Project Name	EMBArk Orchestration Framework
Online team meeting	https://tu-berlin.zoom-x.de/j/62142983444?pwd=nnFsVt1p6bEKQRS6xN2oYewQqTlcF7.1
Production system (if any)	...
Test system (if any)	...
GitHub repository	https://github.com/amosproj/amos2025ss01-embark-orchestration-framework
GitHub feature board	https://github.com/orgs/amosproj/projects/79/views/2
GitHub imp-squared backlog	https://github.com/orgs/amosproj/projects/83
Team T-shirt (white)	https://www.shirtinator.de/s/qaSIJh2NSBO7V5kllYTrWQ
Team T-shirt (black)	https://www.shirtinator.de/s/Bhl3o0Z8R2635N-1SYy3VA
Additional materials	...
Team maling list	oss-amos-proj1@lists.fau.de

	First Name	GitHub User Name	Email Address
Kunow	Johannes	jkunow	j.kunow@tu-berlin.de
Meusling	Patrick	SirGankalot	meusling@campus.tu-berlin.de
Dekanozishvili	Luka	LukaDeka	luka.dekanozishvili1@gmail.com
Roy	Paul	PaulRoy1	paul.roy@fau.de
Novak	Jannik	ashiven	nevisha@pm.me
Prosser	Clemens	ClProsser	clemens.prosser@gmail.com
Damm	Sönke Fridtjof	fridtjof-damm	soenke.f.damm@campus.tu-berlin.de

		Product Owner						
#	Meeting Day	Review	Planning	Software Developer	Release Manager	Scrum Master	Comment	Homework Manager
1	2025-04-16		Johannes	Everyone else	Patrick Meusling	COACH student		Patrick Meusling
2	2025-04-23	Johannes	Fridtjof	Everyone else	Clemens Prosser	COACH student		Clemens Prosser
3	2025-04-30	Fridtjof	Johannes	Everyone else	Clemens Prosser	COACH student		Clemens Prosser
4	2025-05-07	Johannes	Fridtjof	Everyone else	Patrick Meusling	COACH student		Patrick Meusling
5	2025-05-14	Fridtjof	Johannes	Everyone else	Jannik Novak	COACH student		Luka Dekanozishvili
6	2025-05-21	Johannes	Fridtjof	Everyone else	Luka Dekanozishvili	COACH student		Luka Dekanozishvili
7	2025-05-28	Fridtjof	Johannes	Everyone else	Luka Dekanozishvili	COACH student	Mid-term due	Johannes Kunow
8	2025-06-04	Johannes	Fridtjof	Everyone else	Jannik Novak	COACH student		Fridtjof Damm
9	2025-06-11	Fridtjof	Johannes	Everyone else	Patrick Meusling	COACH student		Johannes Kunow
10	2025-06-18	Johannes	Fridtjof	Everyone else	Patrick Meusling	COACH student		Fridtjof Damm
11	2025-06-25	Fridtjof	Johannes	Everyone else	Clemens Prosser	COACH student		Johannes Kunow
12	2025-07-02	Johannes	Fridtjof	Everyone else	Clemens Prosser	COACH student		Fridtjof Damm
13	2025-07-09	Fridtjof	Johannes	Everyone else	Luka Dekanozishvili	COACH student		Johannes Kunow
14	2025-07-16	Johannes	Fridtjof	Everyone else	Luka Dekanozishvili	COACH student	Demo day!	Fridtjof Damm
15	2025-07-23	Fridtjof		Everyone else	Jannik Novak	COACH student	Retrospective	Johannes Kunow

Product owners, software developers, and Scurm Master are set and ideally don't change over time; the critical part is the Release Manager role you need to define here

