Project Name	Kubernetes Inventory Taker
Online team meeting	https://fau.zoom.us/j/66067362855
Production system (if any)	not available
Test system (if any)	not available
GitHub repository	https://github.com/amosproj/amos2023ss04-kubernetes-inventory-taker
GitHub feature board	amos2023ss04-feature-board (github.com)
GitHub impediments backlog	amos2023ss04-impediments-backlog (github.com)
Team T-shirt (white, male)	https://www.shirtinator.de/t-shirts/gestalten/t-shirt-bedrucken#/load/share/87425b24-a18c-4528-b35d-81df816ca277
Team T-shirt (black, male)	https://www.shirtinator.de/t-shirts/gestalten/t-shirt-bedrucken#/load/share/42541114-1bf4-4ca6-89c2-2040abfcfe20
Team T-shirt (white, female)	https://www.shirtinator.de/t-shirts/gestalten/t-shirt-bedrucken#/load/share/05d839cc-d89e-4063-a1e2-a2ddb9deff98
Team T-shirt (black, female)	https://www.shirtinator.de/t-shirts/gestalten/t-shirt-bedrucken#/load/share/077e16f6-e925-4331-9371-fb165b676607
Additional materials	...

Last Name	First Name	GitHub User Name	Email Address	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Degen	Jan	jandegen	jan.degen@fau.de							
Kramer	Philipp	PhlppKrmr	philipp.kramer@fau.de							
Zabka	Stefan	vringar	zabka@campus.tu-berlin.de	Unavailable	Unavailable	Actively Working	Actively Working			
Jochens	Nikolas	njochens	nj@andaco.de							
Ali	Muhammad Fahad	muhdfahadali	mfahad.ali@fau.de	Actively Working	Actively Working	Actively Working	Unavailable	Unavailable		
Gehrlein	Lucas	lifesbest23	l.gehrlein@campus.tu-berlin.de	Actively Working				Unavailable		Actively Working
Wiesend	Simon	smnws	simon.wiesend@fau.de							
Kwiatek	Anna	IceFlovver	anna.kwiatek@fau.de							
Aziri	Ali	freefreep	ali.aziri@campus.tu-berlin.de							

#	Meeting Day	Product Owner	Software Developer	Release Manager	Scrum Master	Comment
1	2023-04-12	Philipp Kramer, Jan Degen	Everyone else	N/A	Anna Kwiatek	
2	2023-04-19	Philipp Kramer, Jan Degen	Everyone else	N/A	Anna Kwiatek	
3	2023-04-26	Philipp Kramer, Jan Degen	Everyone else	Lucas Gehrlein	Anna Kwiatek	
4	2023-05-03	Philipp Kramer, Jan Degen	Everyone else	Stefan Zabka	Anna Kwiatek	
5	2023-05-10	Philipp Kramer, Jan Degen	Everyone else	Stefan Zabka	Anna Kwiatek	
6	2023-05-17	Philipp Kramer, Jan Degen	Everyone else	Ali Aziri	Anna Kwiatek	
7	2023-05-24	Philipp Kramer, Jan Degen	Everyone else	Nikolas Jochens	Anna Kwiatek	Mid-term due
8	2023-05-31	Philipp Kramer, Jan Degen	Everyone else	Nikolas Jochens	Anna Kwiatek	
9	2023-06-07	Philipp Kramer, Jan Degen	Everyone else	Fahad Ali	Anna Kwiatek	
10	2023-06-14	Philipp Kramer, Jan Degen	Everyone else	Simon Wiesend	Anna Kwiatek	
11	2023-06-21	Philipp Kramer, Jan Degen	Everyone else	Lucas Gehrlein	Anna Kwiatek	
12	2023-06-28	Philipp Kramer, Jan Degen	Everyone else	Simon Wiesend	Anna Kwiatek	
13	2023-07-05	Philipp Kramer, Jan Degen	Everyone else	Fahad Ali	Anna Kwiatek	
14	2023-07-12	Philipp Kramer, Jan Degen	Everyone else	Stefan Zabka	Anna Kwiatek	Demo day!
15	2023-07-19	Philipp Kramer, Jan Degen	Everyone else	tbd.	Anna Kwiatek	Retrospective

