

# Process Models

Part 2: Specialized, Unified, and Disciplined Approaches

---

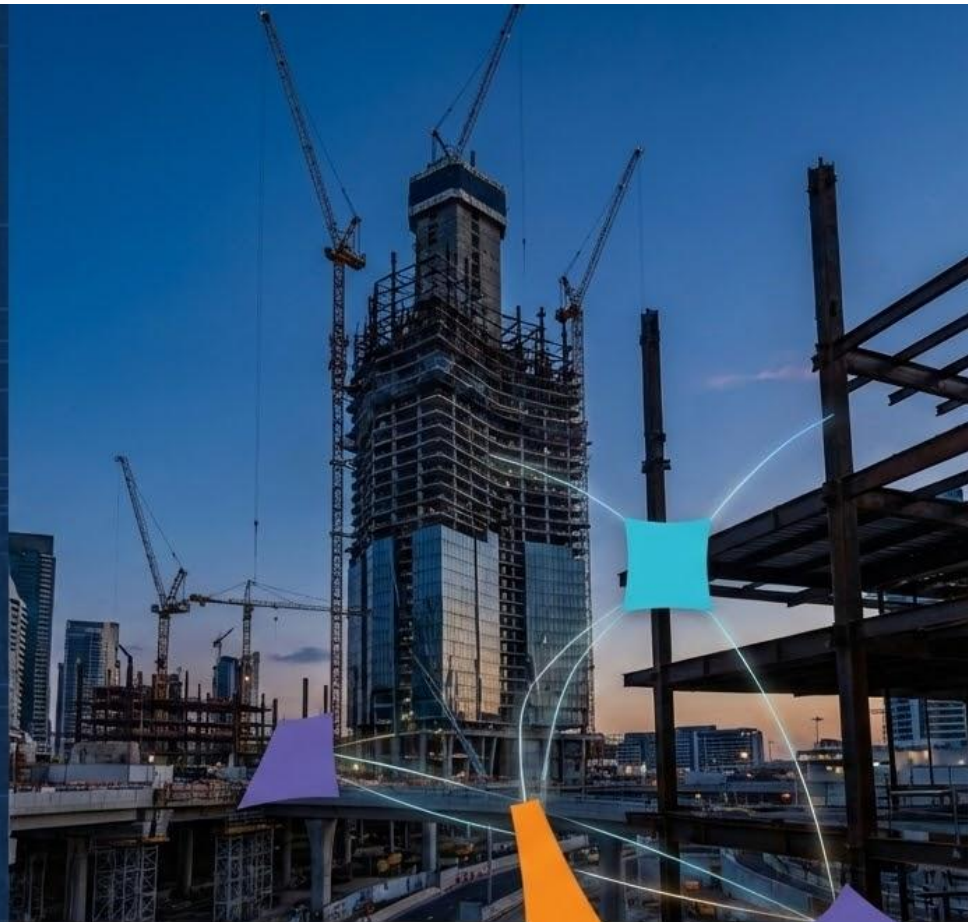
Software Engineering | Chapter 2

# BEYOND THE BASICS

---

"We've seen linear and evolutionary models. But what about systems built from Lego blocks? Or systems where correctness is a matter of life and death?"

Today, we explore **Specialized**, **Unified**, and **Disciplined** process models designed to address these 21st-century challenges.



