

<	...
Online team meeting	<a href="https://fau.zoom-x.de/j/9776061710?pwd=9BPbcHYQaVEf6L0IH3xbsSeNzajvJ0.1">https://fau.zoom-x.de/j/9776061710?pwd=9BPbcHYQaVEf6L0IH3xbsSeNzajvJ0.1</a>
Production system (if any)	...
Test system (if any)	...
GitHub repository	<a href="https://github.com/amosproj/amos2024ws03-android-zero-instrumentation">https://github.com/amosproj/amos2024ws03-android-zero-instrumentation</a>
GitHub feature board	<a href="https://github.com/orgs/amosproj/projects/72/views/2">https://github.com/orgs/amosproj/projects/72/views/2</a>
GitHub imp-squared backlog	<a href="https://github.com/orgs/amosproj/projects/76">https://github.com/orgs/amosproj/projects/76</a>
Team T-shirt (white)	<a href="https://www.shirtinator.de/s/OaDrwZ0JQ9WL1QrhmOU7KA">https://www.shirtinator.de/s/OaDrwZ0JQ9WL1QrhmOU7KA</a>
Team T-shirt (black)	<a href="https://www.shirtinator.de/s/Ou9CCXOBQIW04aOC_Hov6g">https://www.shirtinator.de/s/Ou9CCXOBQIW04aOC_Hov6g</a>
Additional materials	...
Team mailing list	oss-amos-proj3@lists.fau.de

Last Name	First Name	GitHub User Name	Email Address
Krug	Maximilian	HaruspexSan	krugm03@gmail.com
Ayach	Mohammed Tamim	Tamemo99	Tamemayash@gmail.com / Ayachmoh@hu
Bretting	Luca	luca-dot-sh	luca.bretting@fau.de
Seidl	Robin	mr-kanister	robin.seidl@fau.de (main) / 68117355+Mr-k
Labroussis	Christos	clabrous	c.labroussis1@gmail.com
Hilgers	Felix	fhilgers	felix.hilgers@fau.de
Weissshuhn	Tom	der-whity	tom.weissshuhn@fau.de
Schlicht	Franz	ffranzgitHub	franz.schlicht@fau.de
Nawlo	Ali	alinawlo	ali.nawlo@campus.tu-berlin.de
Zinn	Benedikt	BenediktZinn	benedikt.wh.zinn@gmail.com

#	Meeting Day	Product Owners	Software Developer	Release Manager	Scrum Master	Comment
1	2022-10-16	Mohammed Tamim Ayach	Everyone else		Maximilian Krug	
2	2022-10-23	Ali Nawlo	Everyone else	Maximilian Krug	Maximilian Krug	
3	2022-10-30	Mohammed Tamim Ayach	Everyone else	Benedikt Zinn	Maximilian Krug	
4	2022-11-06	Ali Nawlo	Everyone else	Tom Weißhuhn	Maximilian Krug	
5	2022-11-13	Mohammed Tamim Ayach	Everyone else	Robin Seidl	Maximilian Krug	
6	2022-11-20	Ali Nawlo	Everyone else	Franz Schlicht	Maximilian Krug	
7	2022-11-27	Mohammed Tamim Ayach	Everyone else	Benedikt Zinn	Maximilian Krug	Mid-term due
8	2022-12-04	Ali Nawlo	Everyone else	Robin Seidl	Maximilian Krug	
9	2022-12-11	Mohammed Tamim Ayach	Everyone else	Luca Bretting	Maximilian Krug	
10	2023-01-11	Ali Nawlo	Everyone else		Maximilian Krug	
11	2023-01-18	Mohammed Tamim Ayach	Everyone else		Maximilian Krug	
12	2023-01-25	Ali Nawlo	Everyone else		Maximilian Krug	
13	2023-02-01	Mohammed Tamim Ayach	Everyone else		Maximilian Krug	
14	2023-02-08	Ali Nawlo	Everyone else		Maximilian Krug	Demo day!
15	2023-02-15	Mohammed Tamim Ayach	Everyone else		Maximilian Krug	Retrospective
Product owners, software developers, and Scrum Master are set and ideally don't change over time; the critical part is the Release Manager role you need to define here						

<b>Goals 1</b>	
	Completing the objective and task given by our IP, becoming a well rounded team in the meantime
<b>Meeting norms 2</b>	Be punctual (with a 5min pardon time)
	Max. two times missing from IP meeting
	not having the camera off two consecutive times
<b>Working norms 2</b>	Don't push to main, keep main in working order
	Dependencies are a team effort
	all tests must pass
	criticism via pull/merge requests
<b>Coordination norms 2</b>	PR with one other member
	max keeps meetings on track
<b>Communication norms 2</b>	communication via discord - team meeting via zoom
	document major changes
<b>Consideration norms 2</b>	be respectfull
	small disagreement, discuss and vote
<b>Cont. improvement norms 2</b>	team meeting for tracking team's progress -> standup emails for gathering intel
	pushing non functional changes will trigger a workshop
<b>Rewards 1</b>	have cake together
<b>Sanctions 1</b>	Otheres choose a random virtual background
<b>Signatures</b>	
Scrum Master	Maximilian Krug
Product owner	Mohammed Tamim Ayach
Product owner	Ali Nawlo
Software developer	Luca Bretting
Software developer	Benedikt Zinn
Software developer	Christos Labroussis
Software developer	Robin Seidl
Software developer	Franz Schlicht
Software developer	Felix Hilgers
Software developer	Tom Weißhuhn
	<a href="https://oss.cs.fau.de/wp-content/uploads/2014/04/Team-Contract-Explanation-and-Examples.pdf">https://oss.cs.fau.de/wp-content/uploads/2014/04/Team-Contract-Explanation-and-Examples.pdf</a>

