

B.TECH ME • 6TH SEMESTER

Session Jan – May 2026

# Product Cycle and CAD Systems

Computer Aided Design Lecture 1 Module 1

PREPARED BY

**Dinesh Kumar**


Assistant Professor

**D Y PATIL INTERNATIONAL UNIVERSITY**

Precision Engineering  
School of Continuing Education  
COMPUTER AIDED DESIGN

# Learning Objectives

## & Course Agenda

-  Understand the complete product cycle from concept to End-of-Life (EOL)
-  Master design processes, engineering methodologies, and CAD systems
-  Explore CAD architecture, hardware components, and system integration
-  Derive and apply 2D/3D transformations and coordinate systems

## COURSE PROGRESSION

 Product Lifecycle

 Concurrent Eng.

 CAD Systems

 Graphics Basics

 3D Transforms

# Product Cycle

## End-to-End Overview

### Lifecycle Phases

- **Opportunity:** Market needs & VOC
- **Concept & Design:** R&D, Prototyping
- **Manufacturing:** Production ramp-up
- **Use & Service:** Maintenance & Updates
- **EOL:** Recycle, Remanufacture, Dispose

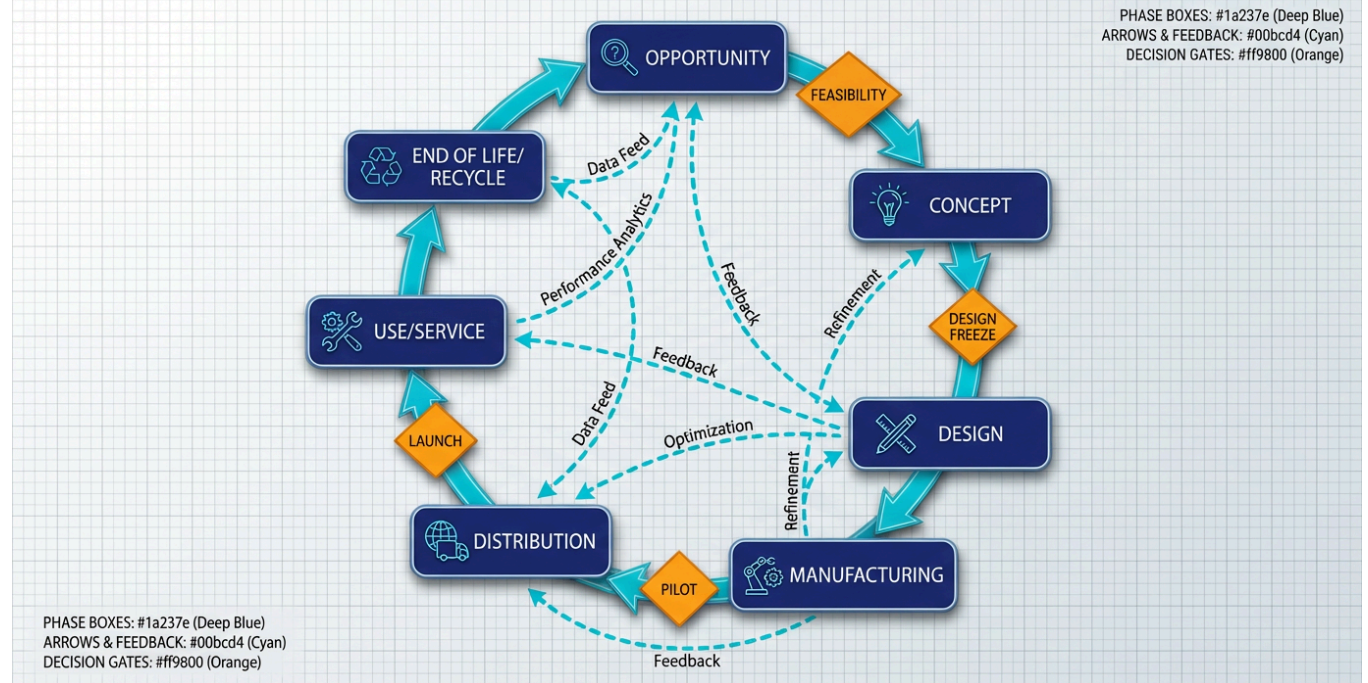
### Key Stage Gates

- **Feasibility:** Technical & Economic check
- **Design Freeze:** Specs locked for tooling
- **Pilot Run:** Validation of process
- **Launch:** Full scale commercialization

### KEY PERFORMANCE INDICATORS

Time-to-Market Cost Quality Reliability Sustainability

## CIRCULAR PRODUCT LIFECYCLE with DIGITAL THREAD & DECISION GATES



# Product Lifecycle

## PHASES & GOVERNANCE



### 1. Concept Phase

Analyze market needs, capture Voice of Customer (VOC), and perform competitive analysis. Output: Product Mission Statement.



### 2. Development Phase

System design, detailed engineering, prototyping, and validation (V-Model). Defining requirements and constraints.



### 3. Production Phase

Manufacturing ramp-up, Supply chain setup, SPC (Statistical Process Control), and PPAP (Production Part Approval Process).