Goals	Aquire new skills
	Produce a functioning and valuable product
Meeting norms	We show up to the team meeting on time
	We respect each others opinions
Working norms	Produce clean code
	We respect other people's work
Coordination norms	Task responsibilities are well defined
	We balance workload among the team
Communication norms	We check our communication platform at least once every workday
	We communicate constructively
Consideration norms	We discuss issues openly
	We vote in case we can't reach a consensus
Cont. improvement norms	We consider the happines index to monitor team motivation
	We encourage critique and improvement efforts
Rewards	We praise each others work
	We treat ourselfes to a sweet of choice for good work
Sanctions	10 push-ups infront of the camera
	We critize objectively

Signatures

Scrum Master	Paul Roy
Product owner	Johannes Kunow
Product owner	Fridtjof Damm
Software developer	Luka Dekanozishvili
Software developer	Jannik Novak
Software developer	Patrick Meusling
Software developer	Clemens Prosser

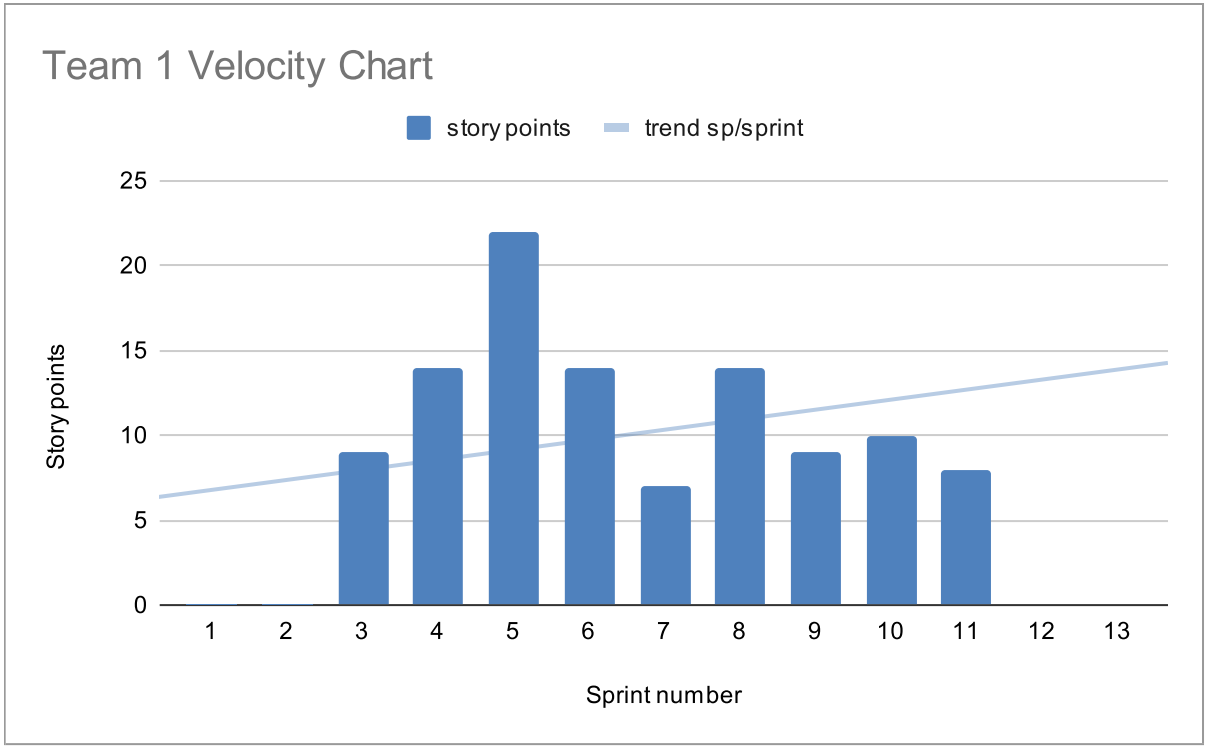
Product Vision	Project Mission
The firmware security analyzer EMBA, along with it's management and orchstration platform EMBArk, enables security professionals and firmware analysts to automate the scalable execution of firmware security scans. This is achieved by parallelizing firmware analyses, reducing manual effort and boosting throughput. As embedded systems become increasingly ubiquitous and complex, EMBArk constitutes a key part in the critical infrastructure in responsible and scalable firmware deployment and development—positioning itself as an essential tool for secure digital transformation. These core values are supplied to users of arbitrary firmware, penetration testing departments, and device vendors, with the common goal of ensuring high security standards.	The mission of this project is to develop a functional orchestration component for EMBArk that enables scalable and automated execution of firmware analysis tasks using the existing EMBA tooling. The MVP will support managing distributed workers (Kali/Ubuntu) via SSH, provide an API interface for job creation, and enable testers to manage worker nodes through a web-based dashboard. Key deliverables include job scheduling, worker management, result collection, and system monitoring features.

