Project Name	amos2022ss02-audit-chain
Online team meeting	https://fau.zoom.us/j/63420352046?pwd=TElTeFZoTDFXNVRYUWVCR1ZpQmVPdz09
Production system (if any)	...
Test system (if any)	...
GitHub repository	https://github.com/amosproj/amos2022ss02-audit-chain
GitHub kanban board (project)	https://github.com/amosproj/amos2022ss02-audit-chain/projects
Team T-shirt (white)	https://www.shirtinator.de/loadBasket/1CxTZ1RYcrz
Team T-shirt (black)	https://www.shirtinator.de/loadBasket/1CxTZ1RYcrz
Additional materials	https://drive.google.com/drive/folders/1Hf5l4I0L0jxXnNfANrMft7RYdwUBlvr4?usp=sharing

Last Name	First Name	GitHub User Name	Email Address
Schmidt	David	Lavicola	david.dav.schmidt@fau.de
Shanabhag	Gajanana	gdshanbhag	gajanana.shanabhag@fau.de
Mazzini	Francesco	francescomazzini	francesco.mazzini@fau.de
Linkies	Sebastian	jaRulez	sebastian.linkies@fau.de
Papadaki	Anastasia	annipap	anastasia.p.papadaki@fau.de
Rehm	Ronja	ronjarehm	ronja.rehm@fau.de
Khalid	Muhammad Ibrahim	ibs337	ibrahim.khalid@fau.de
Srikhaolan	Charinee	CharineeSrikhaolan	Charinee.Srikhaolan@fau.de
D'Ercoli	Chiara	cdercoli	chiara.dercoli@fau.de

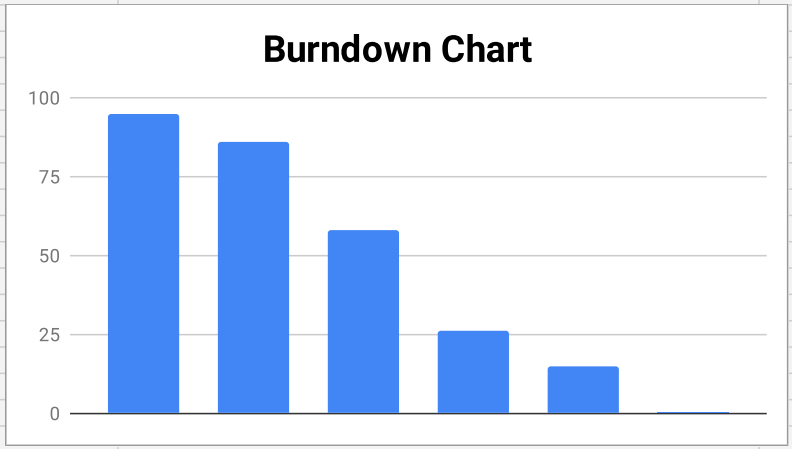
Goals	Learning objectives: Gaining knowledge about agile methods and continous improvement of required skill set (i.e. coding). Moreover, interpersonal relationships are also an important objective. To meet all team members with respect and working a focused but pleasure working atmosphere. Finally, we aim to achieve our defined project goals and work closely and efficient with our industry partner.
Meeting norms	Mandatory team meeting time is on wednesday at 12.30 for our sprint session, which takes 90 minutes. Additionally, we will meet weekly our industry partner to discuss our progress and requirements and meeting eventually a second time per week if required.
Working norms	Team members should discuss objectively and decisions have to be made unanimously. Overall attendance have to above > 80% and team members have to be punctual (not later than 5 minutes!). Criticism should be formulated in a constructive manner and and in case of different opinions, a compriss have to be find. We stick to our definded coding guidelines, our sprint sessions are conducted regarding to the SCRUM rules and in case of major changes in code we will notify all software developers. If team member face troubles with each others, the team is expected to support the individuals and find a solution, which fit to their opinions.
Coordination norms	The Scrum Master has the role of the moderator and support the team in lead us through the agenda of our meetings, the roles of Scrum, approach interpersonal problems and help us to deliver required artifacts on time. The Release manager is responsible for the technical part of every sprint - the software developers agree on the Release manager every sprint.
Communication norms	The sprint sessions on wednesday will be held in Zoom. Apart from our regular meeting, the team communicates via Discord, which has to be checked regularly - the response time should be within 24 hours. Personal communication, in case of urgent matters, is always be possible. Our internal files platform will be a Google Drive folder.
Consideration norms	In case of disagreements the team discusses openly but objectively, all opinions are welcome. The majority of votes has to be made upon the impact of the decision and the urgency, in case of not achieving a compromiss the Scrum master needs to be contacted.
Cont. improvement norms	The team progress is tracked how efficiently the team breaks down problems into tasks and if the team is able to solve all issues in the sprint session. SD's giving each other construtive feedback, if there is a more efficient solution or coding style. In case of good team collaboration (Happiness index) and productive working atmosphere we will go out for a beer.
Rewards	Going out for a beer and making each other compliments.
Sanctions	Recurring laxity in commitment of a team member have to be discussed in the team and the Scrum master tries to motivate the team member. In case of continous decline, we contact the assistants of Prof. Riehle. For acceptable excuses, it is encouraged to notify beforehand any problems are faced.
General Norms	Always be kind, respect each other and discuss problems openly. Ask for if assistance of Prof. Riehle if neccessary.
Signature	Sebastian Linkies, Ronja Rehm, David Schmidt, Gajanana Shanabhag, Francesco Mazzini, Anastasia Papadaki, Ibrahim Khalid, Khaled Saifullah, Charinee Srikhaolan