Product Vision	Project Mission
<p>In systems with a high frequency of component changes, it is difficult to determine which component might be causing performance issues and affecting the entire system negatively. This is especially hard if the source code and/or build environment for the components is not present as they might be coming from external suppliers, which means they cannot easily be instrumented. This can result in a lot of communication and extra work.</p> <p>Using eBPF allows for tracking some of these issues at the kernel level, where for example blocking calls are made and can be tracked. It allows for hooking into Sys-Calls as well as calls to other userspace or kernel-level functions (uprobes and kprobes), all without needing to modify application code. This makes it possible to track down cross-cutting performance issues without needing additional support from the vendor of the component.</p> <p>The information about, for example the length of a blocking calls, can then be passed to various frontends, such as an Android application running on the target hardware or an external sink for displaying the data in visualization software like Grafana.</p>	<p>ZIOFA (Zero Instrumentation Observability for Android) aims to implement observability use cases relevant to performance specified by our industry partner using eBPF. Examples include tracing long-running blocking calls, leaking JNI indirect references or signals like SIGKILL sent to processes, all without instrumenting the observed application itself.</p> <p>The eBPF programs are loaded and unloaded using a backend daemon running as root that will collect metrics and send them to a client. For displaying these metrics to the user, we are implementing an on-device UI that can display visualizations for these use cases and allow for configuration of the enabled use cases, but using a decoupled Client SDK so that future work may easily make the data accessible the external processing.</p>

