OU ASOC

Comprehensive Application Security Analysis

Application Security Orchestration and Correlation

SAST • SCA • Secrets • Quality Gates • SLA

30 Poltavskaya St, P2,, Nizhny Novgorod, 603089, Russia

octounit.ru/en
info@octounit.ru

2026

About us

A team of experts engineering adaptive cybersecurity defence technologies

Our Mission

01

To act proactively and ensure cyber-immunity for our clients

Our Team

02

We are cybersecurity professionals with 10+ years of experience in major international corporations. Our core competencies include SecOps, Threat Intelligence (TI), Incident Response (IR), Red Teaming, and DevOps

Our Approach

03

Our solutions are built upon extensive experience with existing market products and the development of proprietary platforms

300
Pentest projects delivered

5000
Monthly threats analysed

100+
Sources under continuous monitoring, including Dark & Deep Web

~100
Monthly incidents registered

95+
Engagements delivered

15
min response time for OU Intelligence services

25+
Industries successfully covered

Customer industries

Banking | SIS | Telecommunications | Insurance | IT | Healthcare

Services provided

Security assessment

- Web services
- Mobile applications
- Infrastructure
- Source code

Secure Software Development

- End-to-End Implementation
- Strategy development
- Vulnerability Management process design
- SAST implementation

Attack Simulation

- External Pentesting
- Phishing

SOC Consulting

- L1 & L2 as a Service
- SOC evolution strategy
- Process auditing
- Custom detection rule development

Our products

OU Intelligence

Threat Intelligence

Security assessment and continuous security monitoring service

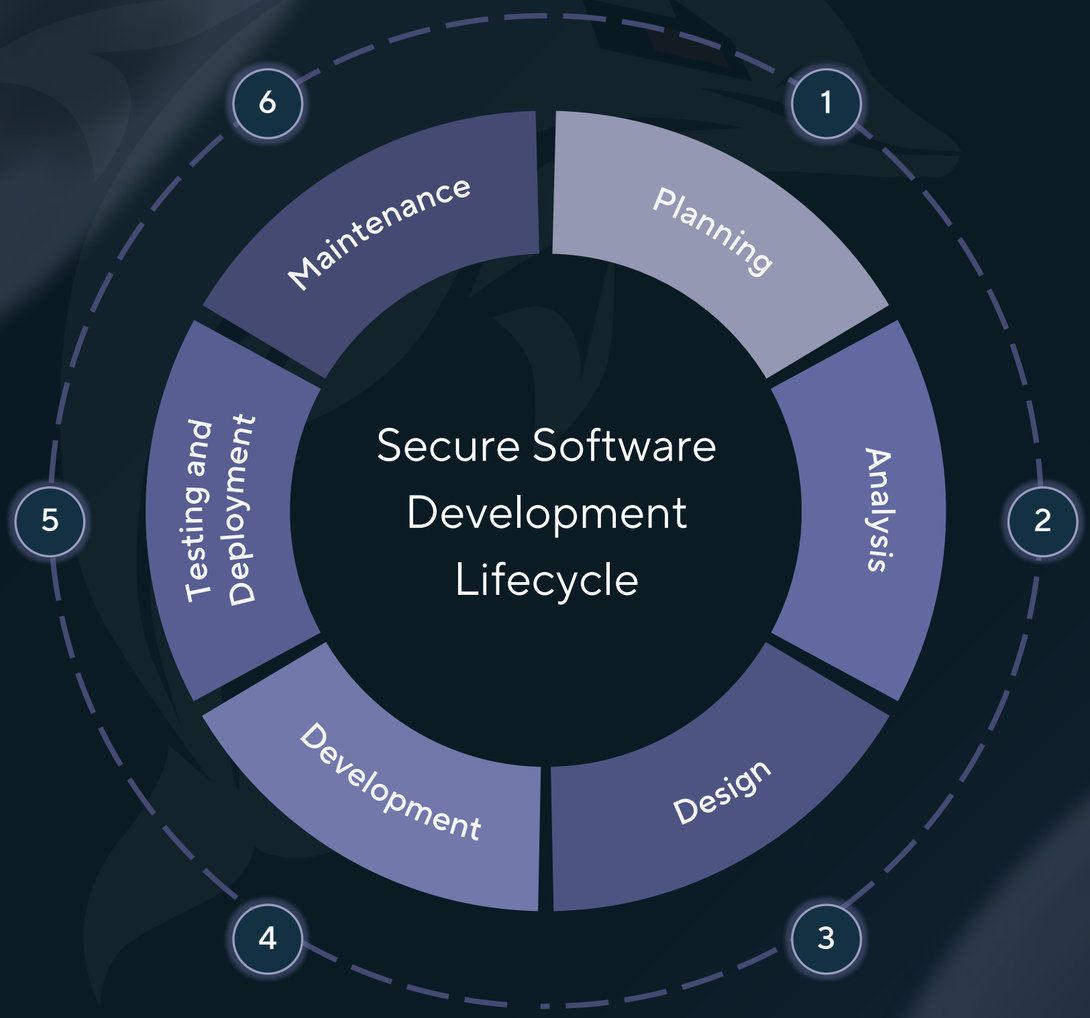
Comprehensive end-to-end application security analysis

OU ASOC

Secure Development

What is SSDLC ?

What is SSDLC ?



SSDLC - is a comprehensive methodology that integrates security practices throughout every phase of the software development lifecycle.

Three Core Principles

Shift-Left

Moving security issue detection to the earliest stages of development.

Automation

Automated security testing.

Continuous Feedback

Constant refinement of security processes based on gathered data and experience

Why implement SSDLC ?

Risk mitigation

A preemptive approach significantly reduces the risk of vulnerability exploitation and data breaches

Release acceleration

Shortening release cycles by integrating security checks directly into development rather than waiting for post-IT testing

