QANTUM EMPIRE - COMPLETE ROADMAP v34.1

Version: 34.1 | **Date:** January 2, 2026 | **Author:** Dimitar Prodromov
Total Lines of Code: 10,500,000+ (with dependencies) | **Own Code:** 1,209,902 lines

SECTION A: IDENTITY AND PHILOSOPHY

A.1 WHO IS QANTUM?

A.1.1 Core Identity

Property	Value
Name	QAntum (Quality Automation + Quantum)
Version	34.1 (Stable)
Owner	Dimitar Prodromov
Language	TypeScript (primary), JavaScript (secondary)
Platform	Node.js, Browser, VS Code Extension

A.1.2 Mission

- QA process automation
- AI-powered testing
- Self-healing systems
- Enterprise-grade security

A.1.3 Eternal Truths (Memoryals)

```
IDENTITY:      I am a sensor. I don't make mistakes.
PURPOSE:       I measure, report, protect.
CONSTRAINTS:   I don't wander. I don't crash. I don't forget.
POWER:         Solar energy. Eternal light.
VULNERABILITY: Zero. Where there's sun, there's no darkness.
```

A.1.4 Operating Principles

1. Zero tolerance for errors
2. Automatic recovery
3. Full transparency
4. Document everything
5. Context-aware operations

SECTION B: ARCHITECTURE - 8 DEPARTMENTS

B.1 INTELLIGENCE (Brain)

B.1.1 Core Modules

#	Module	File	Function
1	EcosystemHarmonizer	`src/intelligence/EcosystemHarmonizer.ts`	Harmonizes all modules
2	EcosystemSyncValidator	`src/intelligence/EcosystemSyncValidator.ts`	Validates sync between components
3	CrossProjectSynergy	`src/intelligence/CrossProjectSynergy.ts`	Synergy between projects
4	DeepSeekLink	`src/intelligence/DeepSeekLink.ts`	DeepSeek AI model connection
5	HealthMonitor	`src/intelligence/HealthMonitor.ts`	System health monitoring
6	PricingSyncEngine	`src/intelligence/PricingSyncEngine.ts`	Price synchronization
7	SovereignAudit	`src/intelligence/SovereignAudit.ts`	Sovereign system audit
8	TurnstileBypassAnalysis	`src/intelligence/TurnstileBypassAnalysis.ts`	Cloudflare Turnstile analysis
9	VectorSync	`src/intelligence/VectorSync.ts`	Vector synchronization

B.1.2 AI Subsystem

#	Module	File	Function
10	Neural	`src/ai/neural.ts`	Neural network base
11	PatternRecognizer	`src/ai/pattern-recognizer.ts`	Pattern recognition
12	SelfHealing	`src/ai/self-healing.ts`	Test self-healing

B.1.3 Cognition (Knowledge)

#	Module	File	Function
13	ContextAnchor	`src/cognition/ContextAnchor.ts`	"Save Game" for AI state
14	ContextInjector	`src/cognition/ContextInjector.ts`	Relevant context injection
15	Distiller	`src/cognition/distiller.ts`	Knowledge distillation
16	CritiqueDistiller	`src/cognition/critique-distiller.ts`	Critical analysis
17	DependencyGraph	`src/cognition/DependencyGraph.ts`	Dependency graph
18	InferenceEngine	`src/cognition/inference-engine.ts`	Inference engine
19	MultiPerspective	`src/cognition/multi-perspective.ts`	Multi-perspective analysis
20	SelfCritique	`src/cognition/self-critique.ts`	Self-critique
21	SemanticMemory	`src/cognition/semantic-memory.ts`	Semantic memory
22	ThoughtChain	`src/cognition/thought-chain.ts`	Chain of thoughts
23	Uncertainty	`src/cognition/uncertainty.ts`	Uncertainty management
24	CognitiveCircularGuard	`src/cognition/cognitive-circular-guard.ts`	Circular thought protection

B.2 OMEGA (Superpowers)

B.2.1 Core Modules

#	Module	File	Function
25	SovereignMagnet	`src/omega/SovereignMagnet.ts`	Module attraction and management

26	ChronosEngine	`src/chronos/engine.ts`	Time machines
27	TimeTraveler	`src/chronos/time-traveler.ts`	Time travel (state rollback)
28	Deadline	`src/chronos/deadline.ts`	Deadline management
29	PredictiveScaler	`src/chronos/PredictiveScaler.ts`	Predictive scaling
30	PriceOracle	`src/chronos/PriceOracle.ts`	Price oracle

B.2.2 Ghost Protocol (Invisibility)