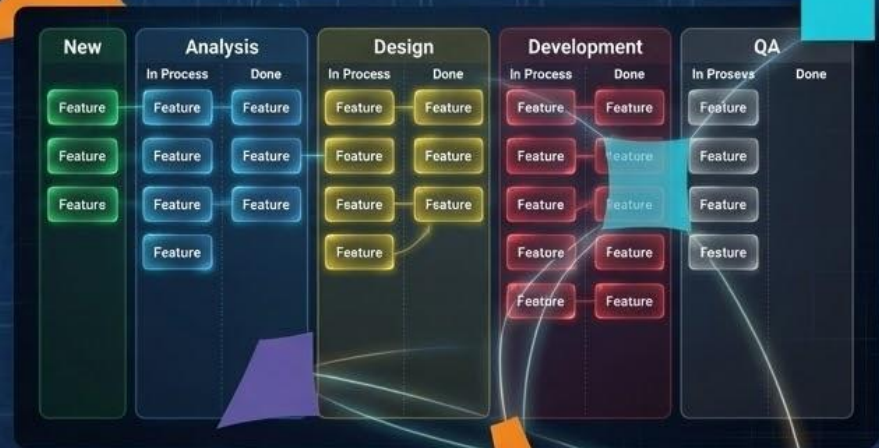
# Concurrent Models

## The Reality of Parallelism

Traditional linear models fail to capture the reality that different components are often in different stages of development simultaneously.

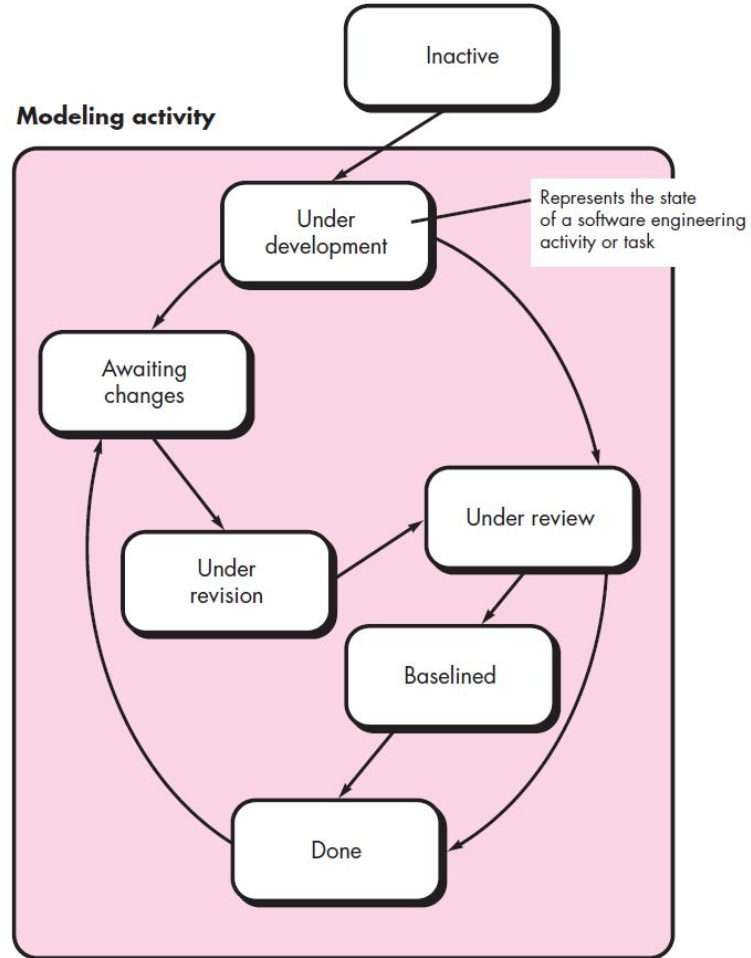
## The State Transition Approach

- Activities exist concurrently in different **states**.
- Example States: *Under Development*, *Awaiting Changes*, *Baselined*.
- **Why?** Accurately models Agile/Iterative workflows.



**FIGURE 2.8**

One element of  
the concurrent  
process model



# Component-Based Development

## "Assemble, Don't Build"

Why build from scratch when you can assemble from pre-built, reusable components (COTS)?



## The Process



**1. Qualification:**  
Do existing components fit?



**2. Adaptation:**  
Modify or "wrap" them.



**3. Composition:**  
Integrate and test.

## Benefits

- ✓ Faster development
- ✓ Reduced cost
- ✓ Higher reliability (with proven components)

## Challenges

- ⚠ Loss of design control
- ⚠ Ongoing vendor dependency
- ⚠ Integration complexities
- ⚠ Potential licensing costs

# Formal Methods

## The Mathematically Rigorous Model

Uses formal mathematical specification languages (like Z or VDM) to describe and verify system requirements before a single line of code is written.