Backdoor prevention

Eliminating the risk of using software with malicious implants or backdoors

High Deception Quality

Improving vulnerability detection through multi-layered comprehensive testing

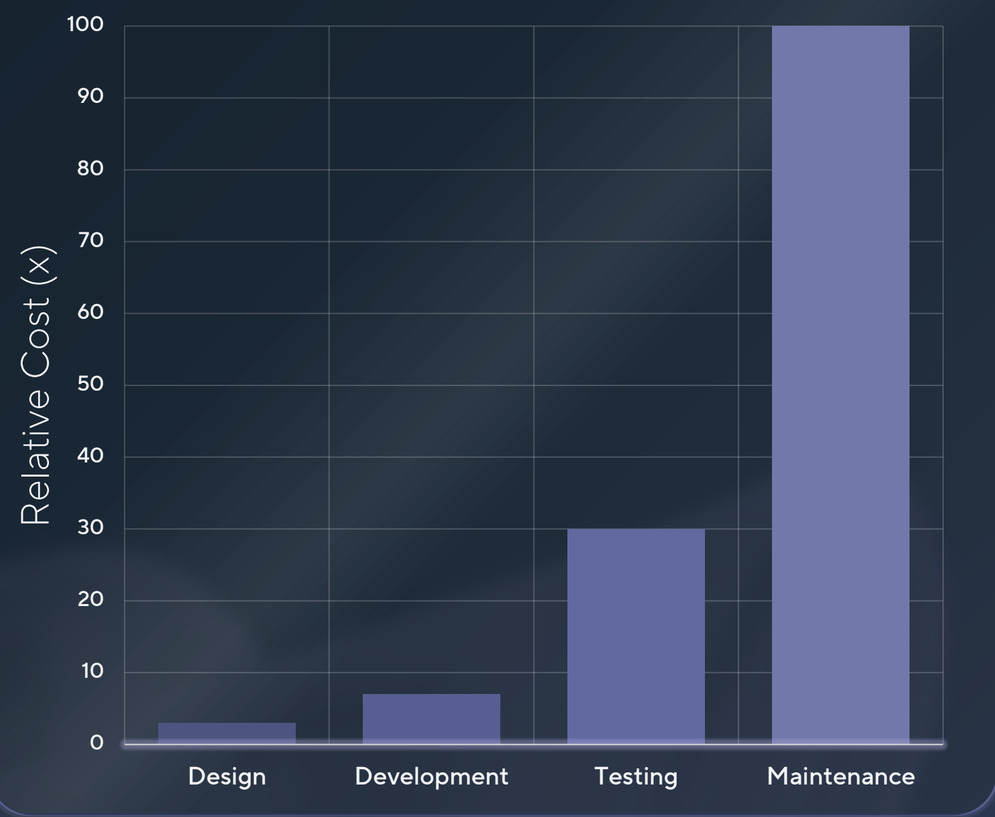
Cost Efficiency

Reducing remediation costs via the "shift-left" approach

Dependency hygiene

Vetting dependencies for undocumented features or unwanted political messaging

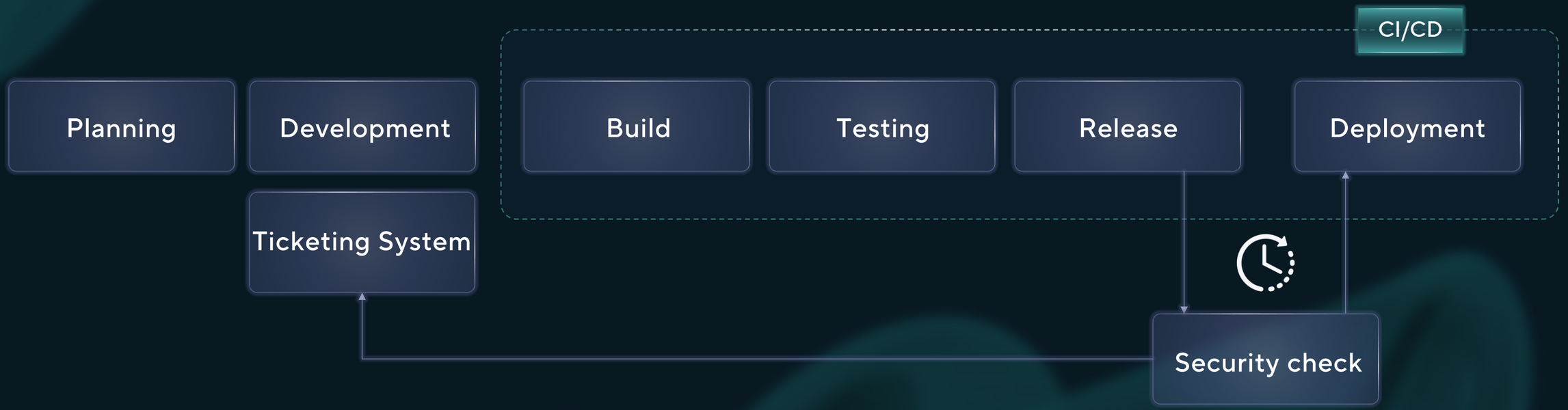
Defect Remediation Cost



The earlier a vulnerability is discovered, the more cost-effective its remediation

Before SSDLC Implementation

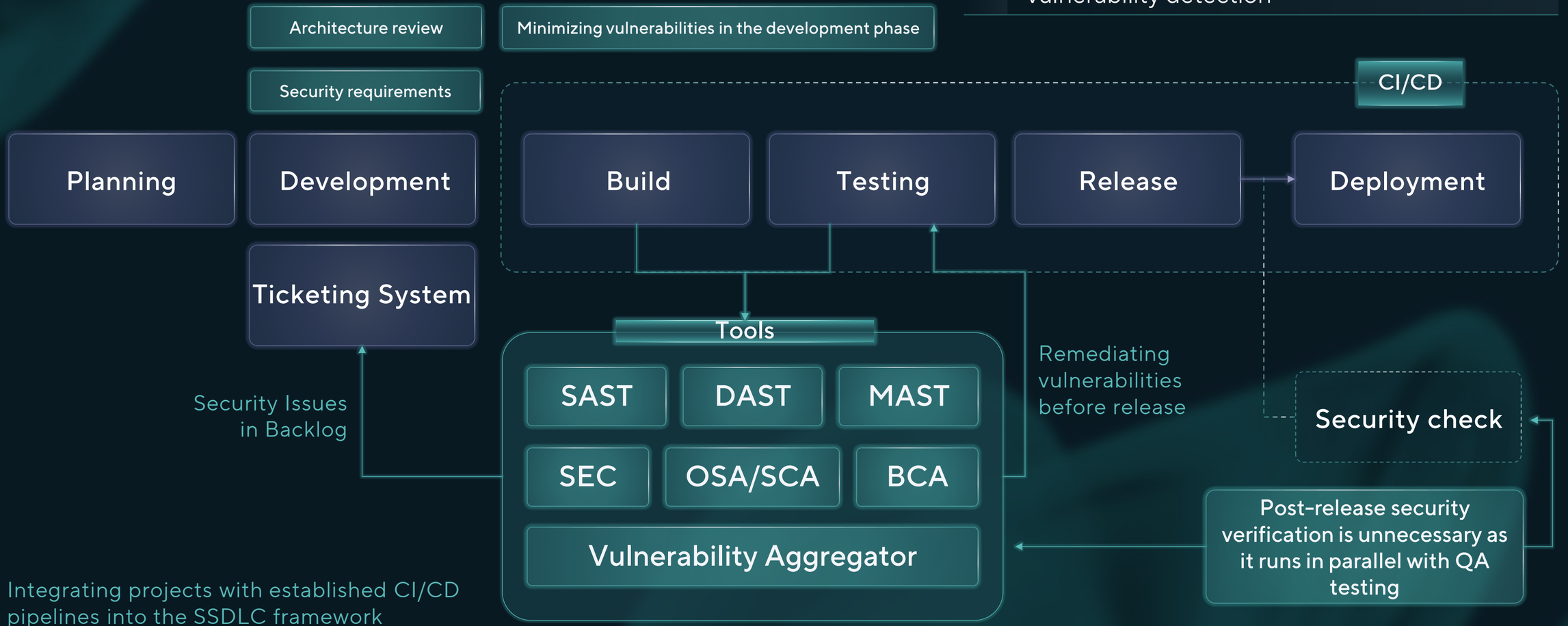
1	Final security checks before go-live
2	Extended project delivery timelines due to multiple passing attempts