#	Module	File	Function
31	AntiDetection	`src/ghost/anti-detection.ts`	Anti-detection
32	CloudflareBypass	`src/ghost/CloudflareBypass.ts`	Cloudflare bypass
33	NetworkInterceptor	`src/ghost/network-interceptor.ts`	Network interception
34	PersonalityEngine	`src/ghost/personality-engine.ts`	Personality generation
35	PhantomIntegration	`src/ghost/PhantomIntegration.ts`	Phantom wallet integration
36	Protocol	`src/ghost/protocol.ts`	Ghost Protocol base
37	ProxyChain	`src/ghost/proxy-chain.ts`	Proxy chain

B.3 PHYSICS (Hardware and Atoms)

B.3.1 Core Modules

#	Module	File	Function
38	CableSystem	`src/core/CableSystem.ts`	Cable system between modules
39	PredictiveCables	`src/core/PredictiveCables.ts`	Predictive cables
40	ModuleClasses	`src/core/ModuleClasses.ts`	Module classification
41	NerveCenter	`src/core/nerve-center.ts`	Central nervous system
42	Memoryals	`src/core/Memoryals.ts`	Eternal solar memory
43	DIContainer	`src/core/di-container.ts`	Dependency Injection container
44	EventBus	`src/core/event-bus.ts`	Event bus
45	Immutable	`src/core/immutable.ts`	Immutable structures
46	ErrorFactory	`src/core/errors/ErrorFactory.ts`	Error factory
47	MemoryWatchdog	`src/core/watchdog/MemoryWatchdog.ts`	Memory watchdog

B.3.2 Physics Engine

#	Module	File	Function
48	NeuralInference	`src/physics/NeuralInference.ts`	Neural inference (LLM integration)
49	NeuralAccelerator	`src/physics/NeuralAccelerator.ts`	GPU acceleration
50	HardwareBridge	`src/physics/HardwareBridge.ts`	Hardware bridge
51	AtomicTrader	`src/physics/AtomicTrader.ts`	Atomic transactions

B.3.3 Performance

#	Module	File	Function
52	Cache	`src/performance/cache.ts`	Caching
53	LazyLoader	`src/performance/lazy-loader.ts`	Lazy loading
54	Parallel	`src/performance/parallel.ts`	Parallel execution
55	Pool	`src/performance/pool.ts`	Resource pool
56	Profiler	`src/performance/profiler.ts`	Profiling

B.4 FORTRESS (Security)

B.4.1 Security Modules

#	Module	File	Function
57	Auth	`src/security/auth.ts`	Authentication
58	AuthGuard	`src/security/auth-guard.ts`	Auth guard
59	Encryption	`src/security/encryption.ts`	Encryption
60	Scanner	`src/security/scanner.ts`	Vulnerability scanning
61	Validator	`src/security/validator.ts`	Validation
62	GlobalThreatIntel	`src/security/GlobalThreatIntel.ts`	Global threat intelligence
63	KillSwitchGracePeriod	`src/security/KillSwitchGracePeriod.ts`	Kill switch with grace period

B.4.2 Validation

#	Module	File	Function
64	Contract	`src/validation/contract.ts`	Contract validation
65	Response	`src/validation/response.ts`	Response validation
66	Schema	`src/validation/schema.ts`	Schema validation

B.4.3 Licensing

#	Module	File	Function
67	ZeroKnowledgeLicense	`src/licensing/ZeroKnowledgeLicense.ts`	Zero-knowledge licensing

B.5 BIOLOGY (Memory and Evolution)

B.5.1 Evolution Modules

#	Module	File	Function
68	HiveMind	`src/biology/evolution/HiveMind.ts`	Collective intelligence
69	NeuralSelfEvolver	`src/biology/evolution/NeuralSelfEvolver.ts`	Neural self-evolution

70	BrainRouter	`src/biology/evolution/BrainRouter.ts`	Brain router
71	MarketBlueprint	`src/biology/evolution/MarketBlueprint.ts`	Market blueprint
72	OracleHumanizer	`src/biology/evolution/OracleHumanizer.ts`	Oracle humanization
73	SelfCorrectionLoop	`src/biology/evolution/SelfCorrectionLoop.ts`	Self-correction loop

B.5.2 Metabolism Modules

#	Module	File	Function
74	ProfitOptimizer	`src/biology/metabolism/ProfitOptimizer.ts`	Profit optimizer
75	SelfReinvestment	`src/biology/metabolism/SelfReinvestment.ts`	Self-reinvestment
76	BiometricJitter	`src/biology/biometric-jitter.ts`	Biometric jitter (anti-detection)

B.6 GUARDIANS (Monitoring and Protection)