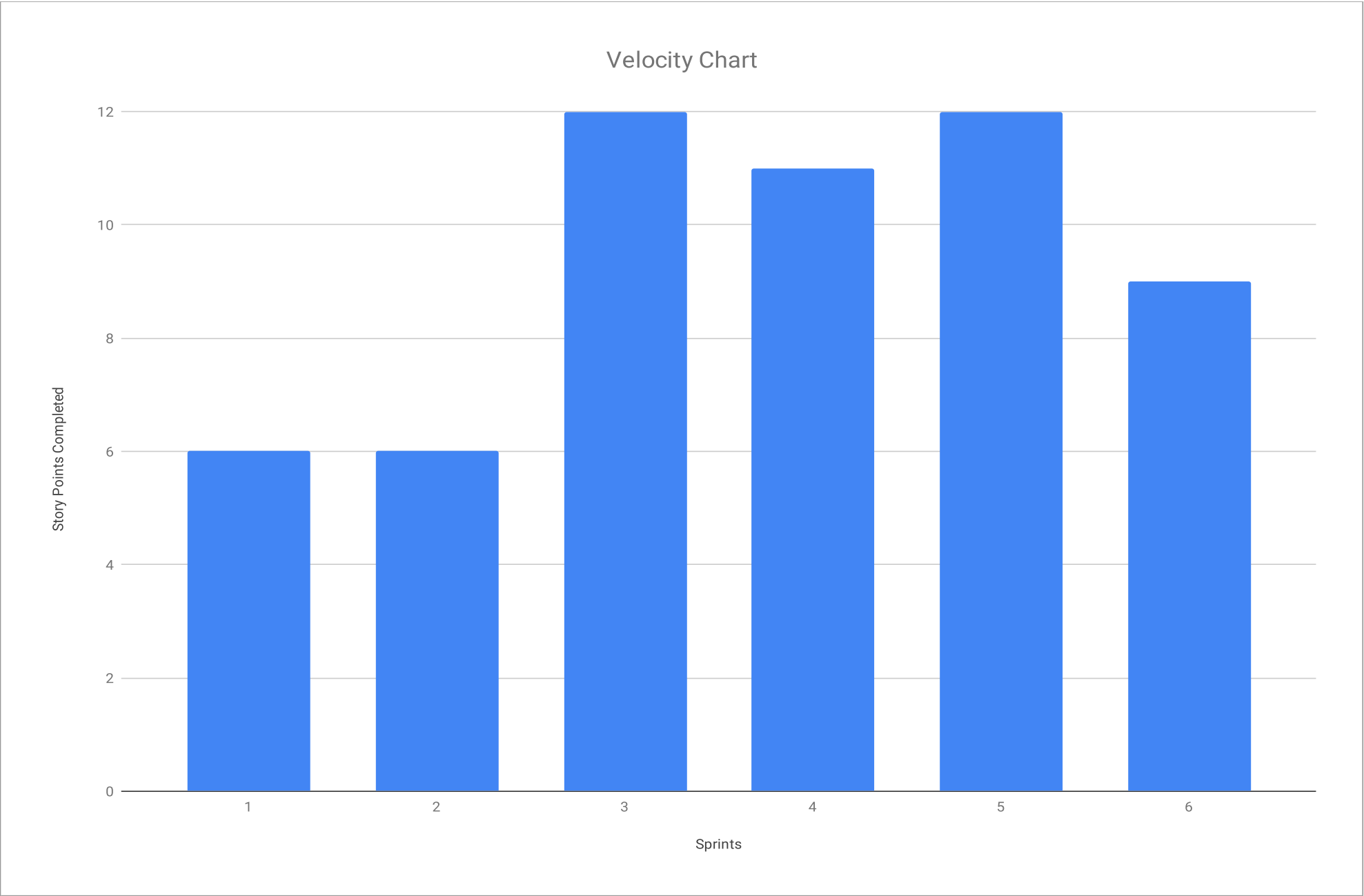
Term	Definition

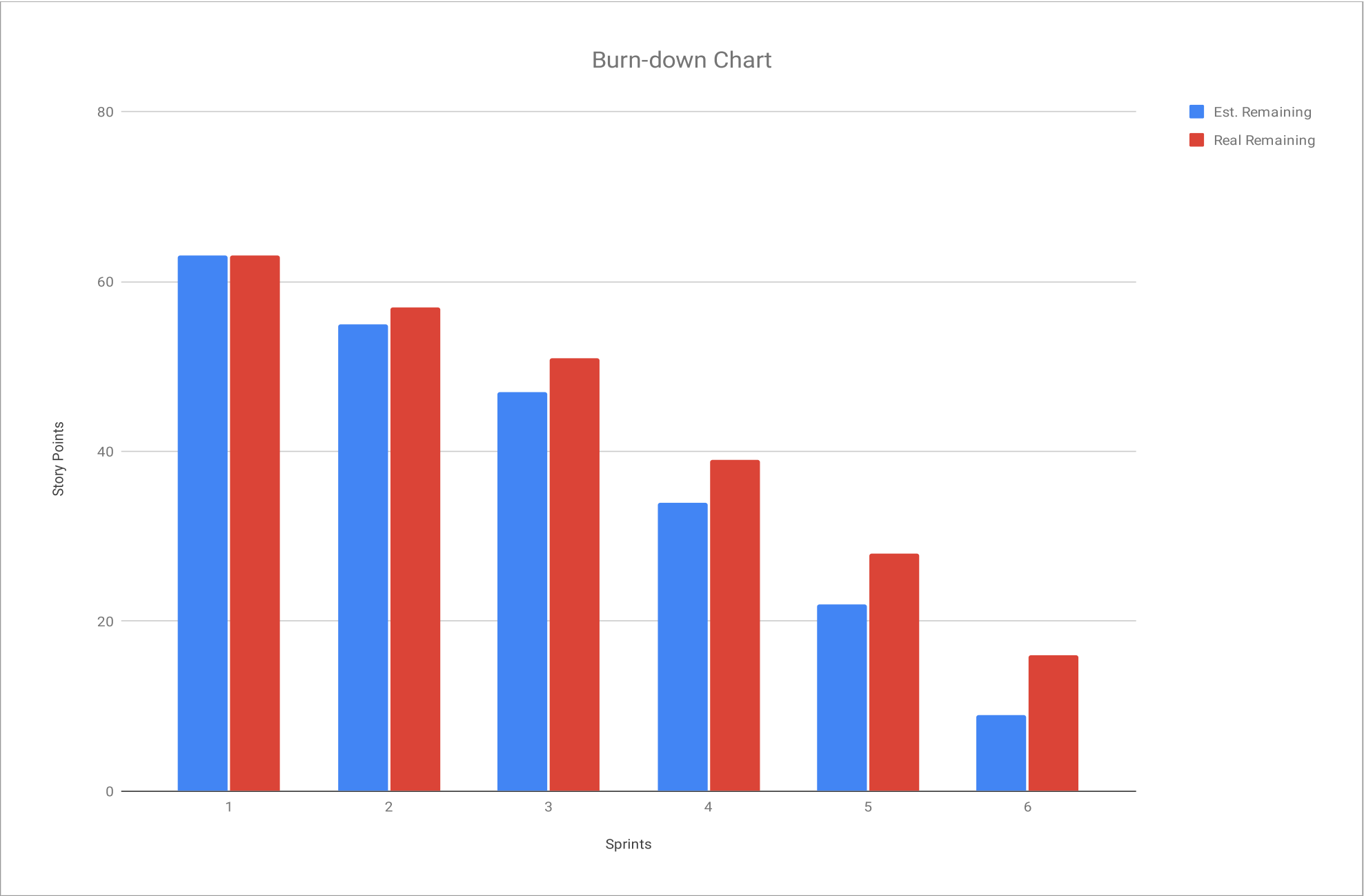
Sprint #	Sprint goal
1	None
2	None
3	None
4	Optional
5	Working, loading, and unloading of eBPF Programs from UI all the way to eBPF
6	Analyzing traffic over Unix Domain Sockets
7	Analyzing user space function calls
8	Finalizing User space function calls
9	
10	
11	
12	
13	
14	
15	

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release						
Total			63	63		
Sprints						
1	Get to know the Team		8	63	6	63
2	Get familiar with eBPF and other required technologies.		8	55	6	57
3	Start Developing, have a UI blueprint and a Backend beginning		13	47	12	51
4	Build a UI and work with Ebpf		12	34	11	39
5	Working, loading, and unloading of eBPF Programs from UI all the way to eBPF		13	22	12	28
6	Analysing traffic over Unix Domain Socket		9	9	9	16
Features						
1	Get to know the Team					
		Brain-storm Architecture	3		1	
		Preperation of Kotlin	3		3	
		Brain-storm ebpf use cases	2		2	
2	Get familiar with eBPF and other required technologies.					
		Docker Container	3		3	
		get information about android processes to list them	3		1	
		set aarch64 als target	1		1	
		use android 13 instead of 15	1		1	
3	Start Developing, have a UI blueprint and a Backend beginning					
		Preparation of CI	3		3	
		find timeseries visualization library	2		2	
		Sbom generation	2		1	
		Generation of sboms doesn't include kotlin	1		1	
		Communcation between Android side and Rust side	5		5	
4	Build a UI and work with Ebpf					
		unix domain socket traffic analysis (research)	5		3	
		Home Screen and Navigation Drawer	2		3	
		EBPF Program extension to load kProbes	3		3	
		Implement frontend load and list programs	2		2	
5	Working, loading, and unloading of eBPF Programs from UI all the way to eBPF					
		kotlin interface for frontend loading and listing programs	1		1	
		test cli client: load and list programs	3		2	
		client library exported to kotlin	2		1	
		Running processes List	3		3	



