Project Name	Financial Product Portfolio Quick-Check
Production system (if any)	...
Test system (if any)	...
GitHub repository	https://github.com/amosproj/amos2021ws05-fin-prod-port-quick-check
GitHub kanban board (project)	https://github.com/amosproj/amos2021ws05-fin-prod-port-quick-check/projects/1
Team T-shirt (white)	https://www.shirtinator.de/loadBasket/TVvtvi-d3Fa
Team T-shirt (black)	https://www.shirtinator.de/loadBasket/TVvtvi-d3Fa
Additional materials	...
Happinessindex	https://happy-amos.appspot.com/Project?project=6194017020674048&course=6014071715397632
Team T-shirt (white)	~~Ivan Butron Sossa~~
Team T-shirt (black)	Maximilian Bartels, Ines Vogel, Lily Hügerich, Alexander Werner, Maximilian Rodiek, Tobias Bernhard, Andreas Ronellenfitsch

Last Name	First Name	GitHub User Name	Email Address
Bartels	Maximilian	Batogami	m.bartels@campus.tu-berlin.de
Bernhard	Tobias	tobbber	to-tb@gmx.de.
~~Butron Sossa~~	~~Ivan Antonio~~	~~ibutrons~~	~~ivanbutronsossa@gmail.com~~
Hügerich	Lily	lilyhuegerich	lilyhueg@gmail.com
Rodiek	Maximilian	m4xrdk	rodiek@campus.tu-berlin.de
Ronellenfitsch	Andreas	aron4	a.ronellenfitsch@campus.tu-berlin.de
Vogel	Ines	InesVogel	ine.vogel@gmail.com
Werner	Alexander	AlexanderW1996	werner.3@campus.tu-berlin.de
Groth	Patrick		patrick.groth@fau.de

Goals	Be Helpful & Respectful
	We and our customers are satisfied with the project results
	Not only satisfied with the product, also with the moral of and atmosphere in the team
	Improve efficiency over the semester
Meeting norms	Team Meetings are mandatory (Thursday, 12:30pm)
	Be on time
	Turn your camera on if possible
	Come prepared
Working norms	Primate of consensus, else voting, majority wins
	Support your team
	Task assignments will be made clear and agreed upon by the end of weekly group meetings.
	Task can solved independently or in groups (pair programming)
	Give constructive criticism, dont make it personal
Coordination norms	Team meetings will be lead by the POs
	Assignments can be chosen based on preference if assignments are not voluntarily chosen the Product owner can assign tasks based on workloads
Communication norms	Our communication channel will be slack and we write respectfully
	Communicate over the corresponding channels
	Response at least in 24h
	Whatsapp as urgent channel
	Any discussion should be viewable to anyone that the discussion pertains.
	If you cannot participate or arrive late, write a message in slack beforehand if possible at least an hour before
	Ask for help if necessary
Consideration norms	Side conversations in separate slack channel
	Don't interrupt each other in the zoom calls
	Be considerate of others people time and communicate at your earliest convience
	If two people have a conflict choose mediator, for more people or fail group discussion
Cont. improvement norms	Kanban board, closed tickets and weekly deadlines are used to track progress
	Test and code metrics
	Assign default reviewer
	Master should always be runnable

	Anyone can start a discussion about outcomes or notices from their assignments.
Rewards	Cake
	Drinks
Sanctions	Write Documentation
	Last place for task assignment
	Sanctions only by vote