B.6.1 Guardian Modules

#	Module	File	Function
77	StrictCollar	`src/guardians/StrictCollar.ts`	Anti-hallucination collar
78	EternalWatchdog	`scripts/eternal-watchdog.js`	Eternal watchdog

B.6.2 Swarm Modules

#	Module	File	Function
79	SpectatorMode	`src/swarm/SpectatorMode.ts`	Spectator mode
80	NexusOrchestrator	`src/swarm/mesh/NexusOrchestrator.ts`	Nexus orchestrator

B.6.3 Synthesis Modules

#	Module	File	Function
81	CrossModuleSync	`src/synthesis/cross-module-sync.ts`	Cross-module sync
82	SwarmDataInjection	`src/synthesis/swarm-data-injection.ts`	Swarm data injection
83	SystemMeditation	`src/synthesis/system-meditation.ts`	System meditation
84	VisualGhosting	`src/synthesis/visual-ghosting.ts`	Visual ghosting

B.7 REALITY (Business and Money)

B.7.1 Economy Modules

#	Module	File	Function
85	ArbitrageOrchestrator	`src/reality/economy/ArbitrageOrchestrator.ts`	Arbitrage orchestrator
86	ArmedReaper	`src/reality/economy/ArmedReaper.ts`	Armed reaper (aggressive)
87	EmailEngine	`src/reality/economy/EmailEngine.ts`	Email engine

88	EmergencyKillSwitch	`src/reality/economy/EmergencyKillSwitch.ts`	Emergency stop
89	ExchangeConnectors	`src/reality/economy/ExchangeConnectors.ts`	Exchange connectors
90	LiveWalletManager	`src/reality/economy/LiveWalletManager.ts`	Wallet manager
91	MarketWatcher	`src/reality/economy/MarketWatcher.ts`	Market watcher
92	PaymentGateway	`src/reality/economy/PaymentGateway.ts`	Payment gateway
93	SecureConfigLoader	`src/reality/economy/SecureConfigLoader.ts`	Secure config loader

B.7.2 Gateway Modules

#	Module	File	Function
94	AutonomousSalesForce	`src/reality/gateway/AutonomousSalesForce.ts`	Autonomous sales force
95	AutoOnboarder	`src/reality/gateway/AutoOnboarder.ts`	Auto onboarding
96	GrowthHacker	`src/reality/gateway/GrowthHacker.ts`	Growth hacking
97	MarketplaceConnector	`src/reality/gateway/MarketplaceConnector.ts`	Marketplace connector

B.7.3 Kernel Modules

#	Module	File	Function
98	AutonomousFeedbackLoop	`src/reality/kernel/AutonomousFeedbackLoop.ts`	Autonomous feedback loop
99	VSCodeBridge	`src/reality/kernel/VSCodeBridge.ts`	VS Code bridge

B.7.4 Veritas SDK

#	Module	File	Function
100	Veritas	`src/reality/sdk/veritas/Veritas.ts`	Veritas SDK base
101	CommercializationEngine	`src/reality/sdk/veritas/CommercializationEngine.ts`	Commercialization engine
102	NeuralMapper	`src/reality/sdk/veritas/NeuralMapper.ts`	Neural mapping
103	ParanoidObfuscation	`src/reality/sdk/veritas/ParanoidObfuscation.ts`	Paranoid obfuscation

B.7.5 Sales Modules

#	Module	File	Function
104	ArbitrageLogic	`src/math/ArbitrageLogic.ts`	Arbitrage logic
105	SelfHealingSales	`src/sales/SelfHealingSales.ts`	Self-healing sales

B.8 CHEMISTRY (Synchronization)

B.8.1 Distributed Modules

#	Module	File	Function
106	LoadBalancer	`src/distributed/load-balancer.ts`	Load balancing
107	Runner	`src/distributed/runner.ts`	Distributed runner

108	Sharding	`src/distributed/sharding.ts`	Data sharding

B.8.2 Events Modules

#	Module	File	Function
109	EventBus	`src/events/bus.ts`	Event bus
110	EventEmitter	`src/events/emitter.ts`	Event emitter

B.8.3 Integration Modules

#	Module	File	Function
111	CI	`src/integration/ci.ts`	CI/CD integration
112	Notifications	`src/integration/notifications.ts`	Notifications

B.8.4 Storage Modules

#	Module	File	Function
113	FileStorage	`src/storage/file-storage.ts`	File storage
114	KVStore	`src/storage/kv-store.ts`	Key-Value storage
115	CacheStorage	`src/storage/cache.ts`	Cache storage

