

# *Rekayasa Perangkat Lunak (Software Engineering)*

Siti Sa'uda, M.Kom.

*Software Engineering: A Practitioner's Approach, 6/e*

## Bab 2

# Proses : Pandangan Umum

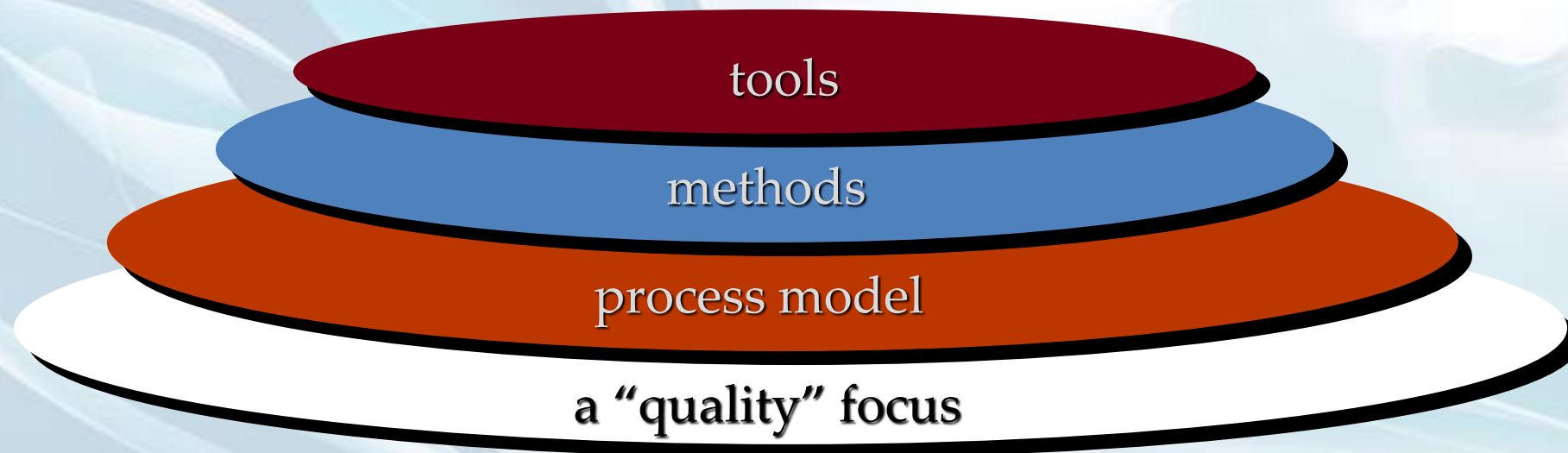
# Definisi

- Pembangunan dan Penggunaan prinsip-prinsip rekayasa dalam rangka mendapatkan perangkat lunak yang ekonomis yang handal dan bekerja efisien pada komputer yang nyata (Fritz Bauer)
- IEEE
  - Aplikasi pendekatan sistematis, disiplin, terkuantifikasi pada pengembangan, operasi, perawatan perangkat lunak, yaitu aplikasi rekayasa pada perangkat lunak
  - Studi pendekatan-pendekatan di atas



# Teknologi Berlapis

## Rekayasa Perangkat Lunak



# Pandangan Umum RPL

- Rekayasa : analisis, desain, konstruksi, verifikasi, dan manajemen entitas teknis (dan sosial)
  - Problem apa yang harus diselesaikan ?
  - Karakteristik entitas apa yang digunakan untuk menyelesaikan masalah ?
  - Bagaimana entitas (dan solusinya) direalisasikan ?
  - Bagaimana entitas di konstruksi ?
  - Pendekatan apa yang digunakan untuk menemukan kesalahan yang dibuat pada desain dan konstruksi entitas ?
  - Bagaimana entitas didukung dalam jangka panjang, dimana koreksi, adaptasi, dan peningkatan selalu diminta pengguna pada entitas

# Tiga Fase Umum RPL

- Fase definisi, fokus pada pertanyaan “apa”
- Fase pengembangan, fokus pada pertanyaan “bagaimana”
- Fase dukungan, fokus pada “perubahan” :
  - Koreksi
  - Adaptasi
  - Peningkatan
  - Pencegahan



# Bingkai Kerja Proses

## Bingkai Kerja Proses

Aktivitas Bingkai Kerja

Tugas-tugas

Produk-produk

milestones & deliverables

QA checkpoints

Aktivitas Payung

# Aktivitas Bingkai Kerja

- Komunikasi
- Perencanaan
- Pemodelan
  - Analisis Kebutuhan
  - Desain
- Konstruksi
  - Menyusun kode
  - Pengujian
- Produksi



# Aktivitas Payung

- Manajemen Proyek PL
- Review Teknik Formal
- Jaminan Mutu PL
- Manajemen Konfigurasi PL
- Persiapan dan Produksi Produk Pekerjaan
- Manajemen Penggunaan Kembali
- Pengukuran
- Manajemen Resiko

# Model Proses : Adaptabilitas

- Aktivitas-aktivitas bingkai kerja akan selalu diaplikasikan pada setiap project, tetapi ....
- Tugas-tugas (dan derajat kekakuan) pada setiap aktivitas akan bervariasi bergantung pada :
  - Tipe proyek
  - Karakteristik proyek
  - Penilaian umum; persetujuan tim proyek

# CMMI

- CMMI menentukan setiap area proses dalam hal “tujuan spesifik” dan “langkah-langkah spesifik” yang dibutuhkan untuk menggapai tujuan-tujuan tersebut.
- *Tujuan-Tujuan Spesifik* membangun karakteristik-karakteristik yang harus ada jika aktivitas yang dilakukan sebuah proses adalah efektif.
- *Langkah-Langkah Spesifik* membuat sebuah tujuan menjadi sekelompok aktivitas-aktivitas yang berkaitan dengan proses.



