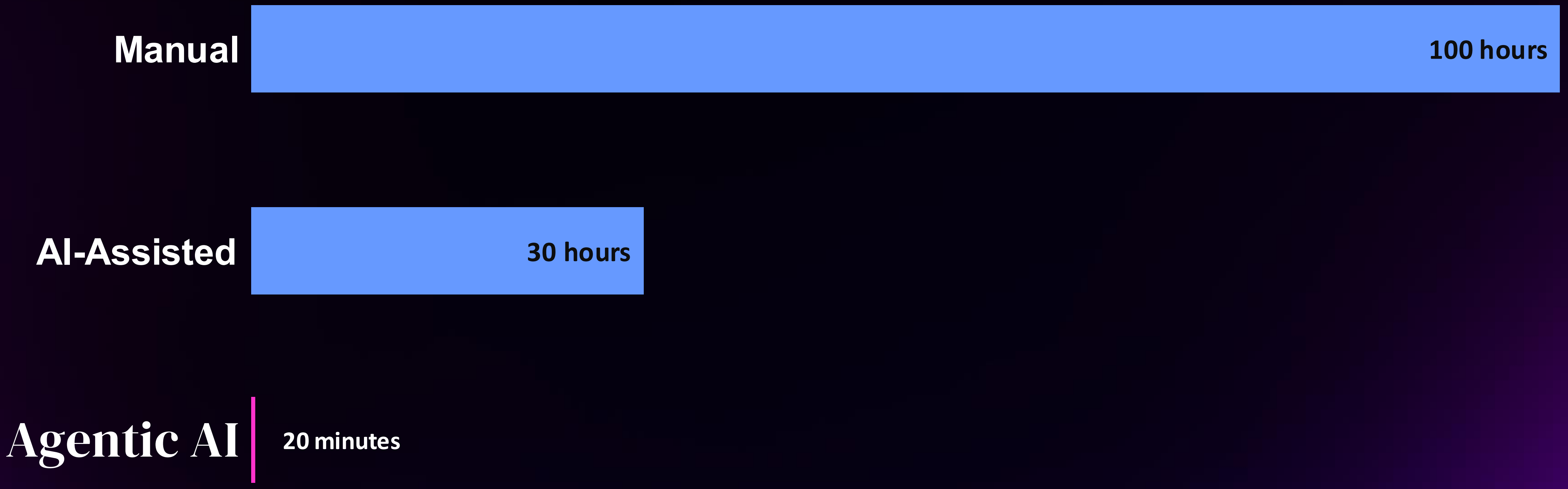
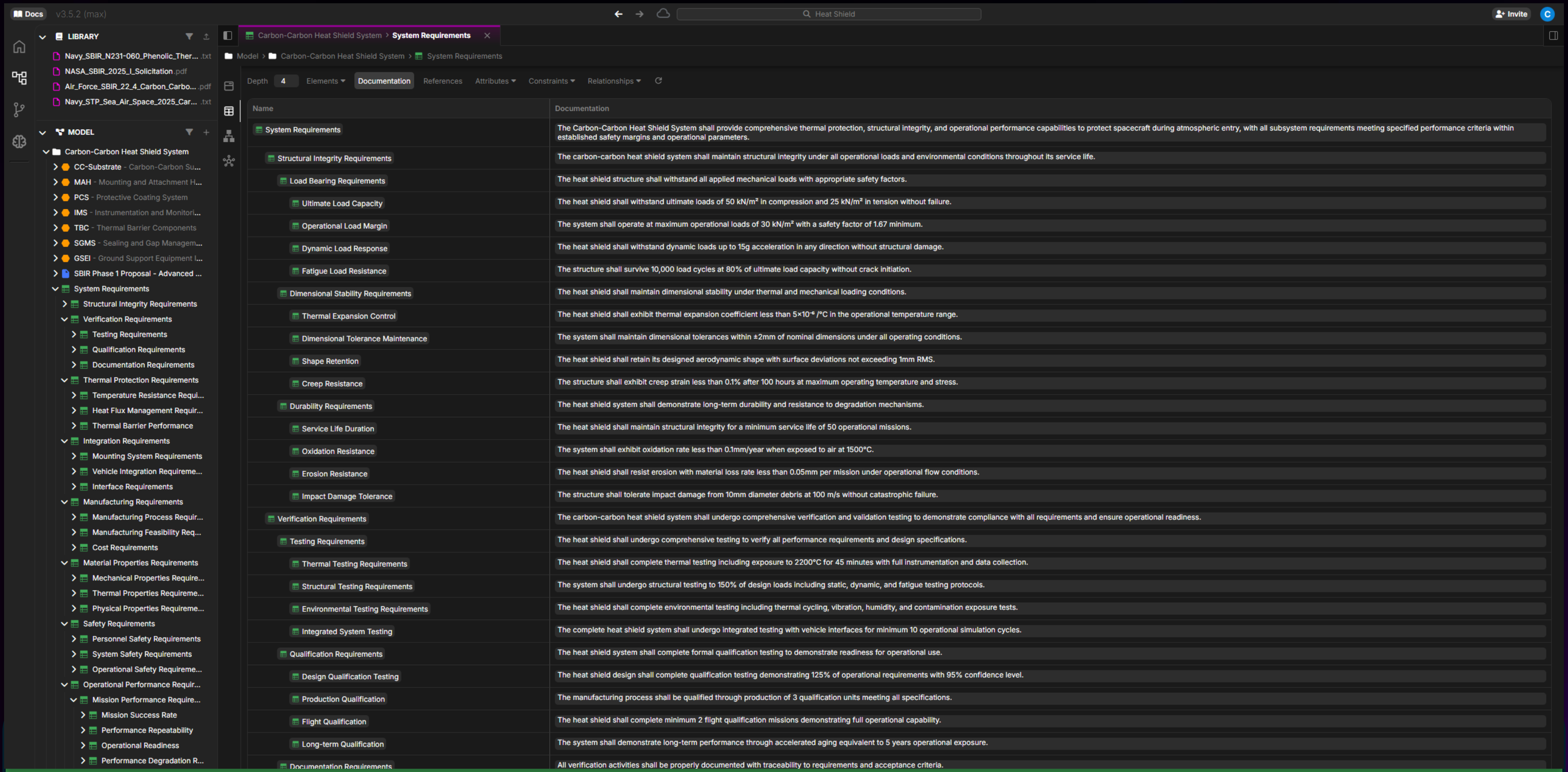
Transforming Systems Engineering Through Agentic AI

Presented by: Dr. Chris Helmerich

CELEDON SOLUTIONS

Creating the System Model





LIBRARY

- Navy_SBIR_N231-060_Phenolic_Ther... .txt
- NASA_SBIR_2025_I_Solicitation .pdf
- Air_Force_SBIR_22_4_Carbon_Carbo... .pdf
- Navy_STP_Sea_Air_Space_2025_Car... .txt

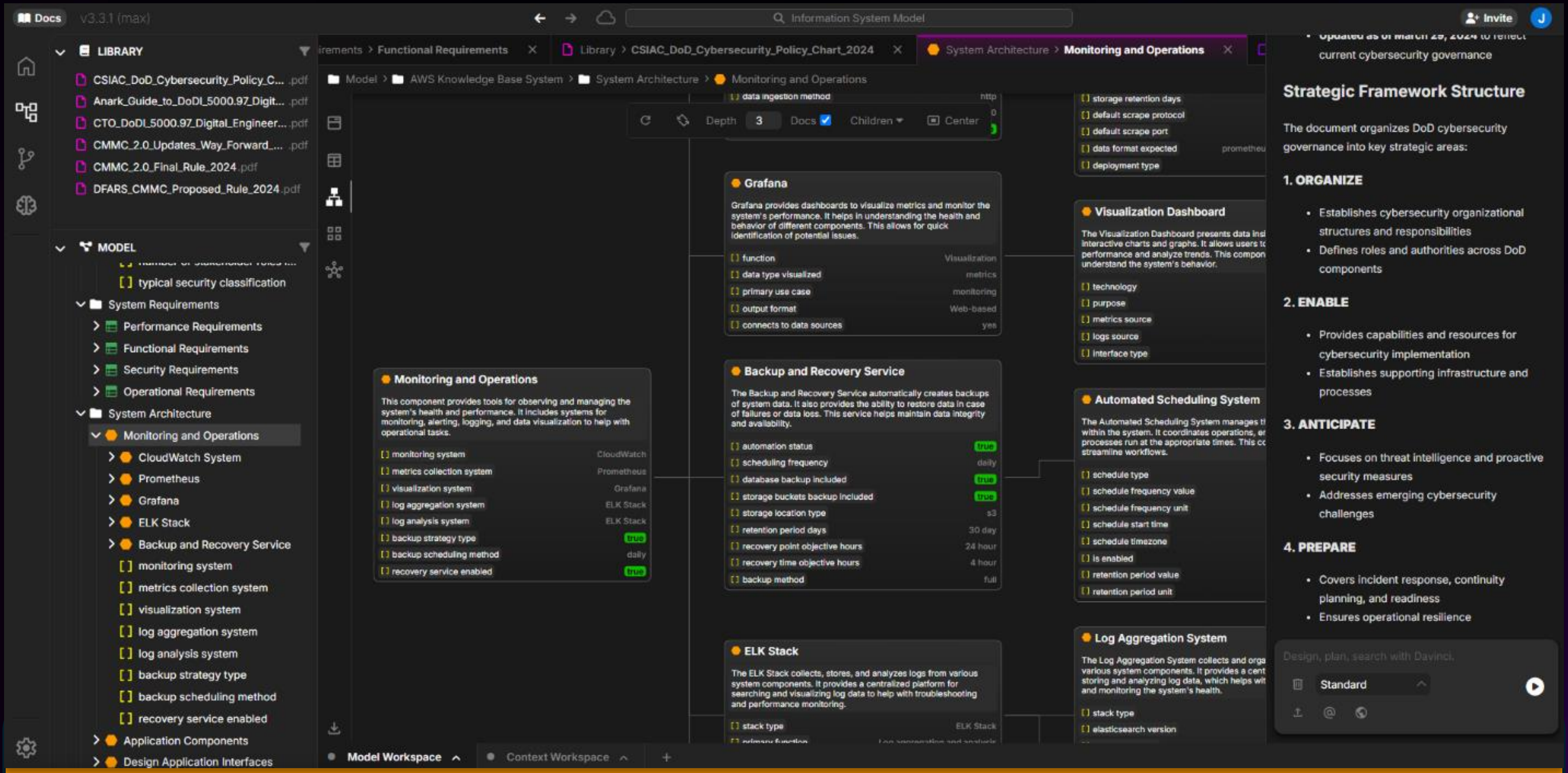
MODEL

- Carbon-Carbon Heat Shield System
  - CC-Substrate - Carbon-Carbon Su...
  - MAH - Mounting and Attachment H...
  - PCS - Protective Coating System
  - IMS - Instrumentation and Monitori...
  - TBC - Thermal Barrier Components
  - SGMS - Sealing and Gap Managem...
  - GSEI - Ground Support Equipment I...
  - SBIR Phase 1 Proposal - Advanced ...
  - System Requirements
    - Structural Integrity Requirements
    - Verification Requirements
      - Testing Requirements
      - Qualification Requirements
      - Documentation Requirements
    - Thermal Protection Requirements
      - Temperature Resistance Requi...
      - Heat Flux Management Requir...
      - Thermal Barrier Performance
    - Integration Requirements
      - Mounting System Requirements
      - Vehicle Integration Requireme...
      - Interface Requirements
    - Manufacturing Requirements
      - Manufacturing Process Requir...
      - Manufacturing Feasibility Req...
      - Cost Requirements
    - Material Properties Requirements
      - Mechanical Properties Require...
      - Thermal Properties Requireme...
      - Physical Properties Requireme...
    - Safety Requirements
      - Personnel Safety Requirements
      - System Safety Requirements
      - Operational Safety Requireme...
    - Operational Performance Requir...
      - Mission Performance Require...
        - Mission Success Rate
        - Performance Repeatability
        - Operational Readiness
        - Performance Degradation R...