Integrating projects with established CI/CD pipelines into the SSDLC framework

After SSDLC Implementation

1	Security checks avoid development bottlenecks by executing in parallel with QA testing
2	Reduced remediation costs through early-stage vulnerability detection

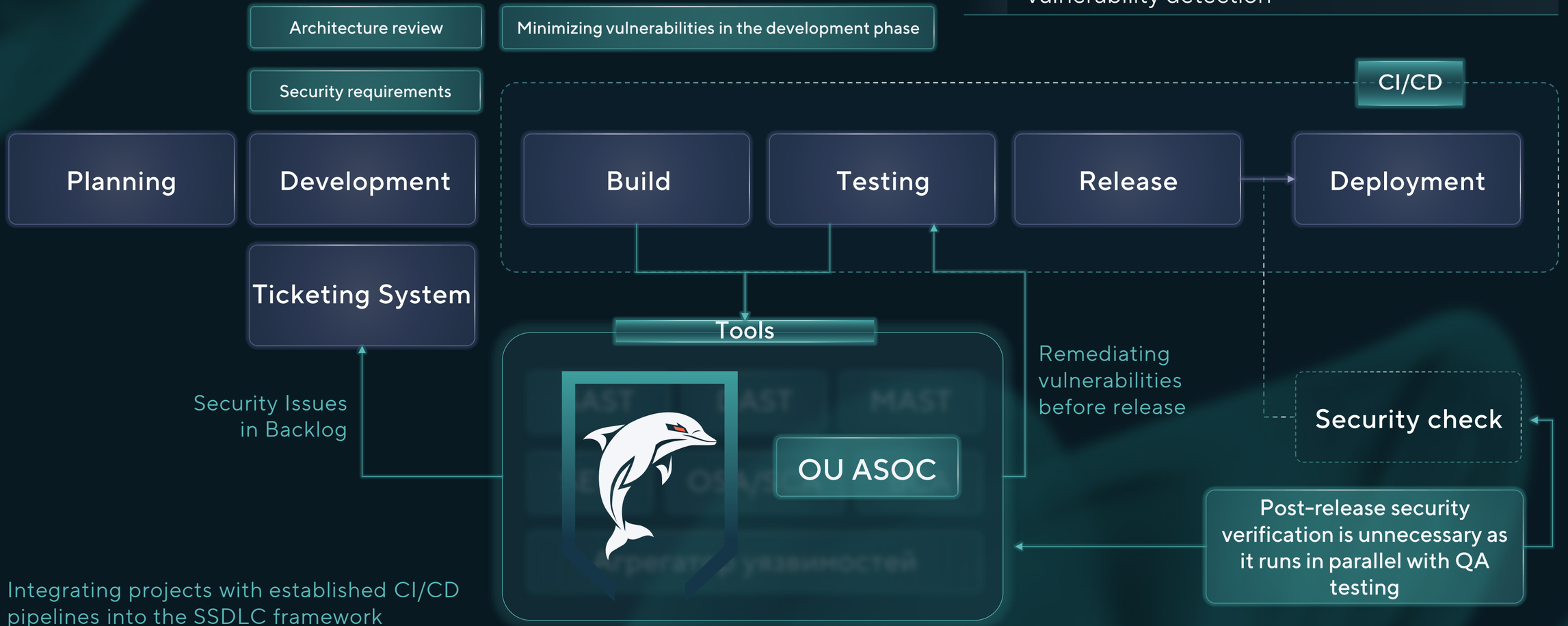


Integrating projects with established CI/CD pipelines into the SSDLC framework

Our solution

After OU ASOC Implementation

1	Security checks avoid development bottlenecks by executing in parallel with QA testing
2	Reduced remediation costs through early-stage vulnerability detection



Integrating projects with established CI/CD pipelines into the SSDLC framework

OU ASOC – The unified security hub

Coverage

SAST	Semgrep, DevSkim, Bearer, GoSec, Bandit and etc.	Secrets	Gitleaks
SCA	Trivy + SBOM + CVE monitoring	Licenses	Identifying restrictive licenses

Core idea

- Consolidate results from 20+ analyzers into a unified format
- Normalize → Deduplicate → Enrich with context
- Automate actions via a visual workflow designer
- Prioritization based on Risk and SLA, not just "quantity"