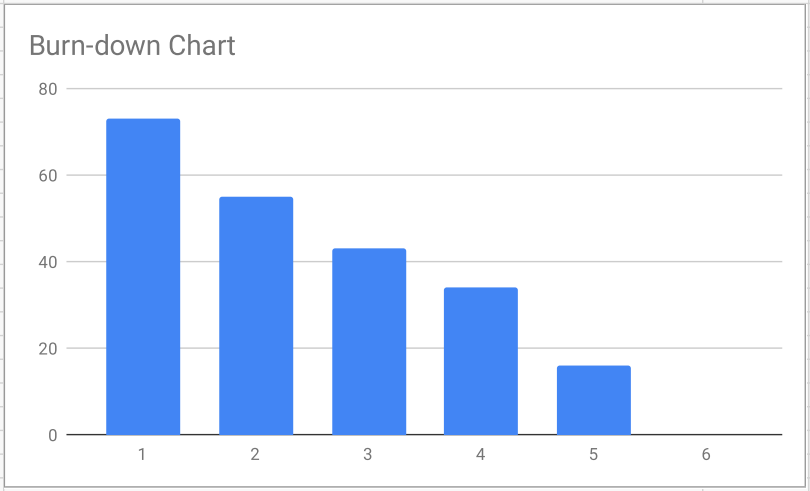
#	Meeting Day	Comment	Coach	Product Owner	Software Developer	Release Manager	Scrum Master
1	2021-10-21		Yes	Max R & Andreas	Everyone else	N/A	Coach
2	2021-10-28		Yes	Max R & Andreas	Everyone else	N/A	Coach
3	2021-11-04		Yes	Max R & Andreas	Everyone else	Ines Vogel	Coach
4	2021-11-11		Yes	Max R & Andreas	Everyone else	Ines Vogel	Coach
5	2021-11-18		Yes	Max R & Andreas	Everyone else	Ines Vogel	Coach
6	2021-11-25		Yes	Max R & Andreas	Everyone else	Ines Vogel	Coach
7	2021-12-02	Mid-project release	Yes	Max R & Andreas	Everyone else	Alexander Werner	Coach
8	2021-12-09			Max R & Max B	Everyone else	Alexander Werner	Max R
9	2021-12-16			Max R & Max B	Everyone else	Alexander Werner	Max B
10	2022-01-13		Yes	Max R & Max B	Everyone else	Alexander Werner	Max R
11	2022-01-20			Max R & Max B	Everyone else	Alexander Werner	Max B
12	2022-01-27			Max R & Max B	Everyone else	Alexander Werner	Max R
13	2022-02-03		Yes	Max R & Max B	Everyone else	Alexander Werner	Max B
14	2022-02-10	Demo day / final release		Max R & Max B	Everyone else	Alexander Werner	Max R
15	2022-02-17	Project retrospective due		Max R & Max B	Everyone else	Alexander Werner	Max B

Product Vision	Project Mission
With growing opportunities of derivating and customizing financial products the product range of financial institutions is emerging rapidly. Hence, the difficulty of good consulting is rising equally while high complexity products drive the largest costs which are outgrowing benefits and revenue of conventional consulting. The Financial Product Portfolio Quick-Check simplifies financial consulting by providing an information-rich project-related overview of financial products. This overview helps banking and other types of financial institutions to understand the risks and opportunities of different financial products, while it tries to maximize profits by identifying overly complex products, which do not contribute to the business volume. Additionally, the product provides useful visualisations annotated with source documents to ensure an increase of a consultants efficiency in assessing the benefit of a product. Published under the MIT license, the aim is to make it accessible for our business partner as well as for any other organization and enable an enormous amount of other use cases.	Our project mission is to provide a web application which allows structured data entry for financial products, comparison and evaluation among products and product variants as well as with source documents annotated visualization. The core feature of the software is a calculation of benchmarks regarding complexity combined with annotated result charts. To meet the objectives mentioned above, we will maintain a helpful and friendly team atmosphere.

Term	Definition
Admin	The admin can create/delete users and assign roles
Consultant	The consultant is the main user of the software, which can conduct analysis, enter ratings and print charts.
Epic	An Epic describes several issues which are grouped as one topic
Evaluation view	An Epic describes several issues which are grouped as one topic
Issue	An issue can describe user stories, bugs, tasks or other types of issues.
Komplexitätskriterium/-treiber	Is a class of questions with the same topic
Kundenschlüssel	The customerID
Marge	The margin whichs displays the y-axis in the in the pie chart
Piechart (äußerer Ring)	The customer complexity, for each Produkt three fields for the percentage of hoch, mittel & gering complex customer is given.
Piechart (innerer Ring)	The distribution of Bewertungen of Produktvarianten for one Produkt.
Piechart (Größe)	The volume of the piechart displays the criteria "credit volume" in the economical evaluation
Produktbereiche	Every product area has to sub areas (private and corporate) & and there are 1 to n products for each product area
Produktschlüssel	The productID
Produkt	Every product has one to n product variants, which are divided in product areas.
Produktvariante	One or many Produktvarianten are forming the Produkt; The Produkt is a Produktvariante of itself; Exp: Sofortdispo, Dispo Fix & Dispo Variabel
Rating	Hoch, Mittel & gering (with colours: red, yellow & green); is set by a consultant
Result view	The webpage with the charts
Projekt	A Projekt has one or many Produkte; Each Projekt has 1-n Produktbereiche
User	A User is ther overall class for Consultant, Admin and Project Manager
User Story	A user story is a describtion of a software functionality
Gesamteinschätzung wirtschaftliche Bewertung	This value is used to determine the economical complexity of Produkt or a Produktvariante; This value is displayed in the inner piechart ring

