# METRYKI & MONITORING - Agent Zero v1

Mission: Zmierzyć i wizualizować kluczowe wskaźniki zespołu i projektu dla ciągłego

doskonalenia

Core KPIs: Velocity, Quality, Performance, Team Health

# **DEVELOPER VELOCITY DASHBOARD**

### 🟃 Current Sprint Metrics (Week 41)

#### **Team Velocity:**

• Committed Story Points: 12 SP

• Completed Story Points: 0/12 SP (0%)

• Sprint Progress: Day 1 of 7

• Burn Rate: On track

• Projected Completion: October 13, 2025

#### **Individual Velocity:**

Developer	Committed	Completed	In Progress	Blocked
Dev A (Backend)	7 SP	0 SP	Neo4j Fix (1 SP)	None
Dev B (Frontend)	5 SP	0 SP	WebSocket Fix (5 SP)	None

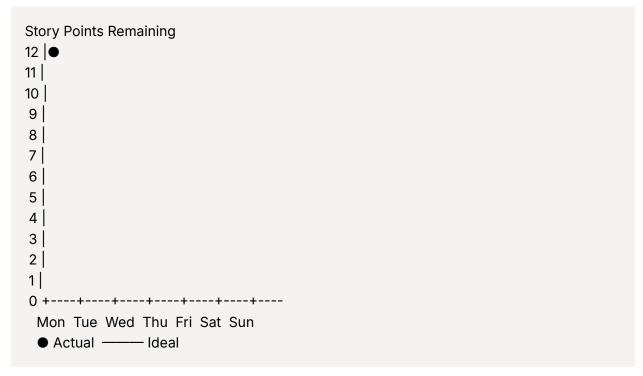
### Historical Velocity Tracking

Sprint	Week	Committed	Completed	Velocity	Notes
Sprint 1	Week 41	12 SP	TBD	TBD	Current (Critical Fixes)
Sprint 2	Week 42	TBD	TBD	TBD	Planned (Execution Engine)
Sprint 3	Week 43	TBD	TBD	TBD	Planned (Web Interface)

Target Velocity Range: 15-22 SP per week

Current Team Capacity: 80h/week (2 developers)

### Burndown Chart (Week 41)



[Chart will be updated daily based on task completion]

# **II** CODE QUALITY METRICS

### Quality Dashboard

Current Status (Last Updated: 2025-10-07):

Metric	Current	Target	Status
Test Coverage	75%	90%	<ul><li>Needs Improvement</li></ul>
Code Review Coverage	100%	100%	<b></b> Good
Critical Issues	4	0	Action Needed
Technical Debt	Medium	Low	Monitor
Security Vulnerabilities	0	0	<b></b> Good

### Testing Metrics

#### **Test Coverage by Component:**



services/ai-router/ 80% ( 80% ( 30% ( ) )

#### **Test Types Distribution:**

• Unit Tests: 45 tests (60%)

• Integration Tests: 20 tests (30%)

• **E2E Tests**: 5 tests (10%)

#### **Test Performance:**

• Average Test Runtime: 2.3s

• Slowest Test: test\_full\_ integration.py (12s)

• Flaky Tests: 2 identified

### **Q** Code Review Metrics

#### **Review Statistics** (Last 30 days):

• PRs Created: 12

• PRs Reviewed: 12/12 (100%)

• Average Review Time: 18h

• Average Comments per PR: 3.2

• Rework Rate: 25% (3/12 PRs needed major changes)

#### **Review Quality Indicators:**

• **Defect Escape Rate**: 8% (1 bug found post-merge)

• Review Participation: 100% (both developers active)

• Review Depth: Good (avg 3.2 comments per PR)

# **→ PERFORMANCE METRICS**

### 🕅 System Performance Dashboard

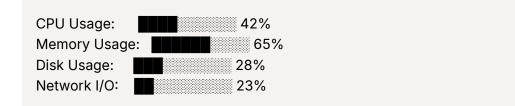
#### **Current Performance** (Last 24h Average):