### 4. In-Service Phase

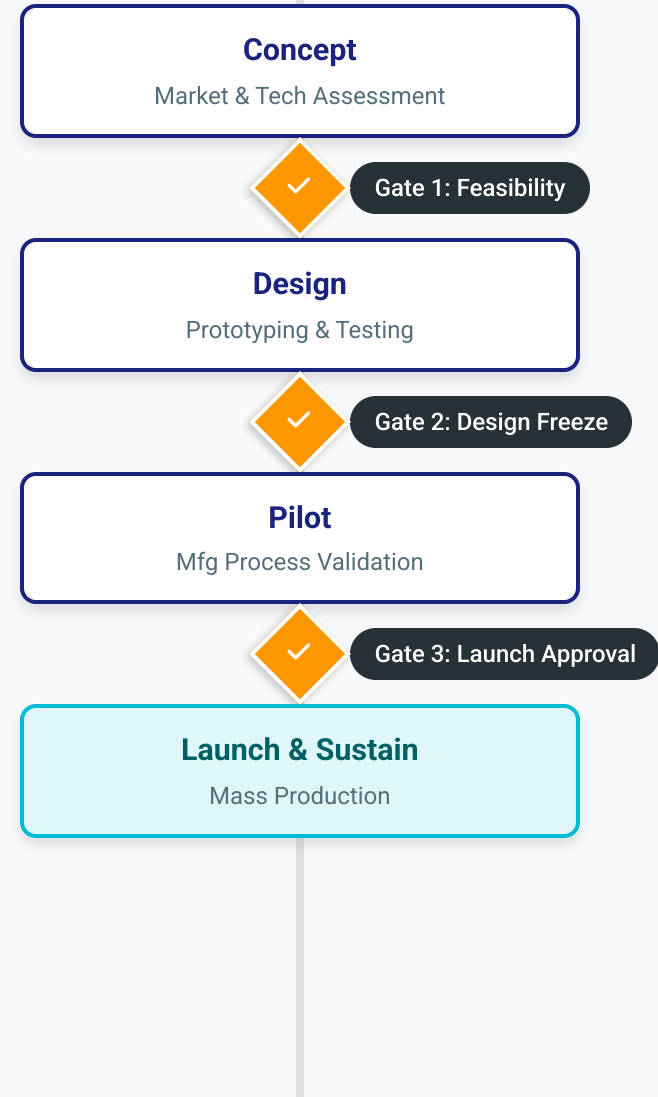
Distribution, customer use, maintenance, software updates, and field data collection for future improvements.



### 5. End of Life (EOL)

Product retirement, remanufacturing, recycling, and ensuring regulatory compliance (e.g., WEEE, RoHS).

## STAGE-GATE GOVERNANCE MODEL



# Design Process

From Problem to Solution

## Systematic Steps

- **Definition:** Problem statement & goals
- **Requirements:** Functional specs & constraints
- **Concept:** Brainstorming & Generation
- **Selection:** Evaluation (Pugh Matrix)
- **Embodiment:** Architecture & Preliminary layout
- **Detail Design:** Technical drawings & specs
- **V&V:** Verification & Validation

## Key Methodologies

Morphological Charts

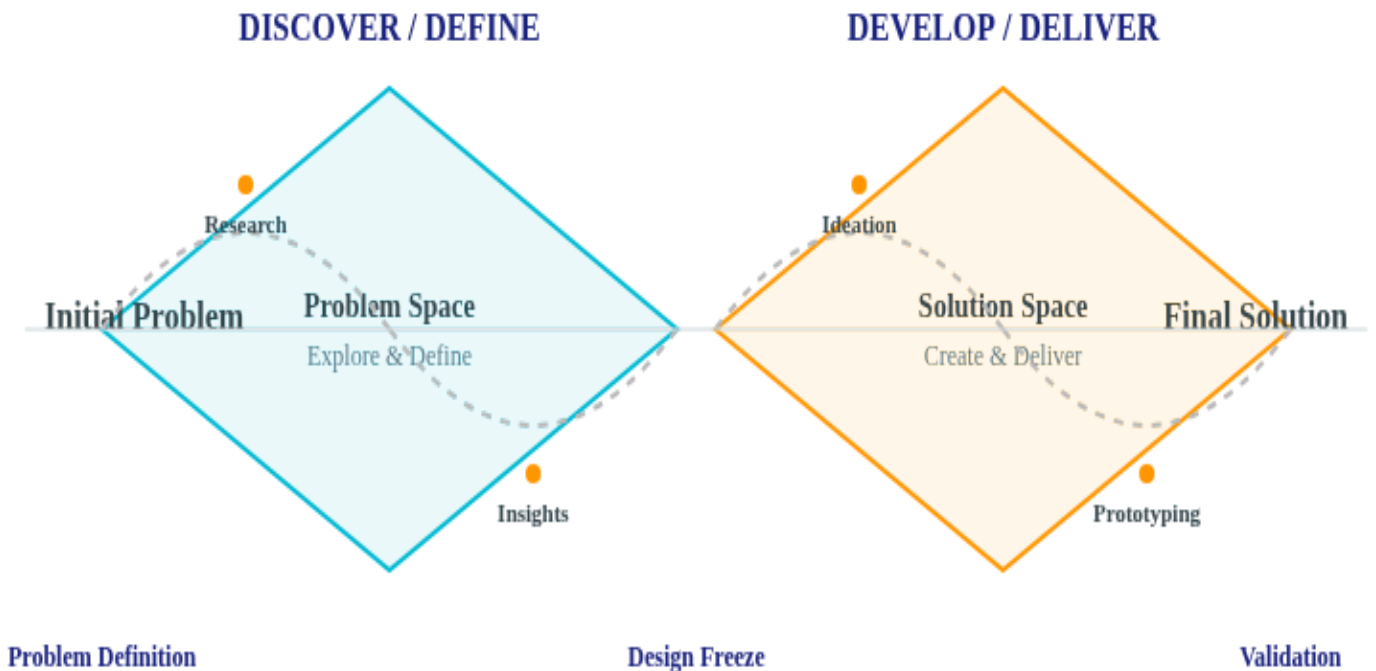
Pugh Matrix

TRIZ

Rapid Prototyping

Brainstorming

SCAMPER



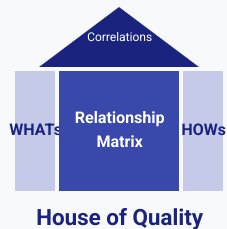
# Design Methods

## Methodologies for Quality, Manufacturability, and Reliability

### QFD

#### Quality Function Deployment

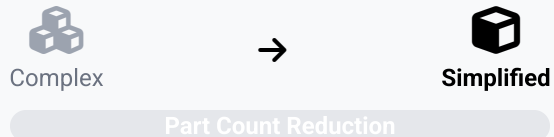
- Translates "Voice of Customer" (VOC) into engineering specs (CTQs).
- Uses "House of Quality" matrix to correlate needs vs. features.
- Prioritizes design requirements based on customer value.