#	Theme	Goal	Feature Name	Est. Size (Feature)	Est. Size (Sprint)	Real Size (Feature)	Real Size (Sprint)	Burn-Down
			TOTAL					
1	**Initial Setup**	**Getting to know & project setup**			**10**		**10**	**83**
			Additional team meeting	2		2		
			#3 Team Logo	1		2		
			#1 T-Shirt Logo	3		2		
			#2 Team Contract	2		1		
			Meeting with industry partner	2		3		
2	**Initial Setups & Research**	**Understand the project & Implement first requirements from industry partner (Docker)**			**24**		**18**	**73**
			#12 Define Initial REST-API	5		3		
			#13 Create Initial Containerized Architecture	5		5		
			#17 Define Minimal Requirements (User Stories)	5		5		
			#15 Upload Research for WebUI	3		1		
			#14 Upload Research for Database Types	3		1		
			#11 Conduct Research for User Management	3		3		
3	**Frontend and Backend Functionalities & Architecture**	**Backend Architecture & first WebUI is implemented**			**18**		**12**	**55**
			#36 Refactor project API	2		1		
			#34 Setup Backend REST API	5		3		
			#30 Set Up Relational Database	3		2		
			#32 Set Up Initial WebUI	5		3		
			#41 Use Case Diagram	1		1		
			#42 Flowchart Diagram	1		1		
			#40 ER-Diagram	1		1		
4	**First Frontend Technical Breakthrough**	**Clear understanding of Project Mission, Product Vision & Definition of done & working on frontend**			**9**		**9**	**43**
			#37 Product Vision	1		1		
			#38 Project Mission	1		1		
			#59 Definitons of Done	1		2		
			#44 Update Glossary	1		2		
			#46 Create View "Project Overview"	5		3		
5	**Main Frontend and Backend Functionalitites**	**Getting a breakthrough with the main funcionalitites of the software**			**24**		**18**	**34**
			#45 Create View "Manage Project"	5		3		
			#47 Create View "Manage Project Members"	5		5		
			#48 Create "Product Overview"	5		5		
			#61 Create "Enter Financial Product Data"	3		1		
			#63 Create "Add Product View"	3		1		
			#64 Create "Enter Complexity Data"	3		3		
6	**Mid Term Release & Testing**	**Getting the last frontend functionalities for a first breakthrough of the software and continue testing the backend**			**21**		**16**	**16**
			#65 Create View "Complexity Evaluation"	5		3		
			#62 Create View "Economic Evaluation"	5		5		
			#79 Testing of "Controller" Folder (Backend)	3		2		

#	Theme	Goal	Feature Name	Est. Size (Feature)	Est. Size (Sprint)	Real Size (Feature)	Real Size (Sprint)	Burn-Down
			#81 Testing of Data Transfer Object ("dto") Folder (Backend)	3		2		
			#83 Testing of "Service" Folder (Backend)	3		3		
			#82 Testing of "Exceptions" Folder (Backend)	2		1		
	Release	**Mid Term**						
	No Sprints	6 .						
	Due Date	02.12.2021						
	Sprint	**Sprint Theme**	**User Stories**	**Est. Size**	**Est. Burndown**	**Real Size**	**Real Burndown**	
	0				106		83	
	1	**Initial Setup**	3,2,1	10	96	10	73	
	2	**More Initial Setups & Research**	12, 13, 17, 14, 15, 11	24	72	18	55	
	3	**Frontend and Backend Functionalities & Architecture**	36, 34, 30, 32, 41, 42, 40	18	54	12	43	
	4	**First Frontend Technical Breakthrough**	37, 38, 59, 44, 46	9	45	9	34	
	5	**Main Frontend and Backend Functionalitites**	45, 47, 48, 61, 63, 64	24	21	18	16	
	6	**Mid Term Release & Testing**	65, 62, 79, 81, 83, 82	21	0	16	0	
	Total			**106**		**83**		



