PROGRAM STUDI
Teknik Informatika S1
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO

MATA KULIAH
Proyek Perangkat Lunak

Review RPL 1

Disusun Oleh

Tim PPL

Definisi Rekayasa Perangkat Lunak

- Rekayasa perangkat lunak adalah penerapan dari pendekatan yang sistematis, disiplin, dapat diukur untuk pengembangan, operasi, dan pemeliharaan perangkat lunak.

[IEEE std 610.12-1990]

- Rekayasa Perangkat Lunak adalah kombinasi teknik, metodologi dan alat bantu yang digunakan untuk membantu pembuatan
  - Suatu sistem perangkat lunak yang berkualitas tinggi (bebas eror)
  - Dalam budget yang ditentukan
  - Sebelum batas waktu yang ditentukan

sementara perubahan terjadi, dapat memenuhi kebutuhan pengguna.

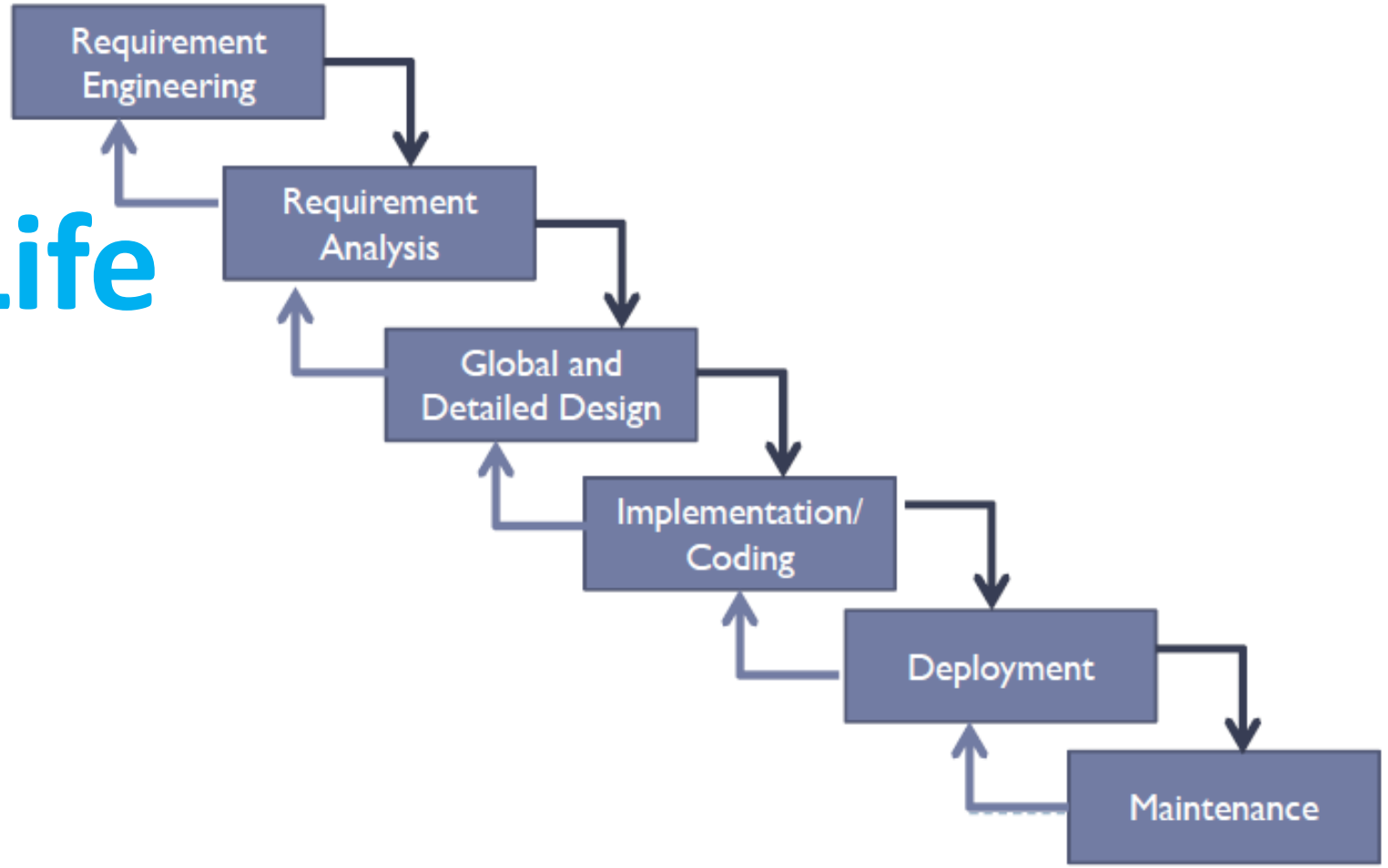
[Bernd Bruegge, Allen Dutoit: "Object-Oriented Software Engineering: Using UML, Patterns, and Java", Prentice Hall, 2003]