#	Meeting Day	Uni	Comment	Product Owner	Software Developer	Release Manager	Scrum Master
1	2022-04-27			Ronja Rehm Sebastian Linkies	Everyone else	N/A	Charinee Srikhaolan
2	2022-05-04			Ronja Rehm Sebastian Linkies	Everyone else	Anastasia Papadaki	Charinee Srikhaolan
3	2022-05-11	Yes		Ronja Rehm Sebastian Linkies	Everyone else	Gajanana Shanabhag	Charinee Srikhaolan
4	2022-05-18			Ronja Rehm Sebastian Linkies	Everyone else	Francesco Mazzini	Charinee Srikhaolan
5	2022-05-25	Yes		Ronja Rehm Sebastian Linkies	Everyone else	Francesco Mazzini	Charinee Srikhaolan
6	2022-06-01			Ronja Rehm Sebastian Linkies	Everyone else	Gajanana Shanabhag	Charinee Srikhaolan
7	2022-06-08	Yes	Mid-term due	Ronja Rehm Sebastian Linkies	Everyone else	Anastasia Papadaki	Charinee Srikhaolan
8	2022-06-15			Ronja Rehm Sebastian Linkies	Everyone else	Muhammad Ibrahim Khalid	Charinee Srikhaolan
9	2022-06-22			Ronja Rehm Sebastian Linkies	Everyone else	Francesco Mazzini	Charinee Srikhaolan
10	2022-06-29	Yes		Ronja Rehm Sebastian Linkies	Everyone else	Chiara D'Ercoli	Charinee Srikhaolan
11	2022-07-06			Ronja Rehm Sebastian Linkies	Everyone else	Anastasia Papadaki	Charinee Srikhaolan
12	2022-07-13			Ronja Rehm Sebastian Linkies	Everyone else	Chiara D'Ercoli	Charinee Srikhaolan
13	2022-07-20	Yes		Ronja Rehm Sebastian Linkies	Everyone else	Anastasia Papadaki	Charinee Srikhaolan
14	2022-07-27		Demo day!	Ronja Rehm Sebastian Linkies	Everyone else	Anastasia Papadaki	Charinee Srikhaolan
15	2022-08-03		Retrospective	Ronja Rehm Sebastian Linkies	Everyone else	Anastasia Papadaki	Charinee Srikhaolan

Product Vision	Project Mission
The long-term goal of our project is to develop a new middleware based on blockchain data structures to guarantee the unchanged, compliant, in sequence, fault tolerant and buffered data flow between any kind of producers and consumers. Events of all kind (i.e. IoT or file systems) are meant to be transmitted securely via the network to enhance the security for end users through tamper-proof events that can be transmitted without any losses.	The mission is to deliver a MVP of a audit-proof recording of file system events to Grau Data. For this purpose there is a central event queue, which receives events from a producer and provides them a consumer, which is built on a underlying blockckain data structure. Finally, the event will be copied in a second consumer and will be deleted from the event queue.