#	Theme	Goal	Feature Name	Est. Size (Feature)	Est. Size (Sprint)	Real Size (Feature)	Real Size (Sprint)	Burn-Down
			TOTAL					
1	**Initial Setup**	**Getting to know & project setup**			**10**		**10**	**83**
			Additional team meeting	2		2		
			#3 Team Logo	1		2		
			#1 T-Shirt Logo	3		2		
			#2 Team Contract	2		1		
			Meeting with industry partner	2		3		
2	**Initial Setups & Research**	**Understand the project & Implement first requirements from industry partner (Docker)**			**24**		**18**	**73**
			#12 Define Initial REST-API	5		3		
			#13 Create Initial Containerized Architecture	5		5		
			#17 Define Minimal Requirements (User Stories)	5		5		
			#15 Upload Research for WebUI	3		1		
			#14 Upload Research for Database Types	3		1		
			#11 Conduct Research for User Management	3		3		
3	**Frontend and Backend Functionalities & Architecture**	**Backend Architecture & first WebUI is implemented**			**18**		**12**	**55**
			#36 Refactor project API	2		1		
			#34 Setup Backend REST API	5		3		
			#30 Set Up Relational Database	3		2		
			#32 Set Up Initial WebUI	5		3		
			#41 Use Case Diagram	1		1		
			#42 Flowchart Diagram	1		1		
			#40 ER-Diagram	1		1		
4	**First Frontend Technical Breakthrough**	**Clear understanding of Project Mission, Product Vision & Definition of done & working on frontend**			**9**		**9**	**43**
			#37 Product Vision	1		1		
			#38 Project Mission	1		1		
			#59 Definitons of Done	1		2		
			#44 Update Glossary	1		2		
			#46 Create View "Project Overview"	5		3		
5	**Main Frontend and Backend Functionalitites**	**Getting a breakthrough with the main funcionalitites of the software**			**24**		**18**	**34**
			#45 Create View "Manage Project"	5		3		
			#47 Create View "Manage Project Members"	5		5		
			#48 Create "Product Overview"	5		5		
			#61 Create "Enter Financial Product Data"	3		1		
			#63 Create "Add Product View"	3		1		
			#64 Create "Enter Complexity Data"	3		3		
6	**Mid Term Release & Testing**	**Getting the last frontend functionalities for a first breakthrough of the software and continue testing the backend**			**21**		**16**	**16**
			#65 Create View "Complexity Evaluation"	5		3		
			#62 Create View "Economic Evaluation"	5		5		
			#79 Testing of "Controller" Folder (Backend)	3		2		
			#81 Testing of Data Transfer Object ("dto") Folder (Backend)	3		2		
			#83 Testing of the first half of "Service" Folder (Backend)	3		3		
			#82 Testing of "Exceptions" Folder (Backend)	2		1		
7	**Refactoring, Tests & Refinement**	**Code Cleanup for Developers & do leftovers**			**31**		**23**	
			#101 Define custom UI theme	3		5		
			#84 Refactor Project Management Backend	13		5		
			#99 Connect Product Overview View with Backend	3		3		
			#131 Fix Edit Mode on Product Overview View	3		2		
			#102 Basic layout for Product Rating View	3		3		
			#130 Refactor Frontend Components	5		3		
			#141 Implement deleteProjectUser	1		2		