### DFM / DFA

#### Design for Mfg / Assembly

- **DFM:** Optimize part manufacturing (tolerances, materials).
- **DFA:** Minimize part count, standardize fasteners, ease of handling.
- Reduces production cost and assembly time.



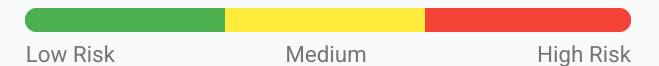
### FMEA

#### Failure Mode & Effect Analysis

- Identify potential failures before they occur.
- **RPN Calculation:**  
Severity × Occurrence × Detection
- Focuses mitigation on high RPN items (Safety/Reliability).

$$RPN = S \times O \times D$$

Risk Priority Number



# Sequential Engineering

Traditional "Over-the-Wall" Approach

## Characteristics

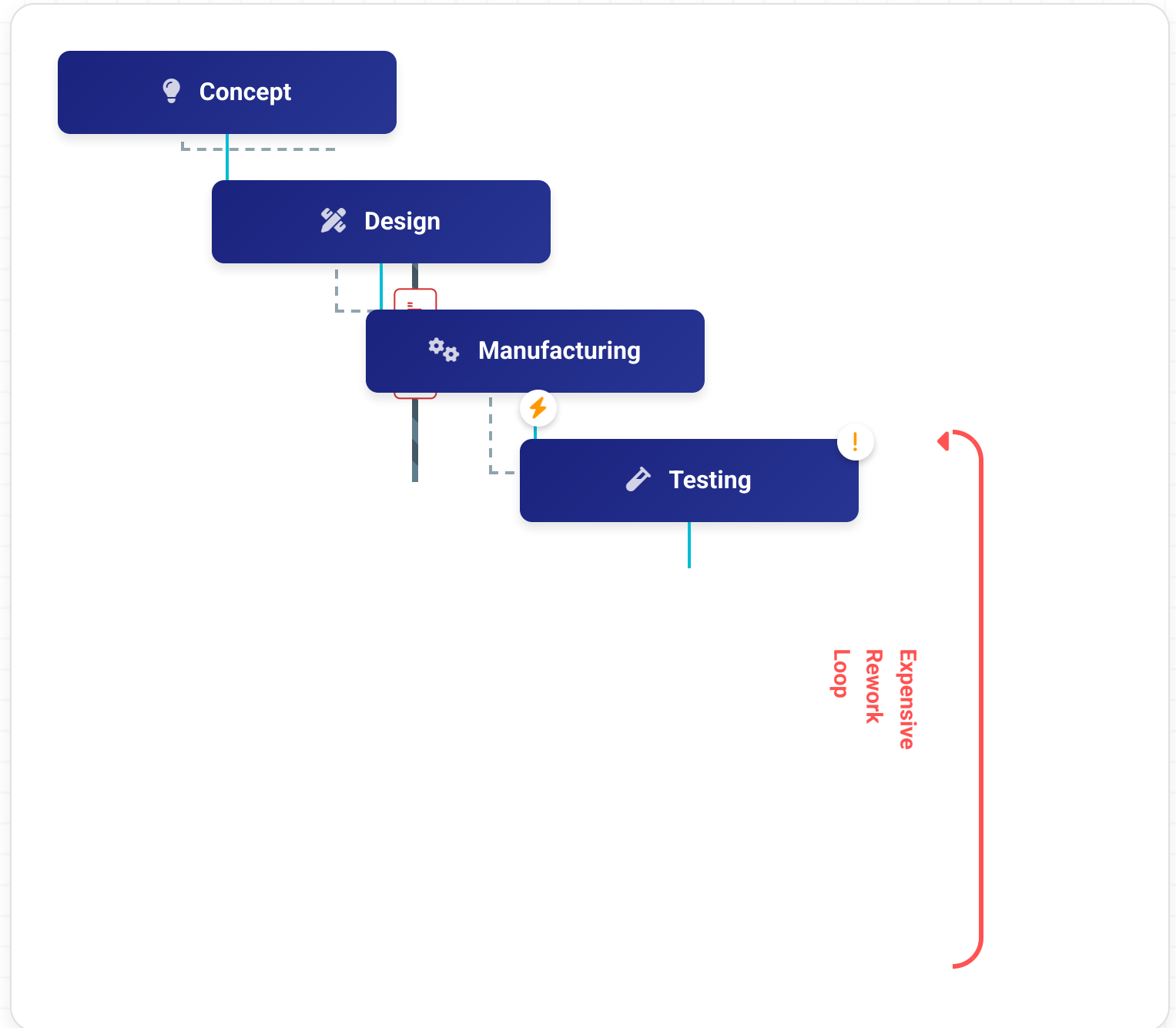
- **Linear Process:** Strict order of phases
- **Siloed Teams:** Limited communication
- **Handoffs:** Completed work passed blindly
- **Late Validation:** Testing only at the end

## Advantages

- Clear ownership & accountability
- Simpler project management
- Well-defined milestones

## Limitations

- Long time-to-market
- Late discovery of design defects
- Expensive rework cycles
- Poor design-for-manufacturing fit