### When to use?

✈ Safety-critical systems (Aircraft, Nuclear).

#### ✔ Advantages:

- Unambiguous specification
- Can prove the absence of certain defect classes.

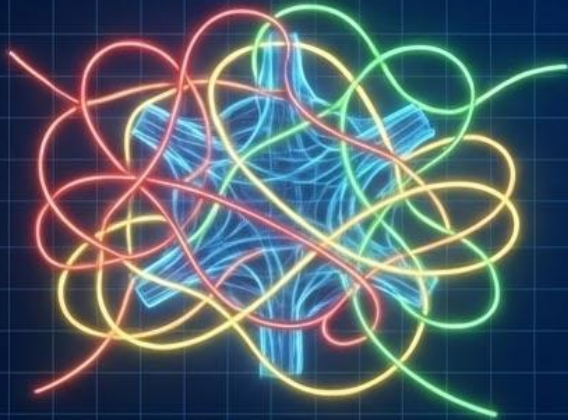
#### ✘ Disadvantages:

- High cost & steep learning curve
- Difficult for stakeholders
- "Overkill" for most business apps.


# Aspect-Oriented Development (AOSD)

## Cross-Cutting Concerns

Some features (security, logging, error handling) cut across many modules, tangling the code.



## The Solution

 **Aspects:** Design and code these concerns independently from the core business logic.



Security



Logging



Error Handling

 **Weaving:**

Use a special compiler to merge the 'Aspects' into the final system at specific points.



# The Unified Process (UP)

An industry-standard, architecture-centric, iterative framework.



1

## Inception

Establish business case & vision.



2

## Elaboration

Refine requirements & baseline architecture.



3

## Construction

Code and build the features.



4

## Transition

Beta testing & deployment.