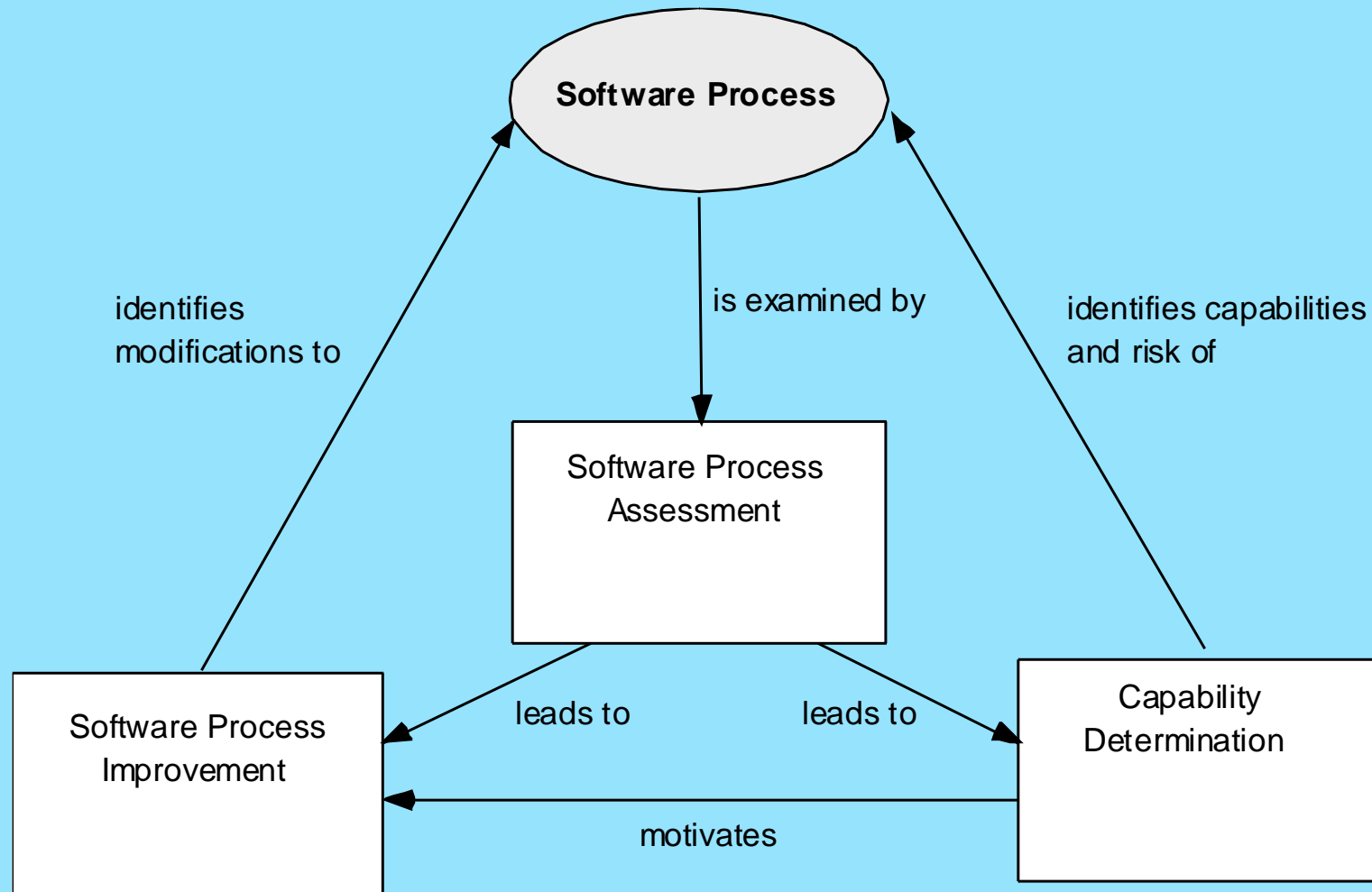
# Pola-Pola Proses

- Pola-pola proses menentukan sekelompok aktivitas, aksi, tugas-tugas pekerjaan, produk-produk pekerjaan dan/atau perilaku yang berkaitan
- Sebuah template digunakan untuk menentukan pattern/pola
- Contoh-contoh umum :
  - Komunikasi pelanggan (sebuah aktivitas proses)
  - Analisis (sebuah aksi)
  - Pengumpulan Kebutuhan (sebuah tugas proses)
  - Review sebuah produk kerja (sebuah tugas proses)
  - Model Desain (sebuah produk kerja)

# Asesmen Proses

- Sebuah proses harus dinilai untuk memastikan bahwa mereka memenuhi sekumpulan kriteria proses dasar yang penting bagi rekayasa PL yang sukses.
- Beberapa pilihan penilaian yang tersedia :
  - SCAMPI
  - CBA IPI
  - SPICE
  - ISO 9001:2000

