last Name	First Name	GitHub User Name	Email Address	WhatsApp Number
Dikov	Hristo	dikovh	hristo.dikov@fau.de	WA-Gruppe bereits vollständig
Dreesens	Philipp	phildree	**philipp.dreesens@fau.de**	WA-Gruppe bereits vollständig
Schilling	Johannes	schlingling	jsh.schilling@fau.de	WA-Gruppe bereits vollständig
Wolfrum	Lukas	lukaswolfrum	lukas.wolfrum@fau.de	WA-Gruppe bereits vollständig
Al-Sheikh	Tawfeek	tawfeeka	toofe.al-sheikh@fau.de	WA-Gruppe bereits vollständig
Rotsching	Lukas	lukas-rotsching	lukas.rotsching@fau.de	WA-Gruppe bereits vollständig
Schwarzmann	Sebastian	MPSebastian	sebastian.schwarzmann@fau.de	WA-Gruppe bereits vollständig
Kurz	Noah	Noah-Kurz	noah.kurz@fau.de	WA-Gruppe bereits vollständig
Schlinger	Johanna	laila-rin	johanna.schlinger@fau.de	WA-Gruppe bereits vollständig

Goals	Involve every teammember to unlock its best potential
	Be helpful and respectful to each other
	Achieve customers satisfaction by providing a solution which covers all requirements
Meeting norms	Nobody is late for a meeting and values the others limited amount of time
	Every team member focuses on the customers satisfaction, not on technical details
	Everyone comes prepared to the meetings
Working norms	Everone is 100% motivated, 100% commited to the team and gives 100% the best to reach the overall project goal
	Everyone works an equaly amount (especiall SD). If someone has too less to do, (s)he has to ask actively for involvement
	We support each other
Coordination norms	Every task is assigned to exactly one person. If the person needs support, (s)he asks actively for it
	Every Taskowner bears the responsibilty, to deliver results until the commited due-date
Communication norms	Every team member checks all message channels at least one time per day in working day and reponses if necessary
	Respect everyone's opinion
	Response at least on the next day
Consideration norms	All product feature decisions are made from the Product Owner
	The team votes for a final decision if the whole projects gets impacted, Disagreements has to be discussed immediatly
Cont. improvement norms	Quality concerns have to be communicated immediatly
	Team spirit concerns have to be communicated immediatly, the Scrum Master has the responsibilty to solve them asap
Rewards	We honor every sprint release
	We drink some beer in periodic physical team building events
Sanctions	Every violation of the team contract has to be communicated immediatly
	The team votes for a sanction TBD
Signed by	Dikov
	Dreesens
	Schilling