Domain Perangkat Lunak

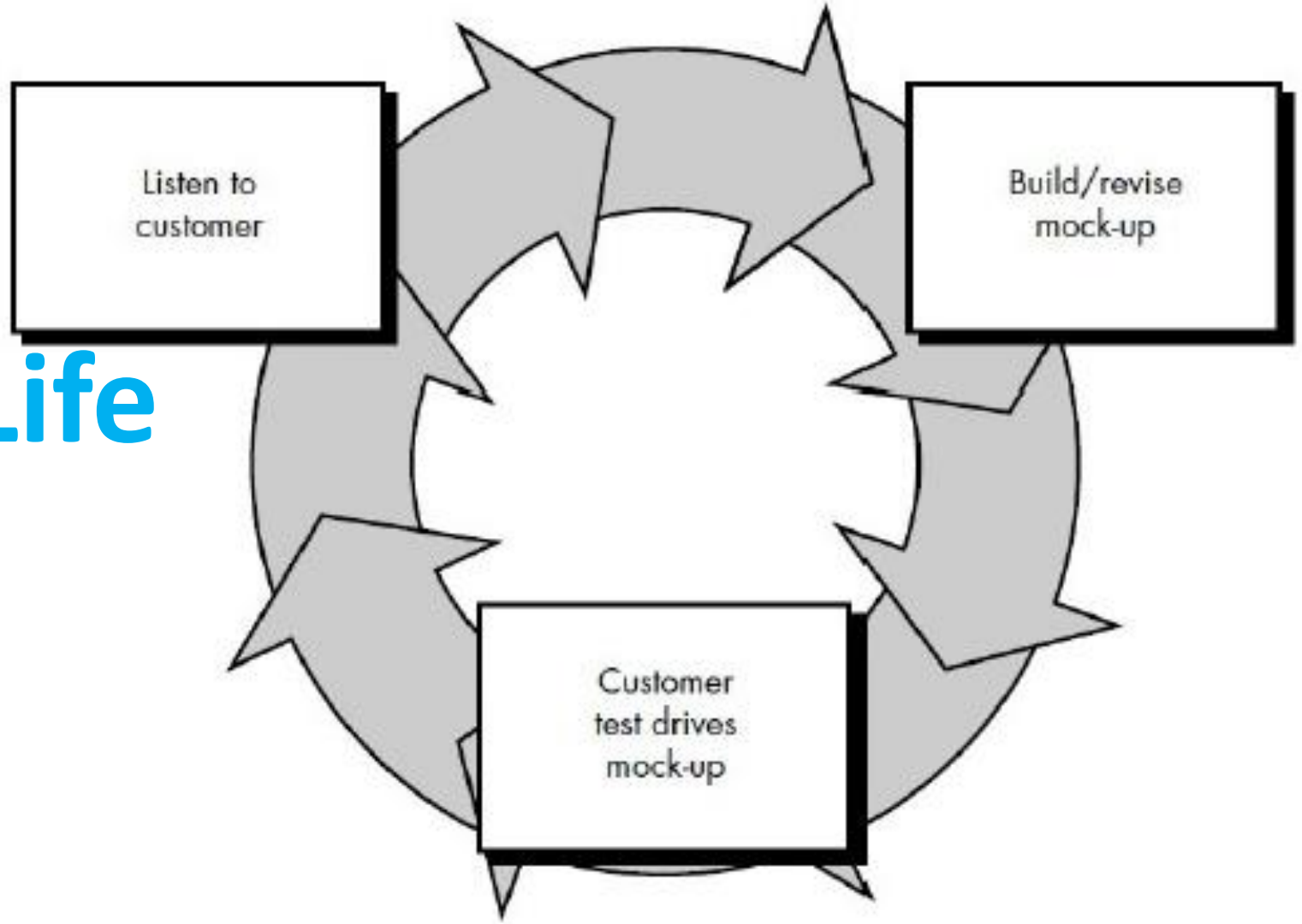
- System software
- Application software
- Engineering or Scientific Software
- Embedded software
- Product-line software (includes entertainment software)
- Web-Applications
- Artificial intelligence software