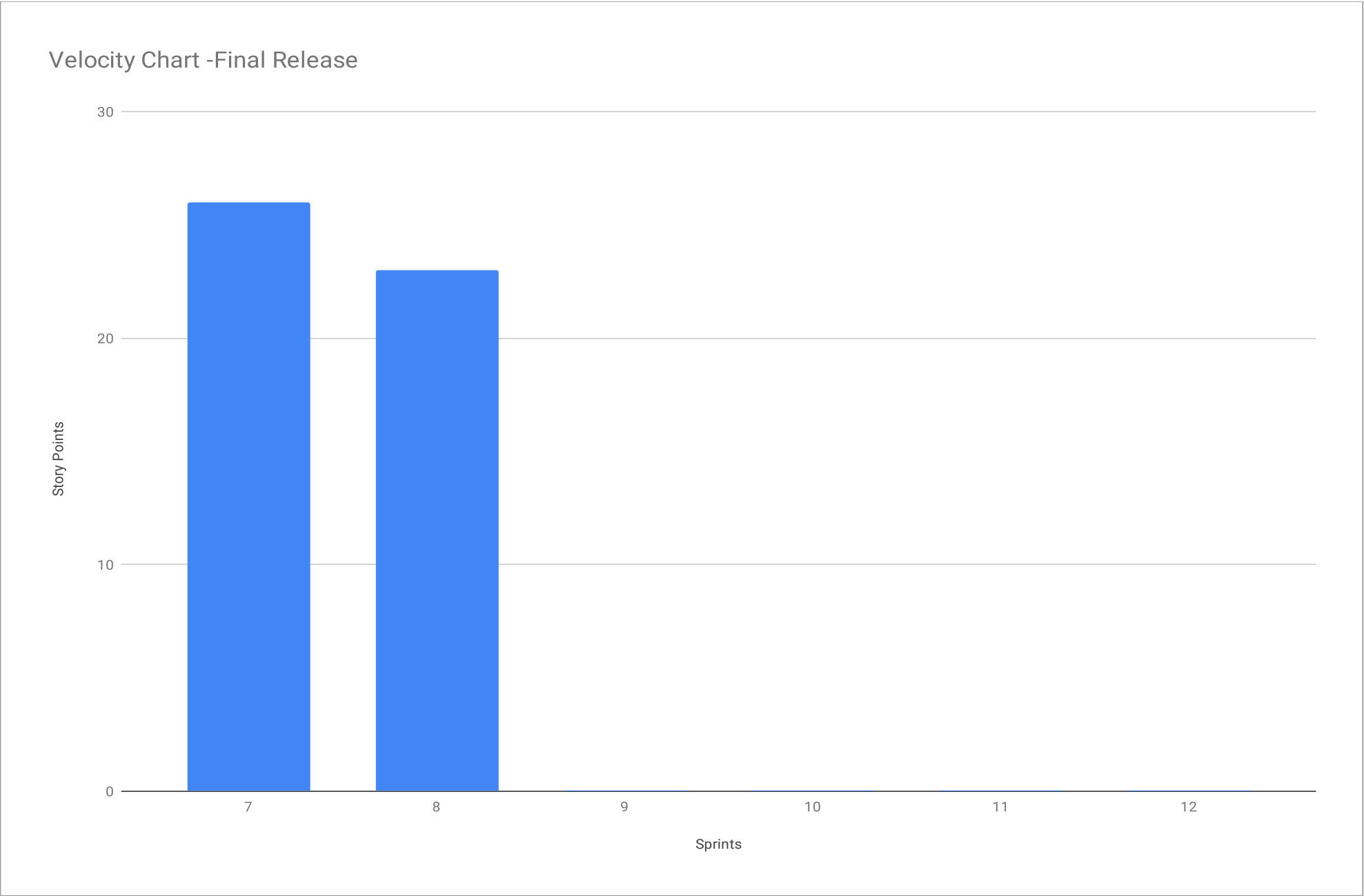
[illegible]