	Wolfrum
	Al-Sheikh
	Rotsching
	Schwarzmann
	Kurz
	Schlinger

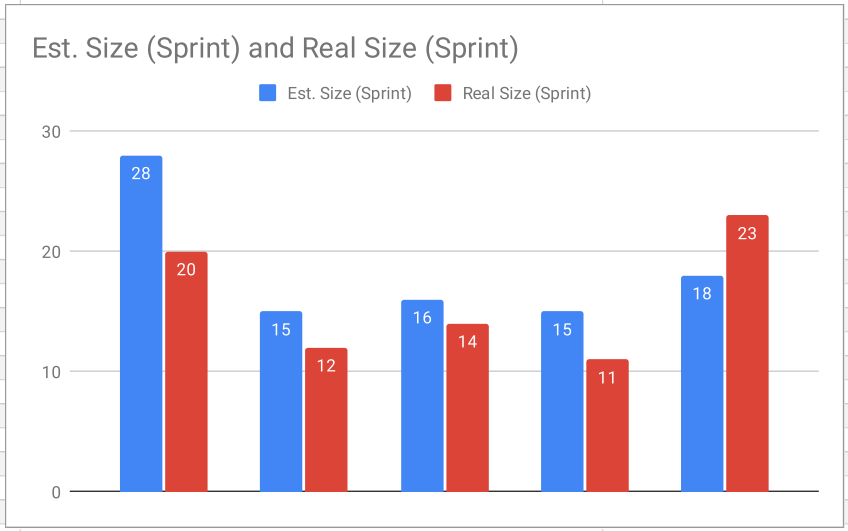
#	Meeting Day	Uni	Comment	Product Owner	Software Developer	Release Manager	Scrum Master
1	2022-04-27			Noah Kurz, Lukas Wolfrum	Everyone else	-	Hristo Dikov
2	2022-05-04			Noah Kurz, Lukas Wolfrum	Everyone else	Johannes Schilling	Hristo Dikov
3	2022-05-11			Noah Kurz, Lukas Wolfrum	Everyone else	Sebastian Schwarzmann	Hristo Dikov
4	2022-05-18			Noah Kurz, Lukas Wolfrum	Everyone else	Tawfeek Al-Sheikh	Hristo Dikov
5	2022-05-25			Noah Kurz, Lukas Wolfrum	Everyone else	Rotsching Lukas	Hristo Dikov
6	2022-06-01			Noah Kurz, Lukas Wolfrum	Everyone else	Philipp Dreesens	Hristo Dikov
7	2022-06-08		Mid-term due	Noah Kurz, Lukas Wolfrum	Everyone else	Johanna Schlinger	Hristo Dikov
8	2022-06-15			Noah Kurz, Lukas Wolfrum	Everyone else	Johannes Schilling	Hristo Dikov
9	2022-06-22			Noah Kurz, Lukas Wolfrum	Everyone else	Sebastian Schwarzmann	Hristo Dikov
10	2022-06-29			Noah Kurz, Lukas Wolfrum	Everyone else	Tawfeek Al-Sheikh	Hristo Dikov
11	2022-07-06			Noah Kurz, Lukas Wolfrum	Everyone else	Rotsching Lukas	Hristo Dikov
12	2022-07-13			Noah Kurz, Lukas Wolfrum	Everyone else	Philipp Dreesens	Hristo Dikov
13	2022-07-20			Noah Kurz, Lukas Wolfrum	Everyone else	Johanna Schlinger	Hristo Dikov
14	2022-07-27		Demo day!	Noah Kurz, Lukas Wolfrum	Everyone else	Johannes Schilling	Hristo Dikov
15	2022-08-03		Retrospective	Noah Kurz, Lukas Wolfrum	Everyone else	Sebastian Schwarzmann	Hristo Dikov