# Penilaian dan Peningkatan





# Proses Personal PL (PSP)

- Rekomendasi 5 aktivitas bingkai kerja :
  - Perencanaan
  - Desain level tinggi
  - Review Desain level tinggi
  - Pengembangan
  - Postmortem
- Penekanan pada kebutuhan software engineer untuk mengidentifikasi kesalahan di awal waktu, dan memahami tipe-tipe kesalahan tersebut

# Proses Tim PL (TSP)

- Setiap proyek diluncurkan menggunakan sebuah script yang mendefinisikan tugas-tugas yang harus diselesaikan
- Tim diarahkan secara mandiri
- Pengukuran dianjurkan
- Pengukuran dianalisis dengan tujuan meningkatkan proses tim

# Tujuan Utama Proses PL : *Kualitas Tinggi*

Ingat:

Kualitas Tinggi = Proyek pendek

Mengapa?

Sedikit/Tidak ada Pekerjaan ulang!



*Software Engineering: A Practitioner's Approach, 6/e*

# Chapter 3

## Prescriptive Process Models

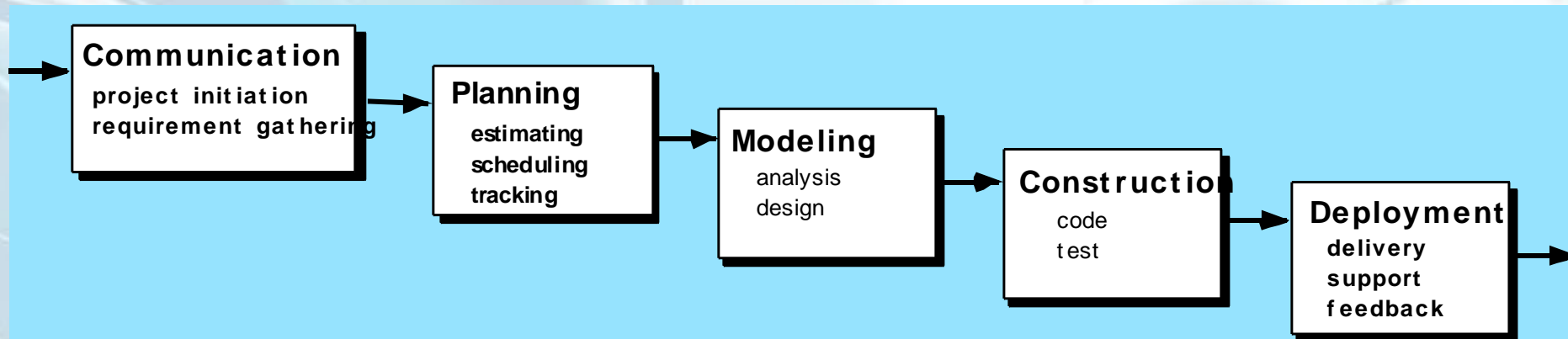
# Prescriptive Models

- Prescriptive process models advocate an orderly approach to software engineering

*That leads to a few questions ...*

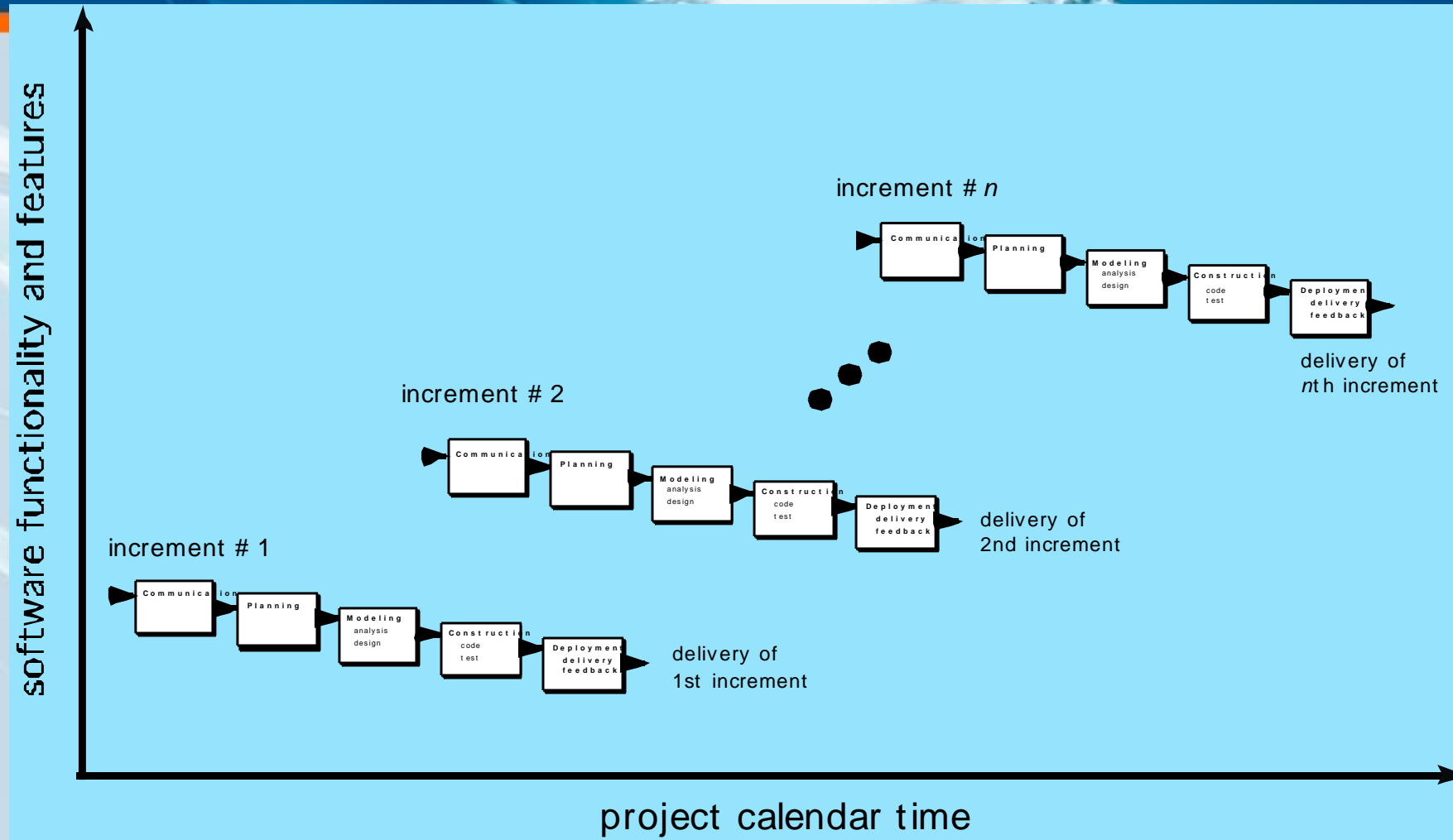
- If prescriptive process models strive for structure and order, **are they inappropriate for a software world that thrives on change?**
- Yet, if we reject traditional process models (and the order they imply) and replace them with something less structured, **do we make it impossible to achieve coordination and coherence in software work?**