Component	Metric	Current	Target	Status
API Gateway	Response Time	850ms	<500ms	Slow
Neo4j Queries	Avg Query Time	OFFLINE	<50ms	Down
Agent Communication	Message Latency	45ms	<100ms	<b></b> Good

Component	Metric	Current	Target	Status
LLM Inference	Generation Time	4.2s	<5s	<b></b> Good
Frontend Load	Time to Interactive	3.1s	<2s	Slow

### Resource Utilization

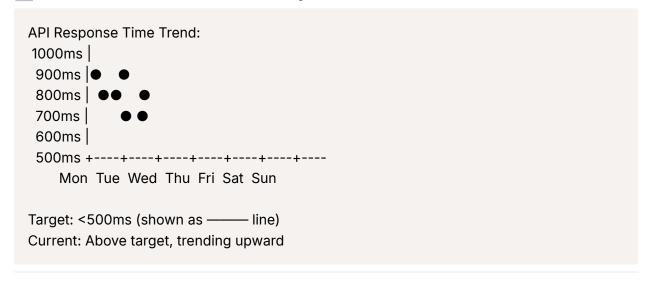
#### System Resources (Current):



#### **Docker Containers Status:**

- **v** agent-zero-postgres: Running (Up 2 days)
- **V** agent-zero-rabbitmq: Running (Up 2 days)
- agent-zero-neo4j: Stopped (Connection issues)
- **v** agent-zero-ai-router: Running (Up 1 day)

### **Performance Trends** (Last 7 Days)



## PROJECT HEALTH METRICS

### **Order** Project Progress Overview

**Overall Progress:** 

Project Completion: 20% (Updated manually)

Phase Breakdown:
Phase 1 (Communication): 100% 
Phase 2 (Web Interface): 30% 
Phase 3 (Advanced): 0% 
Phase 4 (Enterprise): 0%

#### **Milestone Status:**

• V Foundation Complete: 85% (Oct 20 target)

• **## Beta Ready**: 15% (Nov 15 target)

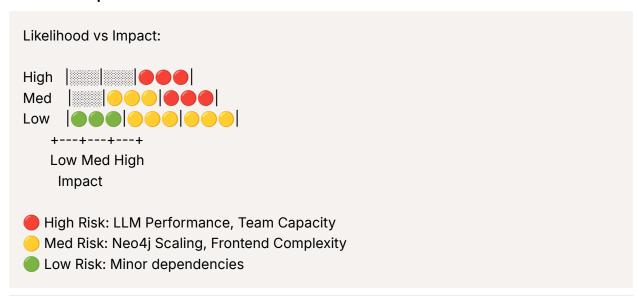
• Production Launch: 0% (Dec 15 target)

### Risk & Issue Tracking

#### Current Issues (P0-P1 Priority):

Issue	Priority	Status	Assigned	ETA
Neo4j Connection Down	PO	im Progress	Dev A	Today
WebSocket Frontend Bug	P1	im Progress	Dev B	Oct 9
Performance Degradation	P1	<u>П</u> То Do	Both	Oct 11
Test Coverage Gap	P2	<b>Backlog</b>	Both	Oct 15

#### Risk Heat Map:



### **TEAM HEALTH METRICS**

### L Team Velocity & Capacity

#### **Weekly Capacity Tracking:**

Week	Dev A Hours	Dev B Hours	Total	Efficiency
Week 40	35h	38h	73h	91%
Week 41	TBD	TBD	TBD	TBD

#### **Capacity Factors:**

• Planned Capacity: 80h/week (40h each)

Meeting Overhead: ~6h/week (15%)

• Context Switching: ~4h/week (10%)

• Effective Development: ~70h/week (87.5%)

