Software Development Plan (SDP)

NASA Farm Navigators: Data-Driven Agricultural Education Platform

Document Version

- **Version:** 2.0
- **Date:** September 2025
- **Project Code:** NFN-2025
- **Challenge:** NASA Space Apps Challenge 2025

---

1. Executive Summary

1.1 Project Vision

NASA Farm Navigators transforms complex NASA satellite data into an engaging educational game that bridges the $43 billion technology adoption gap in agriculture. By gamifying precision farming techniques, we enable farmers, students, and agricultural professionals to understand and apply NASA's freely available Earth observation data for sustainable farming practices.

1.2 Core Innovation

Our solution uniquely addresses the challenge by:

- **Data Depth Understanding**: Going beyond surface-level variables to show interconnections between soil moisture depth variations, temperature layers, and crop health
- **Resolution Awareness**: Teaching players about 30m vs 375m dataset limitations through gameplay
- **Context Sensitivity**: Adapting scenarios for industrial vs smallholder farming realities
- **Offline-First Architecture**: Ensuring accessibility in rural areas with limited connectivity

1.3 Success Metrics

- **Educational Impact**: 85% improvement in data interpretation skills
- **Practical Application**: 40% of players report implementing learned techniques
- **Sustainability Outcomes**: Documented 20-50% water savings in simulations
- **User Engagement:** 15-minute average session, 60% monthly retention

2. Technical Architecture

2.1 Data Integration Strategy

Primary NASA Data Sources

```
┌─────────────────────────────────────────────────────────┐
│           NASA DATA PIPELINE                │
├─────────────────────────────────────────────────────────┤
│                                 │
│ AppEEARS API     Crop-CASMA       Worldview    │
│  ┌──────────────┐    ┌──────────────┐    ┌──────────────┐  │
│ │ MODIS/VIIRS│    │SMAP Moisture│   │ GIBS    │ │
│ │  30-500m │    │  9-11km   │   │Visualization│  │
│  └──────────────┘    └──────────────┘    └──────────────┘  │
│    │        │        │    │
│     └──────────────────────┼──────────────────────┘      │
│          ▼                │
│        ┌────────────────────┐            │
│        │ Data Processor  │          │
│        │ - Resolution Check│          │
│        │ - Depth Analysis │          │
│        │ - Time Alignment │          │
│        └────────────────────┘            │
│          │                │
│          ▼                │
│        ┌────────────────────┐            │
│        │ Game Data Layer  │          │
│        │ - Simplification │          │
│        │ - Caching     │          │
│        │ - Offline Storage │          │
│        └────────────────────┘            │
└─────────────────────────────────────────────────────────┘
```

Data Understanding Matrix

Dataset	Resolution	Depth/Layer	Update Frequency	Game Application
SMAP L3	9km	0-5cm surface	2-3 days	Irrigation timing
SMAP L4	11km	Root zone 0-100cm	Daily	Deep crop water needs
MODIS NDVI	250m	Canopy reflection	8-day composite	Crop health monitoring
Landsat	30m	Surface	16 days	Field-level precision
GPM IMERG	10km	Precipitation	30 minutes	Forecast planning
GLDAS	25km	Multi-layer temp	3-hourly	Heat stress detection

2.2 Technology Stack

Core Architecture

```yaml
Frontend:
  Game Engine: Phaser.js v3.70+
  UI Framework: React 18 (dashboard components)
  State Management: Redux Toolkit
  Visualization: D3.js v7 (data graphs)
  Maps: Leaflet with NASA GIBS tiles
  PWA: Service Worker with Workbox

Backend:
  Runtime: Node.js 20 LTS
  Framework: Express.js with GraphQL
  Database:
    - PostgreSQL 15 (user data, game state)
    - TimescaleDB (time-series satellite data)
    - Redis 7 (session cache, real-time data)
  Queue: Bull MQ (background data processing)

Data Pipeline:
  ETL: Apache Airflow
  Processing: Python with xarray, rasterio
  Storage: AWS S3 with CloudFront CDN
  Format: Cloud Optimized GeoTIFF (COG)

Infrastructure:
  Container: Docker with Kubernetes
  Cloud: AWS (primary) with multi-region
  Monitoring: Prometheus + Grafana
  Logging: ELK Stack
```