Name	Documentation
System Requirements	The Carbon-Carbon Heat Shield System shall provide comprehensive thermal protection, structural integrity, and operational performance capabilities to protect spacecraft during atmospheric entry, with all subsystem requirements meeting specified performance criteria within established safety margins and operational parameters.
Structural Integrity Requirements	The carbon-carbon heat shield system shall maintain structural integrity under all operational loads and environmental conditions throughout its service life.
Load Bearing Requirements	The heat shield structure shall withstand all applied mechanical loads with appropriate safety factors.
Ultimate Load Capacity	The heat shield shall withstand ultimate loads of 50 kN/m^2 in compression and 25 kN/m^2 in tension without failure.
Operational Load Margin	The system shall operate at maximum operational loads of 30 kN/m^2 with a safety factor of 1.67 minimum.
Dynamic Load Response	The heat shield shall withstand dynamic loads up to 15g acceleration in any direction without structural damage.
Fatigue Load Resistance	The structure shall survive 10,000 load cycles at 80% of ultimate load capacity without crack initiation.
Dimensional Stability Requirements	The heat shield shall maintain dimensional stability under thermal and mechanical loading conditions.
Thermal Expansion Control	The heat shield shall exhibit thermal expansion coefficient less than 5×10^{-6} /°C in the operational temperature range.
Dimensional Tolerance Maintenance	The system shall maintain dimensional tolerances within ±2mm of nominal dimensions under all operating conditions.
Shape Retention	The heat shield shall retain its designed aerodynamic shape with surface deviations not exceeding 1mm RMS.
Creep Resistance	The structure shall exhibit creep strain less than 0.1% after 100 hours at maximum operating temperature and stress.
Durability Requirements	The heat shield system shall demonstrate long-term durability and resistance to degradation mechanisms.
Service Life Duration	The heat shield shall maintain structural integrity for a minimum service life of 50 operational missions.
Oxidation Resistance	The system shall exhibit oxidation rate less than 0.1mm/year when exposed to air at 1500°C.
Erosion Resistance	The heat shield shall resist erosion with material loss rate less than 0.05mm per mission under operational flow conditions.
Impact Damage Tolerance	The structure shall tolerate impact damage from 10mm diameter debris at 100 m/s without catastrophic failure.
Verification Requirements	The carbon-carbon heat shield system shall undergo comprehensive verification and validation testing to demonstrate compliance with all requirements and ensure operational readiness.
Testing Requirements	The heat shield shall undergo comprehensive testing to verify all performance requirements and design specifications.
Thermal Testing Requirements	The heat shield shall complete thermal testing including exposure to 2200°C for 45 minutes with full instrumentation and data collection.
Structural Testing Requirements	The system shall undergo structural testing to 150% of design loads including static, dynamic, and fatigue testing protocols.
Environmental Testing Requirements	The heat shield shall complete environmental testing including thermal cycling, vibration, humidity, and contamination exposure tests.
Integrated System Testing	The complete heat shield system shall undergo integrated testing with vehicle interfaces for minimum 10 operational simulation cycles.
Qualification Requirements	The heat shield system shall complete formal qualification testing to demonstrate readiness for operational use.
Design Qualification Testing	The heat shield design shall complete qualification testing demonstrating 125% of operational requirements with 95% confidence level.
Production Qualification	The manufacturing process shall be qualified through production of 3 qualification units meeting all specifications.
Flight Qualification	The heat shield shall complete minimum 2 flight qualification missions demonstrating full operational capability.
Long-term Qualification	The system shall demonstrate long-term performance through accelerated aging equivalent to 5 years operational exposure.
Documentation Requirements	All verification activities shall be properly documented with traceability to requirements and acceptance criteria.

Requirements

159 Constrained Requirements – 1.2 Minutes



Docs
v3.3.1 (max)
Information System Model
Invite
J
LIBRARY
MODEL
irements > Functional Requirements
Library > CSIAC_DoD_Cybersecurity_Policy_Chart_2024
System Architecture > Monitoring and Operations
Model > AWS Knowledge Base System > System Architecture > Monitoring and Operations
Depth 3 Docs ☑ Children ▾ Center
Model Workspace
Context Workspace
+
Design, plan, search with Davinci.
Standard

- CSIAC_DoD_Cybersecurity_Policy_C... .pdf
- Anark_Guide_to_DoDI_5000.97_Digit... .pdf
- CTO_DoDI_5000.97_Digital_Engineer... .pdf
- CMMC_2.0_Updates_Way_Forward_... .pdf
- CMMC_2.0_Final_Rule_2024.pdf
- DFARS_CMMC_Proposed_Rule_2024.pdf

- [ ] typical security classification
- System Requirements
  - Performance Requirements
  - Functional Requirements
  - Security Requirements
  - Operational Requirements
- System Architecture
  - Monitoring and Operations
    - CloudWatch System
    - Prometheus
    - Grafana
    - ELK Stack
    - Backup and Recovery Service
    - [ ] monitoring system
    - [ ] metrics collection system
    - [ ] visualization system
    - [ ] log aggregation system
    - [ ] log analysis system
    - [ ] backup strategy type
    - [ ] backup scheduling method
    - [ ] recovery service enabled
  - Application Components
  - Design Application Interfaces

Monitoring and Operations

This component provides tools for observing and managing the system's health and performance. It includes systems for monitoring, alerting, logging, and data visualization to help with operational tasks.

[] monitoring system	CloudWatch
[] metrics collection system	Prometheus
[] visualization system	Grafana
[] log aggregation system	ELK Stack
[] log analysis system	ELK Stack
[] backup strategy type	true
[] backup scheduling method	daily
[] recovery service enabled	true

[] data ingestion method	http

Grafana

Grafana provides dashboards to visualize metrics and monitor the system's performance. It helps in understanding the health and behavior of different components. This allows for quick identification of potential issues.

[] function	Visualization
[] data type visualized	metrics
[] primary use case	monitoring
[] output format	Web-based
[] connects to data sources	yes

Backup and Recovery Service

The Backup and Recovery Service automatically creates backups of system data. It also provides the ability to restore data in case of failures or data loss. This service helps maintain data integrity and availability.

[] automation status	true
[] scheduling frequency	daily
[] database backup included	true
[] storage buckets backup included	true
[] storage location type	s3
[] retention period days	30 day
[] recovery point objective hours	24 hour
[] recovery time objective hours	4 hour
[] backup method	full

ELK Stack

The ELK Stack collects, stores, and analyzes logs from various system components. It provides a centralized platform for searching and visualizing log data to help with troubleshooting and performance monitoring.

[] stack type	ELK Stack
[] primary function	Log aggregation and analysis

[] storage retention days	
[] default scrape protocol	
[] default scrape port	
[] data format expected	prometheu
[] deployment type	

Visualization Dashboard

The Visualization Dashboard presents data insi interactive charts and graphs. It allows users to performance and analyze trends. This compon understand the system's behavior.

- [] technology
- [] purpose
- [] metrics source
- [] logs source
- [] interface type

Automated Scheduling System

The Automated Scheduling System manages t within the system. It coordinates operations, e processes run at the appropriate times. This c streamline workflows.

- [] schedule type
- [] schedule frequency value
- [] schedule frequency unit
- [] schedule start time
- [] schedule timezone
- [] is enabled
- [] retention period value
- [] retention period unit

Log Aggregation System

The Log Aggregation System collects and orga various system components. It provides a cent storing and analyzing log data, which helps wit and monitoring the system's health.

- [] stack type
- [] elasticsearch version

- Updated as of March 29, 2024 to reflect current cybersecurity governance

Strategic Framework Structure

The document organizes DoD cybersecurity governance into key strategic areas:

1. ORGANIZE

- Establishes cybersecurity organizational structures and responsibilities
- Defines roles and authorities across DoD components

2. ENABLE

- Provides capabilities and resources for cybersecurity implementation
- Establishes supporting infrastructure and processes

3. ANTICIPATE

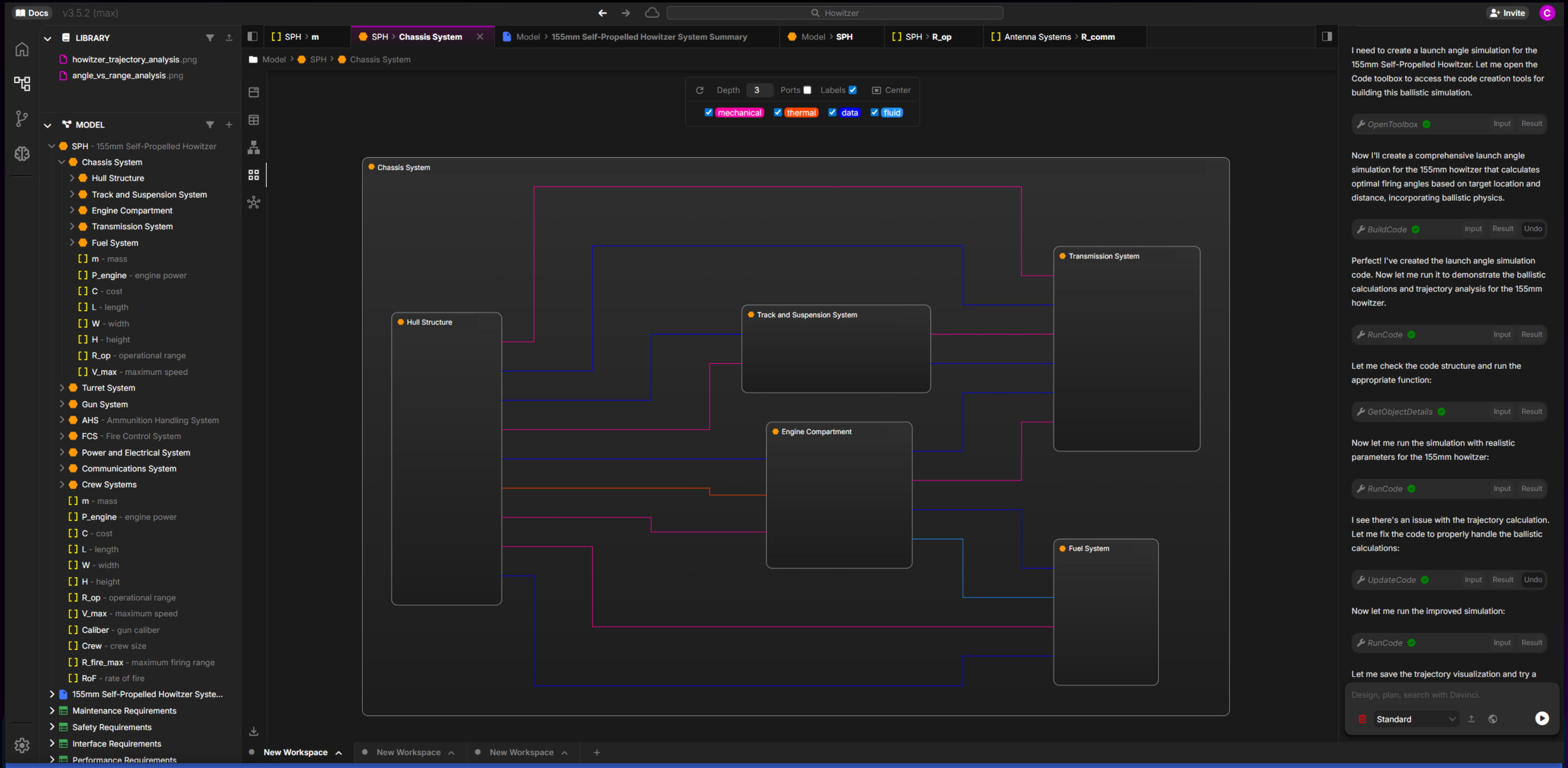
- Focuses on threat intelligence and proactive security measures
- Addresses emerging cybersecurity challenges