# The Waterfall Model

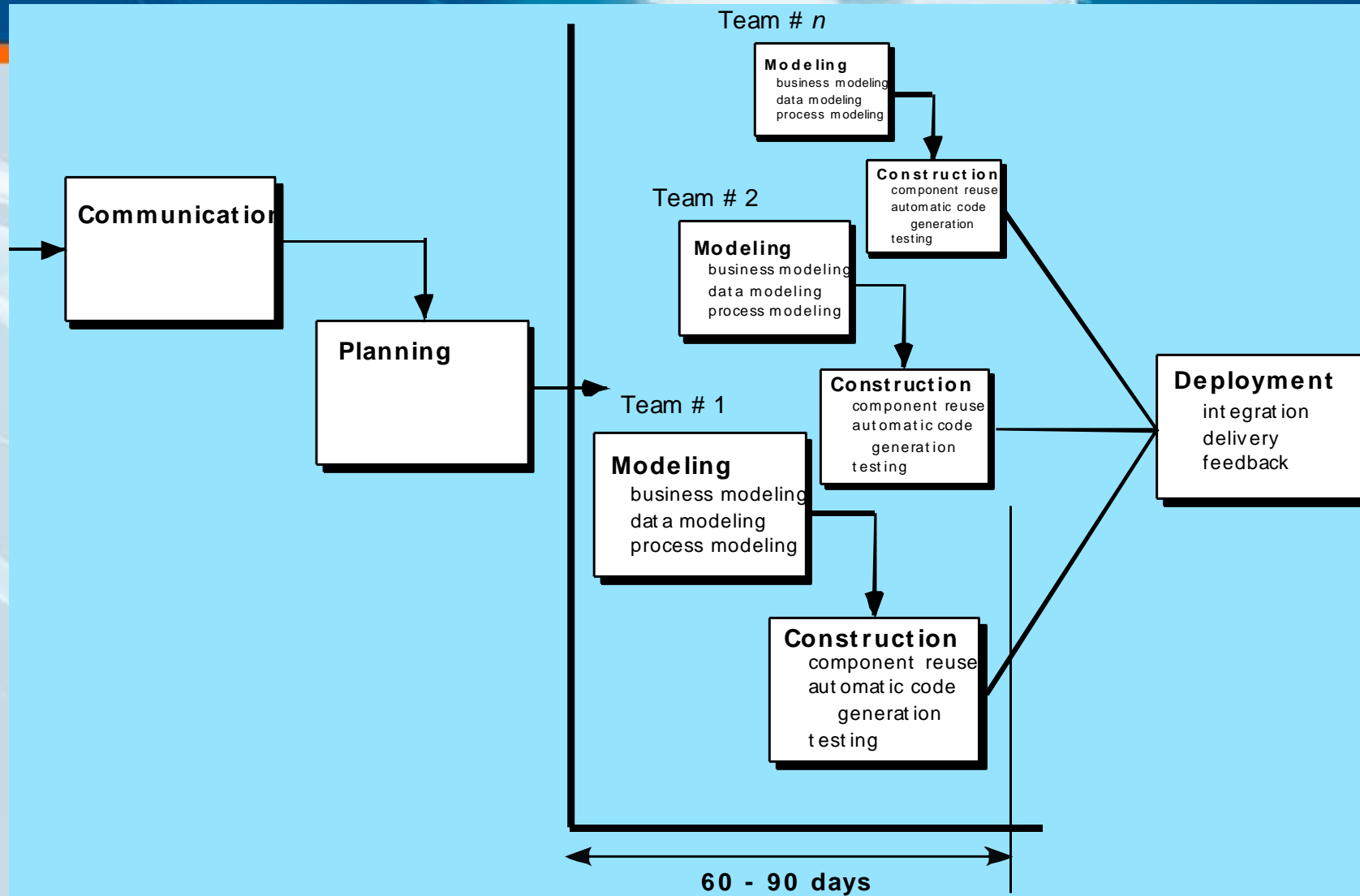




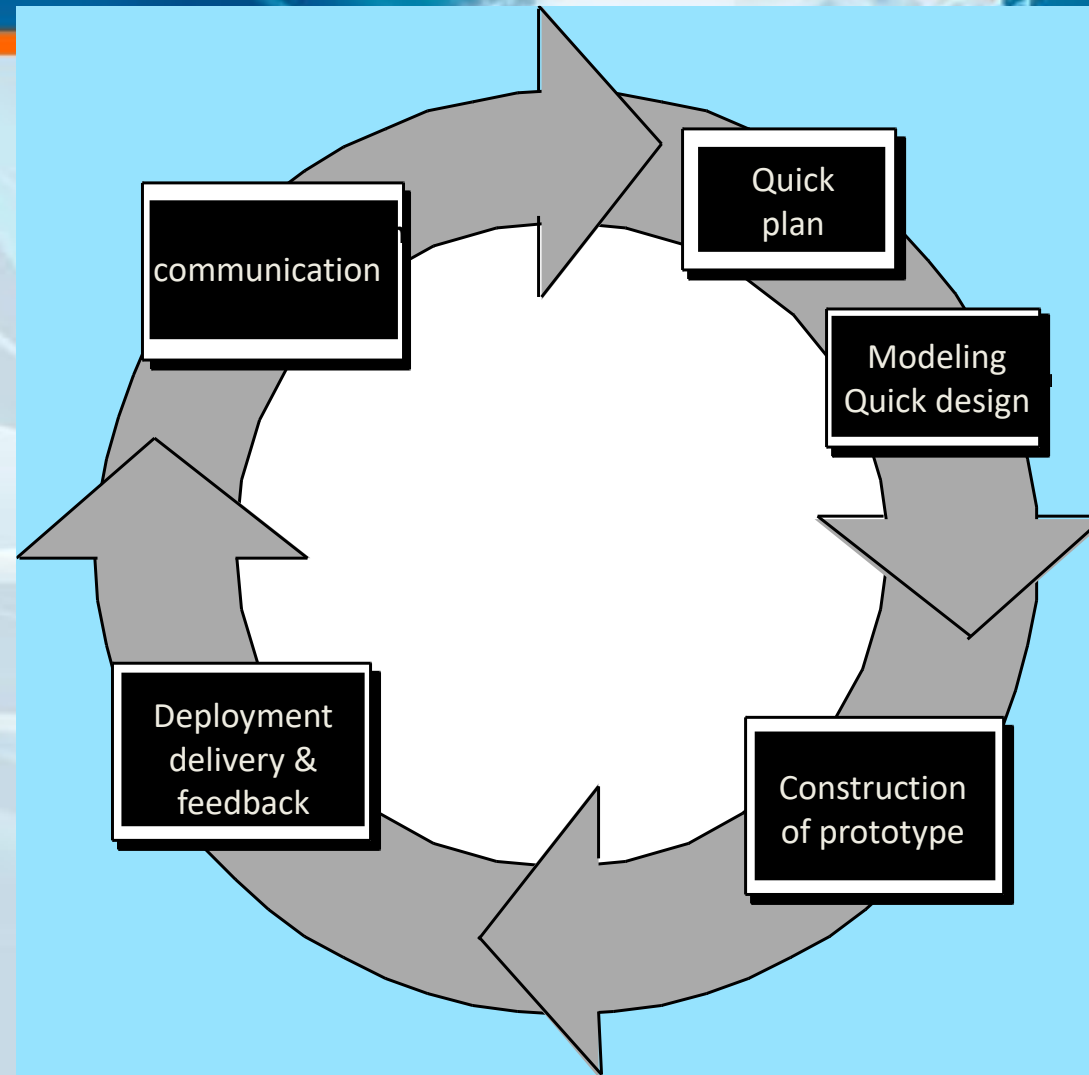
# The Incremental Model



# The RAD Model

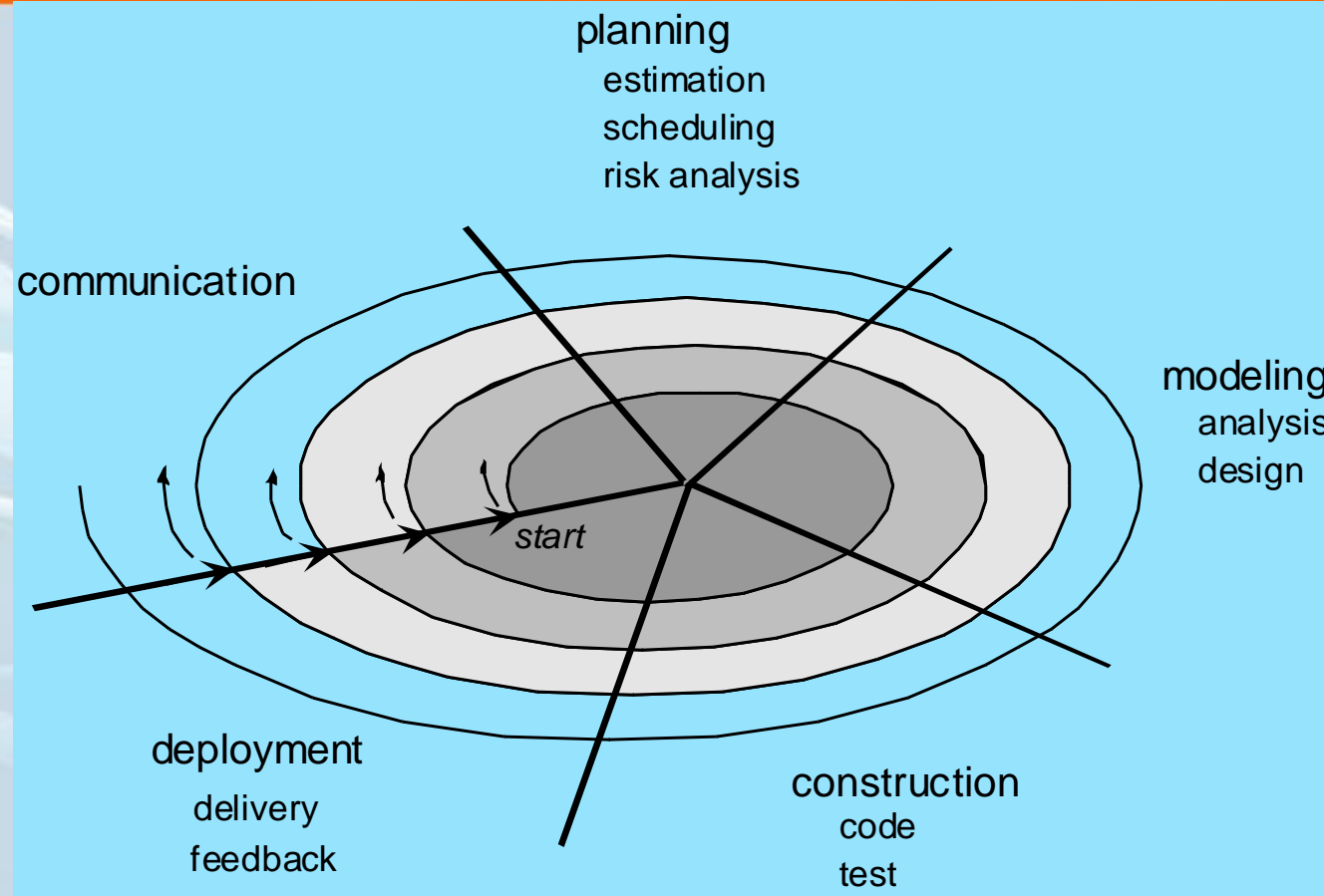


# Evolutionary Models: Prototyping

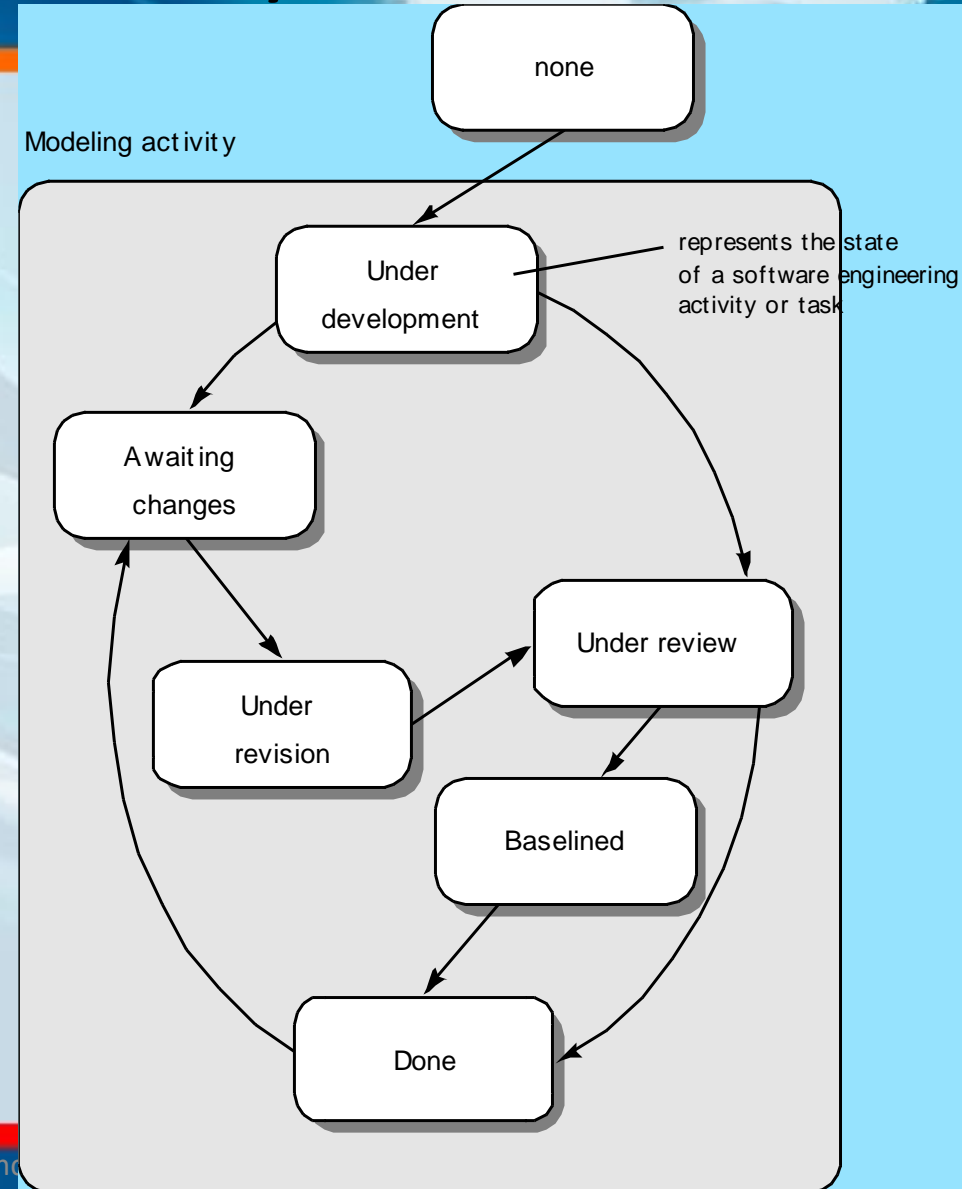




# Evolutionary Models: The Spiral



# Evolutionary Models: Concurrent

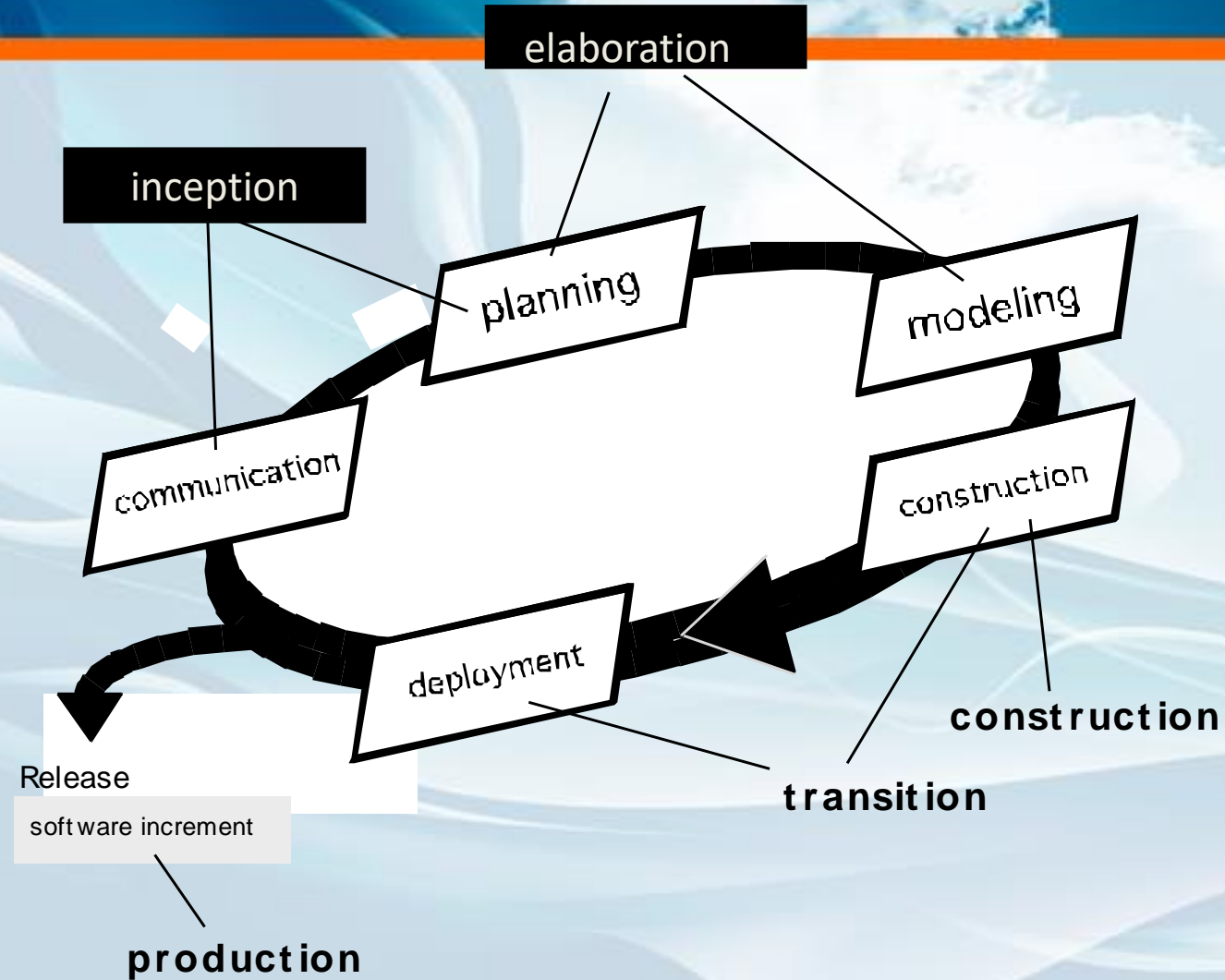