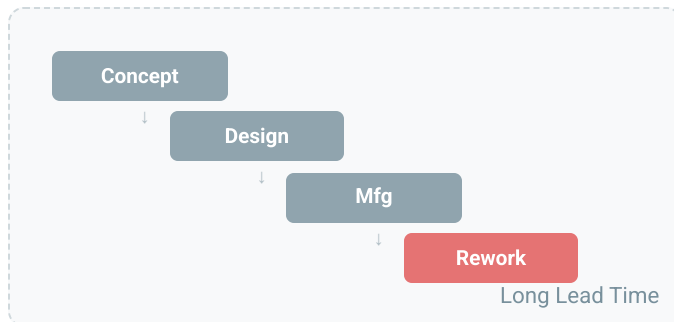
# Concurrent Engineering

## Parallel Workflows vs. Traditional Sequential Handoffs

### Sequential

#### Traditional "Over-the-Wall"

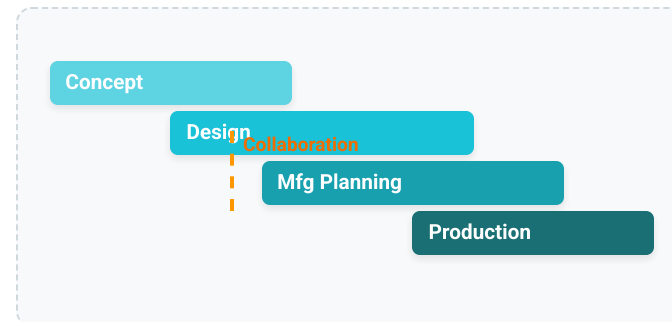
- Linear workflow: Design → Mfg → Test.
- Late discovery of defects leads to costly rework.
- Siloed teams with minimal communication.



### Concurrent

#### Integrated Product Dev

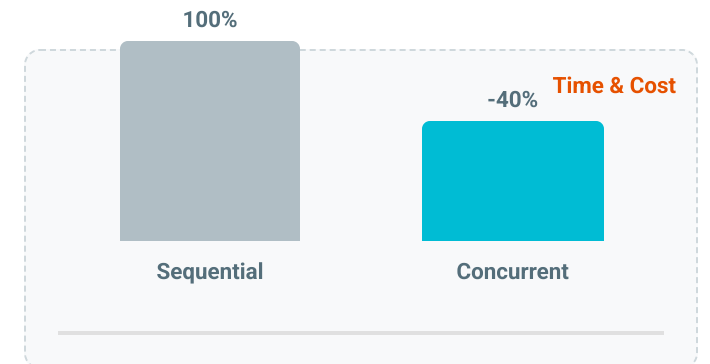
- Parallel execution of phases (overlap).
- Cross-functional teams (Design, Mfg, Mkt).
- Early supplier involvement & feedback.



### Impact

#### Why Switch?

- **Speed:** 30-50% reduced time-to-market.
- **Cost:** Fewer Engineering Change Orders (ECOs).
- **Quality:** Design for Manufacturability (DFM) built-in.



# CAD Systems

## INTRODUCTION, EVOLUTION & APPLICATIONS



### Evolution of CAD

Transitioned from simple 2D electronic drafting to complex 3D Parametric Solid Modeling, and now to Model-Based Definition (MBD) where 3D models carry all PMI data.



### Key Applications

Used extensively for Part & Assembly Design, Complex Surfacing, Simulation Pre-processing (FEA/CFD), and Tooling Design (Molds/Dies).



### Parametric & Feature-Based

Modern CAD uses constraints, dimensions, and relations. Design intent is captured through a history-based feature tree, allowing easy updates.



### Role in Product Cycle

Acts as the central source of truth for the "Digital Thread," feeding data downstream to CAM, CAE, and quality inspection.

## VISUAL PROGRESSION: 2D TO MBD

1

### 2D Drafting

Electronic drawing board, lines & arcs.



2

### 3D Solid Modeling

Wireframe → Surface → Solid geometry.



3

### Parametric Feature-Based

History tree, constraints, associativity.



4

### MBD / PMI

3D Annotated Models (No 2D drawings).



## CORE BENEFITS



Precision



Associativity



Collaboration



Reuse

# CAD Workflow

## Digital Thread Integration

### Workflow Pipeline

- **Input:** Problem Definition & Requirements
- **Modeling:** Parametric 3D CAD Creation
- **Management:** PDM Check-in / Version Control
- **Output:** BOM Generation & 2D Drawings
- **Downstream:** CAM Toolpaths & CAE Sim

### System Interoperability

- **Core Integration:** PLM, ERP, CAM, CAE
- **Change Control:** Revision Mgmt & ECOs