2.3 System Architecture Diagram

```
┌─────────────────────────────────────────────────────────────
│              CLIENT LAYER                  │
├─────────────────────────────────────────────────────────────
│ Browser PWA     Mobile PWA     Desktop App       │
│  ┌──────────────┐    ┌──────────────┐    ┌──────────────┐     │
│ │ Phaser.js │   │ Phaser.js │ │ Electron │    │
│ │  React   │   │  React   │ │  React   │    │
│  └──────────────┘    └──────────────┘    └──────────────┘     │
│       │         │          │         │
```

```
│       └──────────────────────┼──────────────────────┘          │
│              ▼                      │
│          ┌────────────────────────┐               │
│         │ Service Worker │              │
│         │ Offline Cache  │              │
│          └────────────────────────┘               │
└──────────────────────────────────────────────────────────────────
              │
              ▼
┌──────────────────────────────────────────────────────────────────
│           API GATEWAY                  │
├──────────────────────────────────────────────────────────────────
│         GraphQL Federation Router                │
│    ┌────────────────┬────────────────┬────────────────┐          │
│   │ Auth    │ Game    │ Data    │         │
│   │ Service  │ Service  │ Service  │         │
│    └────────────────┴────────────────┴────────────────┘          │
└──────────────────────────────────────────────────────────────────
              │
              ▼
┌──────────────────────────────────────────────────────────────────
│          MICROSERVICES                  │
├──────────────────────────────────────────────────────────────────
│  ┌──────────────┐  ┌──────────────┐  ┌──────────────┐  ┌────────────┐  │
│ │  User  │ │  Farm  │ │ NASA   │ │Analytics│ │
│ │ Manager │ │Simulator │ │ Data   │ │ Engine │ │
│  └──────────────┘  └──────────────┘  └──────────────┘  └────────────┘  │
└──────────────────────────────────────────────────────────────────
              │
              ▼
┌──────────────────────────────────────────────────────────────────
│           DATA LAYER                  │
├──────────────────────────────────────────────────────────────────
│ PostgreSQL   TimescaleDB   Redis    S3       │
│  ┌─────────────┐  ┌──────────────┐  ┌──────────┐  ┌──────────────┐  │
│ │User Data│  │Time Series│  │ Cache │  │Satellite │ │
│  └─────────────┘  └──────────────┘  └──────────┘  └──────────────┘  │
└──────────────────────────────────────────────────────────────────
```

3. Development Methodology

3.1 Agile Framework

- **Methodology:** Scrum with 2-week sprints
- **Team Size:** 14 members
- **Duration:** 16 weeks to MVP, 24 weeks to full release

- **Review Cycles:** Bi-weekly with NASA SMEs and agricultural consultants

3.2 Development Phases

Phase 1: Foundation (Weeks 1-4)

Sprint 1-2: Data Pipeline & Infrastructure

```
Week 1-2:
- NASA Earthdata authentication setup
- AppEEARS API integration
- Crop-CASMA data access implementation
- AWS infrastructure provisioning
- Docker containerization setup

Week 3-4:
- Data resolution handling system
- Multi-depth soil moisture processing
- Temperature layer differentiation
- Offline caching architecture
- Service Worker implementation
```

Deliverables:

- Functional data pipeline processing 5 NASA datasets
- Offline storage supporting 72-hour gameplay
- Data accuracy validation reports

Phase 2: Core Game Engine (Weeks 5-8)

Sprint 3-4: Game Mechanics

```
Week 5-6:
- Phaser.js game engine setup
- Farm grid system (supporting 30m to 11km resolutions)
- Crop growth simulation with real phenology
- Water balance model implementation
- Fertilizer response curves

Week 7-8:
- Livestock management system
- Weather integration with GPM data
- Pest/disease modeling
- Yield prediction algorithms
- Economic simulation
```

Key Features:

- Dynamic resolution switching to show data limitations
- Depth-aware irrigation (surface vs root zone)
- Multi-layer temperature impacts on crops

Phase 3: Educational Framework (Weeks 9-12)

Sprint 5-6: Learning Systems

```
Week 9-10:
- Progressive tutorial system
- Data interpretation challenges
- Resolution awareness mini-games
- Depth understanding scenarios
- Context-switching (industrial vs smallholder)

Week 11-12:
- Real-world case studies integration
- NASA expert video content
- Knowledge assessment system
- Certification pathway
- Classroom management tools
```

