

DevTrack

Development Performance Monitoring System

We develop, integrate, and maintain IT solutions

Development **Performance** Monitoring System:

- ✓ Brings transparency to the development process
- ✓ Analyzes team performance
- ✓ Identifies high-performers
- ✓ Pinpoints production bottlenecks
- ✓ Provides a methodology based on global best practices

Reduces development time by up to

40%

Why does development slow down?

- ❗ Up to **30%** of work time is spent on unproductive tasks
- ❗ **Poor task prioritization** leads to idle time and increased time-to-market
- ❗ Team leads rely on **verbal status** reports from developers
- ❗ There is a risk of engaging employees with **double employment** on projects
- ❗ **Poorly streamlined team processes** complicate workflows, increase delivery time, and reduce quality

The solution – DevTrack

By collecting metrics from code repositories, task trackers, CI/CD tools, test management systems, and knowledge bases

DevTrack delivers transparent analytics of development performance metrics

and provides timely alerts to management about potential issues

What can DevTrack do?



- ✓ Collect data from popular code repositories, CI/CD tools, task trackers, test management systems, and knowledge bases
- ✓ Visualize standard development performance metrics
- ✓ Create custom metrics and analytical reports
- ✓ Integrate with local (Russian) task trackers and other data sources
- ✓ Send team performance reports on a schedule
- ✓ Provide timely alerts about issues

Performance Metrics Analyzed

Developer

- Number and frequency of PRs / commits
- PR size (lines of code)
- Time to first PR review assignment
- PR review duration until first approval
- Time from first approval to merge
- Total PR review duration
- PR maturity (% of rework following review)
- Merge frequency
- Code quality (% of rework)
- Refactoring ratio
- Code security (vulnerabilities identified)
- Deployment time (from merge to release)
- Percentage of commits/PRs linked to a task
- Impact of using AI assistants on development

* Metric set shown is not exhaustive.

Tester

- Number and frequency of defects found
- Percentage of critical defects missed
- Defect density
- Mean time to defect detection
- Number and frequency of new test cases
- Number and frequency of new documents
- Number of new/changed lines/words in documents
- Number of changed test case steps
- Number of test case steps executed
- Number and frequency of new automated tests
- Test case quality (% of rework)
- Automated test quality (% of rework)

Analyst

- Number and frequency of new documents and articles
- Number of new/changed lines/words in documents and articles
- Quality of documents and articles (% of rework)
- Number and frequency of change requests (CRs) for new functionality
- Communication effectiveness
- Requirements stability
- Technical readiness index
- Predictability index
- Risk prevention / mitigation index

Performance Metrics Analyzed

Team Member

- Time spent on a task
- Task estimate accuracy (underestimate / overestimate)
- Percentage of commitments met
- Number and percentage of estimated tasks in progress
- Derived productivity indices (by role)
- Derived quality indices (by role)
- Learning and mentoring index

* Metric set shown is not exhaustive.

Team

- Planning accuracy (underestimate / overestimate)
- Percentage of correctly linked tickets (to Story, to Epic)
- Percentage of tasks in progress that have an estimate
- Percentage of tasks in progress that have an assignee
- Percentage of requirements covered by tests
- Percentage of code covered by unit tests
- Percentage of tests and Quality Gates passed
- Technical debt percentage

DORA

- Lead time for changes
- Development cycle profile broken down by stage (in hours, story points, percentages)
- Deployment frequency
- Time to restore service
- Rollback percentage
- Change failure rate

Solution modules

Connectors to Data Sources

- Ability to connect data from alternative task trackers, knowledge bases, test management systems, and other sources

Reporting

- Standard reports
- Ability to export data to other BI tools
- Ability to configure custom analytics

Notification Center

- Scheduled reports
- Event notifications

AI Assistant

- Metric interpretability
- Recommendations for performance improvement
- Anomaly detection and analysis
- No need to transfer data externally



Metrics must be analyzed comprehensively and complemented with feedback and quality assessment

Implementation results

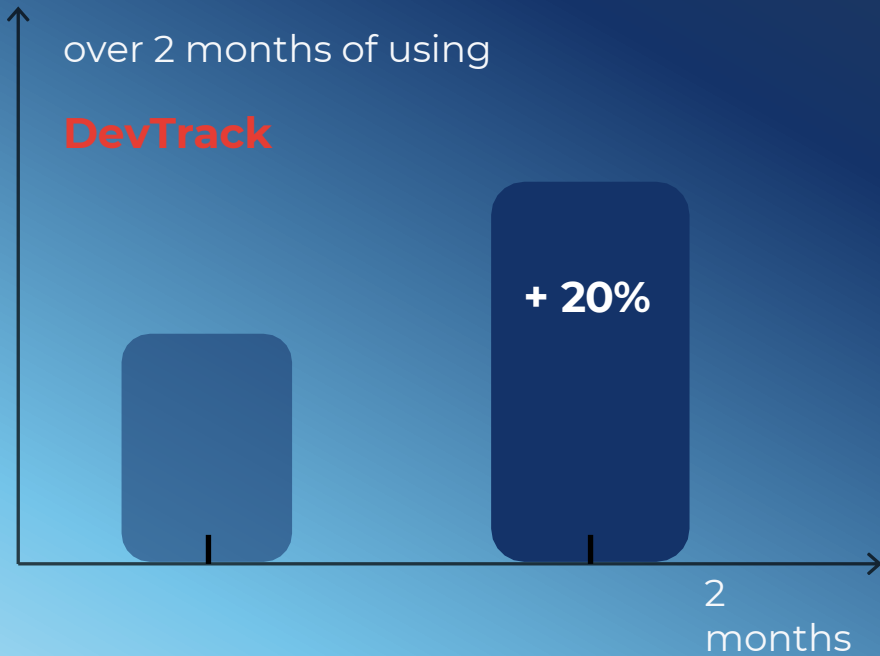
Development team
efficiency has increased

Team
500+
engineers

on **20%**

over 2 months of using

DevTrack



By 2027, the use
of engineering analytics
platforms to improve
developer productivity will
grow **to 50%**



*Gartner, Market Guide to Software
Engineering Intelligence Platforms*

Where to start?

1

Audit

- Upload existing statistical data into DevTrack
- Initial findings and recommendations

3

Analyze Results

- Apply recommendations
- Initial findings and progress against success criteria
- Final report and go / no-go decision for further rollout

2

Pilot

- Identify stakeholders
- Document success criteria and key performance indicators (KPIs)
- Deploy DevTrack within the organization's perimeter
- Collect and analyze metrics
- Validate data and compare against benchmarks
- Formulate recommendations

Implementation

Deploy **DevTrack** organization-wide

4

See how it works for your business

Let's start with an introductory meeting
- we'll discuss your goals and explore the possibilities

Introductory Meeting

Discuss your objectives, demonstrate capabilities, assess applicability

1-2 hours • Free

Solution Demo

Show functionality with examples, answer your questions

2-3 hours • Using your data

Pilot Project

Deploy on a limited perimeter, train your team, evaluate results

2-4 weeks • First results

Rollout

Deploy organization-wide, technical support, continuous improvement

Ongoing • SLA 99.9%

Schedule a meeting

Get a consultation

Ask a question