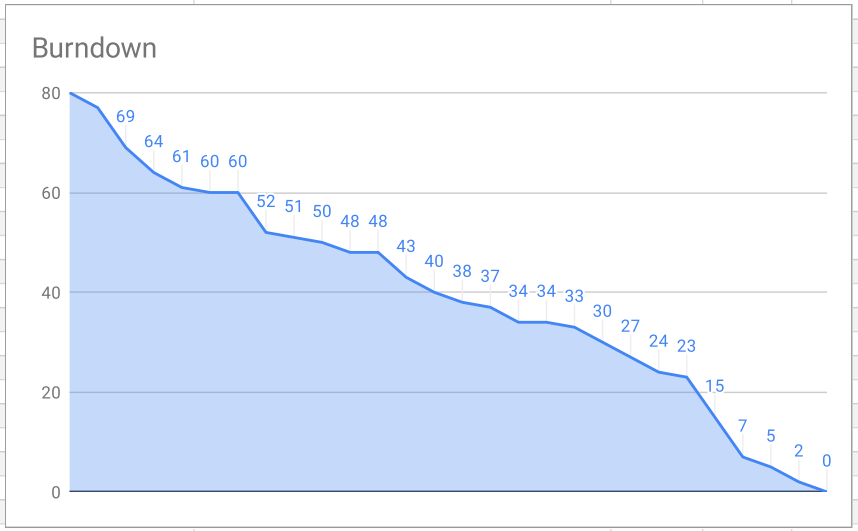
Product Vision	Project Mission
We believe that NFTs have a great potential for companies, artists and private persons. Nevertheless, NFTs are currently either seen as nerdy gadget or rocket science. With this product, we want to change that. We want to demystify NFTs by enabling enthusiast to create NFTs with the least possible effort to focus on their particular use case, not having to think about the NFT creation process itself. Through this, we reach increasing acceptance and distribution of the technology. In the end we want to make the world a tiny bit better by for example ensuring digital property rights, enabling identification in a decentralized manner without being dependend on an administering instance.	The mission of this project is to create a CLI which enables users to create NFTs on the blockchains Polygon, Flow and Solana. Two major use cases will be supported. Firstly, for showcasing how easy NFTs can be created, the CLI has a manual input option where the user can select the blockchain and edit features of the NFT like name and a picture. After that the NFT can be created with a simple command. Secondly, NFTs can be mass produced to include their functionalities in projects. For this, mass deployment can be achieved by defining the features of each NFT individually in a JSON file. The JSON is then read by the CLI and the NFTs are created.