Educational Innovations:

- "Resolution Reality Check": Shows what 30m vs 375m actually means
- "Depth Dive": Teaches difference between surface and root zone moisture
- "Context Matters": Adapts to farm size and resources

Phase 4: Polish & Testing (Weeks 13-16)

Sprint 7-8: Optimization & Launch

```
Week 13-14:
- Performance optimization
- Cross-platform testing
- Accessibility compliance
- Security audit
- Load testing (10,000 concurrent users)

Week 15-16:
- Beta testing with 500+ users
- Educational institution pilots
- NASA review and approval
- Marketing material preparation
- Launch preparation
```

4. Team Structure & Roles

4.1 Core Team Composition

Role	Count	Responsibilities	Required Expertise
Project Manager	1	Overall coordination, NASA liaison	AgTech experience, PMP
Technical Lead	1	Architecture decisions, code review	Full-stack, GIS experience
Game Designer	1	Gameplay mechanics, progression	Educational game design
Backend Engineers	3	API development, data pipeline	Node.js, Python, PostGIS
Frontend Engineers	2	Game development, UI	Phaser.js, React, WebGL
Data Scientists	2	NASA data processing, ML models	Remote sensing, agriculture
UX/UI Designer	1	Interface design, user testing	Game UI, accessibility
Agricultural SME	1	Domain validation, content accuracy	Agronomy, precision ag
NASA Liaison	1	Data access, compliance	Earth observation expertise
DevOps Engineer	1	Infrastructure, deployment	AWS, Kubernetes, CI/CD
QA Engineers	2	Testing, quality assurance	Game testing, automation

4.2 Advisory Board

- NASA Acres representative
- NASA Harvest consortium member
- Precision agriculture farmer
- Agricultural educator
- Climate scientist

5. Risk Management

5.1 Technical Risks

Risk	Impact	Probability	Mitigation Strategy
NASA API rate limits	High	High	Implement intelligent caching, pre-process common queries, use bulk downloads
Data accuracy in simplification	Critical	Medium	Maintain accuracy thresholds, expert validation, show confidence intervals
Resolution confusion by users	High	High	Progressive education, visual demonstrations, clear limitations display
Offline sync conflicts	Medium	Medium	CRDT implementation, clear conflict resolution UI
Performance on low-end devices	High	Medium	Progressive enhancement, quality settings, cloud gaming option

5.2 Educational Risks

Risk	Impact	Probability	Mitigation Strategy
Oversimplification of complex data	Critical	Medium	Layer complexity progressively, maintain scientific accuracy
Misinterpretation of data limitations	High	High	Explicit teaching moments, warning systems, resolution indicators
Surface-level engagement only	High	Medium	Deep gameplay mechanics, real-world connections, expert testimonials
Context mismatch (industrial vs smallholder)	Medium	High	Adaptive scenarios, selectable contexts, localized content

6. Quality Assurance Strategy

6.1 Testing Framework

```yaml
```

```
Unit Testing:
  Coverage: 85% minimum
  Framework: Jest, React Testing Library
  Focus: Data transformations, game logic

Integration Testing:
  Coverage: All NASA API endpoints
  Framework: Postman, Newman
  Focus: Data pipeline accuracy

Game Testing:
  Framework: Playwright, custom game test harness
  Focus: Gameplay mechanics, progression

Performance Testing:
  Tools: K6, Lighthouse
  Targets:
    - Load time: <3 seconds on 3G
    - FPS: 60fps on standard hardware
    - Concurrent users: 10,000

Educational Testing:
  Method: A/B testing with control groups
  Metrics: Knowledge retention, skill application
  Sample: 500+ users across demographics
```

6.2 Data Validation

```python
validation_rules = {
    "SMAP": {
        "range": [0.0, 0.6],  # m³/m³
        "depth_levels": ["surface", "root_zone"],
        "resolution_m": [9000, 11000],
        "null_handling": "temporal_interpolation"
    },
    "NDVI": {
        "range": [-1.0, 1.0],
        "cloud_mask": True,
        "quality_flags": ["good", "marginal"],
        "composite_period": 8  # days
    }
}
```

