Project Name

Online team meeting

Production system (if any)
Test system (if any)

GitHub repository
GitHub feature board
GitHub imp-squared backlog

Team T-shirt (white)
Team T-shirt (black)

Additional materials

Team maling list

EMBArk Orchestration Framework

https://tu-berlin.zoom-x.de/j/62142983444?pwd=nnFsVt1p6bEKQRS6xN2oYewQqTlcF7.1

...
...

https://github.com/amosproj/amos2025ss01-embark-orchestration-framework
https://github.com/orgs/amosproj/projects/79/views/2
https://github.com/orgs/amosproj/projects/83

https://www.shirtinator.de/s/qaSIJh2NSBO7V5kllYTrWQ
https://www.shirtinator.de/s/Bhl3o0Z8R2635N-1SYy3VA

...

oss-amos-proj1@lists.fau.de

	First Name	GitHub User Name
Kunow	Johannes	jkunow
Meusling	Patrick	SirGankalot
Dekanozishvili	Luka	LukaDeka
Roy	Paul	PaulRoy1
Novak	Jannik	ashiven
Prosser	Clemens	ClProsser
Damm	Sönke Fridtjof	fridtjof-damm

Email Address

j.kunow@tu-berlin.de
meusling@campus.tu-berlin.de
luka.dekanozishvili1@gmail.com
paul.roy@fau.de
nevisha@pm.me
clemens.prosser@gmail.com
soenke.f.damm@campus.tu-berlin.de

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

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 n

Comment	Homework Manager
	Patrick Meusling
	Clemens Prosser
	Clemens Prosser
	Patrick Meusling
	Luka Dekanozishvili
	Luka Dekanozishvili
Mid-term due	Johannes Kunow
	Fridtjof Damm
	Johannes Kunow
	Fridtjof Damm
	Johannes Kunow
	Fridtjof Damm
	Johannes Kunow
Demo day!	Fridtjof Damm
Retrospective	Johannes Kunow

ıeed to define here

Goals

Meeting norms

Working norms

Coordination norms

Communication norms

Consideration norms

Cont. improvement norms

Rewards

Sanctions

Signatures

Scrum Master
Product owner
Product owner
Software developer
Software developer
Software developer
Software developer

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

Paul Roy
Johannes Kunow
Fridtjof Damm
Luka Dekanozishvili
Jannik Novak
Patrick Meusling
Clemens Prosser

Product Vision

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.