Goals	- Get in touch with Kubernetes, Docker - Everyone feels comfortable working with it - Earning something - Deliver working application - Satisfy the client - We succeed as a team or we fail as a team
Meeting norms	- Create meeting protocol, maintainer is the current release manager - Zoom for Teammeeting 12:30 - Dont use Zoom Chat, instead use Mattermost channel Teammeeting - Stay within meeting schedule, dont extend meetings heavily
Working norms	- Help each other - Main branch is runnable - No direct pushes to main - Code is reviewed by other software developers - Everyone picks up open active topics proactivly - Every developer has to do at least 1 story point per sprint
Coordination norms	- Decline meetings if you're not able to joiin - Inform team if you're late - Use comments on GitHub issues for coordination on user stories
Communication norms	- Mattermost for team internl communication - We interact polity and respectfully with each other - Everyone can always speak up and state their mind - In case of problems talk to each other - Cameras are always on with happy faces
Consideration norms	- Less tasks for people with exams - Majority vote for hard decisions - Every opinion is weighted equally
Cont. improvement norms	- Perform retrospective after sprints - Talk openly about what went wrong respectful manner
Rewards	- Good sprints give research budgets which is spent on reasearch project related stuff
Sanctions	- Sad smileys for late people
Signature	Jan Degen

	Stefan Zabka
	Muhammad Fahad Ali
	Lucas Gehrlein
	Nikolas Jochens
	Simon Wiesend
	Philipp Kramer
	Anna Kwiatek

Product Vision	Project Mission
Our vision is to create a tool that empowers DevOps teams to manage their Kubernetes clusters with ease, providing them with a single platform for inventory visibility, deep insights, real-time analytics, and intelligent monitoring and alerting. With KIT, teams can optimize their resources and streamline their operations, resulting in faster deployment times, better performance, and happier customers.	Product Mission for KIT: Simplifying Kubernetes Management and Monitoring for DevOps Teams At KIT (Kubernetes Inventory Taker), our mission is to simplify Kubernetes management and monitoring for DevOps teams. We want to provide a user-friendly tool that gives you a clear, real-time view of your Kubernetes resources, from deployments and pods to containers, so you can easily manage and optimize your infrastructure. Here's what we're all about: - Inventory Visibility: We'll provide you with a comprehensive view of your Kubernetes inventory, showing you the state, health, and configuration of your deployments, pods, and containers in a single, easy-to-navigate interface. No more complex logs or guesswork - we'll make it simple and intuitive for you to understand your resources. - Insights and Analysis: We'll help you gain deep insights into your Kubernetes resources with powerful analytics and analysis. Get real-time information on resource utilization, health status, and configuration changes, so you can quickly identify and resolve issues, track changes over time, and optimize your resources for better performance. - User-friendly Web Frontend: We believe in making KIT easy to use and visually appealing. Our web frontend is designed to be user-friendly and responsive, with

Term	Definition
KIT	Project Name short for Kubernetes Inventory Taker

Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release							
	Total			79	49	75	
Sprints							
1	Project kick-off	Get in touch with the team, project and AMOS		0	79	0	75
2	Project Setup	Create application architecture and setup project components		21	79	22	75
3	Application Domain Modelling	Model the application domain as well as create drafts for the UI		19	58	17	53
4	Application Domain Model Implementation & CICD architecture	Implement the first components of the application domain and prepare CICD pipeline		23	39	23	36
5	CICD, Testing and UI design			16	16	13	13
Features							
1	**Project kick-off**	Get in touch with the team, project and AMOS					
			Create & agree on team contract				
			Initialize planning documents				
			Fill happiness index				
			Design team logo				
			Submit team T-Shirt size preference				
2	**Project Setup**	Create application architecture and setup project components					
			Initialize base Website	8		8	
			Initialize database	2		1	
			Initialize proxy container	3		5	
			Introduction to used technologies	8		8	

Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
			Create software architecture description				
			Initialize software bill of materials				
3	**Application Domain Modelling**	Model the application domain as well as create drafts for the UI					
			Create wire draft for UI	5		8	
			Design database scheme	8		5	
			Apply database scheme to database	3		1	
			Initialize project docker-compose	3		3	
4	**Application Domain Model Implementation & CICD architecture**	Implement the first components of the application domain and prepare CICD pipeline					
			Create base implementation for resource information collection	5		5	
			Create base container detail page	8		8	
			Create coding guidelines	5		5	
			Decide on CICD architecture	5		5	
			Create product vision				
			Create product mission				
			Create definition of done				
5	**CICD, Testing and UI design**	Setup CICD and testing framework					
			Update wire draft	3		3	
			Adjust database scheme to history approach	5		5	
			Implement CICD	8		5	
			Create mid-release plan				
			Create build process video				