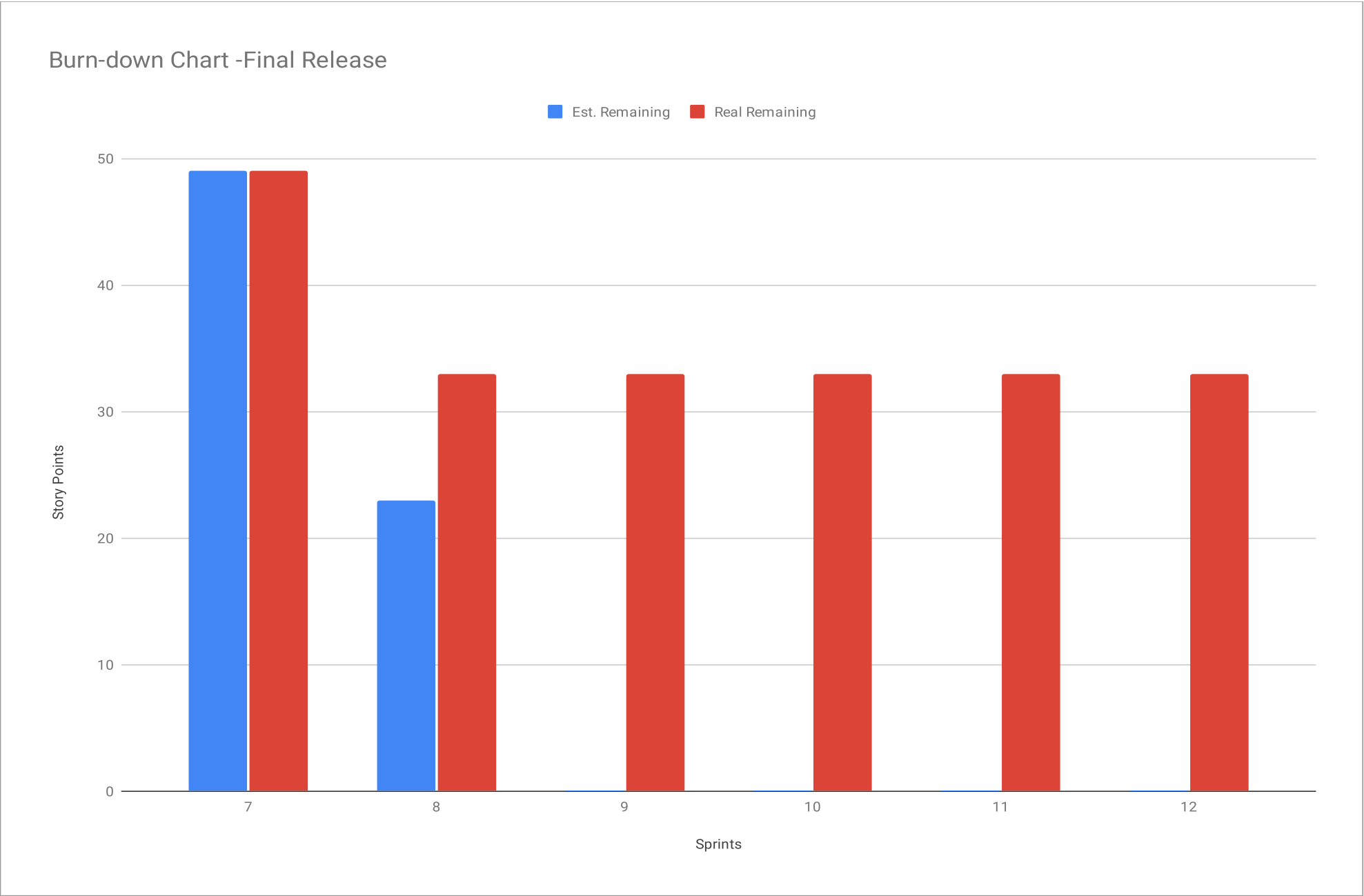




Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release						
Total			49	49		
Sprints						
7	Analyzing user space function calls		26	49	16	49
8	Finalizing User Space Function Calls		23	23	0	33
9			0	0	0	33
10			0	0	0	33
11			0	0	0	33
12			0	0	0	33
Features						
7	Analyzing user space function calls					
		setup Uprobe Analysis	2		5	
		Configure Uprobe Analysis	1		-	
		Uprobe Events in Frontend	2		-	
		Collect Uprobe events	1		-	
		defining metrics for the visualization screen	3		3	
		setup ebpf uprobes	3		-	
		Dex/Oat Symbols	5		-	
		Project Refactored	9		8	
8	Finalizing User Space Function Calls					
		Configure Uprobe Analysis	2			
		Uprobe Events in Frontend	1			
		Collect Uprobe events	1			
		setup ebpf uprobes	3			
		Dex/Oat Symbols	5			
		Refactoring ebpf Programs	1			
		Refactoring Configuration API	2			
		Refactoring pIDs to uint32	2			
		Refactoring Collection of events in Daemon	2			
		Uprobe Analysis: Frontend Show Symbols	2			
		Uprobe Analysis: Frontend Show Uprobe Events	2			
9						

[illegible]





#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done
	<div>1. Code for Components has been written.<div>a. The code does comply to the naming conventions of the used programming language</div><div>b. Code has been completed</div><div>c. Unclear code parts are provided with a short comment, to explain what this part is supposed to do.</div></div> <div>2. Developers submit a screenshots of the finished feature as a comment to the related issue</div> <div>3. Feature has been reviewed by another team member</div> <div>4. Feature has been merged and closed</div>	<div>1. Finished issues are marked as done</div> <div>2. Code is tested and deployed</div> <div>3. A short demo is available for each sprint (this is compliant with point 3 in DoD for Feature) so it can be the screenshots or a small video or even a short-live presentation</div> <div>4. Bill of Material is kept in a current state</div>	<div>1. Team agrees on which features to be released</div> <div>2. Features have been tested and reviewed by other team member</div> <div>3. Documentations are kept updated</div> <div>4. A short demo featuring major features is provided</div>