7. Innovation Features

7.1 Beyond Basic Implementation

1. **Multi-Resolution Comparison Tool**: Side-by-side view of same field at different resolutions
2. **Depth Profile Visualizer**: 3D soil moisture profile showing all layers
3. **Uncertainty Quantification**: Shows confidence intervals on all predictions
4. **Climate Scenario Generator**: Uses NASA climate projections for future planning
5. **Peer Learning Network**: Connect with real farmers using similar data

7.2 Advanced Educational Elements

- **NASA Scientist Mentorship**: Monthly live Q&A sessions
- **Real Farm Twinning**: Partner with actual farms for data comparison
- **Certification Program**: NASA-endorsed precision agriculture certificate
- **Research Mode**: Access to raw data for advanced users

8. Sustainability & Scalability

8.1 Business Model

```
Revenue Streams:
- Freemium: Basic game free, advanced features $4.99/month
- Educational Licenses: $500/year per institution
- Certification Program: $99 per certificate
- Sponsorships: AgTech company partnerships
- Grants: NASA, USDA, NSF funding

Cost Structure:
- Infrastructure: $5,000/month (AWS)
- Data Storage: $2,000/month (S3, CDN)
- Development: $150,000 (initial)
- Maintenance: $50,000/year
- Marketing: $30,000/year
```

8.2 Scaling Strategy

- **Year 1**: 10,000 users, focus on U.S. market
- **Year 2**: 100,000 users, expand to Canada, Mexico
- **Year 3**: 500,000 users, global expansion
- **Year 4**: 1M+ users, white-label for organizations

9. Compliance & Standards

9.1 Required Compliance

- **COPPA**: Age verification, parental consent for <13
- **FERPA**: Educational records protection
- **Section 508**: Accessibility for government use
- **WCAG 2.1 AA**: Web accessibility standards
- **NASA Data Use**: Attribution, no endorsement implied

9.2 Data Standards

- **OGC Standards**: WMS, WFS, WCS for geospatial data
- **CF Conventions**: NetCDF climate data
- **STAC**: SpatioTemporal Asset Catalogs
- **ISO 19115**: Metadata standards

10. Success Criteria

10.1 Technical Success

- Data accuracy: 95% correlation with source
- System uptime: 99.9%
- Response time: <200ms API, <100ms game actions
- Offline capability: 72+ hours
- Cross-platform compatibility: 100%

10.2 Educational Success

- Knowledge assessment improvement: 80%
- Practical application: 40% users report real-world use
- Completion rate: 70% finish tutorial
- Engagement: 15-minute average session
- Retention: 60% monthly active users

10.3 Impact Metrics

- Water savings demonstrated: 20-50%
- Yield optimization shown: 10-30%
- Fertilizer reduction: 15-40%
- Carbon footprint reduction: 20%
- ROI demonstration: 3:1 minimum

11. Documentation Requirements

11.1 Technical Documentation

- API documentation (OpenAPI 3.0)
- Data processing algorithms
- Game mechanics formulas
- Infrastructure as Code (Terraform)
- Deployment procedures

11.2 Educational Documentation

- Educator's guide
- Student workbook
- Data interpretation manual
- Best practices handbook
- Case study compilation

11.3 User Documentation

- Getting started guide
- Video tutorials
- FAQ system
- Community forum
- Help center

12. Project Timeline

```
mermaid
```

```
gantt
    title NASA Farm Navigators Development Timeline
    dateFormat YYYY-MM-DD
    section Phase 1
    Data Pipeline         :2025-01-01, 2w
    Infrastructure Setup   :2025-01-15, 2w
    section Phase 2
    Game Engine          :2025-02-01, 2w
    Core Mechanics       :2025-02-15, 2w
    section Phase 3
    Educational System    :2025-03-01, 2w
    Content Creation      :2025-03-15, 2w
    section Phase 4
    Testing & Polish      :2025-04-01, 2w
    Beta Launch         :2025-04-15, 2w
    section Release
    Public Launch       :2025-05-01, 1d
```

Appendices

Appendix A: NASA Data Source Details

Comprehensive list of all NASA datasets, access methods, and update frequencies.

Appendix B: Agricultural Models

Scientific basis for crop growth, water balance, and yield prediction algorithms.

Appendix C: Educational Framework

Learning objectives mapped to Common Core and NGSS standards.

Appendix D: Technical Architecture

Detailed system design documents, database schemas, and API specifications.