# Still Other Process Models

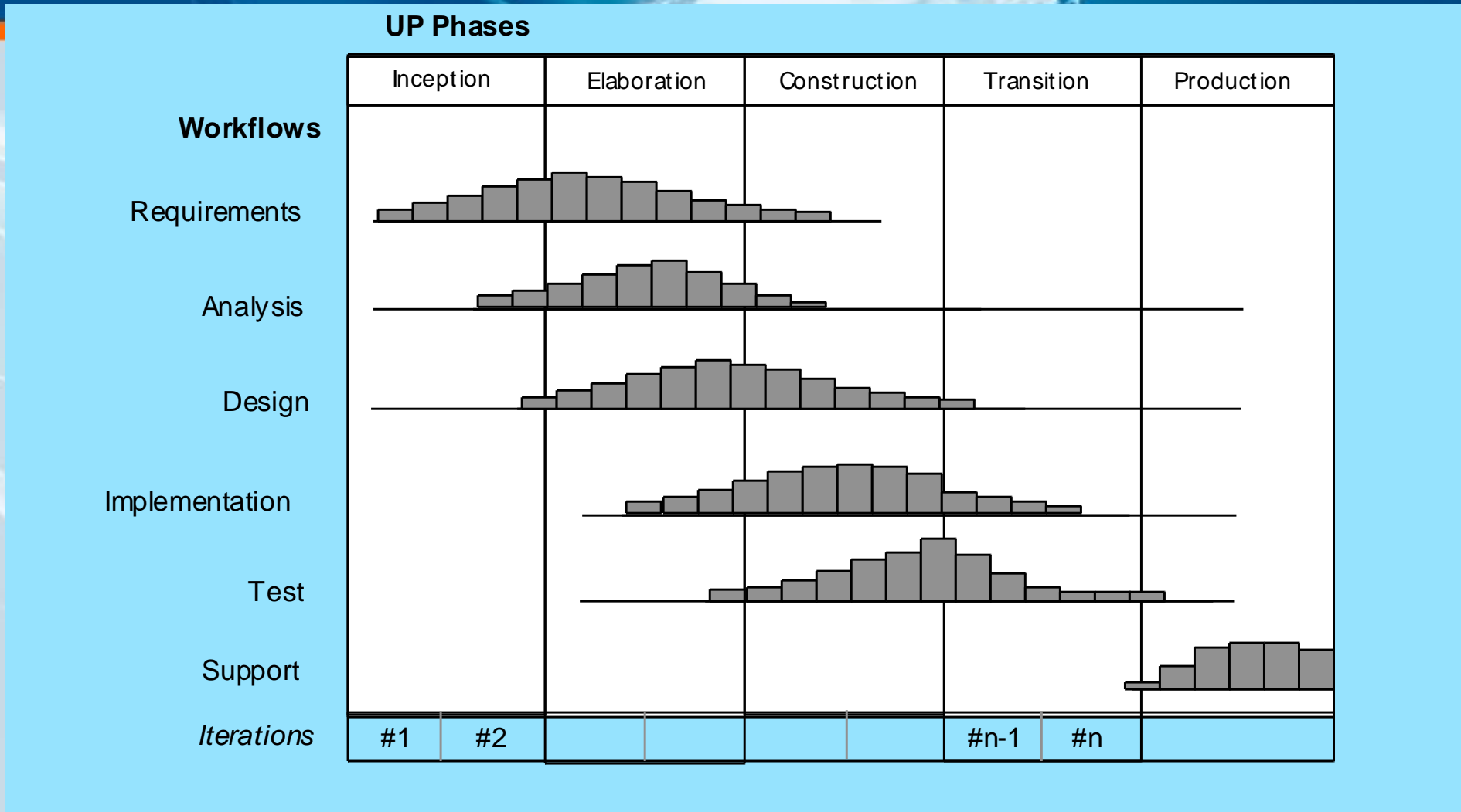
- **Component based development**—the process to apply when reuse is a development objective
- **Formal methods**—emphasizes the mathematical specification of requirements
- **AOSD**—provides a process and methodological approach for defining, specifying, designing, and constructing *aspects*
- **Unified Process**—a “use-case driven, architecture-centric, iterative and incremental” software process closely aligned with the Unified Modeling Language (UML)



# The Unified Process (UP)



# UP Phases



# UP Work Products

## Inception phase

Vision document  
Initial use-case model  
Initial project glossary  
Initial business case  
Initial risk assessment.  
Project plan,  
phases and iterations.  
Business model,  
if necessary.  
One or more prototypes

## Elaboration phase

Use-case model  
Supplementary requirements  
including non-functional  
Analysis model  
Software architecture  
Description.  
Executable architectural  
prototype.  
Preliminary design model  
Revised risk list  
Project plan including  
iteration plan  
adapted workflows  
milestones  
technical work products  
Preliminary user manual

## Construction phase

Design model  
Software components  
Integrated software  
increment  
Test plan and procedure  
Test cases  
Support documentation  
user manuals  
installation manuals  
description of current  
increment

## Transition phase

Delivered software increment  
Beta test reports  
General user feedback