#### NEUTRAL FORMATS

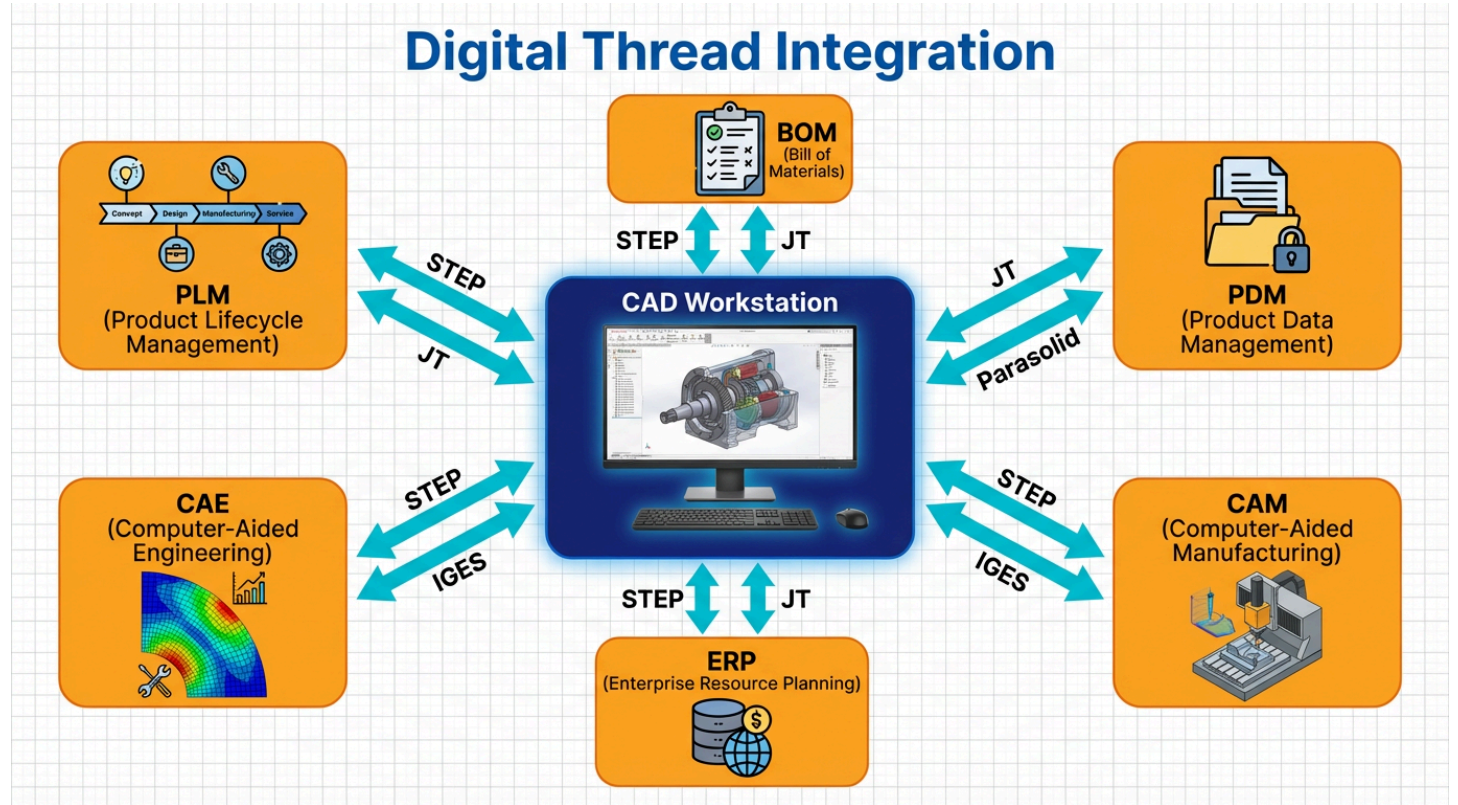
STEP

IGES

JT

Parasolid

## Digital Thread Integration



# CAD Hardware

## WORKSTATION ARCHITECTURE & REQUIREMENTS



### Processor (CPU)

High clock speed (>4.0 GHz) is crucial for parametric modeling (linear). Multi-core helps with rendering and simulation.

**Rec: Intel Core i9/Xeon or AMD Ryzen 9**



### Graphics (GPU)

Professional cards offer certified drivers for stability and ECC VRAM. Handles viewport rotation and shading.

**Rec: NVIDIA RTX A-Series / Quadro (≥ 8GB VRAM)**



### Memory (RAM)

Holds full assembly data in active memory. Prevents swapping to disk which kills performance.

**Rec: 32GB (Entry) - 64GB+ (Complex Assemblies)**

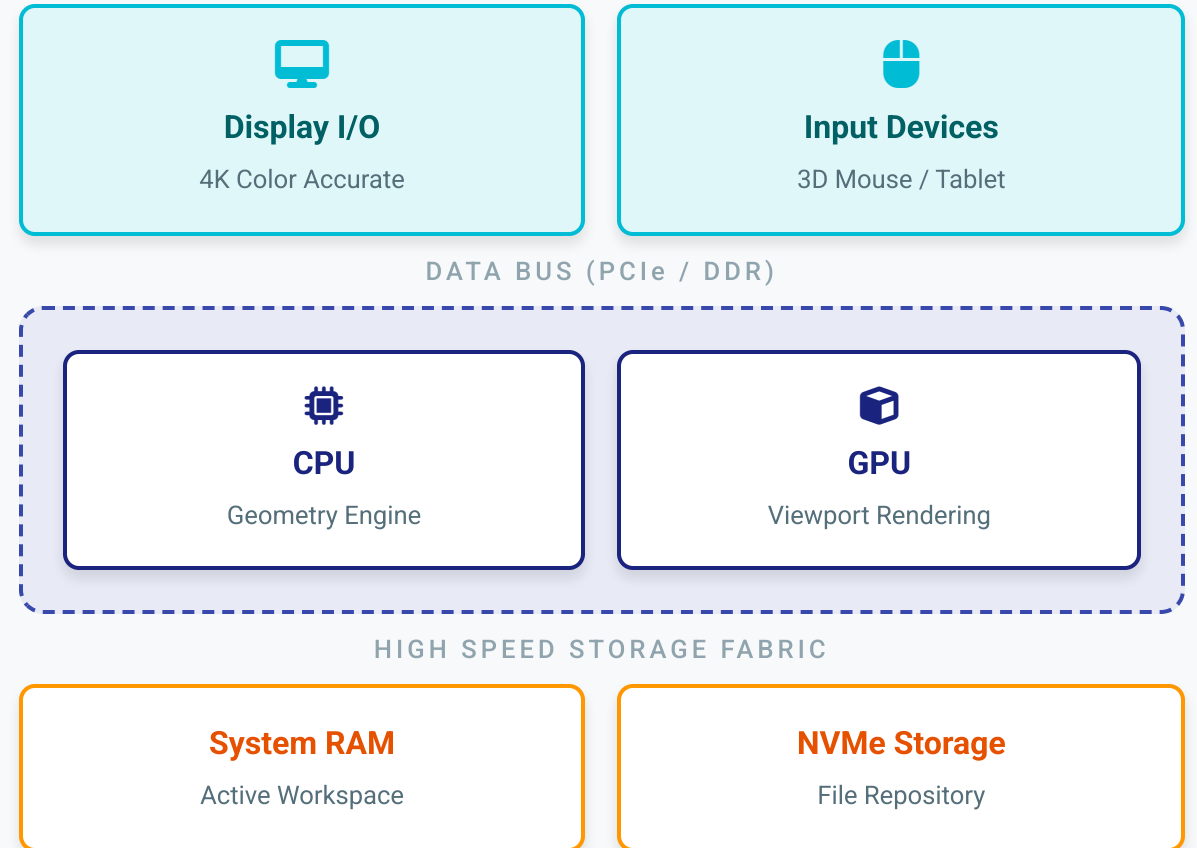


### Storage & I/O

Fast loading of large assemblies. High-res display for detail.