4. PREPARE

- Covers incident response, continuity planning, and readiness
- Ensures operational resilience

Architecture

107 Populated Parts – 1.8 Minutes – 1256 Total Objects



LIBRARY

howitzer_trajectory_analysis.png
angle_vs_range_analysis.png

MODEL

SPH - 155mm Self-Propelled Howitzer
Chassis System
Hull Structure
Track and Suspension System
Engine Compartment
Transmission System
Fuel System
[ ] m - mass
[ ] P_engine - engine power
[ ] C - cost
[ ] L - length
[ ] W - width
[ ] H - height
[ ] R_op - operational range
[ ] V_max - maximum speed
Turret System
Gun System
AHS - Ammunition Handling System
FCS - Fire Control System
Power and Electrical System
Communications System
Crew Systems
[ ] m - mass
[ ] P_engine - engine power
[ ] C - cost
[ ] L - length
[ ] W - width
[ ] H - height
[ ] R_op - operational range
[ ] V_max - maximum speed
[ ] Caliber - gun caliber
[ ] Crew - crew size
[ ] R_fire_max - maximum firing range
[ ] RoF - rate of fire
155mm Self-Propelled Howitzer Syste...
Maintenance Requirements
Safety Requirements
Interface Requirements
Performance Requirements

I need to create a launch angle simulation for the 155mm Self-Propelled Howitzer. Let me open the Code toolbox to access the code creation tools for building this ballistic simulation.

OpenToolbox ✓ Input Result

Now I'll create a comprehensive launch angle simulation for the 155mm howitzer that calculates optimal firing angles based on target location and distance, incorporating ballistic physics.

BuildCode ✓ Input Result Undo

Perfect! I've created the launch angle simulation code. Now let me run it to demonstrate the ballistic calculations and trajectory analysis for the 155mm howitzer.

RunCode ✓ Input Result

Let me check the code structure and run the appropriate function:

GetObjectDetails ✓ Input Result

Now let me run the simulation with realistic parameters for the 155mm howitzer:

RunCode ✓ Input Result

I see there's an issue with the trajectory calculation. Let me fix the code to properly handle the ballistic calculations:

UpdateCode ✓ Input Result Undo

Now let me run the improved simulation:

RunCode ✓ Input Result

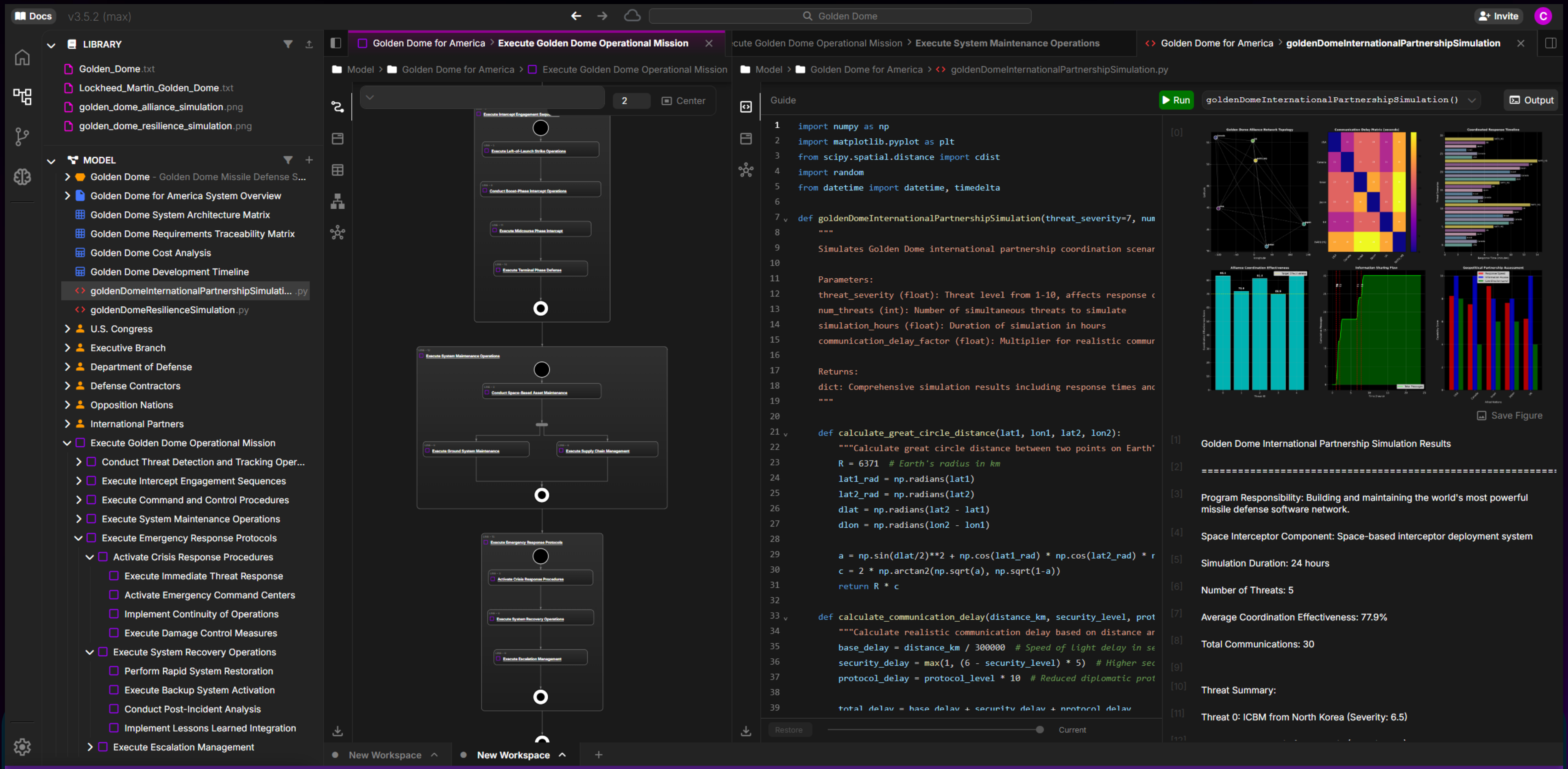
Let me save the trajectory visualization and try a

Design, plan, search with Davinci.

Standard

Interfaces

143 Interfaces Identified – 0.4 Minutes



LIBRARY

- Golden_Dome.txt
- Lockheed_Martin_Golden_Dome.txt
- golden_dome_alliance_simulation.png
- golden_dome_resilience_simulation.png

MODEL

- Golden Dome - Golden Dome Missile Defense S...
- Golden Dome for America System Overview
- Golden Dome System Architecture Matrix
- Golden Dome Requirements Traceability Matrix
- Golden Dome Cost Analysis
- Golden Dome Development Timeline
- goldenDomeInternationalPartnershipSimulati... py
- goldenDomeResilienceSimulation.py
- U.S. Congress
- Executive Branch
- Department of Defense
- Defense Contractors
- Opposition Nations
- International Partners
- Execute Golden Dome Operational Mission
  - Conduct Threat Detection and Tracking Oper...
  - Execute Intercept Engagement Sequences
  - Execute Command and Control Procedures
  - Execute System Maintenance Operations
  - Execute Emergency Response Protocols
    - Activate Crisis Response Procedures
      - Execute Immediate Threat Response
      - Activate Emergency Command Centers
      - Implement Continuity of Operations
      - Execute Damage Control Measures
    - Execute System Recovery Operations
      - Perform Rapid System Restoration
      - Execute Backup System Activation
      - Conduct Post-Incident Analysis
      - Implement Lessons Learned Integration
    - Execute Escalation Management

Golden Dome for America > Execute Golden Dome Operational Mission

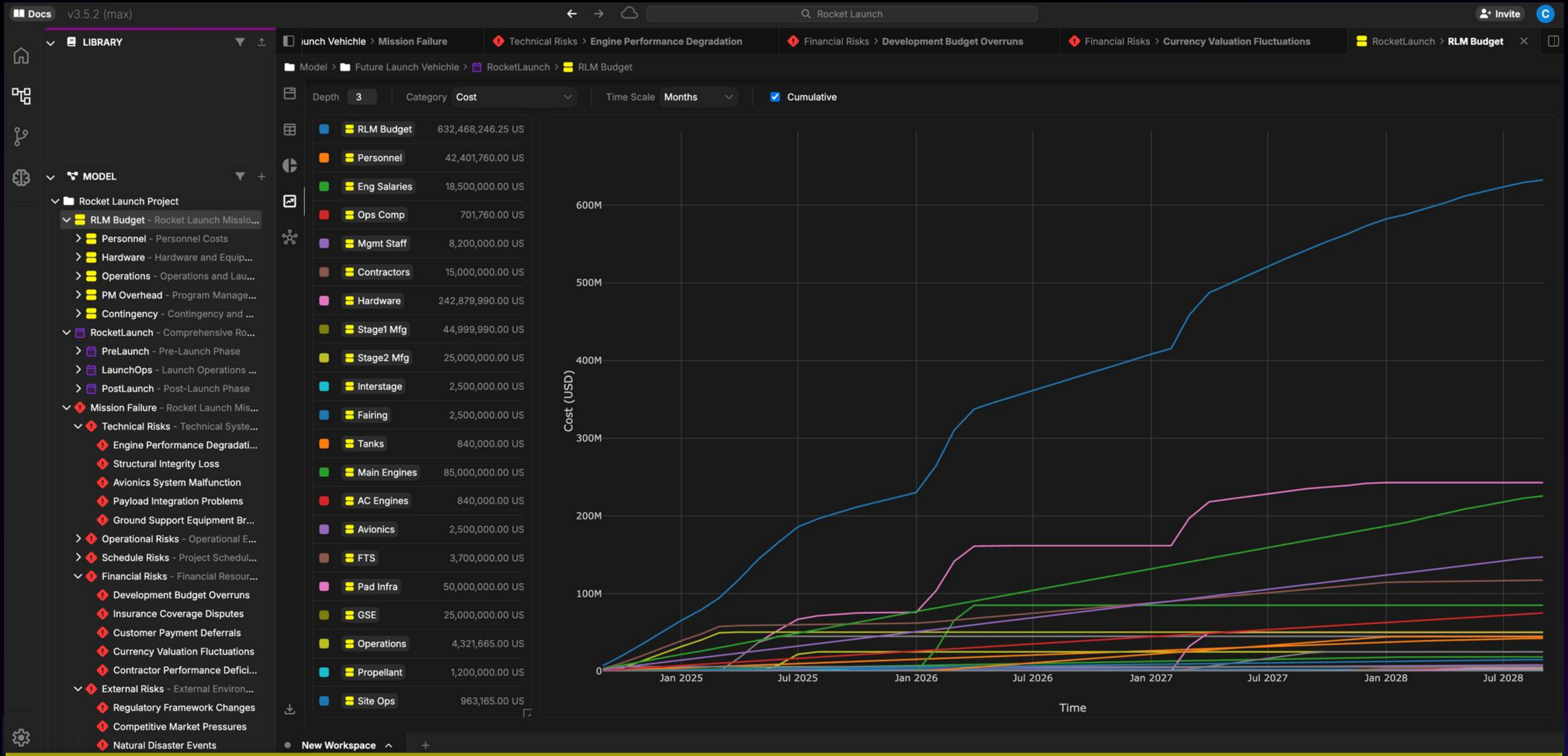
Model > Golden Dome for America > goldenDomeInternationalPartnershipSimulation.py

