Automaspec

AI-Powered Test Specification & Automation

Student: Roman Radchenko (Group JS-22)
Supervisor: Volha Kuzniatsova
Date: January 7, 2026

1. The Problem: The "Silo" Effect

- **Documentation Drift**: Requirements in Notion/Jira diverge from the code.
- **Manual Traceability**: Hard to map which business specs are covered by which tests.
- **Execution Silos**: Results are hidden in CI/CD logs (especially critical for E2E visibility).

2. The Problem: Business Impact

Operational Gaps

- ❌ High maintenance overhead.
- ❌ No single source of truth for QA.
- ❌ Fragmented AI workflows.

Consequences

- Slow time-to-market.
- Increased regression risks.
- Team frustration from manual sync.

3. The Solution: Unified QA Engine

- **Single Source of Truth**: Centralized specification management.
- **Context-Aware AI**: LLM fed with real project requirements.
- **Live CI/CD Sync**: Status mapped directly to business specs.
- **Multi-Tenant**: Secure isolation for multiple organizations.

4. High-Level Architecture

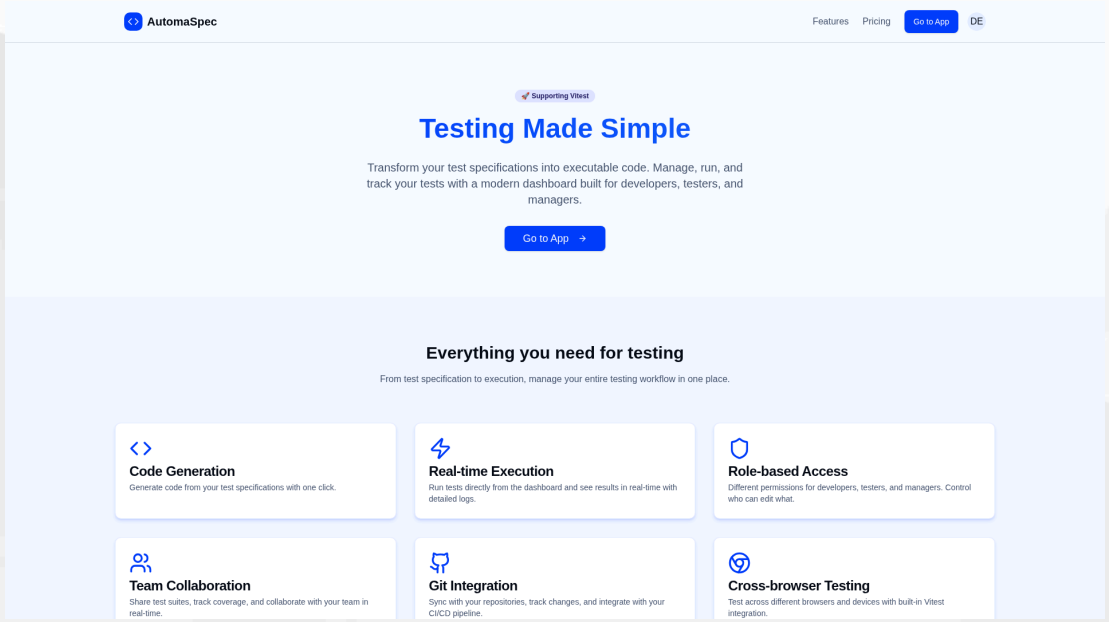
The Modern Stack

- **Frontend**: Next.js 16 (App Router)
- **Backend**: oRPC (Contract-First API)
- **Database**: Turso (Distributed SQLite)
- **AI**: Vercel AI SDK (LLM Bridge)

Web Client (Next.js 16 / React 19)
⇅
oRPC API Layer (Server Logic)
AI Services
Gemini / OpenRouter
Data Tier
Turso + Drizzle