Data pipeline

Git / CI
OAuth • Webhooks • Schedulers

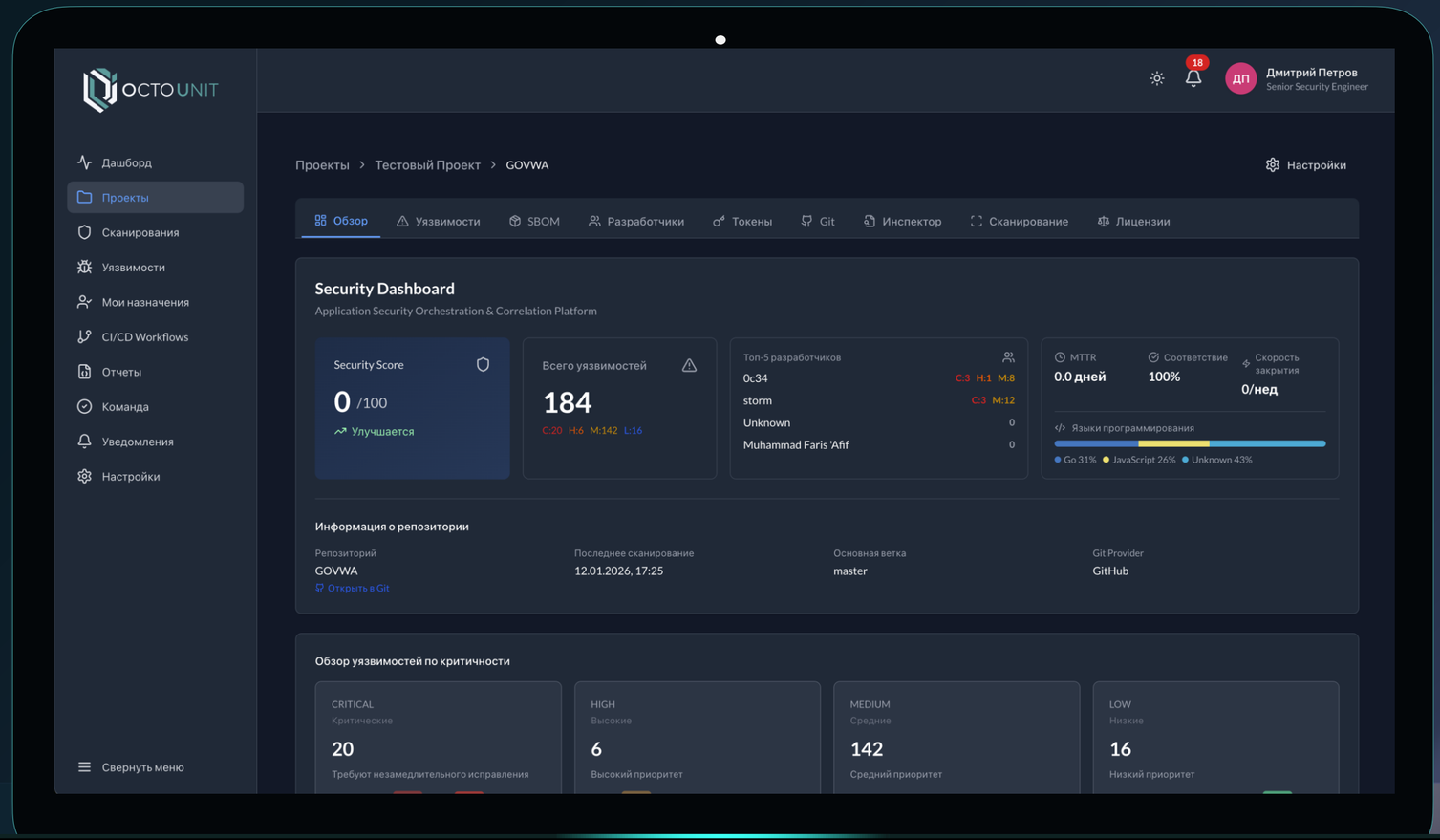
Orchestration
Scanner execution and parallelism

Correlation
Normalization and unique ID assignment

Dedup + Enrich
Deduplication and enrich with context

Governance
RBAC • SLA • Quality Gates

Outputs
UI • Reports • API • Tickets



OU ASOC unifies processes and tools

Secrets Detection

Gitleaks integration with deduplication to prevent token/key leaks

Quality Gates

Custom policies (e.g., Critical=0), severity-based thresholds, and automated escalation

SAST (20+ Analyzers)

Semgrep, DevSkim, Bearer, GoSec, Bandit, SpotBugs
Unified result format

SCA + SBOM

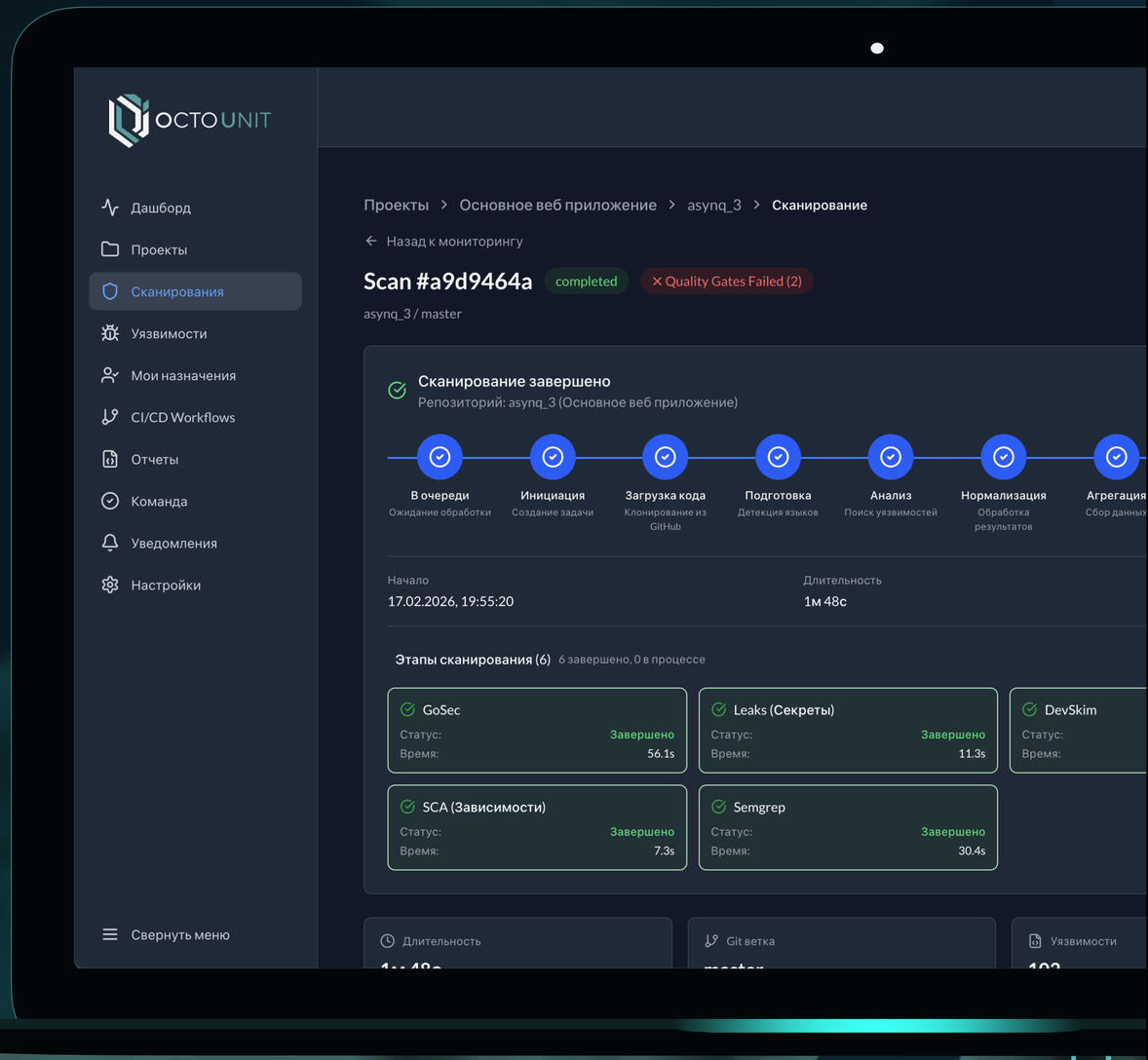
CVE monitoring, license compliance, and transitive dependency analysis (npm, pip, Maven, Go)