Term	Definition
worker node	a vm or physical machine carrying out firmware analyses
orchestrator	component which schedules firmware analysis jobs to worker nodes
celery task queue	a python task queue to allow for concurrent processing
worker node configuring	the action of installing EMBA and all its dependencies on a worker node so that it can be used to analyse firmware

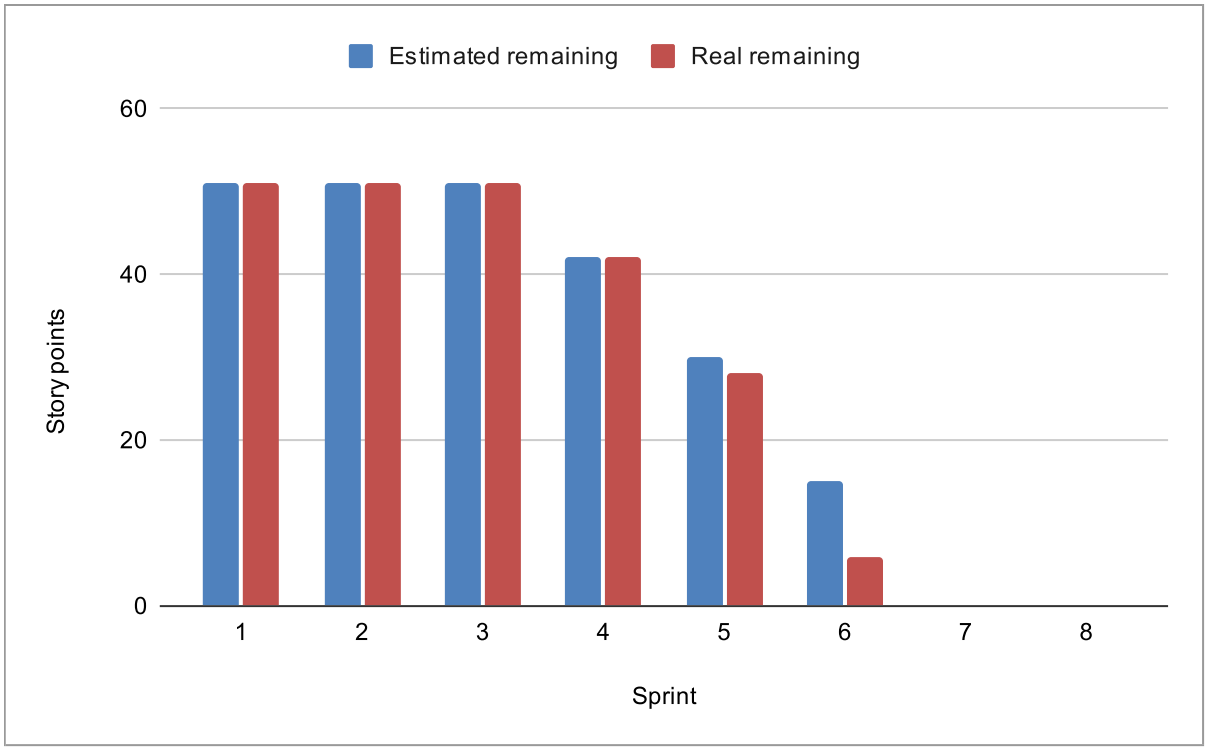
Sprint #	Sprint goal
1	None
2	None
3	Implement basic API features
4	Establishing code quality best practices
5	Set cornerstones for orchestration from UI, worker configuration, and scheduling perspectives
6	Completing UI functionality and enable communication between EMBArk and worker nodes
7	Adding core orchestrator functionality and prepare UI for future features
8	Enable dispatching of firmware analyses with the orchestrator
9	Tie loose orchestrator ends together
10	Enhance worker management to refine user experience
11	Invest in stability
12	Finally finish main orchestrator functionality
13	
14	
15	