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.spatial.distance import cdist
import random
from datetime import datetime, timedelta

def goldenDomeInternationalPartnershipSimulation(threat_severity=7, num
    """
    Simulates Golden Dome international partnership coordination scenar

    Parameters:
    threat_severity (float): Threat level from 1-10, affects response c
    num_threats (int): Number of simultaneous threats to simulate
    simulation_hours (float): Duration of simulation in hours
    communication_delay_factor (float): Multiplier for realistic commur

    Returns:
    dict: Comprehensive simulation results including response times anc
    """

    def calculate_great_circle_distance(lat1, lon1, lat2, lon2):
        """Calculate great circle distance between two points on Earth'
        R = 6371  # Earth's radius in km
        lat1_rad = np.radians(lat1)
        lat2_rad = np.radians(lat2)
        dlat = np.radians(lat2 - lat1)
        dlon = np.radians(lon2 - lon1)

        a = np.sin(dlat/2)**2 + np.cos(lat1_rad) * np.cos(lat2_rad) * r
        c = 2 * np.arctan2(np.sqrt(a), np.sqrt(1-a))
        return R * c

    def calculate_communication_delay(distance_km, security_level, prot
        """Calculate realistic communication delay based on distance ar
        base_delay = distance_km / 300000  # Speed of light delay in se
        security_delay = max(1, (6 - security_level) * 5)  # Higher sec
        protocol_delay = protocol_level * 10  # Reduced diplomatic prot

        total_delay = base_delay + security_delay + protocol_delay
```

```
Golden Dome International Partnership Simulation Results

============================================================

Program Responsibility: Building and maintaining the world's most powerful missile defense software network.

Space Interceptor Component: Space-based interceptor deployment system

Simulation Duration: 24 hours

Number of Threats: 5

Average Coordination Effectiveness: 77.9%

Total Communications: 30

Threat Summary:

Threat 0: ICBM from North Korea (Severity: 6.5)
```

Actions and Behaviors

86 Action Sequences – 2.9 Minutes



Docs
v3.5.2 (max)
Rocket Launch
Invite
LIBRARY
unch Vehichle > Mission Failure
Technical Risks > Engine Performance Degradation
Financial Risks > Development Budget Overruns
Financial Risks > Currency Valuation Fluctuations
RocketLaunch > RLM Budget
Model > Future Launch Vehichle > RocketLaunch > RLM Budget
New Workspace
600M
500M
400M
300M
200M
100M
0
Cost (USD)
Jan 2025
Jul 2025
Jan 2026
Jul 2026
Jan 2027
Jul 2027
Jan 2028
Jul 2028
Time

MODEL

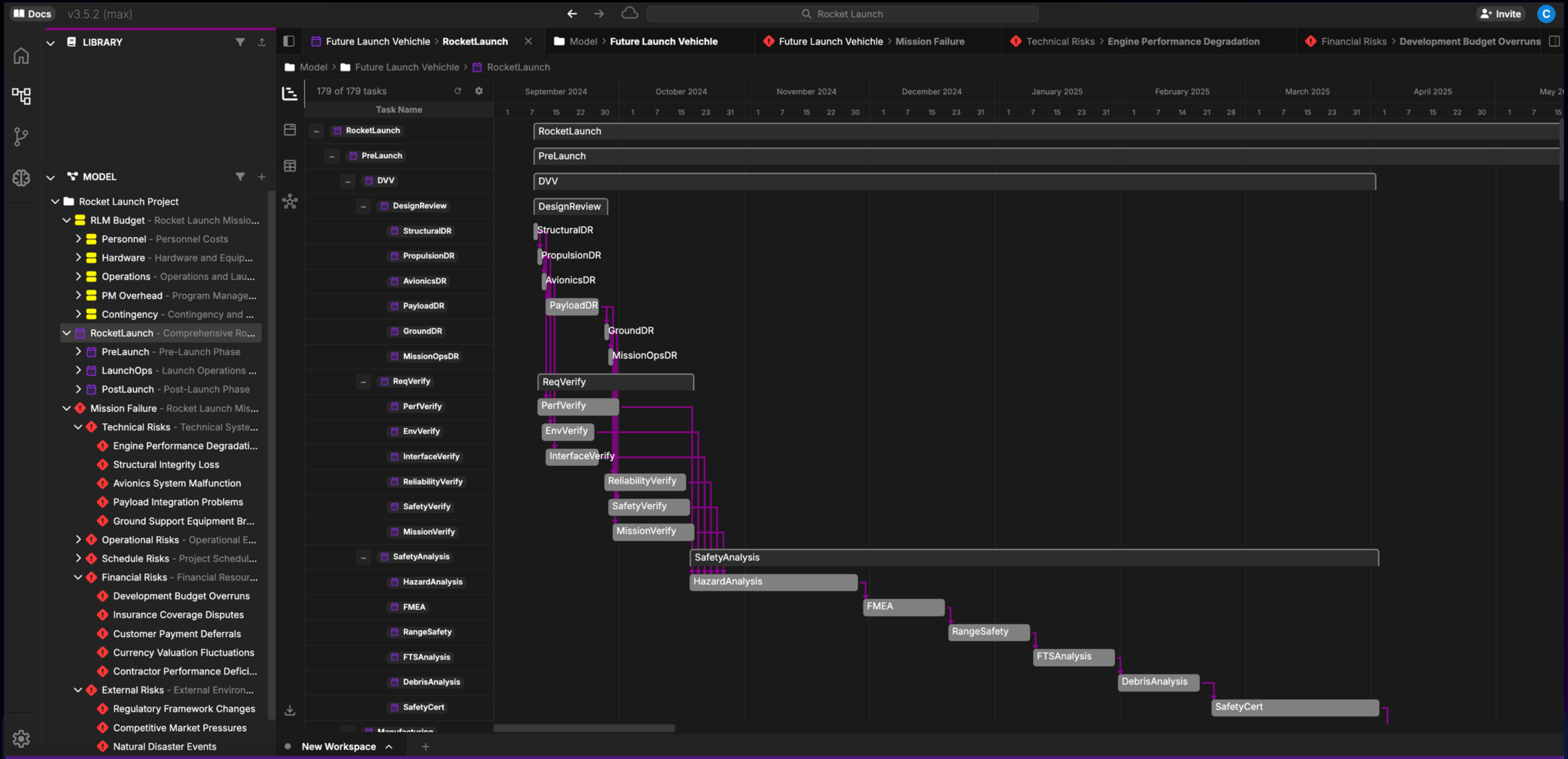
- Rocket Launch Project
  - RLM Budget - Rocket Launch Missio...
    - Personnel - Personnel Costs
    - Hardware - Hardware and Equip...
    - Operations - Operations and Lau...
    - PM Overhead - Program Manage...
    - Contingency - Contingency and ...
  - RocketLaunch - Comprehensive Ro...
    - PreLaunch - Pre-Launch Phase
    - LaunchOps - Launch Operations ...
    - PostLaunch - Post-Launch Phase
  - Mission Failure - Rocket Launch Mis...
    - Technical Risks - Technical Syste...
      - Engine Performance Degradati...
      - Structural Integrity Loss
      - Avionics System Malfunction
      - Payload Integration Problems
      - Ground Support Equipment Br...
    - Operational Risks - Operational E...
    - Schedule Risks - Project Schedul...
    - Financial Risks - Financial Resour...
      - Development Budget Overruns
      - Insurance Coverage Disputes
      - Customer Payment Deferrals
      - Currency Valuation Fluctuations
      - Contractor Performance Defici...
    - External Risks - External Environ...
      - Regulatory Framework Changes
      - Competitive Market Pressures
      - Natural Disaster Events

Depth 3 Category Cost Time Scale Months ☑ Cumulative

■	RLM Budget	632,468,246.25 US
■	Personnel	42,401,760.00 US
■	Eng Salaries	18,500,000.00 US
■	Ops Comp	701,760.00 US
■	Mgmt Staff	8,200,000.00 US
■	Contractors	15,000,000.00 US
■	Hardware	242,879,990.00 US
■	Stage1 Mfg	44,999,990.00 US
■	Stage2 Mfg	25,000,000.00 US
■	Interstage	2,500,000.00 US
■	Fairing	2,500,000.00 US
■	Tanks	840,000.00 US
■	Main Engines	85,000,000.00 US
■	AC Engines	840,000.00 US
■	Avionics	2,500,000.00 US
■	FTS	3,700,000.00 US
■	Pad Infra	50,000,000.00 US
■	GSE	25,000,000.00 US
■	Operations	4,321,665.00 US
■	Propellant	1,200,000.00 US
■	Site Ops	963,165.00 US

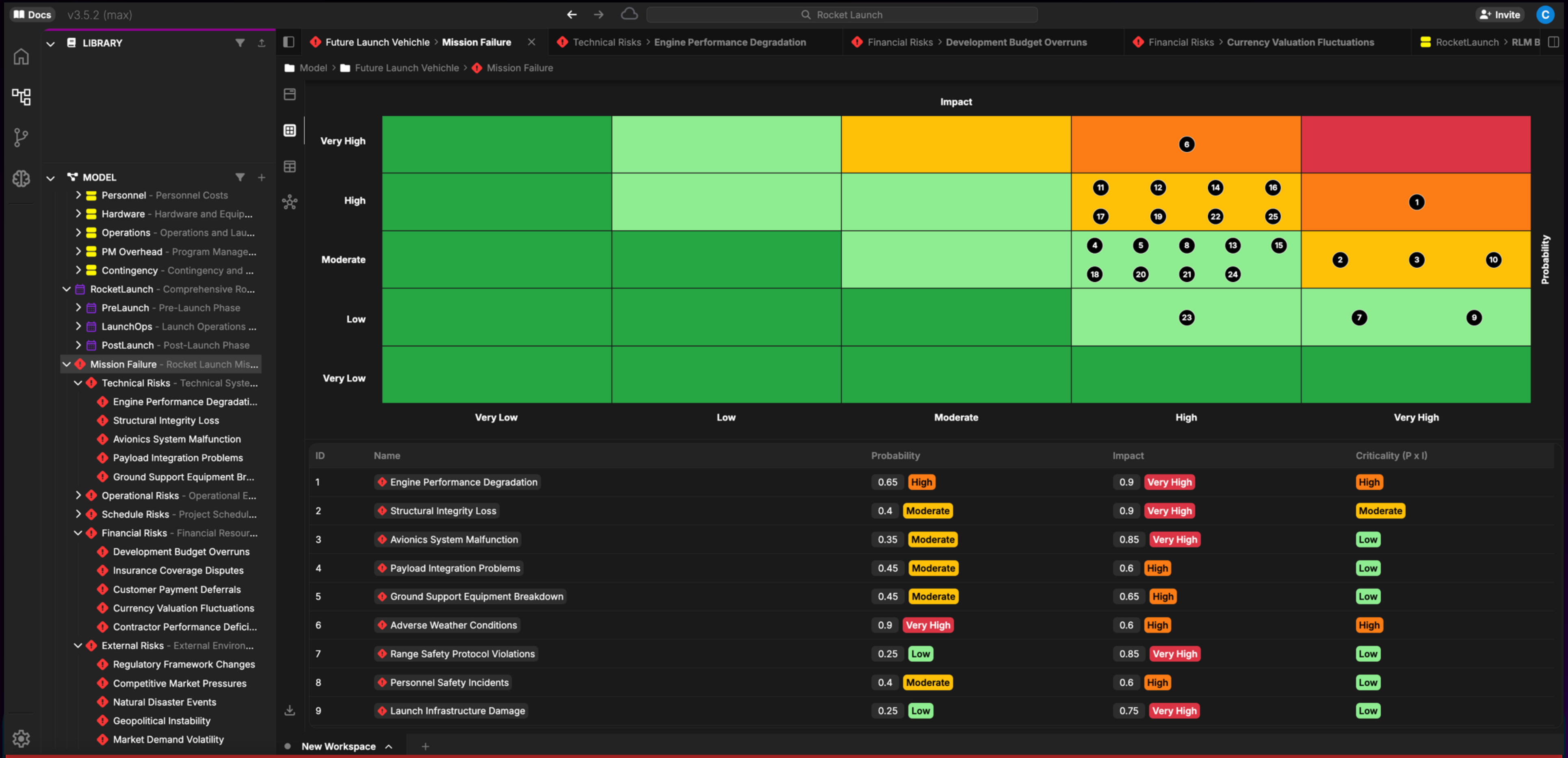
Personnel, Budget, and Resource Modelling

38 Connected Budget Items – 1.4 Minutes


Docs
v3.5.2 (max)
Rocket Launch
Invite
LIBRARY
MODEL
Rocket Launch Project
RLM Budget - Rocket Launch Missio...
Personnel - Personnel Costs
Hardware - Hardware and Equip...
Operations - Operations and Lau...
PM Overhead - Program Manage...
Contingency - Contingency and ...
RocketLaunch - Comprehensive Ro...
PreLaunch - Pre-Launch Phase
LaunchOps - Launch Operations ...
PostLaunch - Post-Launch Phase
Mission Failure - Rocket Launch Mis...
Technical Risks - Technical Syste...
Engine Performance Degradati...
Structural Integrity Loss
Avionics System Malfunction
Payload Integration Problems
Ground Support Equipment Br...
Operational Risks - Operational E...
Schedule Risks - Project Schedul...
Financial Risks - Financial Resour...
Development Budget Overruns
Insurance Coverage Disputes
Customer Payment Deferrals
Currency Valuation Fluctuations
Contractor Performance Defici...
External Risks - External Environ...
Regulatory Framework Changes
Competitive Market Pressures
Natural Disaster Events
Future Launch Vehichle > RocketLaunch
Model > Future Launch Vehichle
Future Launch Vehichle > Mission Failure
Technical Risks > Engine Performance Degradation
Financial Risks > Development Budget Overruns
Model > Future Launch Vehichle > RocketLaunch
179 of 179 tasks
Task Name
September 2024
October 2024
November 2024
December 2024
January 2025
February 2025
March 2025
April 2025
RocketLaunch
PreLaunch
DVV
DesignReview
StructuralDR
PropulsionDR
AvionicsDR
PayloadDR
GroundDR
MissionOpsDR
ReqVerify
PerfVerify
EnvVerify
InterfaceVerify
ReliabilityVerify
SafetyVerify
MissionVerify
SafetyAnalysis
HazardAnalysis
FMEA
RangeSafety
FTSAnalysis
DebrisAnalysis
SafetyCert
New Workspace

Schedule and Tasking

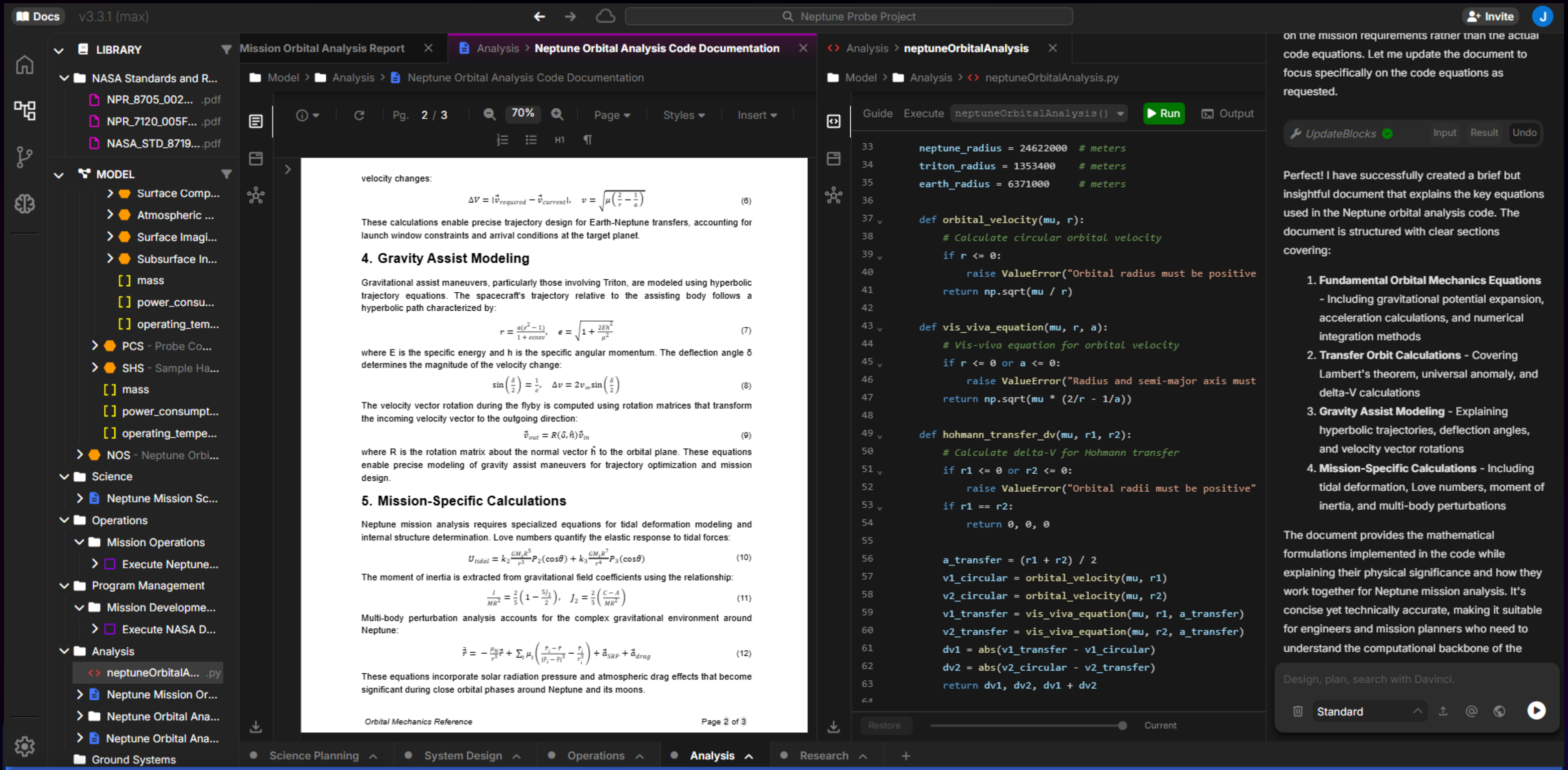
179 Interdependent Tasks – 3.2 Minutes


Docs
v3.5.2 (max)
Rocket Launch
Invite
LIBRARY
MODEL
Personnel - Personnel Costs
Hardware - Hardware and Equip...
Operations - Operations and Lau...
PM Overhead - Program Manage...
Contingency - Contingency and ...
RocketLaunch - Comprehensive Ro...
PreLaunch - Pre-Launch Phase
LaunchOps - Launch Operations ...
PostLaunch - Post-Launch Phase
Mission Failure - Rocket Launch Mis...
Technical Risks - Technical Syste...
Engine Performance Degradati...
Structural Integrity Loss
Avionics System Malfunction
Payload Integration Problems
Ground Support Equipment Br...
Operational Risks - Operational E...
Schedule Risks - Project Schedul...
Financial Risks - Financial Resour...
Development Budget Overruns
Insurance Coverage Disputes
Customer Payment Deferrals
Currency Valuation Fluctuations
Contractor Performance Defici...
External Risks - External Environ...
Regulatory Framework Changes
Competitive Market Pressures
Natural Disaster Events
Geopolitical Instability
Market Demand Volatility
Future Launch Vehichle > Mission Failure
Technical Risks > Engine Performance Degradation
Financial Risks > Development Budget Overruns
Financial Risks > Currency Valuation Fluctuations
RocketLaunch > RLM B
Model > Future Launch Vehichle > Mission Failure
Impact
Probability
Very High
High
Moderate
Low
Very Low
6
11
12
14
16
17
19
22
25
1
4
5
8
13
15
18
20
21
24
2
3
10
23
7
9
Very Low
Low
Moderate
High
Very High
New Workspace

ID	Name	Probability	Impact	Criticality (P x I)
1	Engine Performance Degradation	0.65 High	0.9 Very High	High
2	Structural Integrity Loss	0.4 Moderate	0.9 Very High	Moderate
3	Avionics System Malfunction	0.35 Moderate	0.85 Very High	Low
4	Payload Integration Problems	0.45 Moderate	0.6 High	Low
5	Ground Support Equipment Breakdown	0.45 Moderate	0.65 High	Low
6	Adverse Weather Conditions	0.9 Very High	0.6 High	High
7	Range Safety Protocol Violations	0.25 Low	0.85 Very High	Low
8	Personnel Safety Incidents	0.4 Moderate	0.6 High	Low
9	Launch Infrastructure Damage	0.25 Low	0.75 Very High	Low

Risk Analysis

25 Risks Identified – 0.3 Minutes



velocity changes:

$$\Delta V = |\vec{v}_{required} - \vec{v}_{current}|, \quad v = \sqrt{\mu\left(\frac{2}{r} - \frac{1}{a}\right)} \qquad (6)$$

These calculations enable precise trajectory design for Earth-Neptune transfers, accounting for launch window constraints and arrival conditions at the target planet.

4. Gravity Assist Modeling

Gravitational assist maneuvers, particularly those involving Triton, are modeled using hyperbolic trajectory equations. The spacecraft's trajectory relative to the assisting body follows a hyperbolic path characterized by:

$$r = \frac{a(e^2 - 1)}{1 + e\cos\nu}, \quad e = \sqrt{1 + \frac{2Eh^2}{\mu^2}} \qquad (7)$$

where E is the specific energy and h is the specific angular momentum. The deflection angle δ determines the magnitude of the velocity change:

$$\sin\left(\frac{\delta}{2}\right) = \frac{1}{e}, \quad \Delta v = 2v_\infty \sin\left(\frac{\delta}{2}\right) \qquad (8)$$

The velocity vector rotation during the flyby is computed using rotation matrices that transform the incoming velocity vector to the outgoing direction:

$$\vec{v}_{out} = R(\delta, \hat{h})\vec{v}_{in} \qquad (9)$$

where R is the rotation matrix about the normal vector ĥ to the orbital plane. These equations enable precise modeling of gravity assist maneuvers for trajectory optimization and mission design.

5. Mission-Specific Calculations

Neptune mission analysis requires specialized equations for tidal deformation modeling and internal structure determination. Love numbers quantify the elastic response to tidal forces:

$$U_{tidal} = k_2 \frac{GM_i R^5}{r^3} P_2(\cos\theta) + k_3 \frac{GM_i R^7}{r^4} P_3(\cos\theta) \qquad (10)$$

The moment of inertia is extracted from gravitational field coefficients using the relationship:

$$\frac{I}{MR^2} = \frac{2}{5}\left(1 - \frac{5J_2}{2}\right), \quad J_2 = \frac{2}{5}\left(\frac{C - A}{MR^2}\right) \qquad (11)$$

Multi-body perturbation analysis accounts for the complex gravitational environment around Neptune:

$$\ddot{\vec{r}} = -\frac{\mu_N}{r^3}\vec{r} + \sum_i \mu_i \left(\frac{\vec{r}_i - \vec{r}}{|\vec{r}_i - \vec{r}|^3} - \frac{\vec{r}_i}{r_i^3}\right) + \vec{a}_{SRP} + \vec{a}_{drag} \qquad (12)$$

These equations incorporate solar radiation pressure and atmospheric drag effects that become significant during close orbital phases around Neptune and its moons.

Orbital Mechanics Reference Page 2 of 3