Type	Link / reference

#	Context	Name	Version	License	Comment
1	Gradle Plugin	org.cyclonedx.bom	1.10.0	APACHE-2.0	<a href="https://github.com/CycloneDX/cyclonedx-gradle-plugin">https://github.com/CycloneDX/cyclonedx-gradle-plugin</a>
2	Gradle Plugin	nl.littlerobots.version-catalog-update	0.8.5	APACHE-2.0	<a href="https://github.com/littlerobots/version-catalog-update-plugin">https://github.com/littlerobots/version-catalog-update-plugin</a>
3	Gradle Plugin	com.github.ben-manes.versions	0.51.0	APACHE-2.0	<a href="https://github.com/ben-manes/gradle-versions-plugin">https://github.com/ben-manes/gradle-versions-plugin</a>
4	Gradle Plugin	com.android.application	8.6.0	APACHE-2.0	<a href="https://maven.google.com/web/index.html?q=com.android.applicat#com.android.application:com.android.application:gradle-plugin:8.6.0">https://maven.google.com/web/index.html?q=com.android.applicat#com.android.application:com.android.application:gradle-plugin:8.6.0</a>
5	Gradle Plugin	com.ncorti.ktfmt.gradle	0.21.0	MIT	<a href="https://github.com/corinico/ktfmt-gradle">https://github.com/corinico/ktfmt-gradle</a>
6	Gradle Plugin	org.jetbrains.kotlin.plugin.compose	2.1.0	APACHE-2.0	<a href="https://github.com/JetBrains/compose-multiplatform">https://github.com/JetBrains/compose-multiplatform</a>
7	Gradle Plugin	org.jetbrains.kotlin.android	2.1.0	APACHE-2.0	<a href="https://github.com/JetBrains/kotlin">https://github.com/JetBrains/kotlin</a>
8	Android UI	androidx.activity:activity-compose	1.9.3	APACHE-2.0	<a href="https://maven.google.com/web/index.html?q=androidx.activity#androidx.activity:activity-compose:1.9.3">https://maven.google.com/web/index.html?q=androidx.activity#androidx.activity:activity-compose:1.9.3</a>
9	Android UI	androidx.compose:compose-bom	2024.11.00	APACHE-2.0	<a href="https://maven.google.com/web/index.html?q=androidx.compose#androidx.compose:compose-bom:2024.11.00">https://maven.google.com/web/index.html?q=androidx.compose#androidx.compose:compose-bom:2024.11.00</a>
10	Android UI	androidx.core:core-ktx	1.15.0	APACHE-2.0	<a href="https://maven.google.com/web/index.html?q=androidx.core#androidx.core:core-ktx:1.15.0">https://maven.google.com/web/index.html?q=androidx.core#androidx.core:core-ktx:1.15.0</a>
11	Android UI	androidx.lifecycle:lifecycle-runtime-ktx	2.8.7	APACHE-2.0	<a href="https://maven.google.com/web/index.html?q=androidx.life#androidx.lifecycle:lifecycle-runtime-ktx:2.8.7">https://maven.google.com/web/index.html?q=androidx.life#androidx.lifecycle:lifecycle-runtime-ktx:2.8.7</a>
12	Android DI	io.insert-koin:koin-android	4.0.0	APACHE-2.0	<a href="https://github.com/InsertKoinIO/koin">https://github.com/InsertKoinIO/koin</a>
13	Android DI	io.insert-koin:koin-androidx-compose	4.0.0	APACHE-2.0	<a href="https://github.com/InsertKoinIO/koin">https://github.com/InsertKoinIO/koin</a>
14	Android DI	io.insert-koin:koin-core	4.0.0	APACHE-2.0	<a href="https://github.com/InsertKoinIO/koin">https://github.com/InsertKoinIO/koin</a>
15	Android Test	io.insert-koin:koin-test-junit4	4.0.0	APACHE-2.0	<a href="https://github.com/InsertKoinIO/koin">https://github.com/InsertKoinIO/koin</a>
16	Android Test	androidx.test.espresso:espresso-core	3.6.1	APACHE-2.0	<a href="https://maven.google.com/web/index.html?q=androidx.test.es#androidx.test.espresso:espresso-core:3.6.1">https://maven.google.com/web/index.html?q=androidx.test.es#androidx.test.espresso:espresso-core:3.6.1</a>
17	Android Test	androidx.test.ext:junit	1.2.1	APACHE-2.0	<a href="https://maven.google.com/web/index.html?q=androidx.test.ext#androidx.test.ext:junit:1.2.1">https://maven.google.com/web/index.html?q=androidx.test.ext#androidx.test.ext:junit:1.2.1</a>
18	Android Test	junit:junit	4.13.2	EPL-1.0	<a href="https://github.com/junit-team/junit4">https://github.com/junit-team/junit4</a>
19	Rust Ebpf	aya	0.13.1	MIT OR APACHE-2.0	<a href="https://github.com/aya-rs/aya">https://github.com/aya-rs/aya</a>
20	Rust Ebpf	aya-ebpf	0.1.1	MIT OR APACHE-2.0	<a href="https://github.com/aya-rs/aya">https://github.com/aya-rs/aya</a>
21	Rust Ebpf	aya-log	0.2.1	MIT OR APACHE-2.0	<a href="https://github.com/aya-rs/aya">https://github.com/aya-rs/aya</a>
22	Rust Ebpf	aya-log-ebpf	0.1.1	MIT OR APACHE-2.0	<a href="https://github.com/aya-rs/aya">https://github.com/aya-rs/aya</a>
23	Rust Ebpf	libc	0.2.168	MIT OR APACHE-2.0	<a href="https://github.com/rust-lang/libc">https://github.com/rust-lang/libc</a>
24	Rust Errors	anyhow	1.0.0	MIT OR APACHE-2.0	<a href="https://github.com/dtolnay/anyhow">https://github.com/dtolnay/anyhow</a>
25	Rust Build	cargo_metadata	0.19.1	MIT	<a href="https://github.com/oli-obk/cargo_metadata">https://github.com/oli-obk/cargo_metadata</a>
26	Rust Build	clap	4.5.23	MIT OR APACHE-2.0	<a href="https://github.com/clap-rs/clap">https://github.com/clap-rs/clap</a>
27	Rust Build	which	7.0.0	MIT	<a href="https://github.com/harryfe/which-rs">https://github.com/harryfe/which-rs</a>
28	Rust Logging	env-logger	0.11.5	MIT OR APACHE-2.0	<a href="https://github.com/rust-cli/env_logger">https://github.com/rust-cli/env_logger</a>
29	Rust Logging	log	0.4.22	MIT OR APACHE-2.0	<a href="https://github.com/rust-lang/log">https://github.com/rust-lang/log</a>
30	Rust Async	tokio	1.42.0	MIT	<a href="https://github.com/tokio-rs/tokio">https://github.com/tokio-rs/tokio</a>
31	Rust Async	tokio-stream	0.1.17	MIT	<a href="https://github.com/tokio-rs/tokio">https://github.com/tokio-rs/tokio</a>
32	Rust API	prost	0.13.4	APACHE-2.0	<a href="https://github.com/tokio-rs/prost">https://github.com/tokio-rs/prost</a>
33	Rust API	tonic	0.12.3	MIT	<a href="https://github.com/hyperium/tonic">https://github.com/hyperium/tonic</a>
34	Rust API	tonic-build	0.12.3	MIT	<a href="https://github.com/hyperium/tonic">https://github.com/hyperium/tonic</a>