Vuln Management

Status tracking, assignments, bulk actions, timeline, and Git blame integration

Reports & Compliance

Executive summary, technical, compliance, trends
PDF/Excel/CSV/JSON



Analyzers

The modular architecture enables easy integration of additional analyzers

Category	Analyzers	Languages
Universal	Semgrep, DevSkim, Bearer	30+ languages
Python	Bandit	Python 2/3
Go	GoSec	Go
JavaScript/TypeScript	ESLint Security, NodeJsScan	JS, TS, Node.js
Java	PMD, SpotBugs, Infer	Java, Kotlin
Kotlin	Detekt	Kotlin
C/C++	Cppcheck, Flawfinder, Infer, Joern	C, C++
PHP	Psalm	PHP 7/8
Ruby	Brakeman	Ruby on Rails
Secrets	Gitleaks	All files
Dependencies	Trivy	All ecosystems

OU ASOC

Core Components

Workflows

Custom workflows: triggers (commits/PR/cron), execution steps (scans/notifications/ticketing), webhooks, and routine automation.

Notifications

Events: new vulnerabilities, scan completion, SLA violations, and Quality Gate failures

Workspaces & Repos

Multi-tenancy, data isolation, GitHub/GitLab repository import, branch/commit/contributor monitoring, API tokens with TTL and granular permissions.

Auth & RBAC

SSO: OAuth2/OIDC/SAML (Yandex, Google, Azure AD, Okta, etc.), roles and detailed permissions, invitations, and activity auditing

Quality Gates & SLA

Severity-based thresholds, blocking insecure releases, remediation deadline monitoring, policy violation reports, and security alerts.

Analytics

Dashboards by severity/author/language/time, MTTR, remediation trends, tool effectiveness, and PDF/CSV/JSON export.

Scanning

On-demand and scheduled scans, branch/commit tracking, real-time status updates, scan history and results comparison, cancellation/restart capability, and S3-based artifact storage

Vulnerability Mgmt

Vulnerability normalization and prioritization, task assignment and commenting, Jira/Yandex Tracker integrations, status synchronization, and remediation recommendations.

Integrations & Extensibility

Git Scheduler, Webhooks/API, plugins and custom scripts, extensibility tailored to the customer's tech stack.

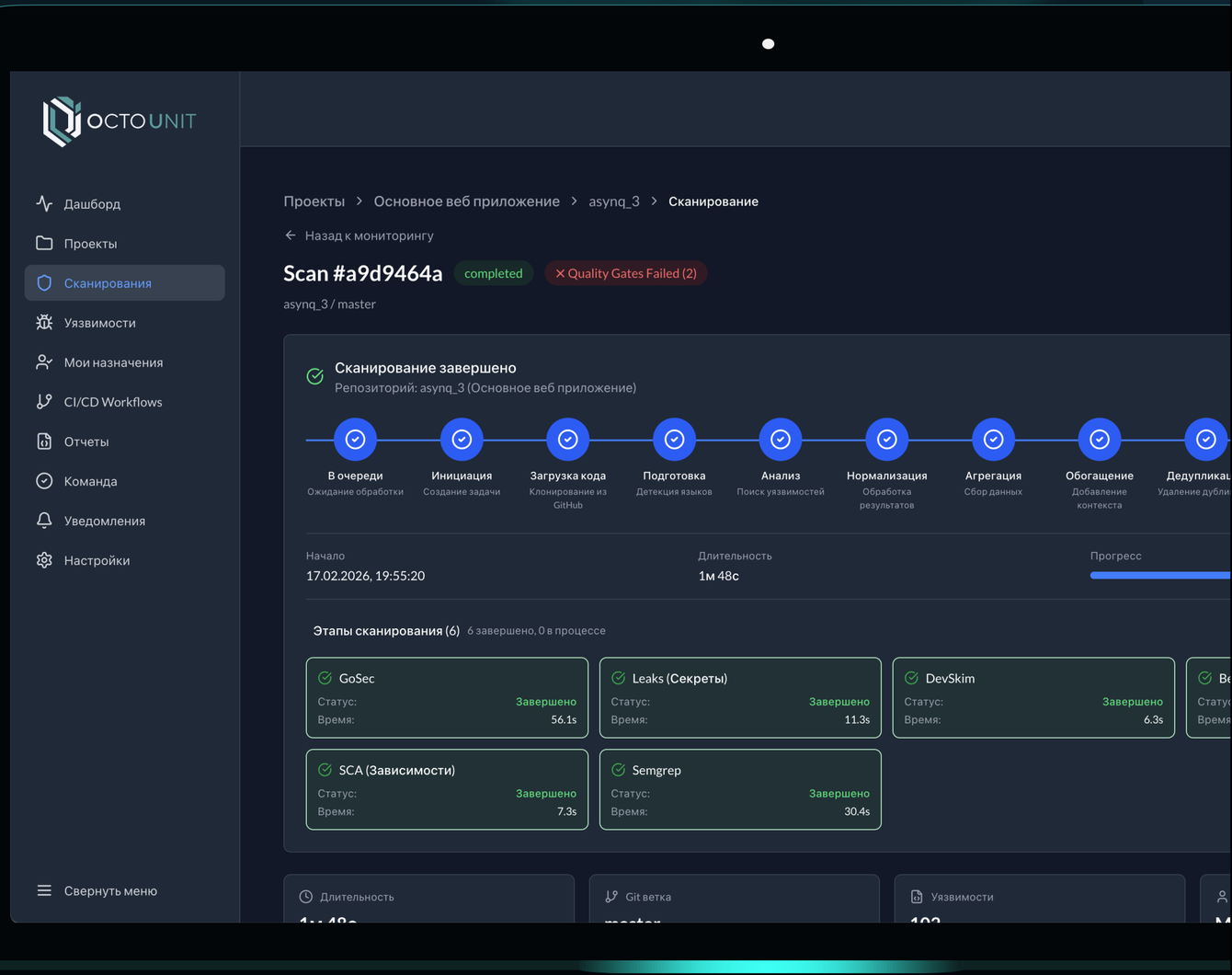
Scanning

Scan execution details & Quality Gates

Full scan tracing
Queue → Analysis → Normalization → Enrichment → Deduplication → Persistence

Tool statuses and per-step execution times; visibility into errors and timeouts (e.g., NodeJsScan)

Dedicated Quality Gate failure status for CI/CD enforcement



Quality Gates and SLA

Quality Gate an automated security verification mechanism that determines whether scan results comply with predefined quality and security standards.