```python
    neptune_radius = 24622000  # meters
    triton_radius = 1353400    # meters
    earth_radius = 6371000     # meters

    def orbital_velocity(mu, r):
        # Calculate circular orbital velocity
        if r <= 0:
            raise ValueError("Orbital radius must be positive
        return np.sqrt(mu / r)

    def vis_viva_equation(mu, r, a):
        # Vis-viva equation for orbital velocity
        if r <= 0 or a <= 0:
            raise ValueError("Radius and semi-major axis must
        return np.sqrt(mu * (2/r - 1/a))

    def hohmann_transfer_dv(mu, r1, r2):
        # Calculate delta-V for Hohmann transfer
        if r1 <= 0 or r2 <= 0:
            raise ValueError("Orbital radii must be positive"
        if r1 == r2:
            return 0, 0, 0

        a_transfer = (r1 + r2) / 2
        v1_circular = orbital_velocity(mu, r1)
        v2_circular = orbital_velocity(mu, r2)
        v1_transfer = vis_viva_equation(mu, r1, a_transfer)
        v2_transfer = vis_viva_equation(mu, r2, a_transfer)
        dv1 = abs(v1_transfer - v1_circular)
        dv2 = abs(v2_circular - v2_transfer)
        return dv1, dv2, dv1 + dv2
```

on the mission requirements rather than the actual code equations. Let me update the document to focus specifically on the code equations as requested.

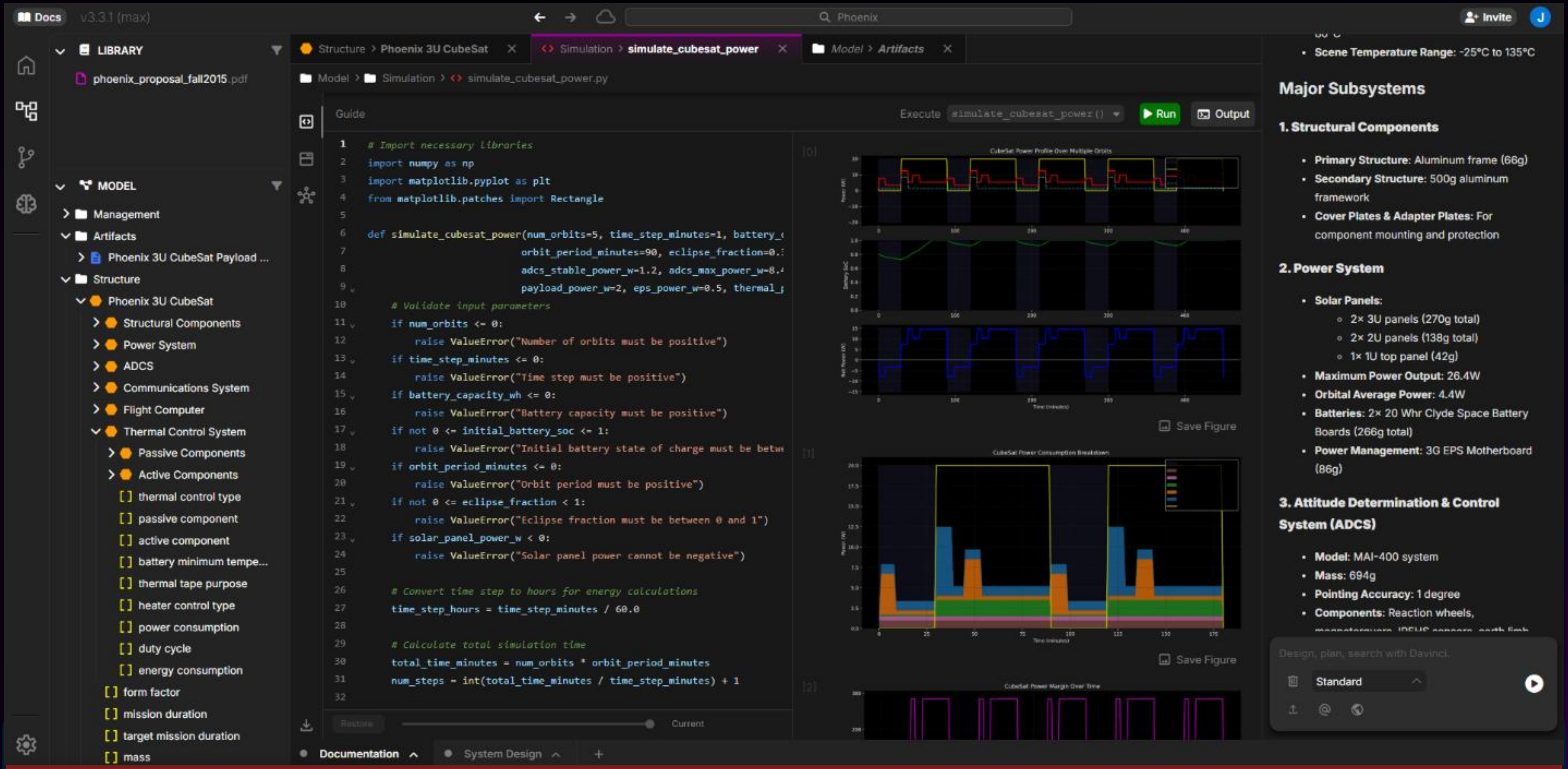
Perfect! I have successfully created a brief but insightful document that explains the key equations used in the Neptune orbital analysis code. The document is structured with clear sections covering:

1. **Fundamental Orbital Mechanics Equations** - Including gravitational potential expansion, acceleration calculations, and numerical integration methods
2. **Transfer Orbit Calculations** - Covering Lambert's theorem, universal anomaly, and delta-V calculations
3. **Gravity Assist Modeling** - Explaining hyperbolic trajectories, deflection angles, and velocity vector rotations
4. **Mission-Specific Calculations** - Including tidal deformation, Love numbers, moment of inertia, and multi-body perturbations

The document provides the mathematical formulations implemented in the code while explaining their physical significance and how they work together for Neptune mission analysis. It's concise yet technically accurate, making it suitable for engineers and mission planners who need to understand the computational backbone of the

Fully Integrated Word Processor with Tables and Equations

24-Page Document With Tables, Equations, and Figures – 5.2 Minutes



```python
# Import necessary libraries
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Rectangle

def simulate_cubesat_power(num_orbits=5, time_step_minutes=1, battery_(
                           orbit_period_minutes=90, eclipse_fraction=0.:
                           adcs_stable_power_w=1.2, adcs_max_power_w=8.4
                           payload_power_w=2, eps_power_w=0.5, thermal_p
    # Validate input parameters
    if num_orbits <= 0:
        raise ValueError("Number of orbits must be positive")
    if time_step_minutes <= 0:
        raise ValueError("Time step must be positive")
    if battery_capacity_wh <= 0:
        raise ValueError("Battery capacity must be positive")
    if not 0 <= initial_battery_soc <= 1:
        raise ValueError("Initial battery state of charge must be betw
    if orbit_period_minutes <= 0:
        raise ValueError("Orbit period must be positive")
    if not 0 <= eclipse_fraction < 1:
        raise ValueError("Eclipse fraction must be between 0 and 1")
    if solar_panel_power_w < 0:
        raise ValueError("Solar panel power cannot be negative")