Term	Definition
Non-fungible Token / NFT	Digital asset stored in a blockchain. Every NFT can have an owner. The owner is entitled to sell or use the asset. It differs from usual crypto currency that every NFT is individual / non fungible. The individuallity is often associated with a hashed picture stored in the NFT contract.
Wallet	A digital purse for crypto-assets, NFTs and other crypto currencies.
Minting	Process of turning the information from the CLI into a valid NFT. The process includes writing the NFT contract on the blockchain
CLI	Command line interface. Program to take input parameters from the user, store them temporarily and then write them on the contract.
(Smart) Contract	Intelligent contract based on computer protocolls. Is the underlying technology of NFTs and used to store the information like the associated picture and the owner.
Blockchain	A cryptographically secured append-only database which history is immutable.
Ethereum	Cryptocurrency and blockchain which is used to create smart contracts and pay assets.
Flow	Cryptocurrency and blockchain which is used to create smart contracts and pay assets.
Solana	Cryptocurrency and blockchain which is used to create smart contracts and pay assets.
Mainnet	A blockchain which is used for real world applications and transactions.
Testnet	A blockchain which is used for development purposes only.
Hash	Individual result of an hash function. Can be used to identify pictures and associate them to NFTs.

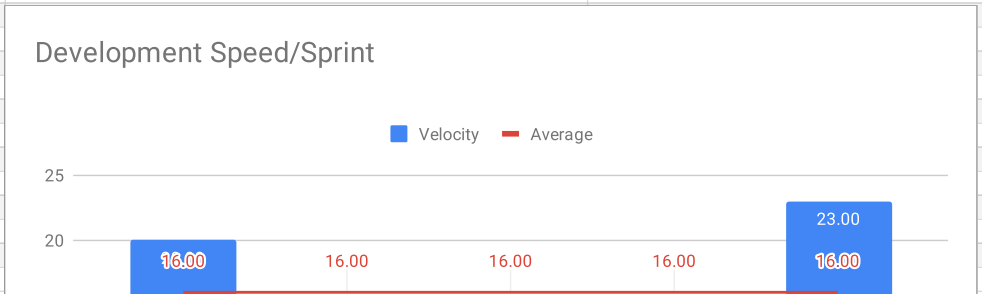
#	Theme	Goal	Feature Name	Est. Size (Feature)	Est. Size (Sprint)	Real Size (Feature)	Real Size (Sprint)	Burn-Down
1	**Organizational Project Setup**	**Get to Know Each Other and the Tasks**						80
			Sum					
2	**Technical Project Setup**	**Create Setup for Devops & CLI**						
			#1 Setup CLI Basic Architecture	8		3		77
			#3 Mint NFT on Ether Based Chain	8		8		69
			#11 Create Devops Setup	8		5		64
			#7 Setup CLI	3		3		61
			#9 CLI Greets User	1		1		60
			Sum		28		20	
3	**Frontend & Backend Architecture and Functionality**	**Create a Backend that can Interact with the CLI**						
			#20 NFT Settings CLI	3				60
			#27 Refactoring Backend	8		8		52
			#16 Homescreen for CLI	1		1		51
			#10 Help Function CLI	1		1		50
			#17 Blockchain Selection CLI	2		2		48
			Sum		15		12	
4	**Frontend & Backend Connection**	**Extend the Functionality of the Frontend with the Backend**						
			#31 Research Pic to Hash Conversion	3				48
			#41 Interface for Backend Frontend Communication	5		5		43
			#39 Solidity Contract for NFTs	3		3		40
			#21 Create NFT Function CLI	1		2		38
			#20 NFT Settings CLI	1		1		37
			#40 Connections to Frontend for NFT Information	3		3		34
			Sum		16		14	
5	**Feature Refinement and Wallet Connect**	**Add the Possibility to Connect with a Wallet and Refine Existing Minting Features**						
			#49 Select Wallet Feature CLI	2				34
			#45 Handling Sensitive Data	1		1		33
			#30 Logical Structure for Wallet Integragtion	5		3		30
			#31 Research Pic to Hash Conversion	3		3		27
			#29 Research "Add Wallet Feature"	3		3		24
			#50 "Add Wallet Feature" in CLI	1		1		23
			Sum		15		11	
6	**Picture Hashing and Refinement for Mid-Project Release**	**Add the Possibility to Create a Hash from a Picture and Polish Existing Features for Release**						
			#62 Submit Deliverables Sprint-06	8		8		15
			#67 Connect Middleware to Front- and Backend	3		8		7
			#32 Integrate Piniata as IPFS Service	3		2		5
			#33 Gas fee prediction	3		3		2
			#51 Update CLI Texts	1		2		0
			Sum		18		23	

#	Theme	Goal	Feature Name	Est. Size (Feature)	Est. Size (Sprint)	Real Size (Feature)	Real Size (Sprint)	Burn-Down

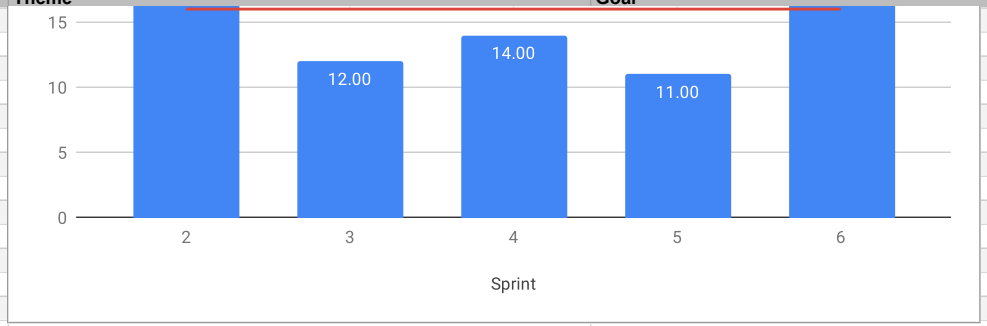




Sprint	Velocity	Average
2	20.00	16.00
3	12.00	16.00
4	14.00	16.00
5	11.00	16.00
6	23.00	16.00