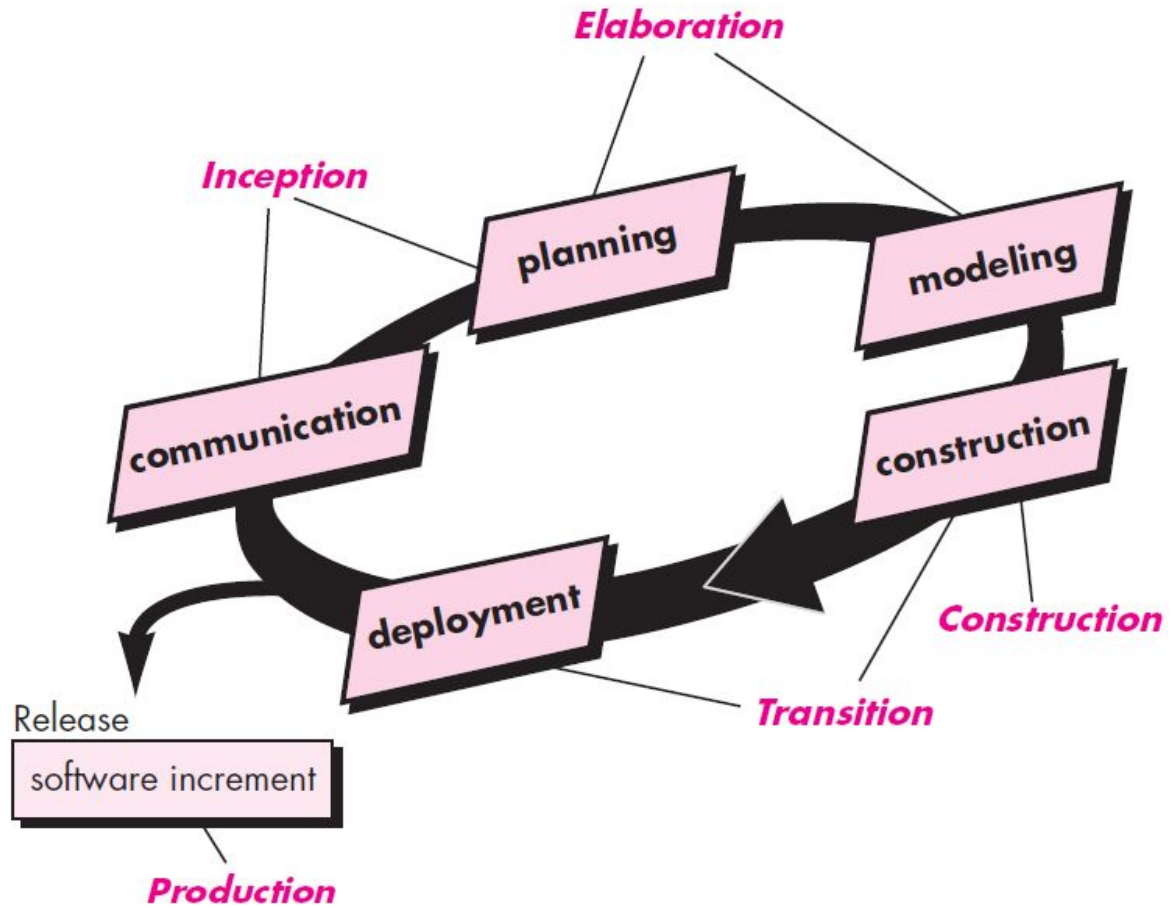
5

## Production

Live use & support.

**FIGURE 2.9**

The Unified Process



# Process Discipline



## Personal Software Process (PSP)

**Focus:** The Individual

**Goal:** Make engineers more effective through personal measurement.

**Practices:** Plan work, track time/defects, and improve personal estimation. It builds the foundation for team success.



## Team Software Process (TSP)

**Focus:** The Team

**Goal:** Build self-directed, high-performance teams.

**Practices:** Teams use data from individual PSPs to plan, manage, and own their project commitments.

# Process Technology

## CASE Tools

### Upper-CASE

Supports early activities  
(Analysis & Design Modeling).



### Lower-CASE

Supports later activities  
(Coding, Debugging, Testing).



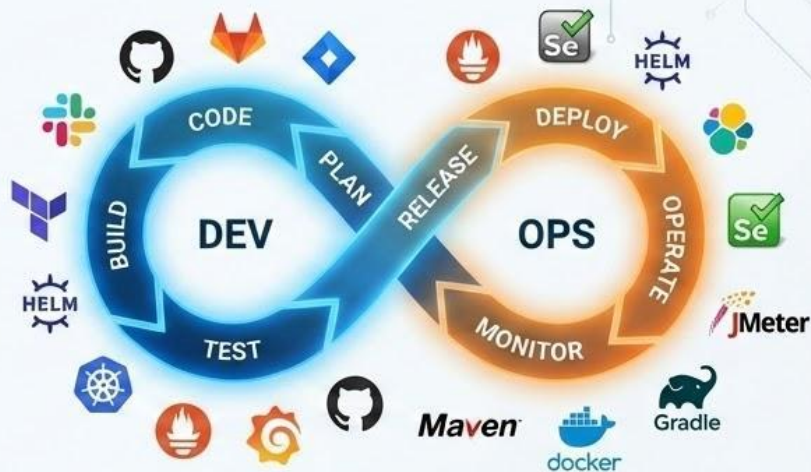
### Integrated-CASE

A full environment from  
analysis to code generation.



## The Modern Ecosystem

- Integrated Development Environments (IDEs)
- DevOps Pipelines (CI/CD)
- Project Management Tools (e.g., Jira)
- Model-Driven Engineering (MDE)



# The Fundamental Relationship

The quality of the product is directly determined by the quality of the process used to build it.



A **chaotic** process leads to an unpredictable product.



A **disciplined** process enables high-quality engineering.

# Key Takeaways



## Concurrent Models

Realistically depict the parallel nature of modern development.



## Specialized Models

CBD, Formal Methods, and AOSD solve specific problems like reuse, safety, and complexity.



## The Unified Process

A robust framework structuring work into 5 distinct phases.



## Discipline

PSP and TSP focus on the human element, ensuring quality through measurement.