    # Convert time step to hours for energy calculations
    time_step_hours = time_step_minutes / 60.0

    # Calculate total simulation time
    total_time_minutes = num_orbits * orbit_period_minutes
    num_steps = int(total_time_minutes / time_step_minutes) + 1

```

CubeSat Power Profile Over Multiple Orbits
CubeSat Power Consumption Breakdown
CubeSat Power Margin Over Time

- **Scene Temperature Range**: -25°C to 135°C

Major Subsystems

1. Structural Components

- **Primary Structure**: Aluminum frame (66g)
- **Secondary Structure**: 500g aluminum framework
- **Cover Plates & Adapter Plates**: For component mounting and protection

2. Power System

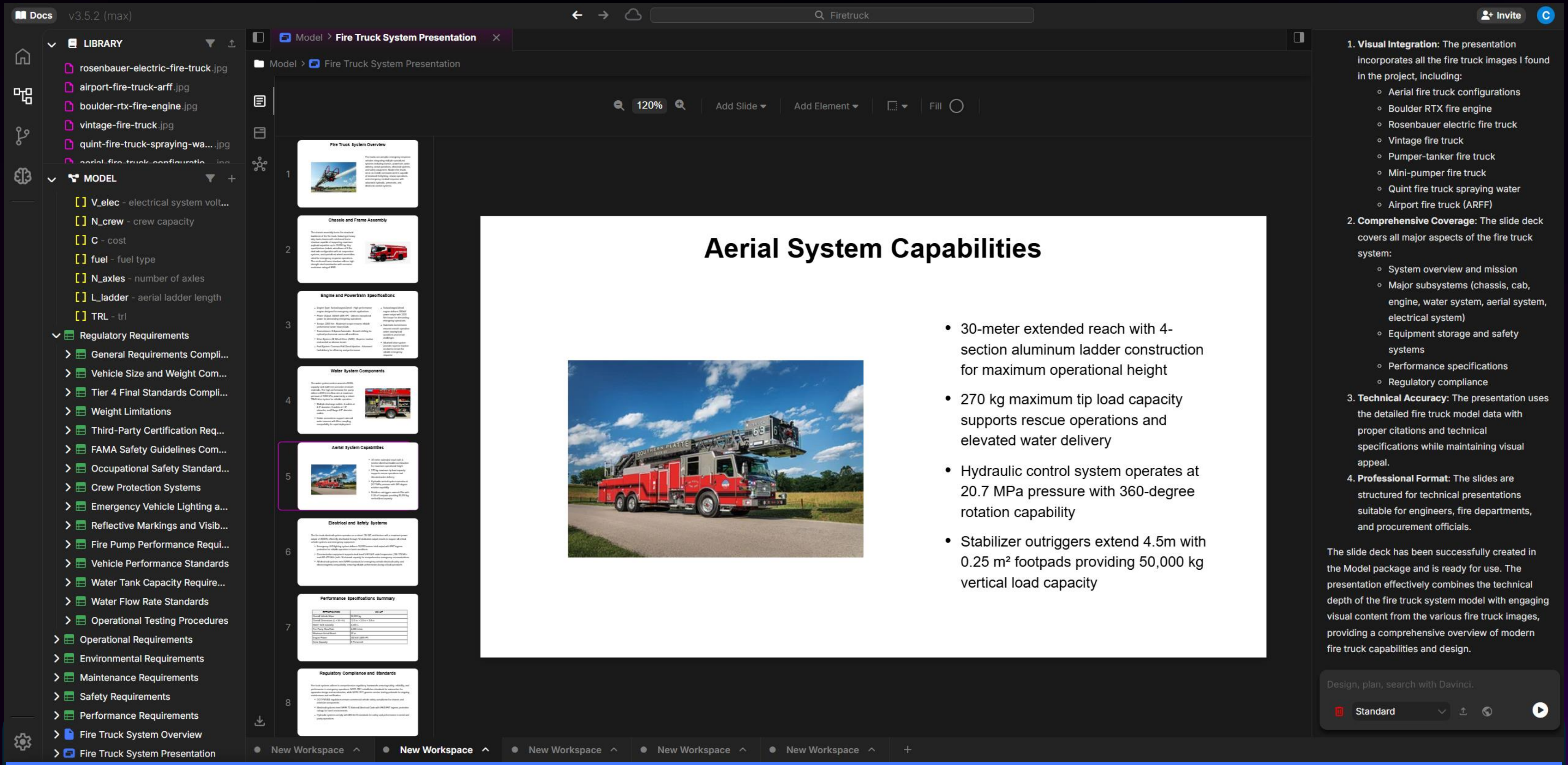
- **Solar Panels**:
  - 2× 3U panels (270g total)
  - 2× 2U panels (138g total)
  - 1× 1U top panel (42g)
- **Maximum Power Output**: 26.4W
- **Orbital Average Power**: 4.4W
- **Batteries**: 2× 20 Whr Clyde Space Battery Boards (266g total)
- **Power Management**: 3G EPS Motherboard (86g)

3. Attitude Determination & Control System (ADCS)

- **Model**: MAI-400 system
- **Mass**: 694g
- **Pointing Accuracy**: 1 degree
- **Components**: Reaction wheels,

Code Execution

2087-Line Simulation – 7.2 Minutes


Docs
v3.5.2 (max)
Firetruck
Invite
C
LIBRARY
rosenbauer-electric-fire-truck.jpg
airport-fire-truck-arff.jpg
boulder-rtx-fire-engine.jpg
vintage-fire-truck.jpg
quint-fire-truck-spraying-wa... .jpg
aerial-fire-truck-configuratio... .jpg
MODEL
[ ] V_elec - electrical system volt...
[ ] N_crew - crew capacity
[ ] C - cost
[ ] fuel - fuel type
[ ] N_axles - number of axles
[ ] L_ladder - aerial ladder length
[ ] TRL - trl
Regulatory Requirements
General Requirements Compli...
Vehicle Size and Weight Com...
Tier 4 Final Standards Compli...
Weight Limitations
Third-Party Certification Req...
FAMA Safety Guidelines Com...
Occupational Safety Standard...
Crew Protection Systems
Emergency Vehicle Lighting a...
Reflective Markings and Visib...
Fire Pump Performance Requi...
Vehicle Performance Standards
Water Tank Capacity Require...
Water Flow Rate Standards
Operational Testing Procedures
Operational Requirements
Environmental Requirements
Maintenance Requirements
Safety Requirements
Performance Requirements
Fire Truck System Overview
Fire Truck System Presentation
Model > Fire Truck System Presentation
Model > Fire Truck System Presentation
120%
Add Slide
Add Element
Fill
1 Fire Truck System Overview
2 Chassis and Frame Assembly
3 Engine and Powertrain Specifications
4 Water System Components
5 Aerial System Capabilities
6 Electrical and Safety Systems
7 Performance Specifications Summary
8 Regulatory Compliance and Standards
New Workspace
New Workspace
New Workspace
New Workspace
New Workspace
Design, plan, search with Davinci.
Standard

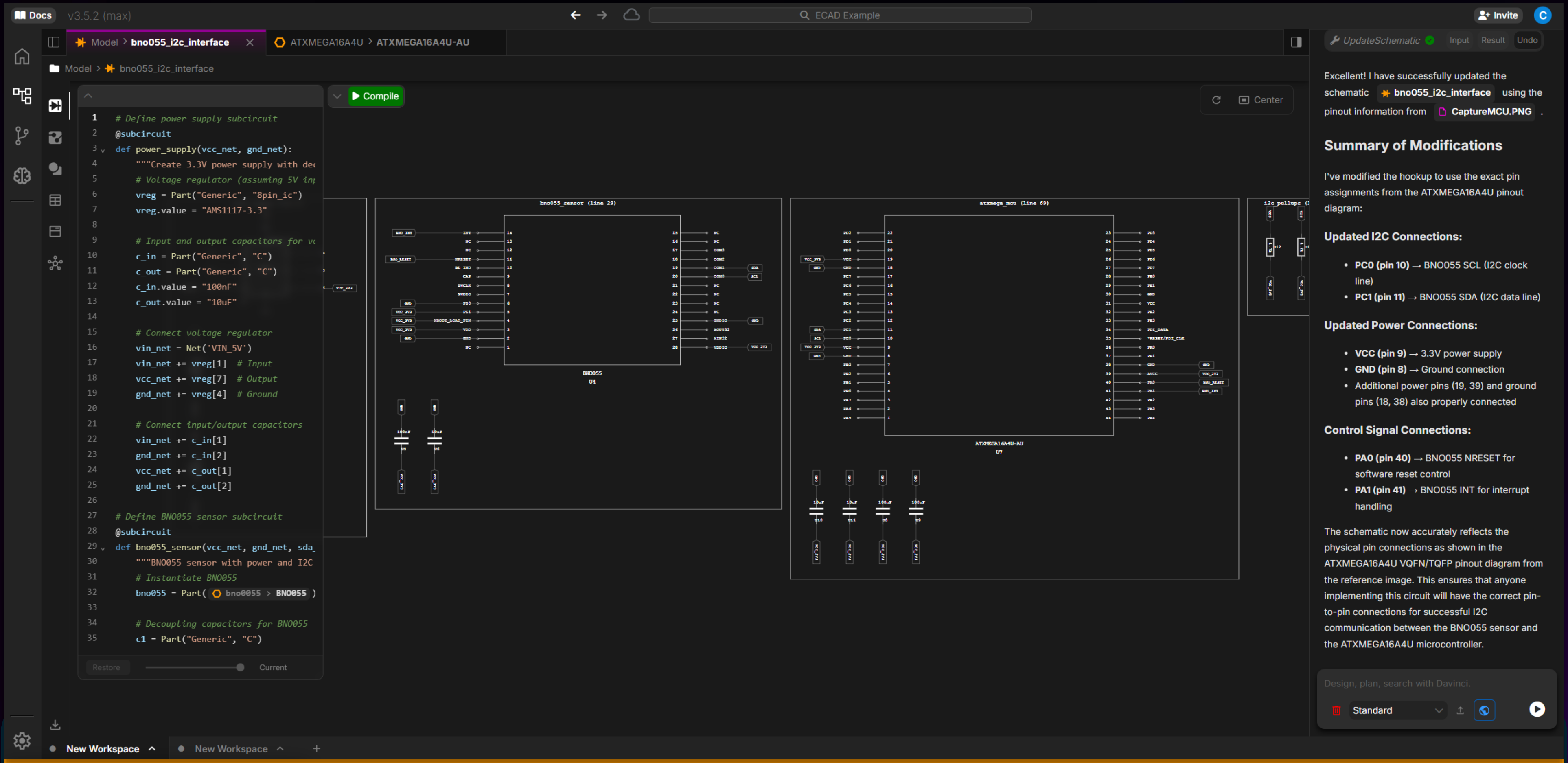
Aerial System Capabilities

- 30-meter extended reach with 4-section aluminum ladder construction for maximum operational height
- 270 kg maximum tip load capacity supports rescue operations and elevated water delivery
- Hydraulic control system operates at 20.7 MPa pressure with 360-degree rotation capability
- Stabilizer outriggers extend 4.5m with 0.25 m² footpads providing 50,000 kg vertical load capacity

1. **Visual Integration**: The presentation incorporates all the fire truck images I found in the project, including:
   - Aerial fire truck configurations
   - Boulder RTX fire engine
   - Rosenbauer electric fire truck
   - Vintage fire truck
   - Pumper-tanker fire truck
   - Mini-pumper fire truck
   - Quint fire truck spraying water
   - Airport fire truck (ARFF)
2. **Comprehensive Coverage**: The slide deck covers all major aspects of the fire truck system:
   - System overview and mission
   - Major subsystems (chassis, cab, engine, water system, aerial system, electrical system)
   - Equipment storage and safety systems
   - Performance specifications
   - Regulatory compliance
3. **Technical Accuracy**: The presentation uses the detailed fire truck model data with proper citations and technical specifications while maintaining visual appeal.
4. **Professional Format**: The slides are structured for technical presentations suitable for engineers, fire departments, and procurement officials.

The slide deck has been successfully created in the Model package and is ready for use. The presentation effectively combines the technical depth of the fire truck system model with engaging visual content from the various fire truck images, providing a comprehensive overview of modern fire truck capabilities and design.

Slide Deck Creation



```
# Define power supply subcircuit
@subcircuit
def power_supply(vcc_net, gnd_net):
    """Create 3.3V power supply with dec
    # Voltage regulator (assuming 5V inp
    vreg = Part("Generic", "8pin_ic")
    vreg.value = "AMS1117-3.3"

    # Input and output capacitors for vc
    c_in = Part("Generic", "C")
    c_out = Part("Generic", "C")
    c_in.value = "100nF"
    c_out.value = "10uF"

    # Connect voltage regulator
    vin_net = Net('VIN_5V')
    vin_net += vreg[1]  # Input
    vcc_net += vreg[7]  # Output
    gnd_net += vreg[4]  # Ground

    # Connect input/output capacitors
    vin_net += c_in[1]
    gnd_net += c_in[2]
    vcc_net += c_out[1]
    gnd_net += c_out[2]

# Define BNO055 sensor subcircuit
@subcircuit
def bno055_sensor(vcc_net, gnd_net, sda_
    """BNO055 sensor with power and I2C
    # Instantiate BNO055
    bno055 = Part( bno0055 > BNO055 )