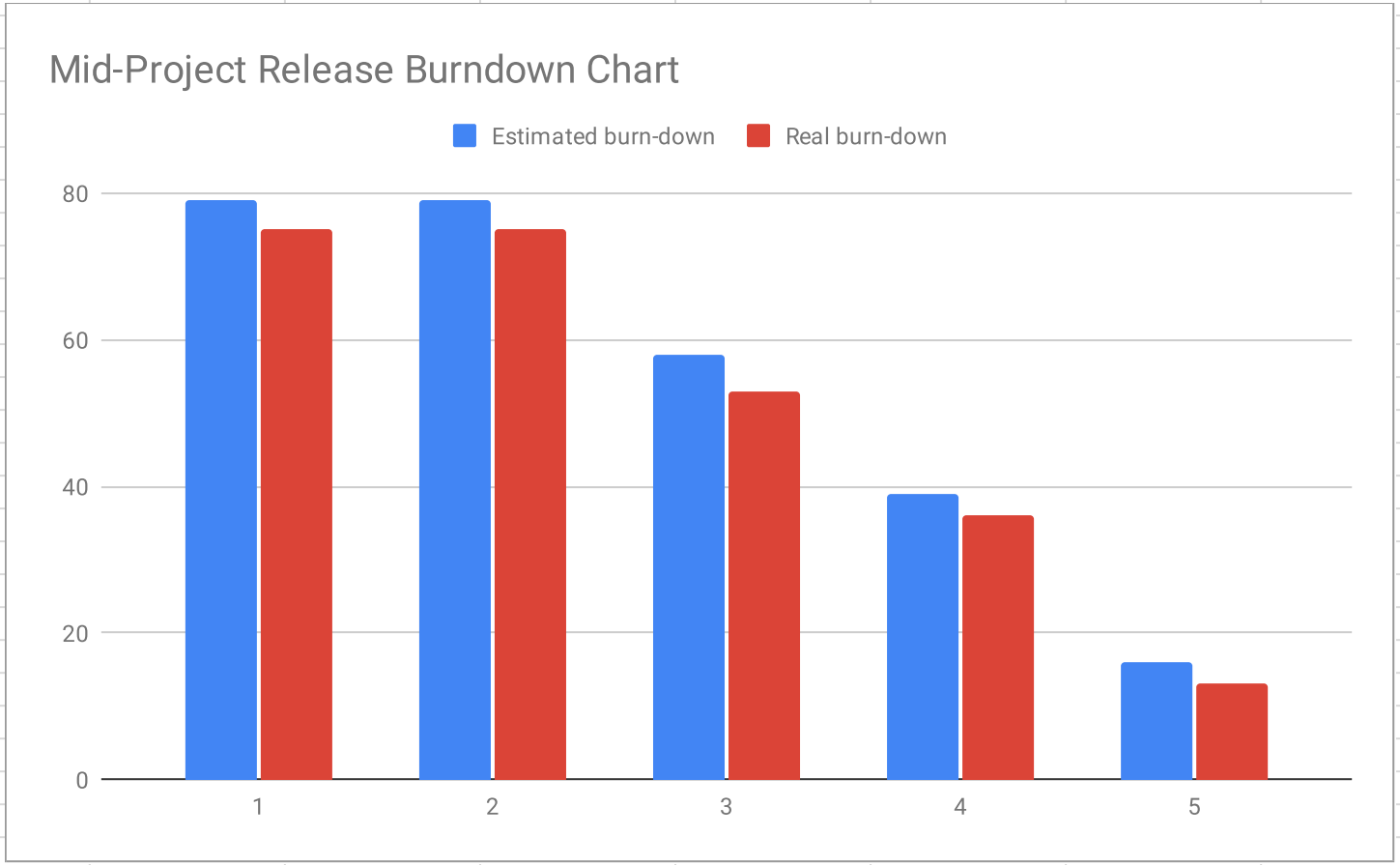
Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release							
	Total			155	49	150	75
Sprints							
1	Project kick-off	Get in touch with the team, project and AMOS		0	155	0	150
2	Project Setup	Create application architecture and setup project components		21	155	22	150
3	Application Domain Modelling	Model the application domain as well as create drafts for the UI		19	134	17	128
4	Application Domain Model Implementation & CICD architecture	Implement the first components of the application domain and prepare CICD pipeline		23	115	23	111
5	CICD, Testing and UI design	Setup CICD and testing framework		16	92	13	88
6	Mid Release	Prepare mid release and documentation documents		18	76	16	75
7	Containers & Navigation	Container list detail views shall be implemented and dynamically loaded from database		0	58	0	59
8	Containers & Navigation 2	Container list detail views shall be implemented and dynamically loaded from database		23	58	25	59
9	Containers	Fine-tune containers in proxy and explorer		17	35	17	34
10	Pods	Introduce Pods to the application		13	18	12	17
11	Pods & Volumes			5	5	5	5
12	Pods & Volumes 2				0		0
13	Final Release				0		0
14	Project Retrospective				0		0
Features							
1	**Project kick-off**	Get in touch with the team, project and AMOS					
			Create & agree on team contract				