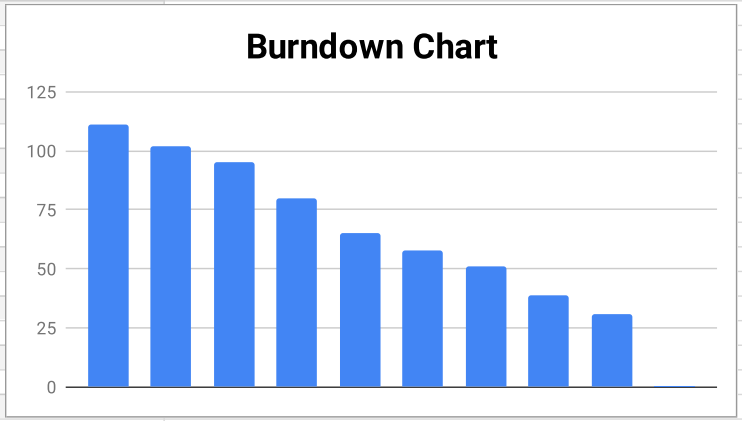
#	Theme	Goal	Feature Name	Est. Size (Feature)	Est. Size (Sprint)	Real Size (Feature)	Real Size (Sprint)	Burn-Down
	Total				92		95	95
1	**Initial Setup**				9		9	86
		Conductive administrative tasks and initiate project						
			Preparing Team Contract	2		2		
			Meeting with Industry Partner	1		1		
			Design Team Logo and T-Shirt	1		1		
			Organizing Release managers and discussing programming experience	1		1		
			Discussing communication strategy and preferred programming language	2		2		
			Create branch strategy in GitHub	2		2		
2	**Fundamentals**				26		28	58
		Programming prototypes of software components						
			Creating Software Architecture	3		5		
			Programm Consumer Prototype	5		5		
			Programm Producer Dummy in Java	3		3		
			Programm Producer Dummy in Python	3		3		
			Setup RabbitMQ as Event Queue	3		3		
			Research of suitable Blockchain technology	8		8		
			Provide Bill of Materials	1		1		
3	**Refinements**				31		32	26
		Evaluating message flow and refinement of software components						
			Facilitate to aggregate messages to one data package	3		5		
			Integrating Blockchain data structure in Consumer Dummy	5		5		
			Recovering of missing data in Producer Dummy	5		3		
			Evaluate secure data sources for Producer Dummy	2		3		
			Revising Blockchain	3		3		
			Check for message integrity	5		5		
			Creating message acknowledgement for Producer Dummy	8		8		
4	**Testing**							
		Creating Unit Tests for software components			11		11	15
			JUnit Test for Producer Dummy	5		5		
			JUnit Test for Consumer Dummy	3		3		
			JUnit Test for Blockchain	3		3		
5	**Documentation**							
		Creating documentation for software components			15		15	0
			Create documentation for repository	8		8		
			Create documentation for Producer Dummy	2		2		
			Create documentation for Consumer Dummy	2		2		
			Create documentation for Blockchain	3		3		

#	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
	Total				119		111	111
6	**Refactoring**							
		Refactoring Project code			11		9	102
			Refactoring code for Producer Dummy	5		3		
			Refactoring code for Consumer Dummy	2		2		
			Refactoring code for Blockchain	2		2		
			Fix bugs on RabbitMQ	2		2		
7	**Testing**							
		Testing Project Code			7		7	95
			Improving Test Coverage Consumer Dummy	2		2		
			JUnit Test Producer Dummy	5		5		
8	**GUI**				16		15	80
		implement GUI	Design GUI	5		5		
			Logic for GUI	5		5		
			Implement methods for the GUI client	3		2		
			GUI itself	3		3		
9	**Blockchain**							
		extend fundamental Blockchain			12		15	65
			persistate Blockchain in files	5		5		
			change storage behavior of Blockchain	5		5		
			optimize Blockchain search time	2		5		
10	**Consumer Dummy**							
		extend fundamental Consumer Dumm			12		7	58
			Extend Consumer Dummy: Acknowledgement	2		2		
			Revise Consumer Dummy with Blockchain	5		3		
			Write Recovery Method/Logic for (Stream)Consumer	5		2		
11	**RabbitMQ**							
		implement functions			8		7	51
			Check for data replication in RabbitMQ	3		2		
			Checking RabbitMQ message flow	5		5		
12	**Whole System**				16		12	39
		improve the code by unifying	Look over code and check consistency	3		3		
			Concurrent Communication Consumer Dummy with RabbitMQ and GUI	5		2		
			Write method for Consumer Dummy to Blockchain	2		1		
			Parameterize Main Consumer/Producer/Blockchain	5		5		
			Final Testing	1		1		
13	**Producer Dummy**							
		add features			5		8	31
			creating dynamic messages with data generator	2		3		
			Analyse and Refactor AggregateMessage: as Vektor maybe	3		5		
14	**Documentation**	update documentation			32		31	0