35	Toolchain	python3	3.12.6	PSF-2.0	<a href="https://docs.python.org/3/license.html">https://docs.python.org/3/license.html</a>
36	Toolchain	rust	1.84.0-nightly	MIT OR APACHE-2.0	<a href="https://www.rust-lang.org/policies/licenses">https://www.rust-lang.org/policies/licenses</a>
37	Toolchain	cargo-ndk	3.5.7	MIT OR APACHE-2.0	<a href="https://github.com/bbqsrc/cargo-ndk">https://github.com/bbqsrc/cargo-ndk</a>
38	Toolchain	protoc	28.2	BSD-3-Clause	<a href="https://github.com/protocolbuffers/protobuf">https://github.com/protocolbuffers/protobuf</a>
39	Toolchain	bpf-linker	0.9.13	MIT OR APACHE-2.0	<a href="https://github.com/aya-rs/bpf-linker">https://github.com/aya-rs/bpf-linker</a>
40	Toolchain	nix	2.18.7	LGPL-2.1	<a href="https://github.com/NixOS/nix">https://github.com/NixOS/nix</a>
41	Toolchain	cyclonedx-cli	0.25.1	APACHE-2.0	<a href="https://github.com/CycloneDX/cyclonedx-cli">https://github.com/CycloneDX/cyclonedx-cli</a>
42	Toolchain	gradle	8.10.2	APACHE-2.0	<a href="https://github.com/gradle/gradle">https://github.com/gradle/gradle</a>
43	Toolchain	openjdk	21.0.3	GPL-2.0-with-classpath-exception	<a href="https://openjdk.org/legal/gplv2+ce.html">https://openjdk.org/legal/gplv2+ce.html</a>
44	Toolchain	android-cmdline-tools	16	android-sdk-license	<a href="https://developer.android.com/studio/terms">https://developer.android.com/studio/terms</a>
45	Toolchain	android-emulator	35.3.6.0	android-sdk-license	<a href="https://developer.android.com/studio/terms">https://developer.android.com/studio/terms</a>
46	Toolchain	android-ndk	28.0.12433566	android-sdk-license	<a href="https://developer.android.com/studio/terms">https://developer.android.com/studio/terms</a>
47	Toolchain	android-tools	35.0.0	android-sdk-license	<a href="https://developer.android.com/studio/terms">https://developer.android.com/studio/terms</a>
48	Toolchain	platform-tools	35.0.2	android-sdk-license	<a href="https://developer.android.com/studio/terms">https://developer.android.com/studio/terms</a>
49	Toolchain	platforms-android	35	android-sdk-license	<a href="https://developer.android.com/studio/terms">https://developer.android.com/studio/terms</a>
50	Rust API	uniffi	0.28.2	MPL-2.0	<a href="https://github.com/mozilla/uniffi-rs">https://github.com/mozilla/uniffi-rs</a>
51	Rust API	thiserror	2.0.6	MIT OR APACHE-2.0	<a href="https://github.com/dtolnay/thiserror">https://github.com/dtolnay/thiserror</a>
52	Gradle Plugin	com.android.library	8.7.2	APACHE-2.0	<a href="https://maven.google.com/web/index.html?q=com.android.lib#com.android.library:com.android.library:gradle-plugin:8.7.2">https://maven.google.com/web/index.html?q=com.android.lib#com.android.library:com.android.library:gradle-plugin:8.7.2</a>
53	Gradle Plugin	org.mozilla.rust-android-gradle.rust-android	0.9.4	APACHE-2.0	<a href="https://github.com/mozilla/rust-android-gradle">https://github.com/mozilla/rust-android-gradle</a>
54	Android Rust	net.java.dev.jna	5.15.0	Apache-2.0 OR LGPL-2.1	<a href="https://github.com/java-native-access/jna">https://github.com/java-native-access/jna</a>
55	Android Navigation	androidx.navigation:navigation-compose	2.8.4	Apache-2.0	<a href="https://maven.google.com/web/index.html?q=androidx.navigation#androidx.navigation:navigation-compose:2.8.4">https://maven.google.com/web/index.html?q=androidx.navigation#androidx.navigation:navigation-compose:2.8.4</a>
56	Android Logging	com.jakewharton.timber	2.8.0	Apache-2.0	<a href="https://github.com/JakeWharton/timber">https://github.com/JakeWharton/timber</a>
57	Android Visualization	com.partykandpatrick.vico.compose	2.0.0-beta.3	Apache-2.0	<a href="https://github.com/patrykandpatrick/vico">https://github.com/patrykandpatrick/vico</a>
58	Android Visualization	com.partykandpatrick.vico.compose-m2	2.0.0-beta.3	Apache-2.0	<a href="https://github.com/patrykandpatrick/vico">https://github.com/patrykandpatrick/vico</a>
59	Android Visualization	com.partykandpatrick.vico.compose-m3	2.0.0-beta.3	Apache-2.0	<a href="https://github.com/patrykandpatrick/vico">https://github.com/patrykandpatrick/vico</a>
60	Android Visualization	com.partykandpatrick.vico.core	2.0.0-beta.3	Apache-2.0	<a href="https://github.com/patrykandpatrick/vico">https://github.com/patrykandpatrick/vico</a>
61	Rust Serialization	serde	1.0.215	MIT OR APACHE-2.0	<a href="https://github.com/serde-rs/serde">https://github.com/serde-rs/serde</a>
62	Rust Serialization	serde-json	1.0.0	MIT OR APACHE-2.0	<a href="https://github.com/serde-rs/json">https://github.com/serde-rs/json</a>
63	Rust Tracing	tracing	0.1.41	MIT	<a href="https://github.com/tokio-rs/tracing">https://github.com/tokio-rs/tracing</a>
64	Rust Tracing	tracing-subscriber	0.3.19	MIT	<a href="https://github.com/tokio-rs/tracing">https://github.com/tokio-rs/tracing</a>
65	Rust System	procs	0.17.0	MIT OR APACHE-2.0	<a href="https://github.com/eminence/procs">https://github.com/eminence/procs</a>
66	Android Visualization	com.google.accompanist:accompanist-drawablepainter	0.15.0	Apache 2.0	<a href="https://github.com/google/accompanist/tree/main/drawablepainter">https://github.com/google/accompanist/tree/main/drawablepainter</a>
67	Rust Async	async-broadcast	0.7.1	MIT OR APACHE-2.0	<a href="https://github.com/smol-rs/async-broadcast">https://github.com/smol-rs/async-broadcast</a>
68	Rust TUI	console	0.15.8	MIT	<a href="https://github.com/console-rs/console">https://github.com/console-rs/console</a>
69	Rust TUI	dialoguer	0.11.0	MIT	<a href="https://github.com/console-rs/dialoguer">https://github.com/console-rs/dialoguer</a>
70	Rust TUI	indicatif	0.17.9	MIT	<a href="https://github.com/console-rs/indicatif">https://github.com/console-rs/indicatif</a>