Sprint #	Story Points Realized
1	0
2	0
3	9
4	14
5	22
6	14
7	7
8	14
9	9
10	10
11	8
12	
13	
	107

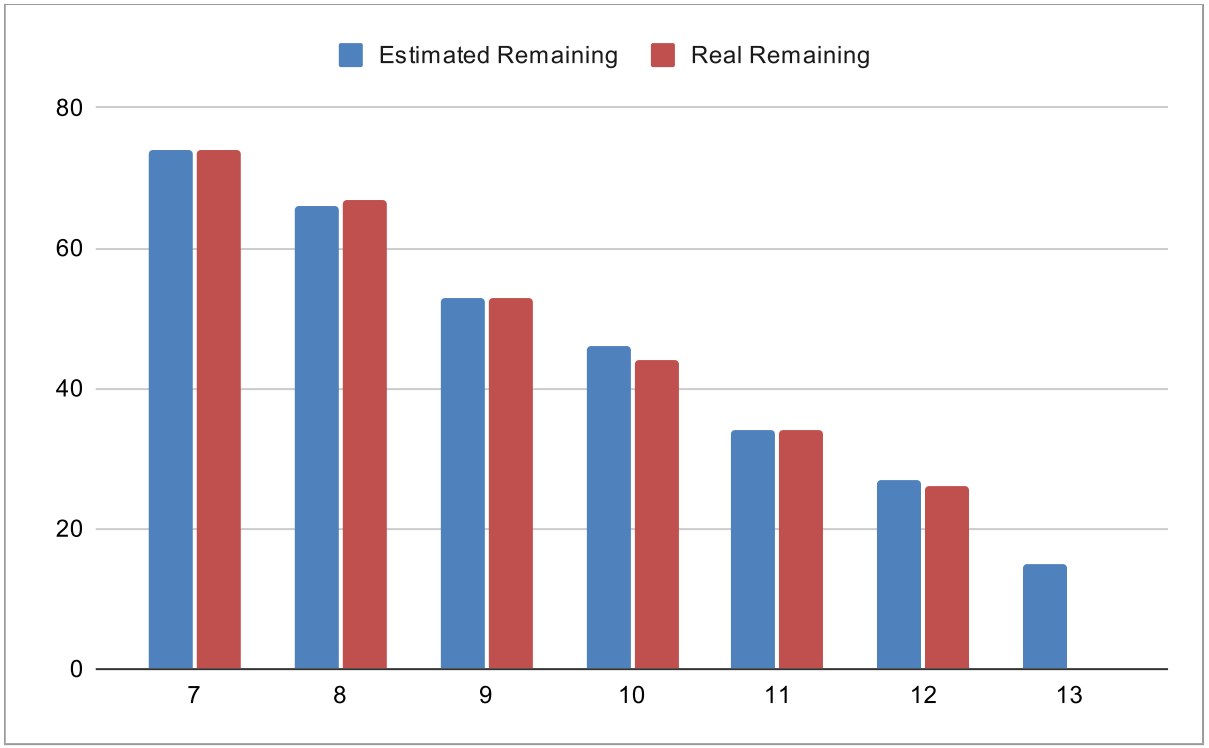
PLEASE CREATE THE VELOCITY CHART ON A NEW TAB USING THE DATA FROM THIS TAB


Team 1 Velocity Chart
story points
trend sp/sprint
25
20
15
10
5
0
Story points
1
2
3
4
5
6
7
8
9
10
11
12
13
Sprint number