#	Theme	Goal	Feature Name	Est. Size (Feature)	Est. Size (Sprint)	Real Size (Feature)	Real Size (Sprint)	Burn-Down
			Update Documentation	1		2		
			Add documenation for User	3		1		
			Update Documentation	2		2		
			Create JAR for execution purpose	2		1		
			Update Github Repo + Description	2		2		
			Discuss and create One-Slide-Summary & presentation for Demo Day	5		5		
			Update and check all the documentations	8		8		
			Demo Day Preparation slides & video	8		8		
			Update Software Architecture Doc	1		1		



Term	Definition
Agile	Teams work in agile iterative cycles, or sprints, and then reassemble frequently to review and adjust their work. By using agile methodologies, you are encouraged to receive frequent feedback and are able to change priorities quickly.
Acceptance Criteria	Acceptance Criteria gives us a clear understanding of what our client needs, so we can reference this against every iteration we deploy.
Awaiting Review	It means when one task is finished and the responsible person will show its work during the meeting. After the presentation, the person must say if the estimated size of the task (that was decided during Planning Poker) is the same or not.
Blockchain	Blockchain is a system of recording information in a way that makes it difficult or impossible to change, hack, or cheat the system.
Bill of materials	A bill of materials (BOM) is a complete list of the materials needed to build a product.
Consummer Dummy	It is a program which "communicates" with the rabbitMQ (Middleware) in order to receive messages
Definition of Done (DoD)	The Definition of Done describes the list of requirements that the team agrees must be met to consider a user story or other backlog item complete.
Docker Engine	Docker is for isolation of an application using containers.
Docker CLI	It is the Command Line Interface using Docker.
Documentation	For a software company, documentation refers to information either embedded in the product or published documentation. It describes what the app does, how it works, and other essential details.
Feature Board	It helps organised our work by creating new futures, checking which are done or not.
Feature	Features are a product’s traits or attributes that deliver value to end-users and differentiate a product in the market.
Git	Git is a distributed version control system with the aim to help our group to develop the software together.
GUI	A graphics-based operating system interface that uses icons, menus and a mouse (to click on the icon or pull down the menus) to manage interaction with the system.
Impediments Backlog	It is a simple list of things, actions and impediments that cause waste in the organization.
In progress	refers to the work items the Scrum Team has started but has not yet. finished
Java	Java is an object-oriented programming language which consists of a development tool for creating code and a runtime environment to run code
Maven	We use Maven for having a report of the test coverage and aplugin to the maven project of the producer dummy to auto generate the javadoc
Planning Poker	Planning poker (also called Scrum poker) helps agile teams estimate the time and effort needed to complete each initiative on their product backlog. The name from this gamified technique is planning poker because participants use physical cards.
Product Backlog	A product backlog is a prioritized list of work for the development team that is derived from the roadmap and its requirements. The most important items are shown at the top of the product backlog so the team knows what to deliver first.
Producer dummy	It operates through the dependencies among its 3 sub-components database, data generator, and persistence mechanism (storage buffer). When the producer dummy is triggered, the data generator catches data events (message) from the database and then forwards it to the client session and persistence mechanism.
Producer owner	The role of a product owner has taken on many different and conflicting definitions.
Project Manager	A project manager is responsible for executing an initiative. Initially, they are responsible for ideation. It is then their responsibility to coordinate components and budgets. The final step is to manage these components until they are delivered. However, PMs aren't involved in choosing which projects to pursue or they do not directly manage most of the resources involved in execution. However, they are ultimately responsible for implementing and delivering the project successfully.
RabbitMQ (Event Queue)	An open source message broker software.
Release Plan	A release plan is a tactical document designed to capture and track the features planned for an upcoming release.