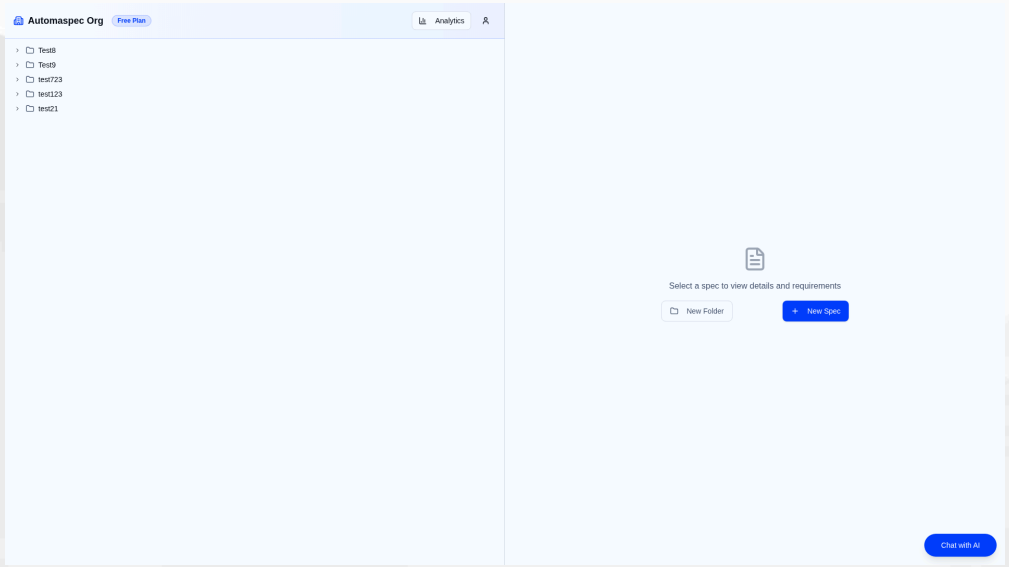
5. Demo: Home & Authentication

- **Landing Page**: Professional overview of capabilities.
- **Auth Flow**: Secure login via Better Auth.
- **Organization**: Seamlessly switch between workspaces.



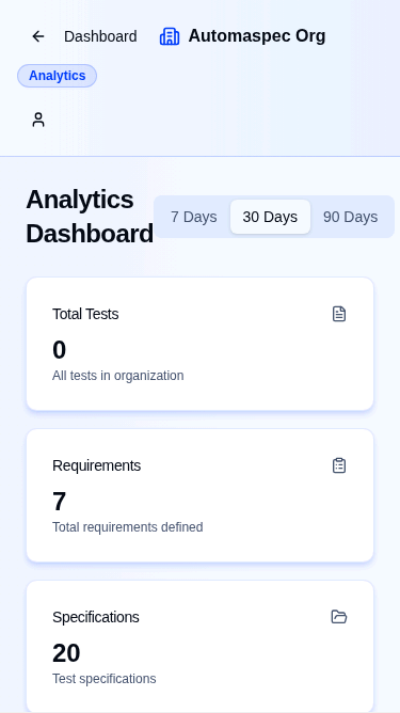
6. Demo: Testing Dashboard

- **Hierarchy**: Navigate folders and specs with ease.
- **AI Side Panel**: Generate test code from requirements.
- **Live Status**: Real-time results from GitHub Actions.



7. Demo: Responsive Design

- **Full Mobile Support**: Check status on the go.
- **Optimized for Tablets**: Full dashboard power.
- **Unified UX**: Consistent across all screen sizes.



8. [Impact] Criterion 1: Business Analysis

- **Strategy**: MoSCoW prioritization & stakeholder analysis.
- **Deliverable**: Full requirements traceability matrix.
- **Rationale**: Needed to ensure MVP focuses on core bottlenecks.