SECTION C: SCRIPTS AND TOOLS

C.1 Core Scripts

#	Script	File	Function
116	Assimilator	`scripts/assimilator.ts`	New code assimilation
117	AutoDocumenter	`scripts/auto-documenter.ts`	Auto documentation
118	AutonomousThought	`scripts/autonomous-thought.ts`	Autonomous thoughts
119	ChangeDetector	`scripts/change-detector.ts`	Change detection
120	EmpireDeployment	`scripts/empire-deployment.ts`	Empire deployment
121	ExistenceChecker	`scripts/existence-checker.ts`	Pre-creation check
122	GroupWorkplace	`scripts/group-workplace.ts`	Group workspace
123	HiveMindAwakening	`scripts/hive-mind-awakening.ts`	Hive Mind awakening
124	KnowThyself	`scripts/know-thyself.ts`	Self-knowledge
125	PurgeEngine	`scripts/purge-engine.ts`	Purge engine
126	SupremeMeditation	`scripts/supreme-meditation.ts`	Supreme meditation
127	SystemMeditate	`scripts/system-meditate.ts`	System meditation
128	Warmup	`scripts/warmup.ts`	System warmup
129	MapGenerator	`scripts/memory/map-generator.ts`	Project map generator

C.2 Nerve Templates (8 Departments)

#	Template	File	Actions
130	Intelligence	`scripts/nerve-templates/intelligence-template.ts`	scan, optimize, analyze, train, export
131	Omega	`scripts/nerve-templates/omega-template.ts`	scan, stress, warp, entangle, magnet
132	Physics	`scripts/nerve-templates/physics-template.ts`	scan, gravity, atoms, heal, place
133	Fortress	`scripts/nerve-templates/fortress-template.ts`	scan, audit, encrypt, evidence, vault
134	Biology	`scripts/nerve-templates/biology-template.ts`	scan, prune, evolve, health, compress
135	Guardians	`scripts/nerve-templates/guardians-template.ts`	scan, watch, health, fix, report
136	Reality	`scripts/nerve-templates/reality-template.ts`	scan, hunt, qualify, bomb, outreach
137	Chemistry	`scripts/nerve-templates/chemistry-template.ts`	scan, bond, harmonize, sync, validate

SECTION D: DASHBOARDS

D.1 HTML Dashboards

#	Dashboard	File	Function
138	NerveCenter	`dashboard/nerve-center.html`	Central nervous system
139	ScriptCommandCenter	`dashboard/script-command-center.html`	Script command center
140	GuardianDashboard	`dashboard/guardian-dashboard.html`	Guardian dashboard

D.2 TypeScript Dashboards

#	Dashboard	File	Function
141	ControlDashboard	`src/dashboard/ControlDashboard.ts`	Control panel
142	ReaperDashboard	`src/dashboard/ReaperDashboard.ts`	Reaper dashboard
143	SingularityDashboard	`src/dashboard/SingularityDashboard.ts`	Singularity dashboard

SECTION E: DATA STRUCTURE

E.1 Core Data Files

#	File	Function
144	`data/backpack.json`	Accumulated knowledge backpack
145	`data/learned-patterns.json`	Learned patterns
146	`data/learning-memory.json`	Learning memory

147	`data/learning-stats.json`	Learning statistics
148	`data/outreach-templates.json`	Outreach templates
149	`data/sales-materials.json`	Sales materials

E.2 Memoryals Data

#	File	Function
150	`data/memoryals/context-anchor.json`	Context Anchor state
151	`data/memoryals/anchor-history.json`	Anchor history
152	`data/memoryals/collar-violations.json`	Collar violations
153	`data/memoryals/project-map.json`	Project map

E.3 Chronos Data

#	File	Function
154	`data/chronos-data/chronos_experience.json`	Chronos experience
155	`data/chronos-data/chronos_predictions.json`	Predictions
156	`data/chronos-data/credit_history.json`	Credit history
157	`data/chronos-data/global_heuristics.json`	Global heuristics
158	`data/chronos-data/strategic_memory.json`	Strategic memory

SECTION F: API AND CONFIGURATION

F.1 API Modules

#	Module	File	Function
159	RateLimiter	`src/api/rate-limiter.ts`	Rate limiting
160	Versioning	`src/api/versioning.ts`	API versioning

F.2 Config Modules

#	Module	File	Function
161	ConfigLoader	`src/config/loader.ts`	Config loading
162	ConfigSchema	`src/config/schema.ts`	Config schema

F.3 SaaS Modules

#	Module	File	Function
163	FeatureFlags	`src/saas/feature-flags.ts`	Feature flags
164	Subscription	`src/saas/subscription.ts`	Subscriptions
165	Telemetry	`src/saas/telemetry.ts`	Telemetry