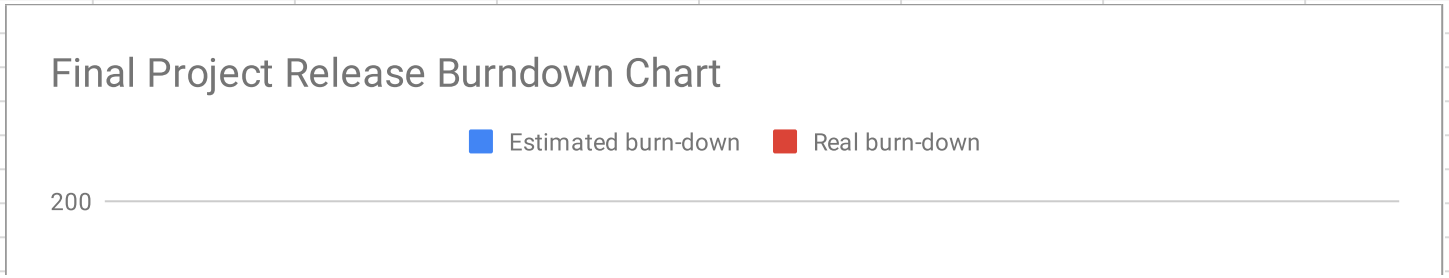
Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
			Initialize planning documents				
			Fill happiness index				
			Design team logo				
			Submit team T-Shirt size preference				
2	**Project Setup**	Create application architecture and setup project components					
			Initialize base Website	8		8	
			Initialize database	2		1	
			Initialize proxy container	3		5	
			Introduction to used technologies	8		8	
			Create software architecture description				
			Initialize software bill of materials				
3	**Application Domain Modelling**	Model the application domain as well as create drafts for the UI					
			Create wire draft for UI	5		8	
			Design database scheme	8		5	
			Apply database scheme to database	3		1	
			Initialize project docker-compose	3		3	
4	**Application Domain Model Implementation & CICD architecture**	Implement the first components of the application domain and prepare CICD pipeline					
			Create base implementation for resource information collection	5		5	
			Create base container detail page	8		8	
			Create coding guidelines	5		5	
			Decide on CICD architecture	5		5	
			Create product vision				
			Create product mission				
			Create definition of done				
5	**CICD, Testing and UI design**	Setup CICD and testing framework					
			Update wire draft	3		3	