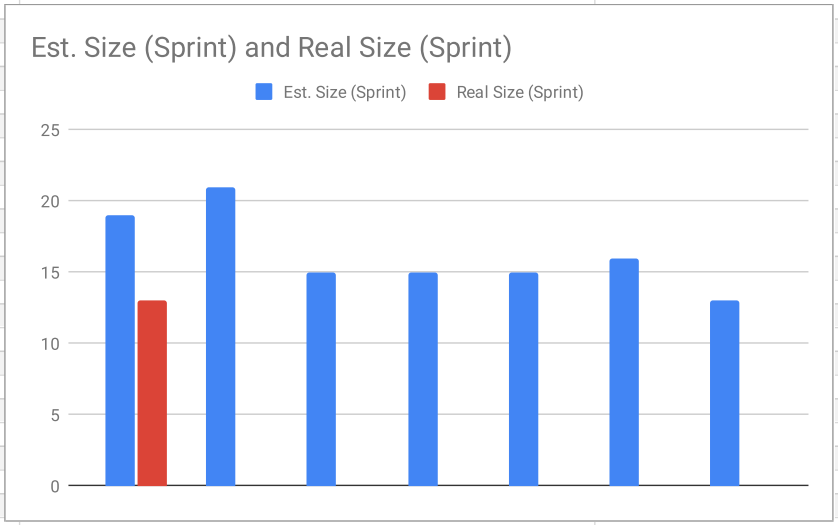


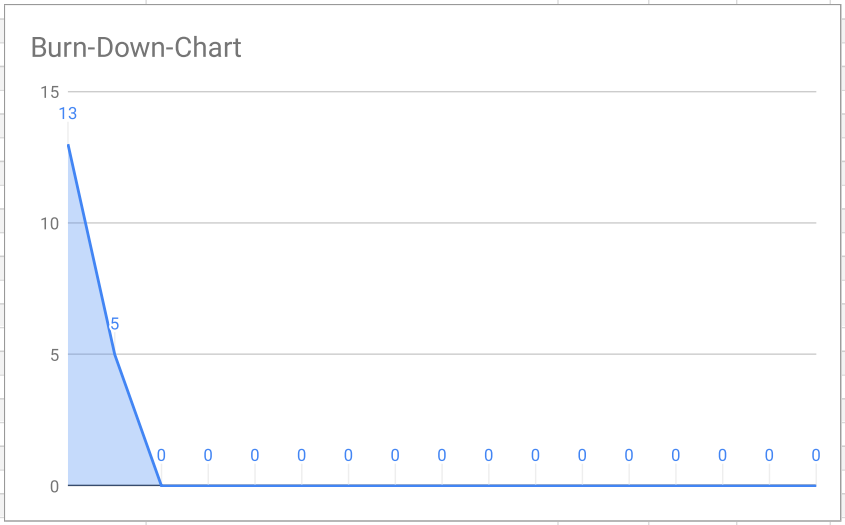
#	Theme	Goal	Feature Name	Est. Size (Feature)	Est. Size (Sprint)	Real Size (Feature)	Real Size (Sprint)	Burn-Down