**Rec: NVMe SSD (Gen4/5), 4K IPS Monitor, SpaceMouse**

## WORKSTATION PERFORMANCE STACK



# CAD System Architecture

## Software Kernel & Geometry Mathematics

### √x NURBS Surface Definition (Non-Uniform Rational B-Spline)

#### GENERAL SURFACE EQUATION S(U, V)

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j} \mathbf{P}_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j}}$$

$\mathbf{P}_{\{i,j\}}$	Control points forming the control net
$w_{\{i,j\}}$	Weights determining the influence of control points
$N_{\{i,p\}}, N_{\{j,q\}}$	B-spline basis functions of degree p and q
$u, v$	Parametric variables in directions U and V

#### B-Rep (Boundary Rep)

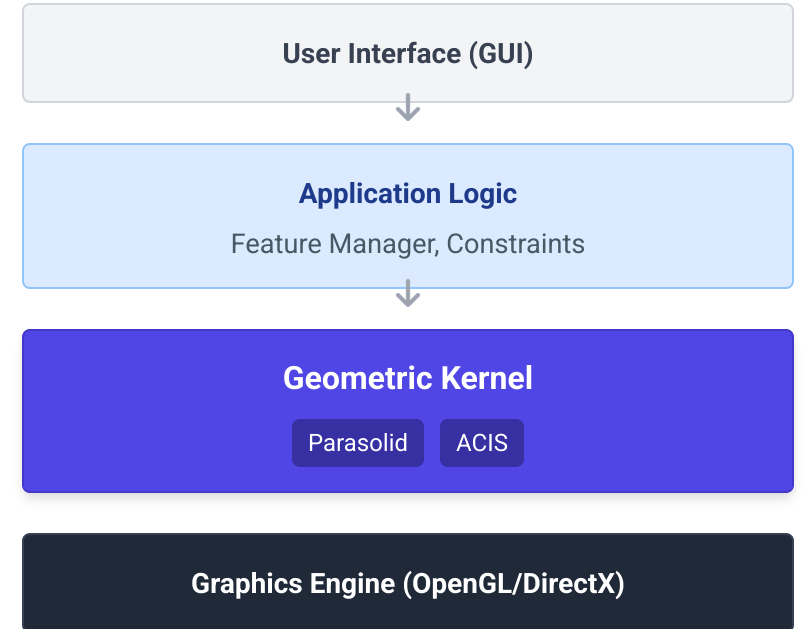
Object defined by topological limits: Vertices, Edges, Faces. Uses Euler-Poincaré Formula:

$$\mathbf{V} - \mathbf{E} + \mathbf{F} = 2 \text{ (for simple convex polyhedra).}$$

#### CSG (Constructive Solid)

Object constructed via Boolean operations (Union  $\cup$ , Intersection  $\cap$ , Difference  $-$ ) on primitives like spheres, cubes, cylinders.

#### Software Architecture



#### Data Structures

- **Half-Edge Structure:** Efficient for mesh traversal.
- **Scene Graph:** Hierarchical arrangement of assembly components.
- **Octree/BSP:** Spatial partitioning for collision detection.

# Computer Graphics

## FUNDAMENTALS & PIPELINE



### Raster vs. Vector

Two fundamental ways to represent 2D images.

- **Raster:** Grid of pixels (Bitmap, JPEG). Resolution dependent. Good for complex textures/photos.
- **Vector:** Geometric primitives (Lines, Curves). Infinite scaling without quality loss. Used in CAD.



### Graphics Buffers

Memory areas storing data for each pixel on the screen.

- **Frame Buffer:** Stores final color values (RGBA).
- **Z-Buffer (Depth):** Stores depth (distance) to handle object occlusion.
- **Stencil Buffer:** Limits rendering area (masking/clipping).



### Shading Models

Methods to calculate light intensity on surfaces.

- **Flat:** One color per polygon face (blocky).
- **Gouraud:** Vertex lighting interpolated across face (smooth, no specular).
- **Phong:** Normal interpolation per pixel (high quality specular highlights).

## THE RENDERING PIPELINE

1

### Model & World Transform

Object Coords → World Space



2

### Viewing & Projection

Camera Space → Clip Space (Frustum)



3

### Rasterization

Primitives → Fragments (Pixels)



4

### Fragment Processing

Shading, Texturing, Depth Test



5

### Display

Frame Buffer → Screen



# Coordinate Systems

## Conversions and Mathematical Definitions

### Spherical Coordinates $(\rho, \theta, \phi)$ to Cartesian $(x, y, z)$

#### TRANSFORMATION EQUATIONS

$$\begin{aligned}x &= \rho \sin \phi \cos \theta \\y &= \rho \sin \phi \sin \theta \\z &= \rho \cos \phi\end{aligned}$$

Where  $\rho$  is radius,  $\theta$  is azimuthal angle (XY plane),  $\phi$  is polar angle (from Z-axis).