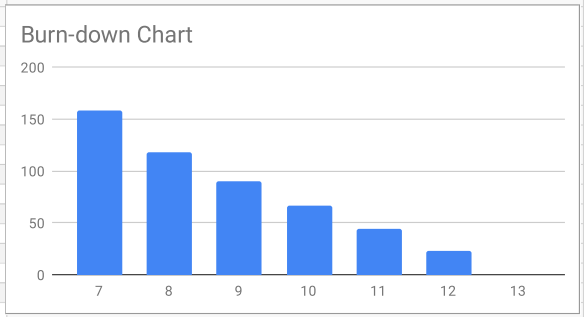
#	Theme	Goal	Feature Name	Est. Size (Feature)	Est. Size (Sprint)	Real Size (Feature)	Real Size (Sprint)	Burn-Down
8	**Product Ratings**	**Implement the possibility to rate the products with scores**			**46**		40	
			#103 Fill the Basic Layout of the Product Rating View with Mock Data	2		2		
			#104 Implement Bearing Point Theme	2		2		
			#142 Allignment between front & backend regarding API endpoints and input/output	5		2		
			#149 Code Style in frontend	2		2		
			#148 State management	5		5		
			#144 Refactor backend	13		13		
			#97 Testing of the second half of the "Controller" Folder (Backend)	5		3		
			#95 Testing of the second half of the "Service" Folder (Backend)	5		5		
			#98 Connect Manage Project View with Backend	3		3		
			#143 Database Upload of Dummy Data	3		2		
			#110 Catalogue for Scores (Backend)	1		1		
9	**Evaluation of results**	**Implement basis for visualisation of results**			**40**		28	
			#109 Make the Functional Site of the Product Rating View Dynamic and Connect it to Backend	5		2		
			#176 Define API for Result view	1		3		
			#116 Make the Basic Layout of the Result View to a Functional Site	3		2		
			#84 Refactor "Score", "ProductAreaController", & "RatingController" functionalities dynamically	2		2		
			#166 Fix disabled testcases with flag (fix)	1		1		
			#168 Fix disabled testcases with flag (figure out)	3		1		
			#167 Add disabled testcases with flag (implement)	3		1		
			#80 Testing of Database ("db") Folder (Backend)	5		1		
			#174 Get/Fetch Data with backend in product overview	3		2		
			#171 Setup Basic Charts/Diagrams	3		5		
			#106 Make the Basic Layout of the Product Rating View to a Functional Site	2		2		
			#169 Implement Tests for ProjectUserService	3		2		
			#105 Write code documentation for backend	2		2		
			#173 Review issues in product backlog	2		1		
			#172 Product Glossary is unclead and definitions are too long	2		1		
10	**Start Visualisation of Results**	**Start with implementation of the core of the software, the visualisation of results in different graphs**			**29**		23	
			#115 Fill the Basic Layout of the Result View with Mock Data	3		3		
			#108 Evaluate project in backend	5		3		
			#195 Connect result figure to product page	2		0		
			#112 Fill the Basic Layout of the Evaluation View with Mock Data	2		2		
			#111 Basic layout for Evaluation View	3		3		
			#194 Product view - Update products	3		3		
			#189 Create Question for Komplexitäts- and wirtschaftliche Bewertung	2		2		
			#190 Enable all API Request for Ratings in frontend	2		1		
			#191 Product area page - Create Products	2		2		
			#193 Product area page - Show product variants for products	3		2		
			#164 Add Actual Values for Progress Bars in Product Overview	2		2		
11	**Finish Visualisation**	**You can see the graphs derived from inserted data**			**24**		23	
			#206 Add functionality to add variants to already existing products	5		3		
			#204 Change Product constructor so that a new product is linked to all questions	2		3		
			#113 Make the Basic Layout of the Evaluation View to a Functional Site	1		1		
			#119 Make the Functional Site of the Evaluation View Dynamic and Connect it to Backend	1		1		
			#87 Fix auto Id counter in db	3		2		
			#210 Restrict access to backend to frontend msgs (CORS issue)	2		1		
			#209 Refactore API Definiton in yaml File	1		1		
			#218 Annotate figures	3		3		
			#208 Finish connecting figures to backend	3		5		
			#205 Redesign product area	3		3		
12	**Fine Tuning & Preparation for Demo Day**	**Prepare the Demo Day; Fix last bugs and do leftovers so that the video can get recorded**			**27**		21	
			#124 Prepare Demo Day Video	3		2		
			#222 Fill final mock data in backend for video	2		1		
			#221 Fix NaN sources bug	3		3		
			#225 Interconnect pages	2		1		
			#223 Input field in rating page updates after every key hit	3		3		
			#220 Fix spelling mistakes in frontend	2		1		
			#224 Delete category column in evaluation page	1		0		