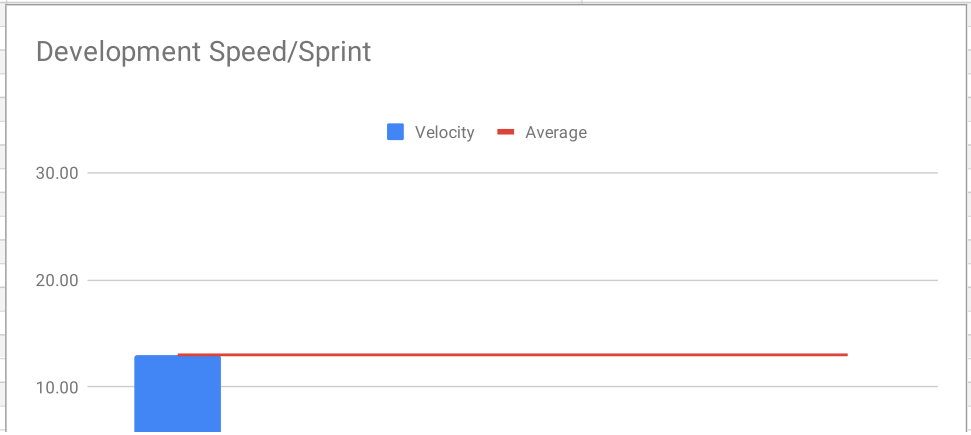
#	Theme	Goal	Feature Name	Est. Size (Feature)	Est. Size (Sprint)	Real Size (Feature)	Real Size (Sprint)	Burn-Down
								13
7	**Add Flow Blockchain & Exception Handling**	**Add the Option to Mint on the Flow Blockchain, also Exception Handling is a necessary refactoring**						
			#4 Research about minting on Flow Blockchain	3				13
			#44 Flow Blockchain Fuction can be called succesfully from CLI	8		8		5
			#66 Exception Handling Middleware/Backend	8		5		0
			Sum		19		13	
8	**Connect Wallet & Add Solana**	**Add the Connect Wallet Option, Implement a log file for the user, Add the Solana Blockchain to the Backend**						
			#72 Wallet integration Middleware	5				0
			#83 Solana Implementation Backend	8				0
			#86 Implement log file	3				0
			#59 Customizable Ethereum Network Parameters	2				0
			#79 Store Pinata password in dotenv	2				
			#81 Show Success message when Pinata Upload is succesful	1				
			Sum		21		0	
9	**Add Flow Blockchain**	**Add the Option to Mint on the Flow Blockchain in Frontend and Backend**						
			Further Resarch on Flow Blockchain	5				0
			Create CLI Input for Flow	2				0
			Create Backend Extension for Flow	8				0
			Sum		15		0	
10	**Add Solana Blockchain**	**Add the Option to Mint on the Solana Blockchain in Frontend and Backend**						
			Further Resarch on Solana Blockchain	5				0
			Create CLI Input for Solana	2				0
			Create Backend Extension for Flow	8				0
			Sum		15		0	
11	**Add Custom Blockchain**	**Add the Option to Easily Add Other Blockchains in Frontend and Backend**						
			Research on other interesting Blockchains	3				0
			Create Config File for Parameters	2				0
			Integrate Config File in Backend	8				0
			Integrate Config File Selection in Frontend	2				0
			Sum		15		0	
12	**Refactoring and Documentation**	**Provide Extensive Documentation and Neat Code Towards the End of the Project**						
			Write User Manual and Technical Documentation	8				
			Refactor the code	8				
			Sum		16			
13	**Refinement for Final Project Release**	**Make Sure that Everything is Ready for the Final Presentation**						
			Extensive Feature Testing	8				
			Prepatation for Presentation	5				
			Sum		13			

#	Theme	Goal	Feature Name	Est. Size (Feature)	Est. Size (Sprint)	Real Size (Feature)	Real Size (Sprint)	Burn-Down

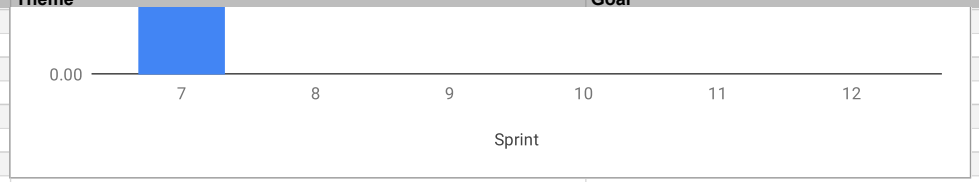




Sprint	Velocity	Average
7	13.00	13.00
8		13.00
9		13.00
10		13.00
11		13.00
12		13.00
13		13.00