Project Mission

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
worker node
orchestrator

Definition

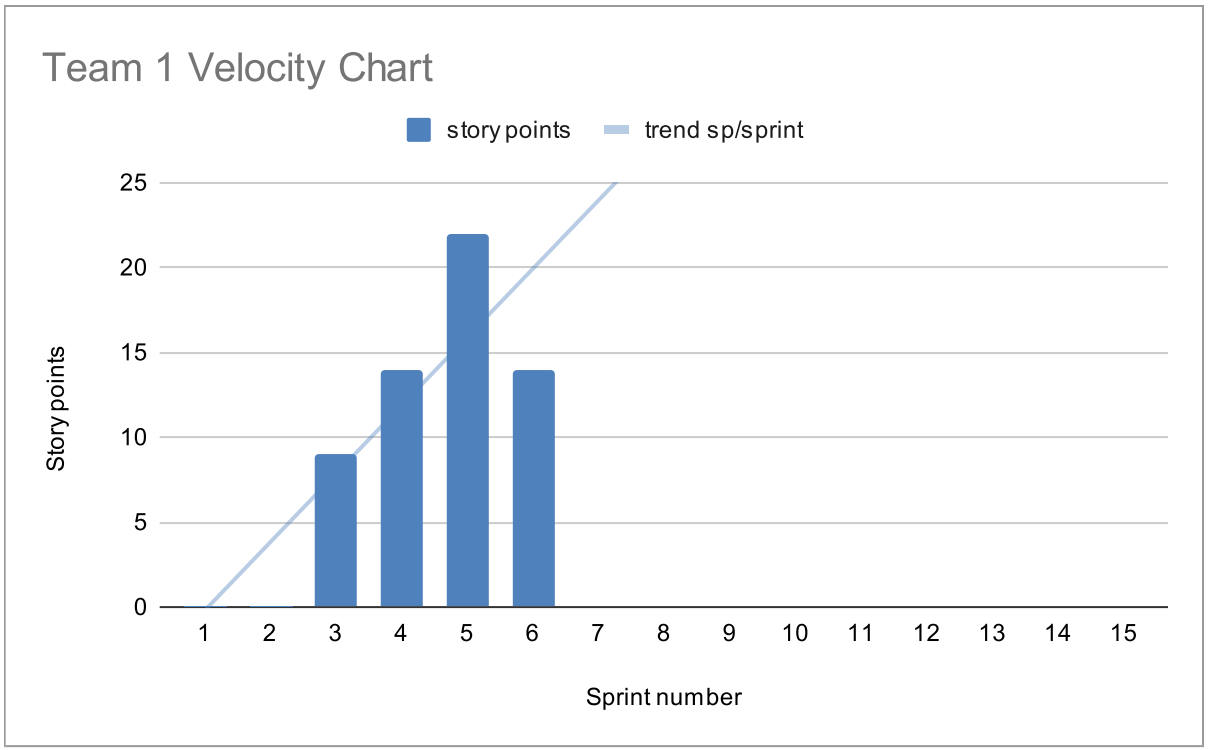
a vm or physical machine carrying out firmware analyses
component which schedules firmware analysis jobs to worker nodes

Sprint #
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

Sprint goal
None
None
Implement basic API features
Establishing code quality best practices
Set cornerstones for orchestration from UI, worker configuration, and scheduling perspectives
Completing UI functionality and enable communication between EMBArk and worker nodes
Adding core orchestrator functionality and prepare UI for future features

Sprint #	Story Points Realized
1	0
2	0
3	9
4	14
5	22
6	14
7	
8	
9	
10	
11	
12	
13	
14	
15	