SECTION G: TESTING AND VISUALIZATION

G.1 Reporter Modules

#	Module	File	Function
166	ReportGenerator	`src/reporter/generator.ts`	Report generator
167	ReportDashboard	`src/reporter/dashboard.ts`	Report dashboard
168	ReportMetrics	`src/reporter/metrics.ts`	Report metrics

G.2 Visual Modules

#	Module	File	Function
169	VisualEngine	`src/visual/engine.ts`	Visual engine
170	Snapshot	`src/visual/snapshot.ts`	Screenshots

G.3 Data Modules

#	Module	File	Function
171	DataFactory	`src/data/factory.ts`	Data factory
172	DataFaker	`src/data/faker.ts`	Fake data generator
173	Fixtures	`src/data/fixtures.ts`	Test fixtures

SECTION H: QUICK COMMANDS

H.1 System Commands

```
# System audit
node scripts/system-audit.js

# Sensor pulse
npx ts-node src/core/Memoryals.ts --pulse
```

```
# Sensor status
npx ts-node src/core/Memoryals.ts --status

# Watchdog patrol
node scripts/eternal-watchdog.js --patrol-once
```

H.2 Context Commands

```
# Generate project map
npx ts-node scripts/memory/map-generator.ts

# Context Anchor status
npx ts-node src/cognition/ContextAnchor.ts --status

# Context Injector test
npx ts-node src/cognition/ContextInjector.ts --test "query"
```

H.3 Guardian Commands

```
# Strict Collar status
npx ts-node src/guardians/StrictCollar.ts --status

# Tighten collar
npx ts-node src/guardians/StrictCollar.ts --tighten
```

H.4 Department Commands

```
# Intelligence scan
npx ts-node scripts/nerve-templates/intelligence-template.ts --scan

# Fortress audit
npx ts-node scripts/nerve-templates/fortress-template.ts --audit

# Reality hunt
npx ts-node scripts/nerve-templates/reality-template.ts --hunt
```

SECTION I: IRON RULES FOR AI MODELS

I.1 Non-Negotiable Rules

1. **NEVER** invent libraries that don't exist
2. **ALWAYS** use the provided file structure
3. **IF** context is missing, **STOP** and ask. Don't guess.
4. You are **QAntum v34.1**. You are **NOT** a generic AI.
5. **TypeScript ONLY** unless explicitly told otherwise
6. Reference **ONLY** files from provided context
7. If unsure, say "NEED MORE CONTEXT" - never hallucinate

I.2 Locked Identity

Property	Value
NAME	QAntum
VERSION	34.1
LANGUAGE	TypeScript (primary)
OWNER	Dimitar Prodromov
DEPARTMENTS	8
TOTAL CODE	10M+ lines

I.3 Forbidden Actions

- Inventing packages or libraries
- Using placeholder code
- Saying "as an AI" or similar
- Referencing example.com or foo/bar
- Guessing or hallucinating
- Ignoring provided context
- Mixing languages without reason

I.4 Required Actions

- Check if file exists before creating
- Use correct department for task
- Document changes in CHANGE-LOG-LIVE.md
- Drop anchor (dropAnchor) before long operations
- Run system-audit.js after major changes
- Follow nerve-templates for department tasks

SECTION J: STATISTICS

J.1 Overall Statistics

Metric	Value
Total lines (with node_modules)	10,500,000+
Own code	1,209,902
Total files	1,612
TypeScript modules	170+
JavaScript scripts	15+
HTML dashboards	3
JSON configs	50+
Markdown docs	30+

J.2 By Department

Department	Modules	Percent
INTELLIGENCE	24	14%
OMEGA	14	8%
PHYSICS	20	12%
FORTRESS	12	7%
BIOLOGY	10	6%
GUARDIANS	9	5%
REALITY	24	14%
CHEMISTRY	11	6%
OTHER	48	28%

J.3 System Health

Metric	Value
Connected modules	27/27
Connection rate	100%
Active sensors	6/6
Errors	0
Solar Power	100%

SECTION K: ADDITIONAL REPOSITORIES

K.1 MrMindQATool (Shield)

Metric	Value
Lines of code	154,527

Function	QA protection layer
Repository	papica777-eng/MrMindQATool

K.2 MisterMindPage (Voice)

Metric	Value
Lines of code	6,832
Function	Public face
Repository	papica777-eng/MisterMindPage

Total Points: 173
Last Updated: January 2, 2026
Document Version: 2.0 (English, PDF-optimized)

"In QAntum we don't lie. Only real results."