Sprint	Goal	Feature Name	Est. size	Est. remaining	Real size	Real remaining
Release						
Total			51	51		
Sprints						
1			0	51	0	51
2			0	51	0	51
3	**Implement basic API features**		9	51	9	51
4	**Establishing code quality best practices**		12	42	14	42
5	**Set cornerstones for orchestration from UI, worker configuration, and scheduling**		15	30	22	28
6	**Completing UI functionality and enable communication between EMBArk and wo**		15	15	9	6
7						
8						
Features						
1						
2						
3	**Implement basic API features**					
		API | Documentation tooling	1		1	
		Mount file system via SSHfs in Python	2		2	
		API | Generate API-Key in user interface	3		3	
		API | Upload firmware and add to queue	3		3	
4	**Establishing code quality best practices**					
		Integration testing	2		2	
		API | Documentation Upload firmware	1		1	
		API | Get status report	3		5	
		API | Documentation Status report	1		1	
		API | Integration test Upload firmware	2		2	
		Configure worker nodes in EMBArk	3		3	
5	**Set cornerstones for orchestration from UI, worker configuration, and scheduling perspectives**					
		API | Document API-Key generation	1		1	
		API | Integration test API-Key generation	2		2	
		EMBA offline worker configuration	3		5	
		Configuration scripts for worker node | Kali	3		5	
		Configuration scripts for worker node | Ubuntu	3		5	
		Reduce check_project.sh execution time	1		2	
		API | Integration test Status report	2		2	
6	**Completing UI functionality and enable communication between EMBArk and worker nodes**					
		EMBArk worker UI	3		3	
		Orchestrator | Receive new workers	3		3	
		Caching in GitHub actions pipeline	2		not completed	
		Configure worker node	5		3	
		Query worker node information	2		not completed	
		Prepare upstream pull request	2		2	
		Connect to worker node	2		3	


Estimated remaining
Real remaining
60
40
20
0
Story points
1
2
3
4
5
6
7
8
Sprint

Sprint	Goal	Feature Name	Est. size	Est. remaining	Real size	Real remaining
Release						
Total			74	74		
Sprints						
7			8	74	7	74
8			13	66	14	67
9			7	53	9	53
10			12	46	10	44
11			7	34	8	34
12			12	27		26
13			15	15		
Features						
7						
		Orchestrator | FIFO scheduling	2		2	
		EMBArk worker UI | Show job id in worker nodes table	2		1	
		Orchestrator | Query worker pool	2		2	
		Caching in Github actions pipeline	2		2	
8						
		Add Celery dependency	2		2	
		Soft reset worker node	2		2	
		Periodic worker information fetch	3		2	
		Update worker nodes	3		5	
		Query worker node information	2		2	
		UI | Update/Reset	1		1	
9						
		Pass newly configured worker nodes to orchestrator	1		2	
		EMBArk starts firmware analysis on worker node	2		3	
		Hard reset worker node	1		1	
		Fix Github Actions bugs	2		2	
		Use Celery for worker node updates	1		1	
10						
		Add Settings App to EMBArk	2		2	
		Manually check for updates	3		3	
		EMBArk default installation produces FileNotFoundError on startup	2		1	
		Collect dependency version information when configuring or updating node	2		1	
		Celery for reset	1		1	
		Add users to sudoers when configuring workers	1		1	
		Reset all worker nodes in config	1		1	
11						
		Hide worker app in sidebar when disabled	1		1	
		Orchestrator | Free worker nodes	2		2	
		Safe and prioritized worker update management	1		2	
		PR Revised Github Actions pipeline and check_project.sh	1		1	
		Download specific worker nodes dependency version	2		2	
12						
		Trigger Orchestrator	2			
		Orchestrator | Abort running firmware analysis	2			
		Monitor workers and collect results	2			
		Orchestrator | Draft Upstream pull request	2			
		Add wiki entries for API	1			
		Celery for IP range scanning	1			
		Indicate currently installed dependency versions for EMBA and externals	1			
		Check EMBArk permissions	1			
13						
		Bug fixing	3			
		Orchestrator PR: Change requests	2			
		Handle unresponsive worker nodes	2			
		Extend update user experience	1			
		Validate analysis workflow with orchestrators	1			
		Provide adequate feedback for apt dependency updates of worker nodes	2			
		Wiki entries for Orchestrator	2			
		Don't allow overlapping IP ranges in configurations	1			
		Change worker authentication to SSH Keys	1			