Term	Definition
Retrospective	A retrospective is a meeting held after a product ships to discuss what happened during the product development and release process, with the goal of improving things in the future based on those learnings and conversations.
Scrum	*It is a framework used within Agile software development that emphasizes flexibility and gives engineers and developers a great deal of autonomy in solving problems without rigorous instruction.*
Scrum Agile Framework	Typically the Scrum agile framework favors moving projects forward via short-term blocks of work called sprints, which are usually confined to two-week intervals.
Sprint	A sprint is a period (in our case was 7 days) in which an agreed-upon set of development tasks takes place.
Sprint Backlog	A sprint backlog is the set of items that a cross-functional product team selects from its product backlog to work on during the upcoming sprint.
Sprint Goal	In the scrum methodology for agile, sprint goals are clear objectives set before the beginning of a sprint. They are set by the product owner and delivery team collaboratively. Sprint goals should be easy to measure and should convey the underlying objective of the items in the sprint backlog.

#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done
1	Feature Code Review has been completed and passed	Feature Code Review has been completed and passed	Feature Code Review has been completed and passed
	a) Code is completly implemented	*a) Code is completly implemented*	*a) Code is completly implemented*
	b) Code is structured according to our coding guidelines and commented	*b) Code is structured according to our coding guidelines and commented*	*b) Code is structured according to our coding guidelines and commented*
	c) Code is checked into repository	*c) Code is checked into repository*	*c) Code is checked into repository*
	d) Documentation is updated	*d) Documentation is updated*	*d) Documentation is updated*
2	Cleanliness of Code	Cleanliness of Code	Cleanliness of Code
3	JUnit Tests have been written and passed (if required)	JUnit Tests have been written and passed (if required)	JUnit Tests have been written and passed (if required)
4	No critical bugs are open	All known bugs are fixed	All known bugs are fixed
5	Feature branch has been tagged and merged	Code has been inclued into the release (candidate)	Software prototype passes external review
6	Feature Code has been included into the release (candidate)		User documentation passes external review
7	Product Owners accept Feature		Developer documentation is available
8	Code Coverage:	Code Coverage:	Code Coverage:
	60% for Features	70% for Sprint Release	80% for Product Release

Type	Link / reference
Main repository	https://github.com/amosproj/amos2022ss02-audit-chain
Blockchain	https://github.com/amosproj/amos2022ss02-audit-chain/blob/main/Documentation/Blockchain/Documentation.md
Branching strategy for Git.docx	https://github.com/amosproj/amos2022ss02-audit-chain/blob/main/Branching_Strategy_Documentation/Branching%20strategy%20for%20(
Deliverables	https://github.com/amosproj/amos2022ss02-audit-chain/tree/main/Deliverables
DocumentationForUser.docx	https://github.com/amosproj/amos2022ss02-audit-chain/blob/main/Documentation/DocumentationForUser.docx
RabbitMQ installation	https://github.com/amosproj/amos2022ss02-audit-chain/tree/main/Documentation/Middleware
README	https://github.com/amosproj/amos2022ss02-audit-chain/blob/main/README.md

\	Context	Name	Version	License	Comment
	Backend	Python	3.10.4	MIT	software is programmed in python
	Backend	Java	8.0 Update 333	MIT	second Producer Dummy
	Message Broker	RabbitMQ	3.10.0	MIT	implements Advanced Message Queuing Protocol
	Virtualisierung	Docker			Docker Cluster due to effective ressource management implemented
	Virtualisierung	Maven Repository			A directory where all the project jars, library jar, plugins or any other project specific artifacts are stored and can be used by Maven easily.
	type of user interface	GUI			A graphics-based operating system interface that uses icons, menus and a mouse (to click on the icon or pull down the menus) to manage interaction with the system.

Last Name	First Name						
Schmidt	**David**	1		**1.00**	**OK**		
Shanabhag	**Gajanana**	1					
Mazzini	**Francesco**	1					
Papadaki	**Anastasia**	1					
D'Ercoli	**Chiara**	1		0	No size		
Khalid	**Muhammad Ibrahim**	1		1	Trivial size		
				2	Small size		
				3	Medium size		
				5	Large size		
				8	Very large size		
				13	Too large (size)		