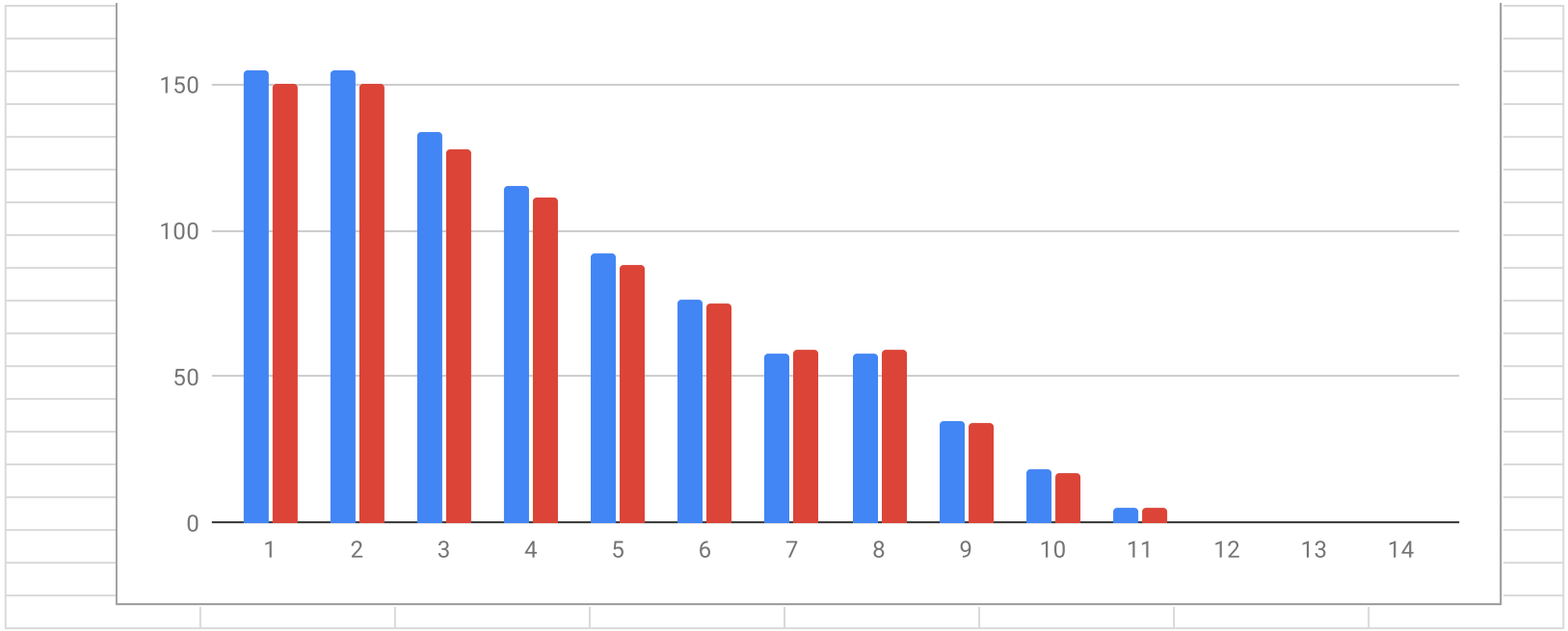
Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
			Adjust database scheme to history approach	5		5	
			Implement CICD	8		5	
			Create mid-release plan				
			Create build process video				
6	**Mid Release**	Prepare mid release and documentation documents					
			Setup frontend testing	5		5	
			Create health indicator	3		3	
			Generate test data for container & pods within database	5		3	
			Create list with static data (container names)	5		5	
			Create user documentation				
			Create build & deploy documentation				
			Create technical design documentation				
7	**Containers & Navigation**	Container list detail views shall be implemented and dynamically loaded from database		0		0	
8	**Containers & Navigation 2**	Container list detail views shall be implemented and dynamically loaded from database					
			Update database schema	5		5	
			Track changes with watchers	8		13	
			Display all containers as list	8		5	
			GHA and Pre-Commit use different configs for golangcilint	2		2	
9	**Containers**	Fine-tune containers in proxy and explorer					
			Collect metadate about containers	5		5	
			Display health state in list views	5		5	
			Rename tables to lower case	1		1	
			Create sidebar navigation	5		5	
			Make the container detail page accessible from the all containers view	1		1	

Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
10	**Pods**						
			Use /api instead of marking pages as force-dynamic	0		0	
			Search containers by name	5		5	
			Display container metadata in container detail page	5		5	
			Pre-commit Doesn't Pass All Relevant File Paths To Eslint	1		1	
			Install ESLint when building next.js application	2		1	
11	**Pods & Volumes**						
			Container Sorting Functionality Not Working Properly After Implementing Search Feature	2		2	
			Run all tests in CI	3		3	
12	**Pods & Volumes 2**						
			Display all pods as list	3			
			Collect pod information from Kubernetes cluister	3			
			Update health states without user interaction	3			
			Collect persistent volumes from cluster	5			
			Show parent pod of containers	1			
			Display pod metadata in the pod detail page	5			
			Search pods by name	3			
			Research failure stories of Kubernetes	5			
			Create demo day video				
			Create demo day slides				
13	**Final Release**						
			Update test data for demo day	3			
			Finalize user documentation				
			Finalize design documentation				
			Finalize build & deploy documentation				