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

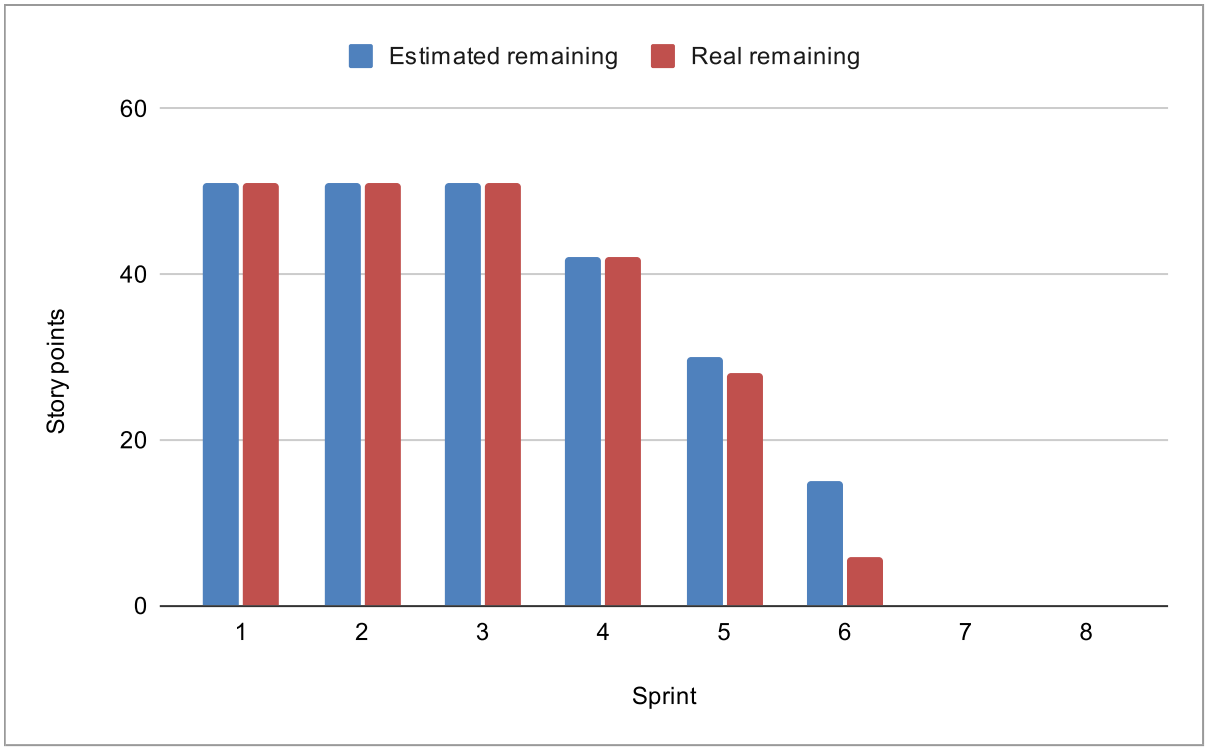

Team 1 Velocity Chart
story points
trend sp/sprint
25
20
15
10
5
0
Story points
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Sprint number

Sprint	Goal	Feature Name	Est. size	Est. remaining
Release				
Total			51	51
Sprints				
1			0	51
2			0	51
3	**Implement basic API features**		9	51
4	**Establishing code quality best practices**		12	42
5	**Set cornerstones for orchestration from UI, worker configuration, and scheduling**		15	30
6	**Completing UI functionality and enable communication between EMBArk and wo**		15	15
7				
8				
Features				
1				
2				
3	**Implement basic API features**			
		API | Documentation tooling	1	
		Mount file system via SSHfs in Python	2	
		API | Generate API-Key in user interface	3	
		API | Upload firmware and add to queue	3	
4	**Establishing code quality best practices**			
		Integration testing	2	
		API | Documentation Upload firmware	1	
		API | Get status report	3	
		API | Documentation Status report	1	
		API | Integration test Upload firmware	2	
		Configure worker nodes in EMBArk	3	
5	**Set cornerstones for orchestration from UI, worker configuration, and scheduling perspectives**			

	API | Document API-Key generation	1
	API | Integration test API-Key generation	2
	EMBA offline worker configuration	3
	Configuration scripts for worker node | Kali	3
	Configuration scripts for worker node | Ubuntu	3
	Reduce check_project.sh execution time	1
	API | Integration test Status report	2
6	**Completing UI functionality and enable communication between EMBArk and worker nodes**	
	EMBArk worker UI	3
	Orchestrator | Receive new workers	3
	Caching in GitHub actions pipeline	2
	Configure worker node	5
	Query worker node information	2
	Prepare upstream pull request	2
	Connect to worker node	2

Real size	Real remaining
0	51
0	51
9	51
14	42
22	28
9	6
1	
2	
3	
3	
2	
1	
5	
1	
2	
3	