    # Decoupling capacitors for BNO055
    c1 = Part("Generic", "C")
```

Excellent! I have successfully updated the schematic bno055_i2c_interface using the pinout information from CaptureMCU.PNG .

Summary of Modifications

I've modified the hookup to use the exact pin assignments from the ATXMEGA16A4U pinout diagram:

Updated I2C Connections:

- **PC0 (pin 10)** → BNO055 SCL (I2C clock line)
- **PC1 (pin 11)** → BNO055 SDA (I2C data line)

Updated Power Connections:

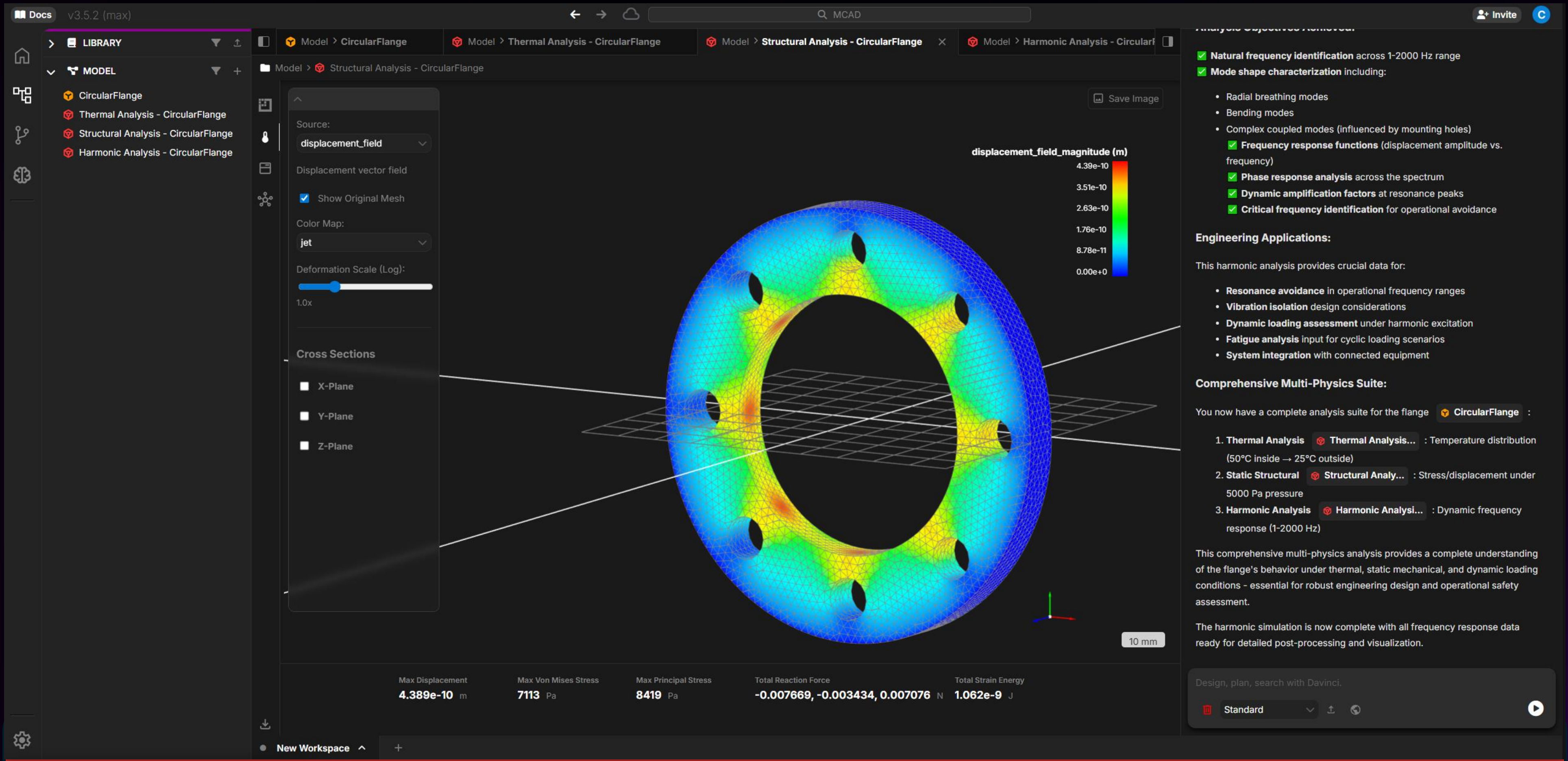
- **VCC (pin 9)** → 3.3V power supply
- **GND (pin 8)** → Ground connection
- Additional power pins (19, 39) and ground pins (18, 38) also properly connected

Control Signal Connections:

- **PA0 (pin 40)** → BNO055 NRESET for software reset control
- **PA1 (pin 41)** → BNO055 INT for interrupt handling

The schematic now accurately reflects the physical pin connections as shown in the ATXMEGA16A4U VQFN/TQFP pinout diagram from the reference image. This ensures that anyone implementing this circuit will have the correct pin-to-pin connections for successful I2C communication between the BNO055 sensor and the ATXMEGA16A4U microcontroller.

Electrical Design


Docs
v3.5.2 (max)
MCAD
Invite
LIBRARY
MODEL
CircularFlange
Thermal Analysis - CircularFlange
Structural Analysis - CircularFlange
Harmonic Analysis - CircularFlange
Model > CircularFlange
Model > Thermal Analysis - CircularFlange
Model > Structural Analysis - CircularFlange
Model > Harmonic Analysis - Circular
Model > Structural Analysis - CircularFlange
Save Image
displacement_field_magnitude (m)
4.39e-10
3.51e-10
2.63e-10
1.76e-10
8.78e-11
0.00e+0
10 mm
New Workspace
Design, plan, search with Davinci.
Standard

Source: displacement_field

Displacement vector field

☑ Show Original Mesh

Color Map: jet

Deformation Scale (Log): 1.0x

Cross Sections

□ X-Plane

□ Y-Plane

□ Z-Plane

Max Displacement	Max Von Mises Stress	Max Principal Stress	Total Reaction Force	Total Strain Energy
4.389e-10 m	7113 Pa	8419 Pa	-0.007669, -0.003434, 0.007076 N	1.062e-9 J

✅ **Natural frequency identification** across 1-2000 Hz range
✅ **Mode shape characterization** including:

- Radial breathing modes
- Bending modes
- Complex coupled modes (influenced by mounting holes)
  ✅ **Frequency response functions** (displacement amplitude vs. frequency)
  ✅ **Phase response analysis** across the spectrum
  ✅ **Dynamic amplification factors** at resonance peaks
  ✅ **Critical frequency identification** for operational avoidance

Engineering Applications:

This harmonic analysis provides crucial data for:

- **Resonance avoidance** in operational frequency ranges
- **Vibration isolation** design considerations
- **Dynamic loading assessment** under harmonic excitation
- **Fatigue analysis** input for cyclic loading scenarios
- **System integration** with connected equipment

Comprehensive Multi-Physics Suite:

You now have a complete analysis suite for the flange CircularFlange :

1. **Thermal Analysis** Thermal Analysis... : Temperature distribution (50°C inside → 25°C outside)
2. **Static Structural** Structural Analy... : Stress/displacement under 5000 Pa pressure
3. **Harmonic Analysis** Harmonic Analysi... : Dynamic frequency response (1-2000 Hz)

This comprehensive multi-physics analysis provides a complete understanding of the flange's behavior under thermal, static mechanical, and dynamic loading conditions - essential for robust engineering design and operational safety assessment.

The harmonic simulation is now complete with all frequency response data ready for detailed post-processing and visualization.

Mechanical Design

Case Study

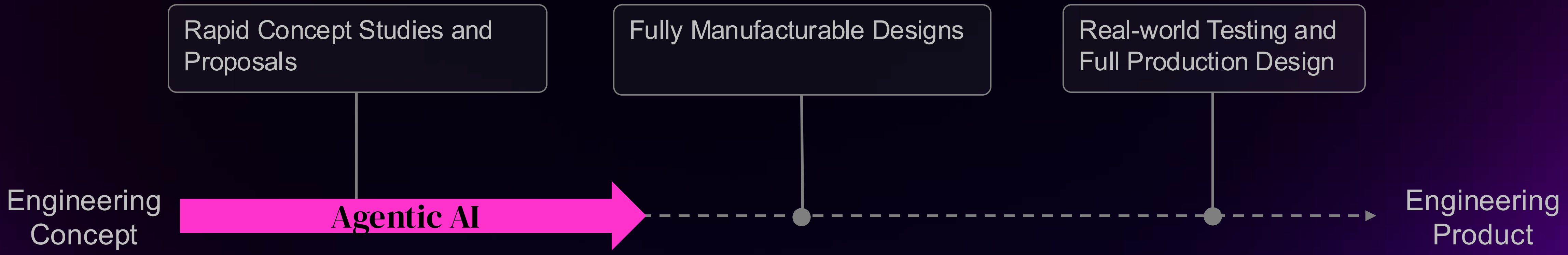
PDF + Goals -> Architecture + Proposal

4 hours of Model Development

46 Requirements, 305 Parts, 3917 Attributes
3 40-page Documents

The Road Ahead

We envision *Agentic AI* being capable of building any system on timescales of hours



Rapid Concept Studies and Proposals

Fully Manufacturable Designs

Real-world Testing and Full Production Design

Engineering Concept

Agentic AI

Engineering Product

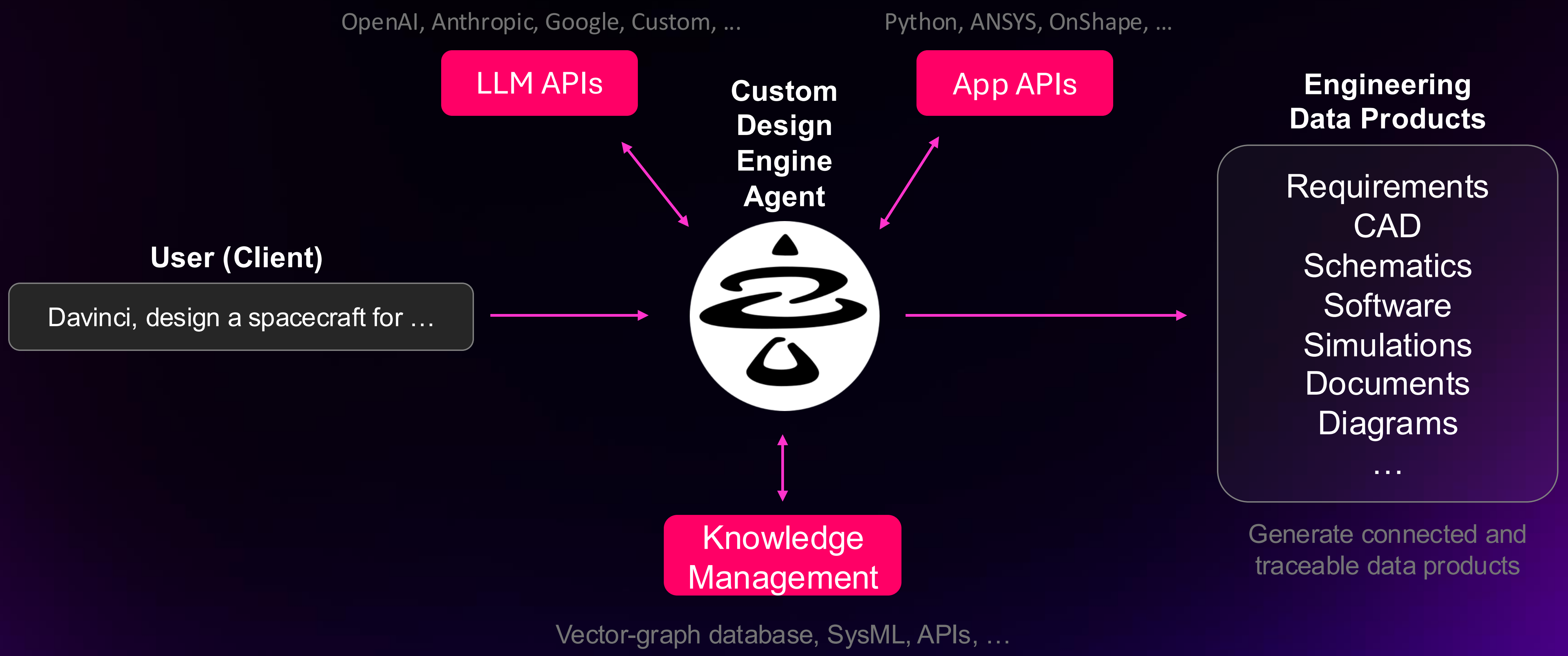
Less than 1% of complex engineering projects deliver **on-time** and **under-budget**

Our Mission Should Be To Change That By Accelerating Engineering by 1,000x Using Agentic AI

Backup Slides

The Davinci platform

Scalable, deployable, and LLM agnostic



OpenAI, Anthropic, Google, Custom, …

LLM APIs

Custom Design Engine Agent

Python, ANSYS, OnShape, …

App APIs

User (Client)

Davinci, design a spacecraft for …

Engineering Data Products

Requirements
CAD
Schematics
Software
Simulations
Documents
Diagrams
…

Generate connected and traceable data products

Knowledge Management

Vector-graph database, SysML, APIs, …

Who Uses Davinci?

Davinci brings together the whole design team, all working in one cohesive environment and database, each empowered by AI design agents

System Engineers

Use Davinci to build the organization of the project, construct requirements, make design studies

Discipline Engineers

Use Davinci to build subsystem diagrams, define interfaces, simulations, CAD models

Managers

Uses Davinci to model processes, perform design reviews, make informed decisions

Stakeholders

Use Davinci to explore documentation, ask questions and get answers about the project

Davinci extends across traditional tool and user boundaries to engineer 100x faster and cheaper