Key capabilities

Flexible verification rules

Severity-based thresholds	configuring the maximum allowable number of vulnerabilities for each criticality level
Security Score	Minimum security score threshold
Custom rules	ability to create custom validation rules

Hierarchical configuration

Level Workspace	global rules for the entire project
Level Repository	per-repository policy customization
Inherit	repositories automatically inherit settings from the workspace level

SLA (Service Level Agreement) — a vulnerability remediation deadline management system featuring automated tracking, escalation, and notifications.

Key capabilities

SLA policies

Severity-based configuration	differentiated remediation timeframes per severity level
Granularity	policies at the workspace or individual repository level
Remediation time	customizable hours/days per vulnerability type

Recommended SLA Deadlines

Severity	Remediation time	SLA
CRITICAL	24 hours	12 hours
HIGH	72 hours	48 hours
MEDIUM	168 hours (7 days)	120 hours
LOW	720 hours (30 days)	120 hours

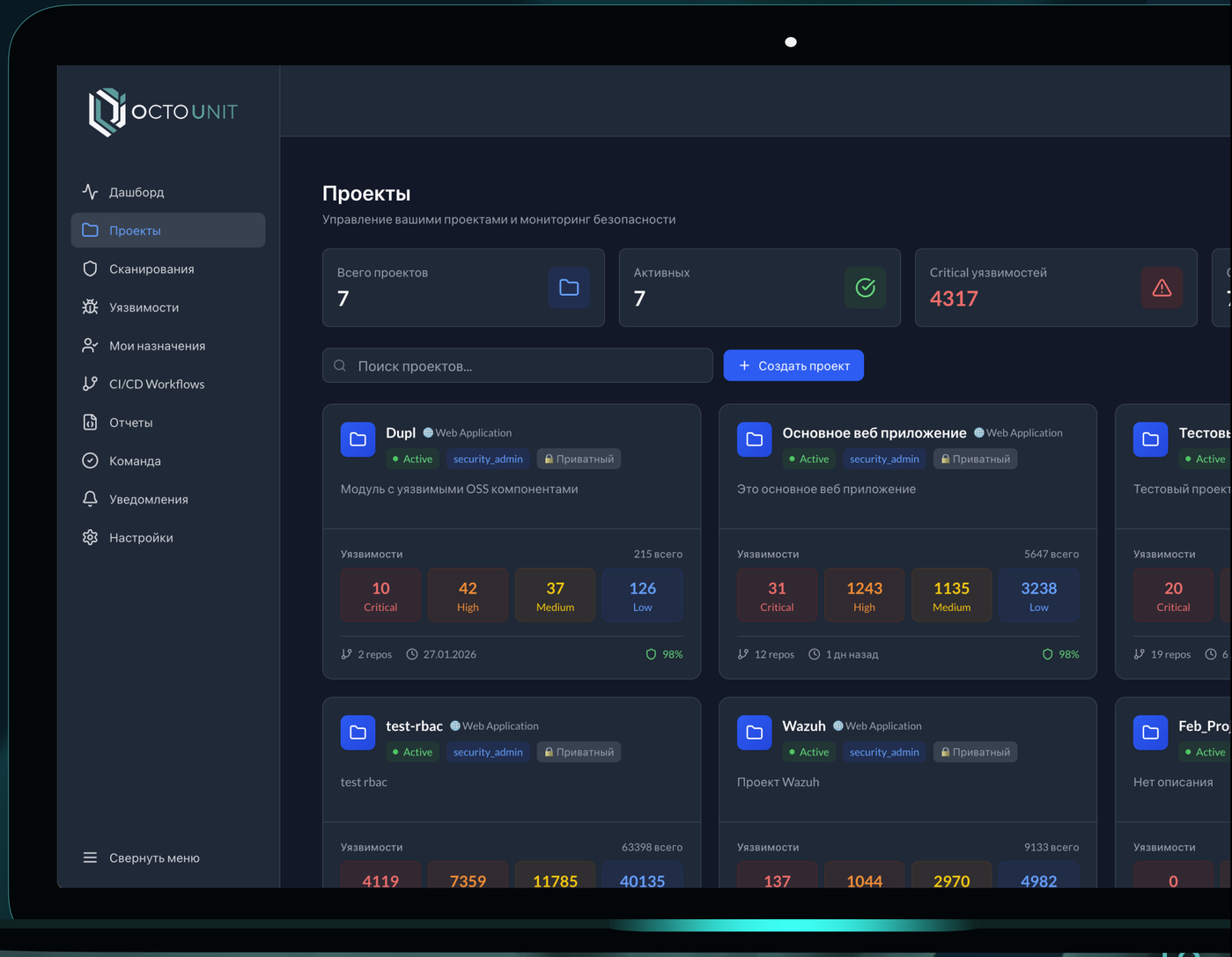
Workspace

Repository list and Risk profile

Repository Cards: status, Git provider, direct link, and vulnerability count by severity.

Repository discovery and creation; rapid overview of security coverage and dev activity.