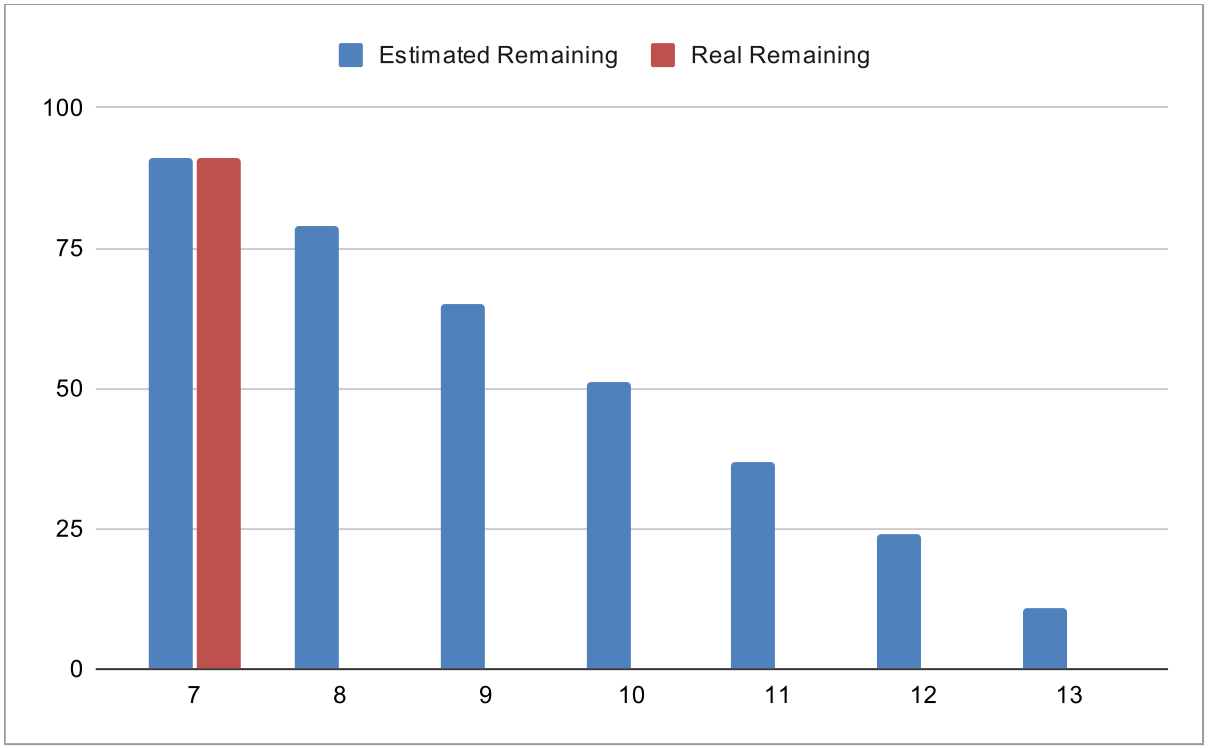
1
2
5
5
5
2
2
3
3
not completed
3
not completed
2
3


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
Release				
Total			91	91
Sprints				
7			12	91
8			14	79
9			14	65
10			14	51
11			13	37
12			13	24
13			11	11
Features				
7				
		Update worker nodes	3	
		Orchestrator | FIFO scheduling	2	
		EMBArk worker UI | Show job id in worker nodes table	2	
		Orchestrator | Query worker pool	2	
		Query worker node information	2	
		UI | Update/Reset	1	
8				
		Add Celery dependency	2	
		EMBArk starts firmware analysis on worker node	2	
		Monitor workers and collect results	3	
		Soft reset worker node	2	
		Hard reset worker node	2	
		Periodic worker information fetch	3	
9				
		Update routinen	8	
		Orchestrator | Abort running firmware analyses	3	
		Trigger orchestrator	3	
10				

	EMBArk ohne Orchestrator	5
	Pull Request	2
	Complete port scan for worker nodes	2
	Priority queue	5
11		
	Refinements	5
	Initialize user, design and build/deploy documentation	3
	Automate worker configuration steps	2
	Pull request: Change requests	3
12		
	Documentation	3
	Final upstream feature PR	5
	Final upstream feature PR: Change requests	5
13		
	Demo preparation	3
	Final upstream wiki PR	5
	Final upstream feature PR: Change requests	3

Real size	Real remaining
	91


Estimated Remaining
Real Remaining
100
75
50
25
0
7
8
9
10
11
12
13

#	Feature Definition of Done	Sprint Release Definition of Done
1	Github actions pipeline runs without errors	Features and changes have been demoed in review
2	If changes are visible to users, documentation is added	Features not covered by unit tests are not negatively impacted by sprints changes
3	Code review passed	
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

Project Release Definition of Done

Build and deployment documentation exists

Software architecture documentation is up to date
Readme is up to date

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

Last Name	First Name	Value		#DIV/0!	#DIV/0!
Meusling	Patrick				
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