### Cylindrical Coordinates $(r, \theta, z)$

#### TRANSFORMATION EQUATIONS

$$\begin{aligned}x &= r \cos \theta \\y &= r \sin \theta \\z &= z\end{aligned}$$

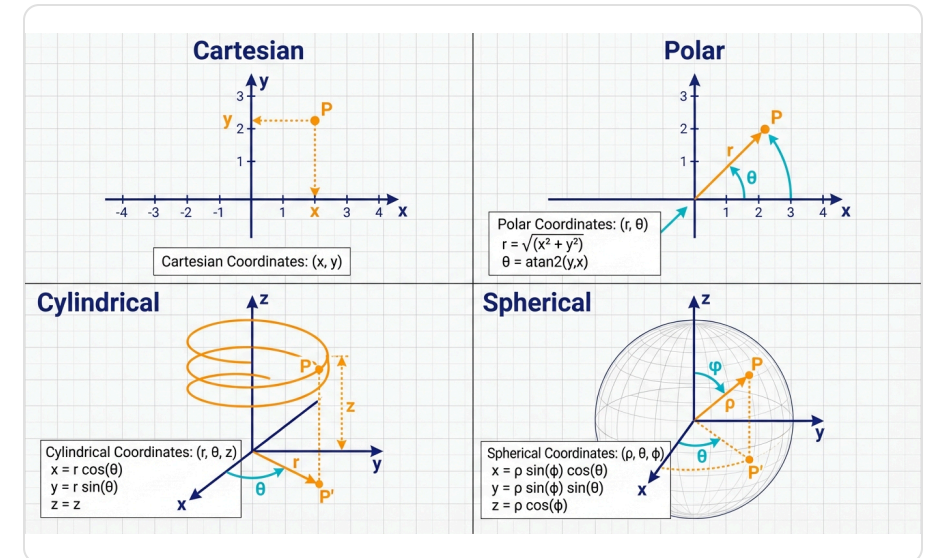
### Polar (2D) $(r, \theta)$

$$\begin{aligned}x &= r \cos \theta \\y &= r \sin \theta\end{aligned}$$

### Inverse (to Polar)

$$\begin{aligned}r &= \sqrt{x^2 + y^2} \\ \theta &= \text{atan2}(y, x)\end{aligned}$$

## System Visualization



Visual comparison of reference frames and basis vectors

## Key Applications

- **Spherical:** Camera rotations, environment mapping, orbital mechanics.
- **Cylindrical:** Modeling shafts, pipes, and rotational symmetry objects.
- **Homogeneous Coords:** Unified handling of translation and rotation (essential for 3D matrices).

### Convention Note

In graphics (OpenGL), Y often points up. In physics/engineering, Z often points up. Always verify the "up" vector and handedness.

# 2D Transformations

## Homogeneous Matrix Operations (3x3)

General Form:  $p' = M \cdot p$

TRANSLATION  $T(T_x, T_y)$

MOVE OBJECT

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

ROTATION  $R(\theta)$

COUNTER-CLOCKWISE ABOUT ORIGIN

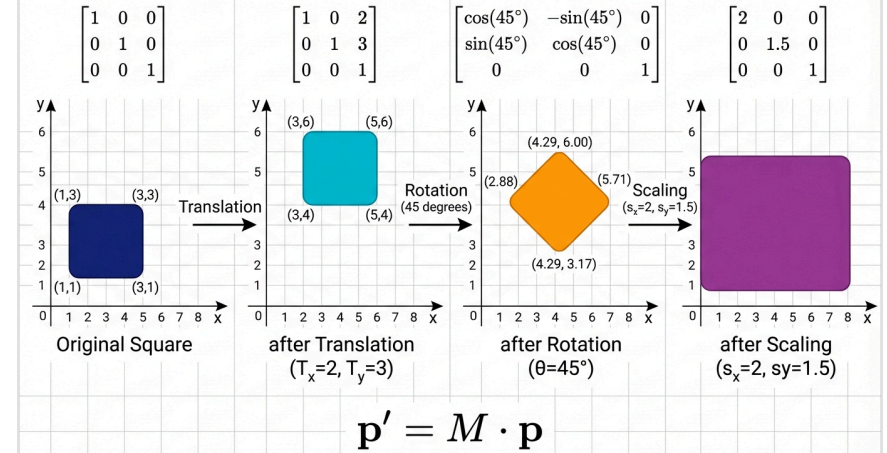
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

SCALING  $S(S_x, S_y)$

$$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

REFLECTION (X-AXIS)

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Step-by-step transformation: Original → Translate → Rotate → Scale

## Composite Transformations

Matrix multiplication is non-commutative ( $A \cdot B \neq B \cdot A$ ). Order matters!

$$M = T \cdot R \cdot S$$

(Scale first, then Rotate, then Translate)

## Why Homogeneous?

- Unifies Translation (addition) with Rotation/Scaling (multiplication) into a single matrix format

# 3D Transformations

## Homogeneous Coordinates (4x4 Matrices)

### Translation Matrix $T(t_x, t_y, t_z)$

$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Applied as: } \mathbf{P}' = T \cdot \mathbf{P}$$

### Rotation Matrices (Right-Hand Rule)

#### X-AXIS ROTATION (ROLL) - $R_x(\alpha)$

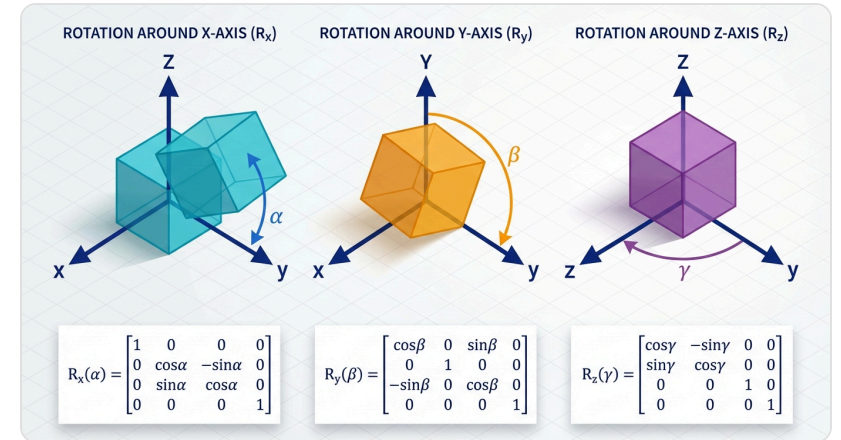
$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

#### Y-AXIS ROTATION (PITCH) - $R_y(\beta)$

$$R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

#### Z-AXIS ROTATION (YAW) - $R_z(\gamma)$

$$R_z(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



### Visualizing Rotations

The diagram demonstrates rotations around the three principal axes. Note that in a right-handed system, positive rotation is counter-clockwise when looking towards the origin from the positive axis.

### Composite Transformation

Matrix multiplication is **non-commutative** ( $A \cdot B \neq B \cdot A$ ). The order of operations matters significantly.

$$M_{total} = T \cdot R_z \cdot R_y \cdot R_x \cdot S$$

Example: Translate after Rotation vs Rotate after Translation yields different results.