#	Context	Name	Version	License	Comment
71	Rust Raw Linux APIS	nix	0.29.0	MIT	<a href="https://github.com/nix-rust/nix">https://github.com/nix-rust/nix</a>
72	Gradle Plugin	com.android.tools.build:gradle	8.7.2	APACHE-2.0	<a href="https://maven.google.com/web/index.html?q=com.android.tools.build#com.android.tools.build:gradle:8.7.2">https://maven.google.com/web/index.html?q=com.android.tools.build#com.android.tools.build:gradle:8.7.2</a>
73	Android Coroutines	org.jetbrains.kotlinx:kotlinx-coroutines-android	1.9.0	APACHE-2.0	<a href="https://github.com/Kotlin/kotlinx.coroutines">https://github.com/Kotlin/kotlinx.coroutines</a>
74	Rust Safety	bytemuck	1.20.0	MIT OR APACHE-2.0 OR Zlib	<a href="https://github.com/Lokathor/bytemuck">https://github.com/Lokathor/bytemuck</a>
75	Rust Concurrency	ractor	0.13.4	MIT	<a href="https://github.com/slawlor/ractor">https://github.com/slawlor/ractor</a>
76	Rust Concurrency	crossbeam	0.8.4	MIT OR APACHE-2.0	<a href="https://github.com/crossbeam-rs/crossbeam">https://github.com/crossbeam-rs/crossbeam</a>
77	Rust Parsing	object	0.36.5	MIR OR APACHE-2.0	<a href="https://github.com/gimli-rs/object">https://github.com/gimli-rs/object</a>

Last Name	First Name	Value					
Krug	Maximilian			#DIV/	#DIV/		
Ayach	Mohammed Tamim			0!	0!		
Bretting	Luca						
Seidl	Robin						
Hilgers	Felix			0	No size		
Weisshuhn	Tom			1	Trivial size		
Schlicht	Franz	?		2	Small size		
Nawlo	Ali			3	Medium size		
Zinn	Benedikt			5	Large size		
				8	Very large size		
				13	Too large (size)		
Team members left							
Labroussis	Christos						
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							