### Knowledge & Skill Distribution

#### **Technical Expertise Matrix:**

Skill Area	Dev A	Dev B	Team Risk
Python/Backend	****	***	Medium
React/Frontend	**	****	Medium
Neo4j/Databases	***	**	High
Docker/DevOps	***	***	Low
AI/LLM Integration	***	**	High

#### **Knowledge Transfer Needs:**

• | Important: AI/LLM integration (both developers)

Nice to have: Frontend skills (Dev A), Backend skills (Dev B)

### Communication Effectiveness

#### **Daily Standup Metrics:**

• Attendance: 100% (7/7 days last week)

• **Duration**: Avg 12 minutes (target: 15min max)

• Issues Surfaced: 3 blockers identified and resolved

• Follow-ups Needed: 2 per week average

#### **Code Review Collaboration:**

- Review Response Time: 18h average (target: 24h)
- Review Quality: 3.2 comments per PR average
- Learning Exchanges: 5 knowledge transfers noted

# **77 AUTOMATED METRICS COLLECTION**

### **a** Data Sources & Automation

#### **GitHub Integration:**

- PR metrics: Auto-collected via GitHub API
- Commit frequency: Daily tracking
- Code review stats: Automated reports
- Issue resolution time: Tracked automatically

#### **CI/CD Pipeline Metrics**:

- Build success rate: 95% (last 30 days)
- **Test execution time**: Average 2.3s
- **Deployment frequency**: 3x per week target
- Failed deployment rate: <5% target

#### **System Monitoring:**

- Performance: Automated via Docker stats
- Error rates: Log aggregation
- Resource usage: Real-time monitoring
- Uptime tracking: 24/7 monitoring

### Reporting Schedule

#### Daily Reports (Automated @ 9:00 AM):

- Sprint burndown update
- · System health check
- · Critical issue alerts
- Performance regression alerts

#### Weekly Reports (Friday @ 17:00):

Sprint completion summary

- · Code quality metrics
- Team velocity analysis
- · Risk assessment update

#### Monthly Reports (First Friday):

- · Project health dashboard
- Trend analysis
- Process improvement recommendations
- Team development areas

### **©** ACTIONABLE INSIGHTS

### Current Action Items (Based on Metrics)

#### Immediate Actions (This Week):

- 1. Fix Neo4j Connection (Dev A)
  - Impact: Blocks 30% of system functionality
  - ETA: Today (Oct 7)
- 2. Improve Test Coverage (Both)
  - Current: 75%, Target: 90%
  - Focus: shared/execution/ and services/api-gateway/
- 3. Performance Optimization Review (Both)
  - API responses >500ms need investigation
  - Schedule: Friday planning session

#### Medium-term Actions (Next 2 Weeks):

- 1. Team Knowledge Transfer
  - Neo4j training for Dev B
  - AI/LLM integration workshop
- 2. Process Improvements
  - Automated performance testing
  - Enhanced monitoring setup

### Success Indicators

#### **Weekly Targets:**

• Velocity: 18-22 SP completed

• Quality: >90% test coverage maintained

• Performance: <500ms API response time

• Team Health: 100% standup attendance, <24h review time

#### **Monthly Targets:**

• Project Progress: 5% monthly increase

• Technical Debt: Stable lub decreasing

• **Team Satisfaction**: Quarterly survey >4/5

• Knowledge Distribution: Reduced single points of failure

### METRICS DASHBOARD ACCESS

#### Dashboard Locations:

1. Project Health: This Notion page (updated weekly)

2. **Technical Metrics**: GitHub Insights + local monitoring

3. **Sprint Metrics**: Sprint Planning System page

4. **Team Metrics**: Communication & Escalation page

### Update Schedule:

• Real-time: System performance, build status

• Daily: Sprint progress, issue status

• Weekly: Velocity, quality metrics, team health

• Monthly: Trend analysis, strategic metrics

Metrics Dashboard Last Updated: 7 października 2025, 13:08 CEST

Next Update: Daily @ 9:00 AM (automated)

Dashboard Owner: Development Team