Enables rapid prioritization of 'on-fire' repositories for immediate response



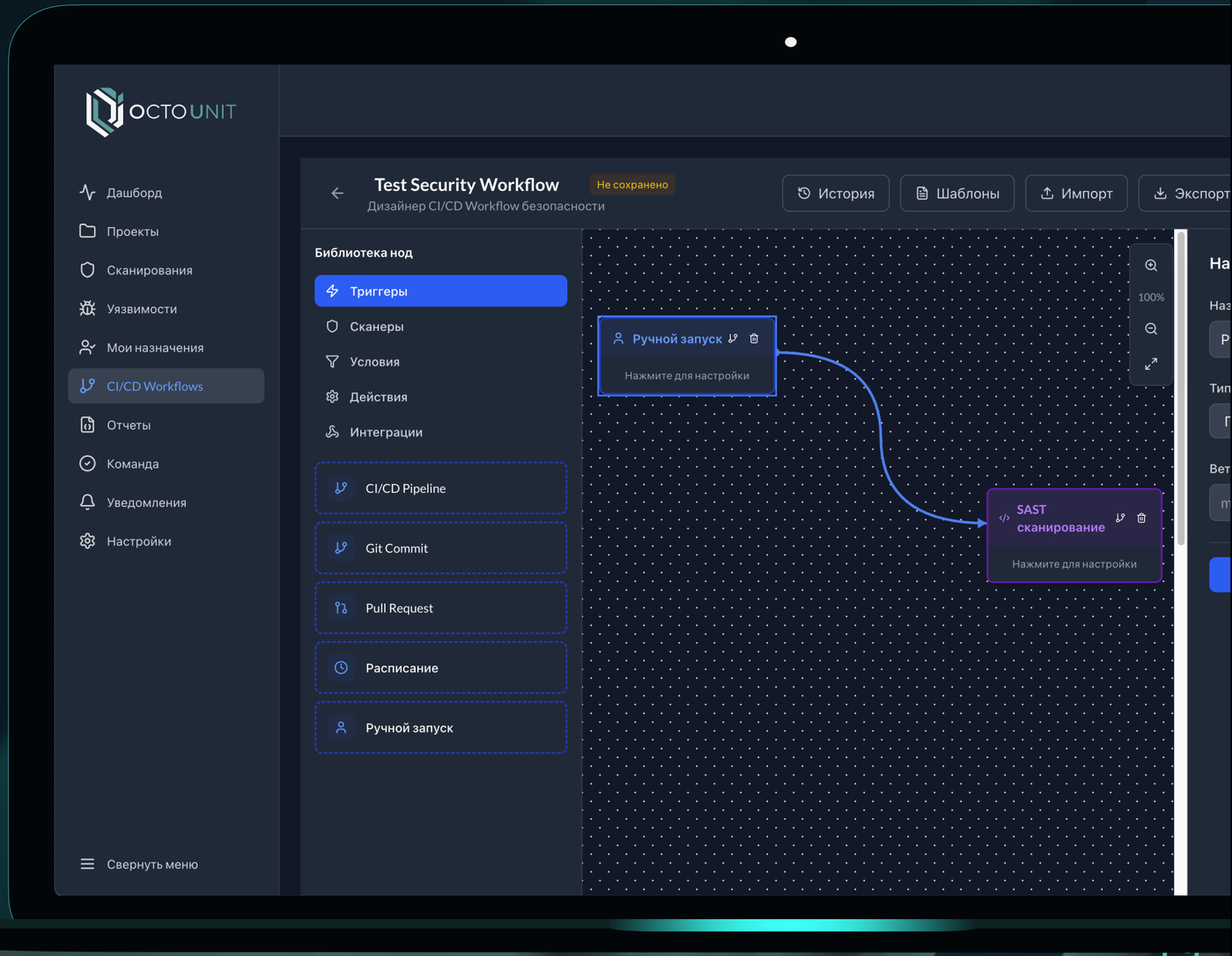
CI/CD Security Workflow

Visual security workflow

Security Workflow Designer:
assembling scenarios from nodes (triggers, scanners, conditions, actions, and integrations)

Flow example:
«On-demand run → SAST scanning».

Lifecycle management:
templates, import/export, saving, and execution



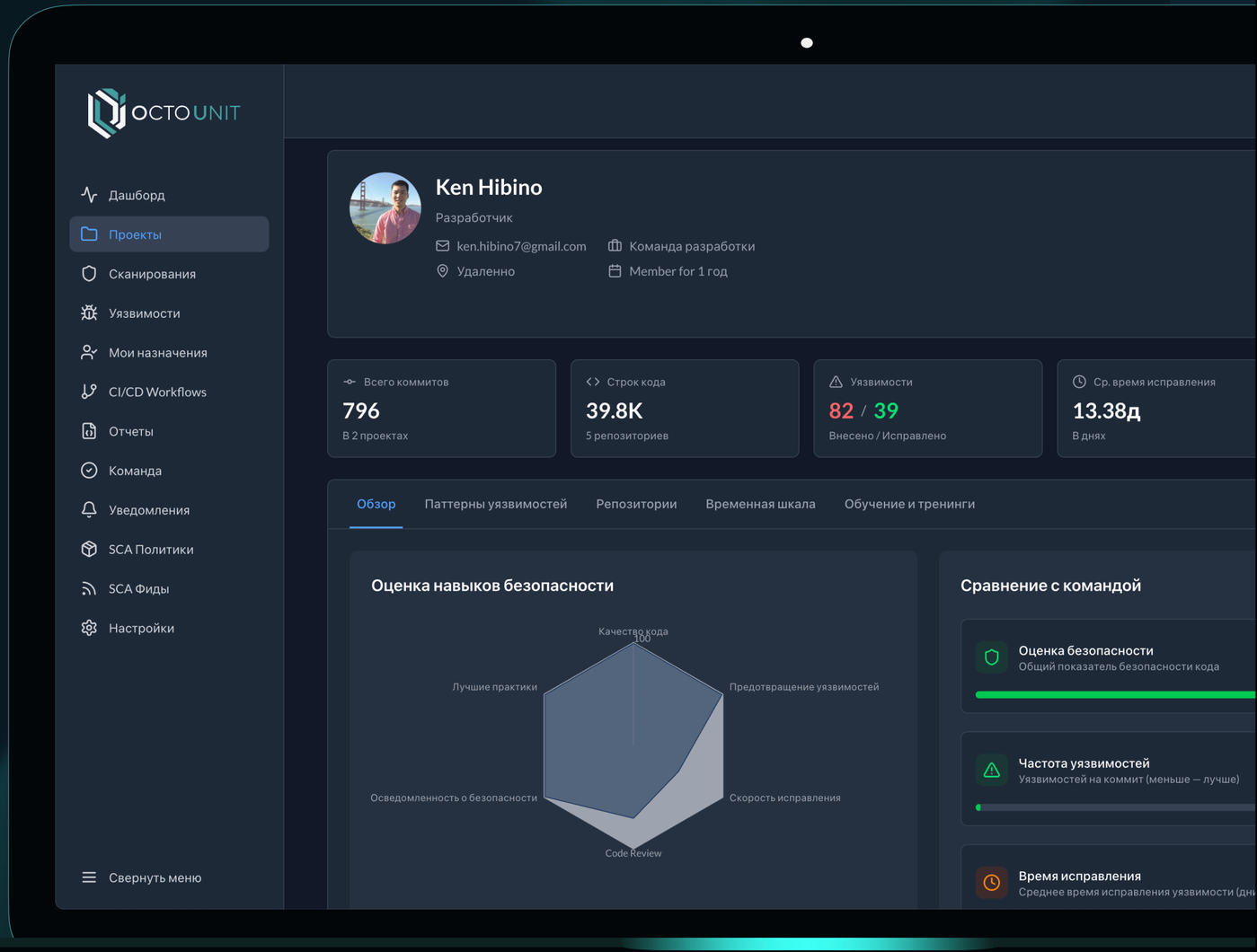
Developer

Profile
and security-performance

Developer Profile: security score, activity metrics (commits/lines of code), and vulnerabilities introduced/remediated.