#	Theme	Goal	Feature Name	Est. Size (Feature)	Est. Size (Sprint)	Real Size (Feature)	Real Size (Sprint)	Burn-Down
			#227 The great bug hunt	8		8		
			#127 GitHub Clean Up	2		1		
			#126 Prepare Demo Day Slides	1		1		
13	**Code Documentation & Project Documentation**	**Finishing the project in a clean & documented state**			**23**		23	
			#192 Write Integration test	5		5		
			#219 Finish Frontend Docker Container	2		2		
			#235 Finish Final Project Release Plan	1		1		
			#245 Finalize user documentation	3		2		
			#246 Finalize technical design document	3		2		
			#180 Write comments for test folder (backend)	3		5		
			#107 Write Java Doc for frontend	3		3		
			#248 Product Area Bug	3		3		
	Release	**Mid Term**						
	No. Sprints	6						
	Due Date	02.12.2021						

	Sprint	Sprint Theme	User Stories	Est. Size	Est. Burndown	Real Size	Real Burndown	
	0				106		83	
	1	**Initial Setup**	3,2,1	10	96	10	73	
	2	**More Initial Setups & Research**	12, 13, 17, 14, 15, 11	24	72	18	55	
	3	**Frontend and Backend Functionalities & Architecture**	36, 34, 30, 32, 41, 42, 40	18	54	12	43	
	4	**First Frontend Technical Breakthrough**	37, 38, 59, 44, 46	9	45	9	34	
	5	**Main Frontend and Backend Functionalitites**	45, 47, 48, 61, 63, 64	24	21	18	16	
	6	**Mid Term Release & Testing**	65, 62, 79, 81, 83, 82	21	0	16	0	
	Total			**106**		**83**		

	Release	**Final Project**
	No Sprints	8
	Due Date	10.02.2022

	Sprint	Sprint Theme	User Stories	Est. Size	Est. Burndown	Real Size	Real Burndown	
					220		181	
	7	**Refactoring, Tests & Refinement**	101, 84, 99, 131, 102, 130, 141	31	189	23	158	
	8	**Product Ratings**	95, 97, 103, 104, 142, 148, 149, 144,	46	143	40	118	
	9	**Evaluation of results**	109, 176, 116, 84, 166, 168, 167, 80, 174, 171, 106, 169, 105, 173, 172	40	103	28	90	
	10	**Start Visualisation of Results**	115, 108, 195, 112, 111, 194, 189, 190, 191, 193, 164	29	74	23	67	
	11	**Finish Visualisation**	206, 204, 113. 119, 87, 210, 209, 218, 208, 205	24	50	23	44	
	12	**Fine Tuning & Preparation for Demo Day**	124, 222, 221, 226, 223, 220, 224, 227, 127, 126	27	23	21	23	
	13	**Code Documentation & Project Documentation**	192, 219, 235, 245, 246,180, 107, 248	23	0	23	0	
	Total			**220**		**181**		



Sprint	Status	Source	Impediment	Resolution
1	Resolved	Maximilian R	Not all members could use the happiness tool	All members posted a screenshot of the happiness tool to Slackto verify the properly used the hapiness tool
1	Resolved	Maximilian R	Late appointment with the industry partner (one day before sprint meeting because of their vacation)	Early communication because they are not on vacation anymore
1	Resolved	Andreas Ronellenfitsch	Chaotic settling-in phase	Defined a working process
1	Resolved	Andreas Ronellenfitsch	Use of designated Slack servers	Definied a Slack server for each topic
2	Resolved	Maximilian R	The feature creation process was a little unstructured	Feature creation process at least 3 days before group meeting
3	Resolved	Maximilian R	External requirements regarding the exact definition of "product" & "product area" was unclear and blocked some tasks	Meeting with Theo on 10.11.2021 solved those terminologies (Glossary)
4	Resolved	Tobi	Code review proces is currently unclear	Was resolved by scrum implicitly
4	Resolved	Ines	Clarification needed on product area definition	See answer from Andreas in slack
4	Resolved	Maximilian R	1 Software Developer (Frontend) dropped out	Splitting the team in fixed frontend and backend teams in the next meeting (18.11)
6	Resolved	Patrick G.	Tickets could not be finished	- Ivan dropped out -> POs support with smaller development tasks - unforseen complexity as well as sickness of a teammembers - watch for one more week and maybe split tickets into frontend/backend if it's not resolved
6	Resolved	Team	Complexity of frontend tickets still uncertain	- Split frontend tickets further (functional -> fancy|different components) - Max is going to support front-end from now on
6	Resolved	Team	Communication between front- and backend	- additional (optional) problem specific meetings to solve issues on monday
7	Resolved	Ines	Changes were not communicated and led to merge conflicts	- Communication before changing something (frontend/backend)
8	Resolved	Max B	Product Glossary is unclead and definitions are too long	- Reworken & shorten the product glossary with at least two persons
8	Resolved	Alex/Tobi	API-endpoint definiton is/was not clear	- Plan Meeting between front- and backend to get a clear definition
9	Resolved	Max R./Tobi	Some issues are based on to old code/ exists to long	- Create smaller issues/ perhaps update old isssues
10	Resolved	Tobi	CORS Config blocked requests to backend	- Make ticket to adress CORS config
11	Resolved	Lily	Some Bugs occured	- Create adressed tickets/ -Click through software and "find" those bugs (manual testing)
12	Resolved	Alex	No overview over all dependecies in backend	-Create ticket or is part of documentation tickets
12	Resolved	Andreas Ronellenfitsch	Bug while creating project	- Create ticket to solve that
12	Resolved	Max R/Max B	Happiness Index does work sometimes	- Won't fix, write email to Riehle