#	Theme	Goal	Feature Name	Est. Size (Feature)	Est. Size (Sprint)	Real Size (Feature)	Real Size (Sprint)	Burn-Down



#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done
	A Feature is Done when all its Acceptance criterias are fulfilled	A Sprint Release is Done when no severe bugs are open	A Project Release is Done when the software runs without any noticeable bugs
	A Feauture is Done when the code has been peer reviewed from the dedicated team	A Sprint Release is Done when the whole software can be executed	A Project Release is Done when the core interests of the costumer are fulfilled
	A Feature is Done when it has been tested		
	A Feature is Done when its core functionality has been commented properly		
	A Feature is Done when it can be fully integrated into the main branch without negative side effect		
	A Feature is Done when its core functionality has been documented properly		

Type	
General Wiki	https://github.com/amosproj/amos2022ss07-nft-playbook/wiki
User Documentation	https://github.com/amosproj/amos2022ss07-nft-playbook/wiki/User-Documentation
Design Documentation	https://github.com/amosproj/amos2022ss07-nft-playbook/wiki/Design-Documentation
Build/Deploy Documentation	https://github.com/amosproj/amos2022ss07-nft-playbook/wiki/Build-Deploy-documentation

\	Context	Name	Version	License	Comment
		@nrwl/cli	14.01.02	MIT	taskexexutor for build system
		@nrwl/eslint-plugin-nx	14.01.02	MIT	linter
		@nrwl/jest	14.01.02	MIT	testing
		@nrwl/js	14.01.02	MIT	java script-Adaption for mono-Repo
		@nrwl/linter	14.01.02	MIT	linter-Adaption for mono-Repo
		@nrwl/node	14.01.02	MIT	node-Adaption for mono-Repo
		@nrwl/workspace	14.01.02	MIT	mono-Repo
		@types/jest	27.04.01	MIT	testing
		@types/node	16.11.07	MIT	NodeJS Framework for Java Script
		@types/inquirer	08.02.01	MIT	interactive command line.
		@typescript-eslint/eslint-plugin	05.18.0	MIT	linter
		@typescript-eslint/parser	05.18.0	MIT	linter
		eslint	08.12.00	MIT	linter
		eslint-config-prettier	08.01.00	MIT	linter
		eslint-plugin-prettier	04.0.0	MIT	linter
		jest	27.05.01	MIT	testing
		nx	14.01.02	MIT	build system
		prettier	02.05.01	MIT	linter
		ts-jest	27.01.04	MIT	typescript for testing
		ts-node	09.01.01	MIT	typescript for node
		typescript	04.06.02	MIT	typescript for java script
		dotenv	16.00.01	BSD-2-Clause	module that loads env. variables from a .env
		chalk	04.01.02	MIT	terminal string styling
		@pinata/sdk	^1.1.25	MIT	API for pinata network
		@solana/solidity	^0.0.20	MIT	API for Ethereum Solidity Contracts
		"@solana/spl-token":	"^0.2.0",	MIT	API for Solana Contract
		"@solana/web3.js":	"^1.44.0",	MIT	API for Solana
		"axios":	"^0.27.2",	MIT	API for HTTP Requests
		"ethers":	"^5.6.5",	MIT	API for Ethereum
		"inquirer":	"^8.2.4",	MIT	CLI API
		"node-fetch":	"^2.6.1",	MIT	Fetch API for node.js
		"solc":	"0.8.14-fixed",	MIT	API for Ethereum Solidity Contracts Compiler
		"tslib":	"^2.3.0"	MIT	Runtime libary for ts helper functions
		@metaplex-foundation/js	^0.11.5	MIT	API for Solana NFT System

Last Name	First Name	Value					
Dikov	Hristo						
Dreesens	Philipp			2.50	NOK		
Schilling	Johannes						
Wolfrum	Lukas						
Al-Sheikh	Tawfeek	2		0	No size		
Rotsching	Lukas	3		1	Trivial size		
Schwarzmann	Sebastian			2	Small size		
Kurz	Noah			3	Medium size		
Schlinger	Johanna			5	Large size		
				8	Very large size		
				13	Too large (size)		
Source:	https://t2informatik.de/wissen-kompakt/planning-poker/						