Team benchmarking and efficiency metrics (e.g., Mean Time to Repair — MTTR)

Foundation for UEBA and secure coaching: identifying behavioural patterns and development opportunities



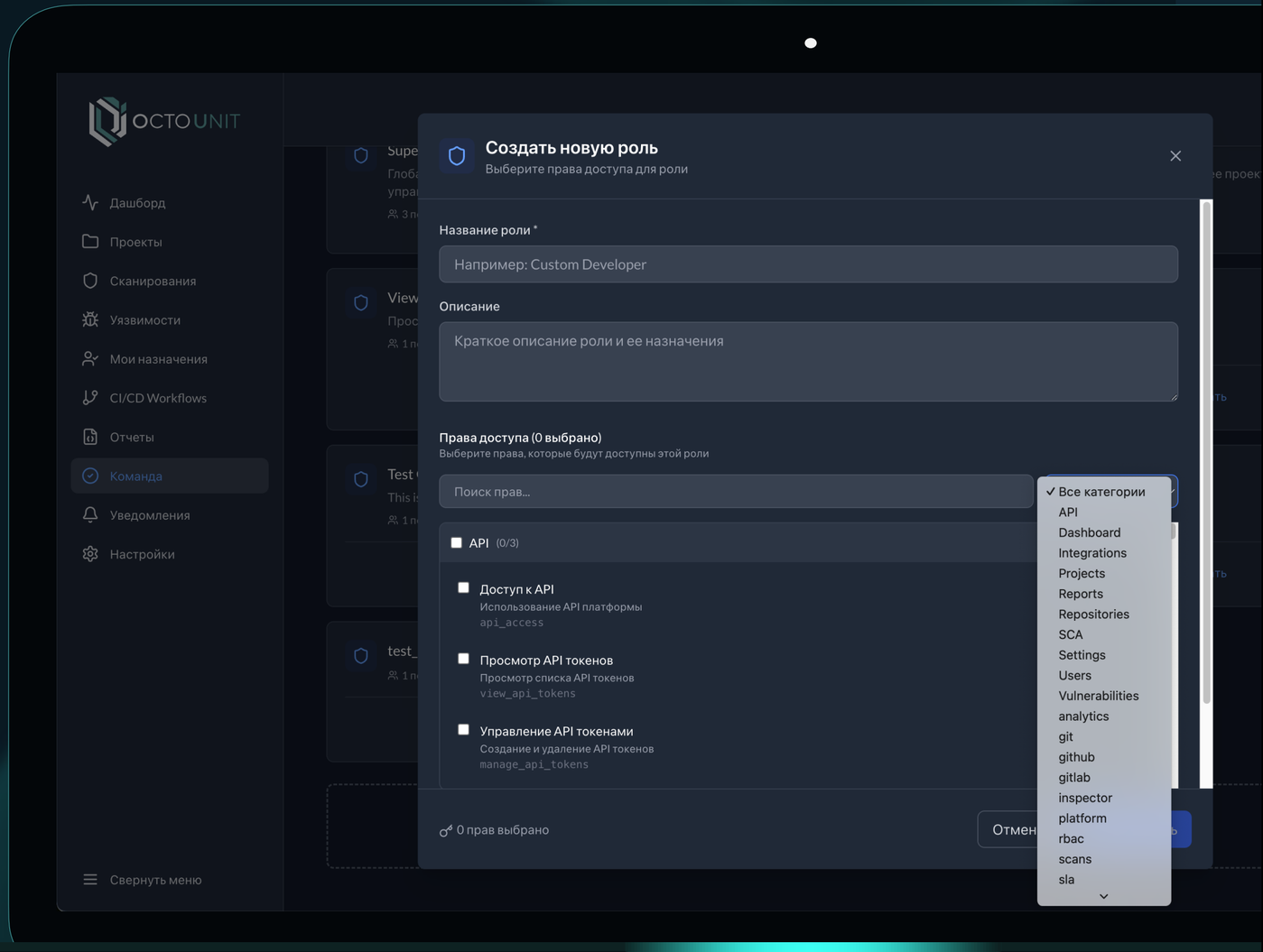
RBAC

Creating new role

Creating custom role: name, description.

Filtering and searching permissions by category (e.g., Repositories, Reports, SCA) for streamlined access control.

Framework for a flexible 'least privilege' access model



Before SSDLC Implementation

The contract fails to account for security requirements

1	Lack of security oversight during the development process
2	Source code is not provided (for certain projects)
3	Contracts do not account for vulnerability remediation

Before SSDLC Implementation

The contract fails to account for security requirements

1	Lack of security oversight during the development process
2	Source code is not provided (for certain projects)
3	Contracts do not account for vulnerability remediation

After OU ASOC Implementation

Security requirements are embedded in the contract and the UAT process

1	Reduced lead time to project completion. Security requirements are established during the contract drafting stage.
2	Reducing the number of vulnerabilities through comprehensive security validation during the UAT phase.

Notification channels

- Individual and team alerts
- Rea-time notifications to dedicated channels
- Third-party system interoperability

Ticketing system integrations

Jira — Automated issue provisioning, bidirectional status sync. Custom fields included.

Event types

- Scan completion
- New critical vulnerabilities
- Weekly reports
- SLA violations
- Quality Gate failures

Executive summary

- Dashboard Metrics
- Analytical Reports
- Security Trends and Remediations Dynamics
- Advanced Data Filtering

CI/CD Workflow

- Visual Workflow Designer
- Node Types: Triggers, Scanners, Conditions, Actions, IntegrationsExecution triggers
- Execution statuses

Authentication and authorization

Method	Decription
Default	Built-in authentication
LDAP/AD	Active Directory integration
SSO (OIDC)	Keycloak, Azure AD, Okta
SSO (SAML)	ADFS
OAuth 2.0	OAuth 2.0 Provider

Final deliverables

Role-Based Access Model

Role	Description	Permissions
Security Admin	Full access	57
Security Analyst	Vulnerability Management	22
Team Lead	Team management	22
Auditor	Read-only & Export Access	17
Developer	Project-level access	12
Viewer	Basic View	3

Demo

Getting started is easy!

01 Select your pilot type: On-Premise / Cloud

02 Selecting up to 3 projects for the pilot program

03 Distribution package delivery and installation assistance

04 Assistance with account setup and project initialization

05 Full trial access to the system for one month

info@octounit.ru
octounit.ru/en

Contact us

We look forward to providing further information regarding our company, products and services.