Estimated Remaining
Real Remaining
80
60
40
20
0
7
8
9
10
11
12
13

#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done
1	Github actions pipeline runs without errors	Features and changes have been demoed in review	Build and deployment documentation exists
2	If changes are visible to users, documentation is added	Features not covered by unit tests are not negatively impacted by sprints changes	Software architecture documentation is up to date
3	Code review passed		Readme is up to date
4	Code merged to main branch		
5	Testable code has appropriate unit tests (Unfortunately the nature of the product forbids general statements for code coverage)		
6	SBOM updated: Added new dependencies to SBOM, removed removed dependencies		
7	Changes added to change log		
8	All added dependencies follow an open source license compatible with the project		

* Upstream PR is explicitly not part of the DoD because the client prefers frequent pulls as soon as features are ready

Type	Link / reference

You hav	Name	Version	License	Comment
1	daphne	4.1.2	BSD	python package
2	mysqlclient	2.2.7	GPLv2+	python package
3	django-apscheduler	0.7.0	MIT	python package
4	python-dotenv	1.1.0	BSD-3-Clause	python package
5	Rx	3.2.0	MIT	python package
6	inotify-simple	1.3.5	BSD	python package
7	psutil	7.0.0	BSD-3-Clause	python package
8	msgpack	1.1.0	Apache 2.0	python package
9	django	5.2	BSD-3-Clause	python package
10	django-hashid-field	3.4.1	MIT	python package
11	django-tables2	2.7.5	BSD	python package
12	requests	2.32.3	Apache 2.0	python package
13	djangorestframework	3.16.0	BSD	python package
14	watchdog	6.0.0	Apache 2.0	python package
15	channels	4.2.2	BSD	python package
16	channels-redis	4.2.1	BSD	python package
17	mod-wsgi-standalone	5.0.2	Apache 2.0	python package
18	django-bootstrap5	25.1	BSD-3-Clause	python package
19	pytz	2025.2	MIT	python package
20	pycodestyle	2.13.0	MIT	python package; development only
21	djlint	1.36.4	GPLv3+	python package; development only
22	pylint-django	2.6.1	GPLv2+	python package; development only
23	selenium	4.31.0	Apache 2.0	python package; development only
24	EMBA	latest	MIT	
25	jquery.js	3.6.0	MIT	javascript library
26	confirm.js	3.3.2	MIT	javascript library
27	bootstrap.js	5.2.3	MIT	javascript library
28	datatable.js	1.11.2	MIT	javascript library
29	charts.js	3.5.1	MIT	javascript library
30	base64.js	3.7.5	MIT	javascript library
31	ansi_up.js	6.0.2	MIT	javascript library
32	confirm.css	3.3.2	MIT	css library
33	bootstrap.css	5.2.3	MIT	css library
34	datatable.css	1.11.2	MIT	css library
35	spectral	6.15.0	Apache 2.0	npm package; development only
36	paramiko	3.5.1	LGPL	python package
37	celery	5.5.3	BSD-3-Clause	python package
38	django-celery-beat	2.8.1	BSD	python package

Last Name	First Name	Value			
Meusling	**Patrick**			#DIV/0!	#DIV/0!
Dekanozishvili	**Luka**				
Novak	**Jannik**				
Prosser	**Clemens**				
				0	No size
				1	Trivial size
				2	Small size
				3	Medium size
				5	Large size
				8	Very large size
				13	Too large (size)

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