Sprint	Theme	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
			Clean-up final release plan				
			Create final-project-release tag				


Mid-Project Release Burndown Chart
Estimated burn-down
Real burn-down
80
60
40
20
0
1
2
3
4
5


Final Project Release Burndown Chart
Estimated burn-down
Real burn-down
200


150
100
50
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14

#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done
	Colours match with color scheme	sprint-XX-release-candidate release created and marked as pre-release	All required features are implemented
	Implementation is unit tested	New Docker images available at Docker Container Registry	User manual is ready
	Code documentation updated (jsdoc or godoc?)	Sprint Review completed	Technical manual is ready
	Changes to database documented at Confluence- Database scheme	Sprint Retrospective completed	No bugs
	Changes to architecture documented at Confluence - Architecture	Pull request to "main" branch merged & closed if there is a new release	
		Real size set for each closed user story	
	Feature has been fully implemented		
	All acceptance criteria were met		
	Product owner approved features		
	All tests are passing		
	Pull request to "develop" branch is merged & closed		
	Bill of materials is updated with new dependencies		

Type	Link / reference

#	Context	Name	Version	License	Comment
1	Proxy	Kubernetes Client Library	v0.27.1	Apache-2.0	https://github.com/kubernetes/client-go/
2	Frontend	Types	multiple	MIT	https://github.com/DefinitelyTyped/DefinitelyTyped/
3	Frontend	Next.js	13.3.4	MIT	https://github.com/vercel/next.js
4	Frontend	typescript	5.0.4	Apache-2.0	https://github.com/microsoft/TypeScript
5	Frontend	tailwindcss	3.3.2	MIT	https://github.com/tailwindlabs/tailwindcss
6	Frontend	swr	2.1.5	MIT	https://github.com/vercel/swr
7	Frontend	react	18.2.0	MIT	https://github.com/facebook/react
8	Frontend	postcss	8.4.23	MIT	https://github.com/postcss/postcss
9	Frontend	eslint	8.39.0	MIT	https://github.com/eslint/eslint
10	Frontend	autoprefixer	10.4.14	MIT	https://github.com/postcss/autoprefixer
11	Frontend	cypress	12.13.0	MIT	https://github.com/cypress-io/cypress
12	Frontend	eslint-plugin-cypress			
13	Proxy	klog	v1.0.0	Apache-2.0	https://github.com/kubernetes/klog
14	Proxy	utils	v0.0.0-20230209194617-a36077c30491	Apache-2.0	https://github.com/kubernetes/utils
15	Proxy	apimachinery	v0.27.2	Apache-2.0	https://github.com/kubernetes/apimachinery
16	Proxy	api	v0.27.2	Apache-2.0	https://github.com/kubernetes/api
17	Proxy	go-yaml/yaml.v2	v2.4.0	MIT and Apache-2.0	https://github.com/go-yaml/yaml
18	Proxy	uptrace/bun	v1.1.14	BSD 2-Clause	https://github.com/uptrace/bun
19	Proxy	golang	v1.20	BSD 3-Clause	https://github.com/golang/go
20	Proxy	golangci-lint	v1.53.3	GPL-3.0	https://github.com/golangci/golangci-lint

#	Context	Name	Version	License	Comment

#	Context	Name	Version	License	Comment

#	Context	Name	Version	License	Comment

a	First Name	Value					
Degen	**Jan**			#DIV/0!	#DIV/0!		
Kramer	**Philipp**						
Zabka	**Stefan**						
Jochens	**Nikolas**						
Ali	**Muhammad Fahad**			0	No size		
Gehrlein	**Lucas**			1	Trivial size		
Wiesend	**Simon**			2	Small size		
Kwiatek	**Anna**			3	Medium size		
Aziri	**Ali**			5	Large size		
				8	Very large size		
				13	Too large (size)		