#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done
	Approved by product owner	Approved by product owner	Approved by product owner
	Tests have been written (e.g. Unit test, integration test etc..)	100% tests passed (Backend)	No critical bugs
	Code has been peer reviewed and approved	No critical bugs	User documentation is written
	No syntax or runtime errors emerged	Merge conflicts solved	75% test coverage (backend)
	Code has to be included in the release candidate	Demo approved by team	Demo approved by team
		Code is deployable using docker	Code is deployable using docker
		Code in a clean state	Approved by business partner
			Code documentation is written
			Build & deployment documentation is written
			Documentation has been peer reviewed

Type	Link / reference
Wiki	https://github.com/amosproj/amos2021ws05-fin-prod-port-quick-check/wiki
Build & Deployment Documentation	https://github.com/amosproj/amos2021ws05-fin-prod-port-quick-check/wiki/Build-&-Deployment-Documentation
Technical Design Documentation	https://github.com/amosproj/amos2021ws05-fin-prod-port-quick-check/wiki/Technical-Design-Documentation
User Documentation	https://github.com/amosproj/amos2021ws05-fin-prod-port-quick-check/wiki/User-Documentation

\	Context	Name	Version	License	Comment
1	Backend framework	Spring (Boot)	2.5.6	Apache License, Version 2.0	Used for REST API and microservice architecture
2	Backend JavaScript Runtime Environment	Node.js	14.16.0	MIT	https://nodejs.org/en/
3	Backend orchestrator	Docker-Compose	1.29.1	Apache License 2.0	For defining and running multi-container Docker applications
4	Backend programming language	Java	17	Oracle JDK	
5	Backend runtime	Docker Engine	20.10.06	Apache License 2.0	Docker engine is used to run applications in a container
6	Backend Testing	JUnit	5	Eclipse Public License	Unit testing framework for the Java programming language
7	Database	HyperSQL	2.6.2	BSD License	Deployment of database applications
8	Deployment	Github Actions		-	CI/CD Tool for running automated tests, building docker images and deployment.
9	Deployment	Docker CLI	20.10.06	Apache License 2.0	Tool to build containerized applications as docker images.
10	Frontend framework	React	17.0.2	MIT License	UI Framework and handler for the view layer of web and mobile apps. It is a Frontend JavaScript library.
11	Frontend programming language	HTML	5	GNU Project License	Markup language for documents designed to be displayed in a web browser.
12	Frontend programming language	CSS	2.1		Style sheet language used for describing the presentation of a document
13	Frontend programming language	JavaScript	ES2015	GNU Project License	Programming language for Web development
14	Java Runtime	OpenJDK	11.0.11	GNU GPL v2	with linking exception

Last Name	First Name	Value					
Bartels	**Maximilian**	5		**5.00**	**OK**		
Bernhard	**Tobias**	5					
~~Butron Sossa~~	**~~Ivan Antonio~~**	5					
Hügerich	**Lily**	5					
Rodiek	**Maximilian**	5		0	No size		
Ronellenfitsch	**Andreas**	5		1	Trivial size		
Vogel	**Ines**	5		2	Small size		
Werner	**Alexander**	5		3	Medium size		
				5	Large size		
				8	Very large size		
				13	Too large (size)		