9. [Impact] Criterion 2: Backend & oRPC

Contract-First API

- **Type Safety**: Zero-runtime errors.
- **Middleware**: Auth & Logging.
- **Scalability**: Deep hierarchies.

```
export const router = orpc.router({
  getSpecs: p
    .input(z.object({ orgId: z.string() }))
    .output(z.array(SpecSchema))
    .handler(async ({ input }) => {
       return db.select()
         .from(specs)
         .where(eq(specs.orgId, input.orgId))
    }),
})
```

10. [Impact] Criterion 3: Database Engineering

Relational Integrity

- **3NF Design**: Multi-tenant isolation.
- **Nesting**: Self-referential folders.
- **Type-safe SQL**: via Drizzle.

```
export const folders = table('folder', {
  id: text('id').primaryKey(),
  parentId: text('parent_id')
    .references(() => folders.id),
  orgId: text('org_id').notNull(),
  name: text('name').notNull(),
})

export const specs = table('spec', {
  id: text('id').primaryKey(),
  folderId: text('folder_id')
    .references(() => folders.id),
  name: text('name').notNull(),
})
```

11. [Impact] Criterion 4: Testing Strategy

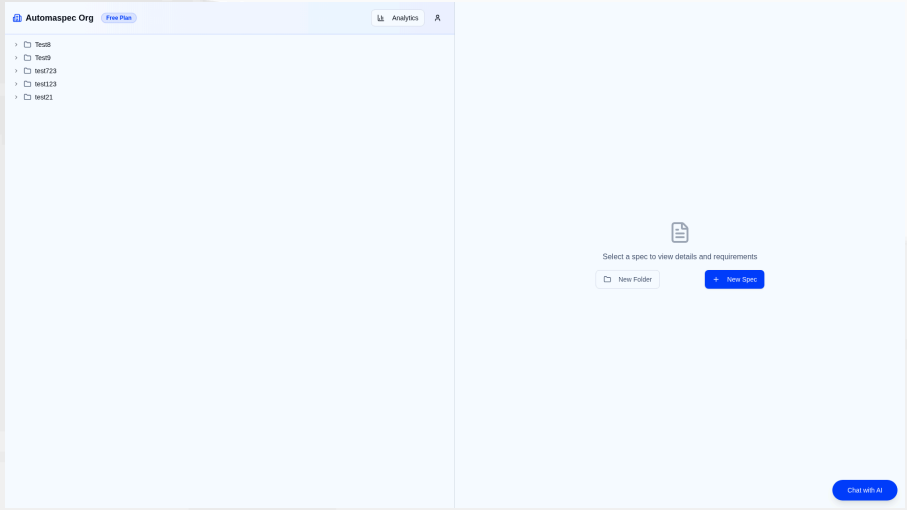
- **Quality Gates**: ≥70% coverage enforced in CI.
- **Playwright E2E**: Critical flow (Auth, Tree Ops) validation.
- **Vitest**: Logic testing & oRPC procedure verification.

Result: A self-documented, high-reliability platform.

12. [Impact] Criterion 5: AI Orchestration

Spec-Driven Gen

- **Mapping**: Specs to Vitest.
- **Streaming**: Real-time feedback.
- **Expert Prompts**: Best practices.



13. [Impact] Criterion 6: Auth & Security

- **Better Auth**: Enterprise-grade session management.
- **Isolation**: Strict data separation at DB level.
- **RBAC**: Permissions (Owner, Admin, Member).

14. [Impact] Criterion 7: DevOps & Scaling

- **Docker**: Optimized multi-stage production builds.
- **CI/CD**: Automated GitHub Actions (Lint/Test/Build).
- **Edge Data**: Low latency via Turso Distributed SQLite.

15. Technical Challenges

Challenge	Solution	Impact
AI Accuracy	Structured context injection.	High code quality.
Hierarchy	Self-referential Drizzle schemas.	Unlimited nesting.
CI/CD Sync	Secure webhook integration.	Real-time status.

16. Results & Future Work

- ✅ **90%** faster test creation via AI.
- ✅ **100%** traceability to Requirements.
- ✅ **70%+** Code Coverage.
- ✅ **Production Ready**: automaspec.vercel.app

17. Q&A

Roman Radchenko (JS-22)

- **Repo**: github.com/automaspec/automaspec
- **App**: automaspec.vercel.app
- **Docs**: /rpc/docs