# Projection & Viewing

## Camera Models & Transformation Matrices

### 👁️ Pin-hole Camera Model (Perspective)

#### BASIC PROJECTION RELATIONSHIP

$$x' = f \frac{x}{z}, \quad y' = f \frac{y}{z} \implies \begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} \approx \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$f$  Focal length (distance from eye to near plane)

$(x,y,z)$  Point in Camera Coordinate System

$w$  Homogeneous component (used for perspective divide)

### 📦 General Perspective Projection Matrix

#### MAP FRUSTUM TO NDC (NORMALIZED DEVICE COORDINATES)

$$M_{persp} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$n, f$  Near and Far clipping planes (z-coordinates)

$l, r$  Left and Right frustum bounds (at near plane)

$t, b$  Top and Bottom frustum bounds (at near plane)

### 📌 Projection Types

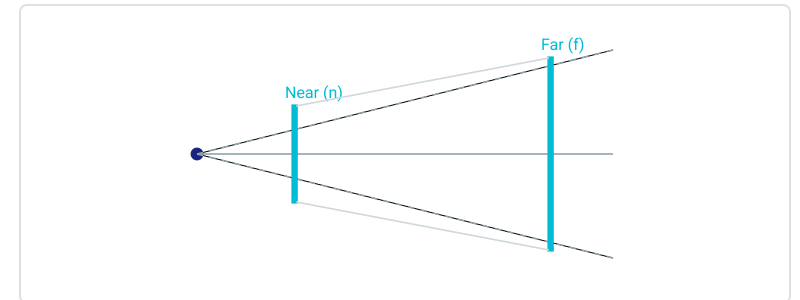
#### Perspective

Objects appear smaller as they get further away. Parallel lines converge at vanishing points. Mimics human vision.

#### Orthographic

Parallel projection. No depth perception scaling. Essential for engineering drawings and CAD measurements.

### 👁️ View Frustum



Volume visible to the camera (Truncated Pyramid)

### 📷 Camera Transform

Transforms world coordinates to camera coords.

$$P_{cam} = R \cdot (P_{world} - C)$$

# Advanced 3D Concepts

## Viewing Pipeline & Precision



### Transformation Pipeline

Object → World → Camera → Clip → NDC → Screen Space sequence.



### Clipping & Culling

Discarding geometry outside frustum and back-facing polygons to optimize rendering.



### Depth Precision (Z-Buffer)

Non-linear distribution of depth values. Importance of Near/Far plane ratio to avoid Z-fighting.



### Coordinate Conventions

Right-handed (OpenGL) vs Left-handed (DirectX) systems affect matrix composition.

## VIEWING PIPELINE

 Object Space

 World Space

 Camera/View

 Clip / NDC

 Screen Space

# Summary

## Key Takeaways & Conclusions

### 🔄 End-to-End Integration

Product success relies on the seamless integration of design, manufacturing, and lifecycle management (PLM) from concept to disposal.

### 🏗️ Concurrent Efficiency

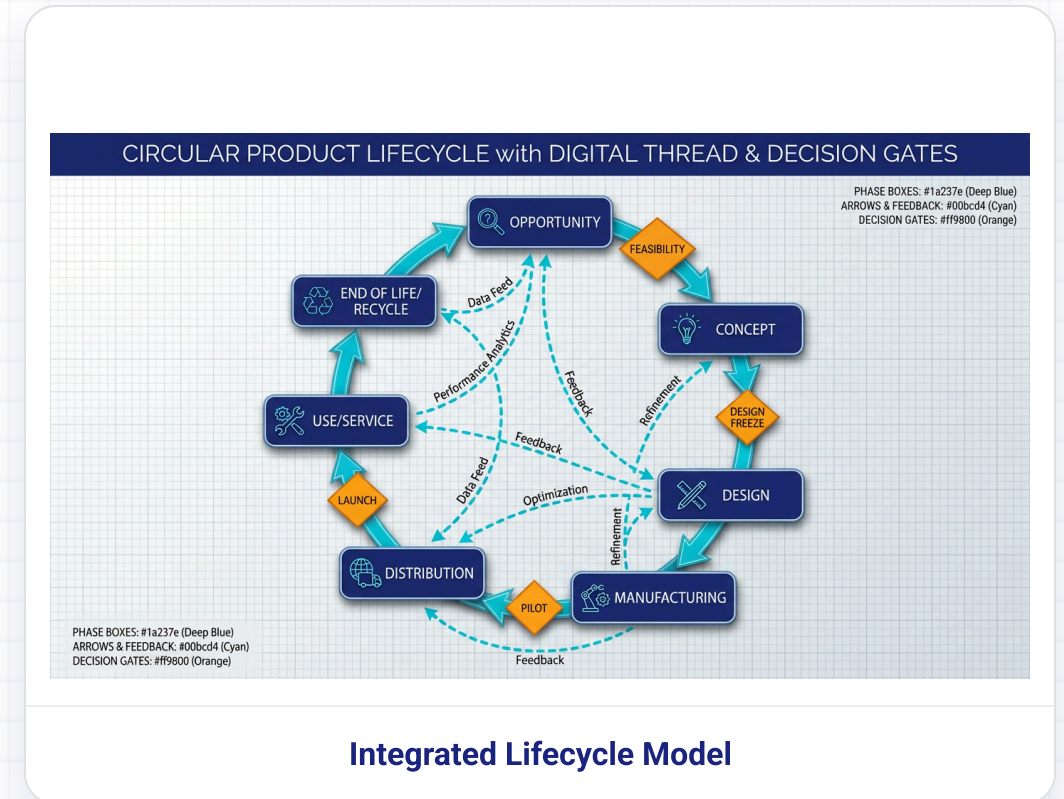
Shifting from sequential to concurrent engineering significantly reduces time-to-market and minimizes late-stage design changes.

### 💻 Digital Foundation

CAD systems and geometric modeling are the backbone of modern engineering, enabling precise simulation and digital twin capabilities.

### 📐 Mathematical Core

Mastery of coordinate systems and matrix transformations is essential for understanding 3D graphics, robotics, and advanced CAD operations.



## Questions & Discussion

Thank you for your attention