Software Development Life Cycle [1]

Waterfall Model



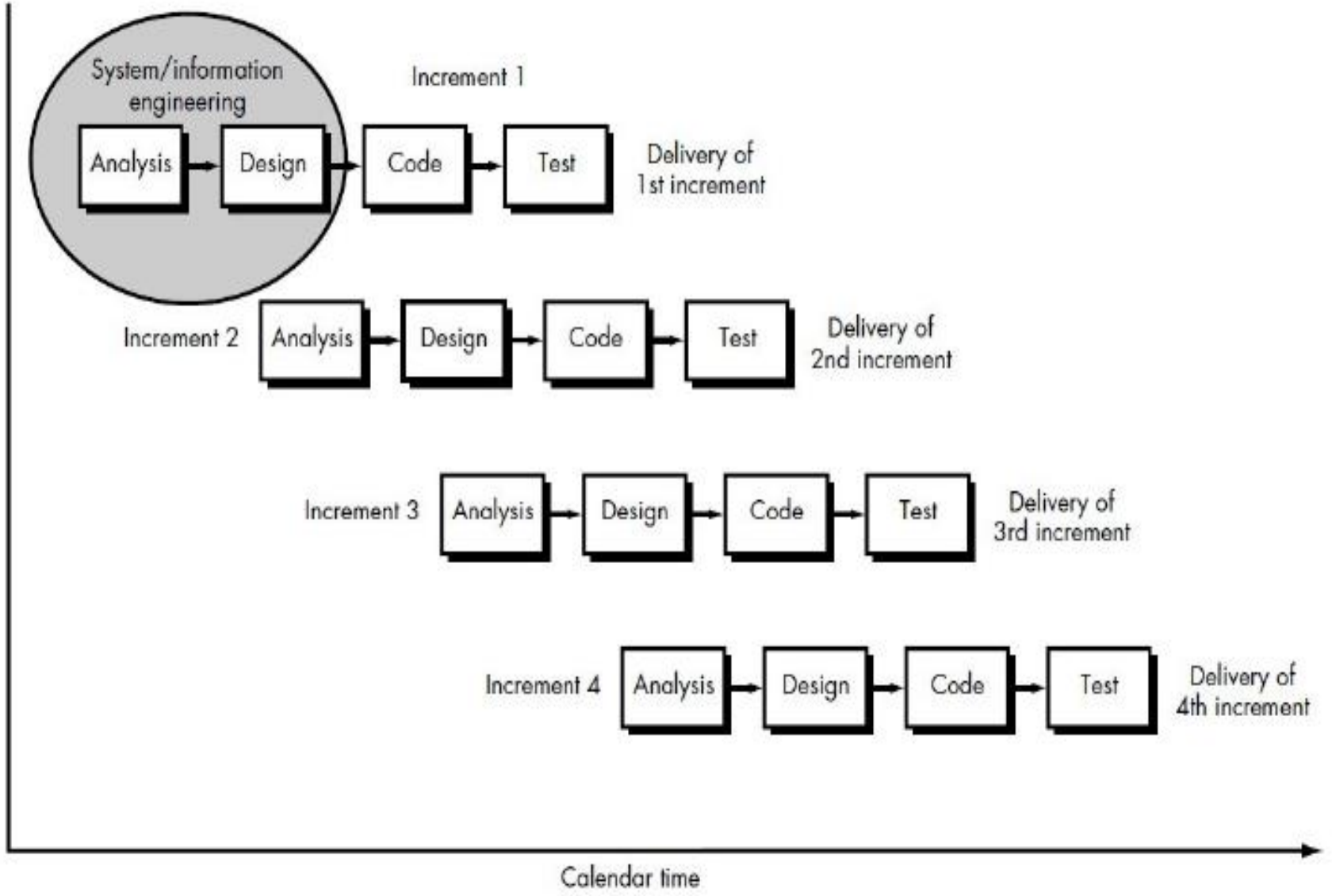
Software Development Life Cycle [2]



Prototyping Model

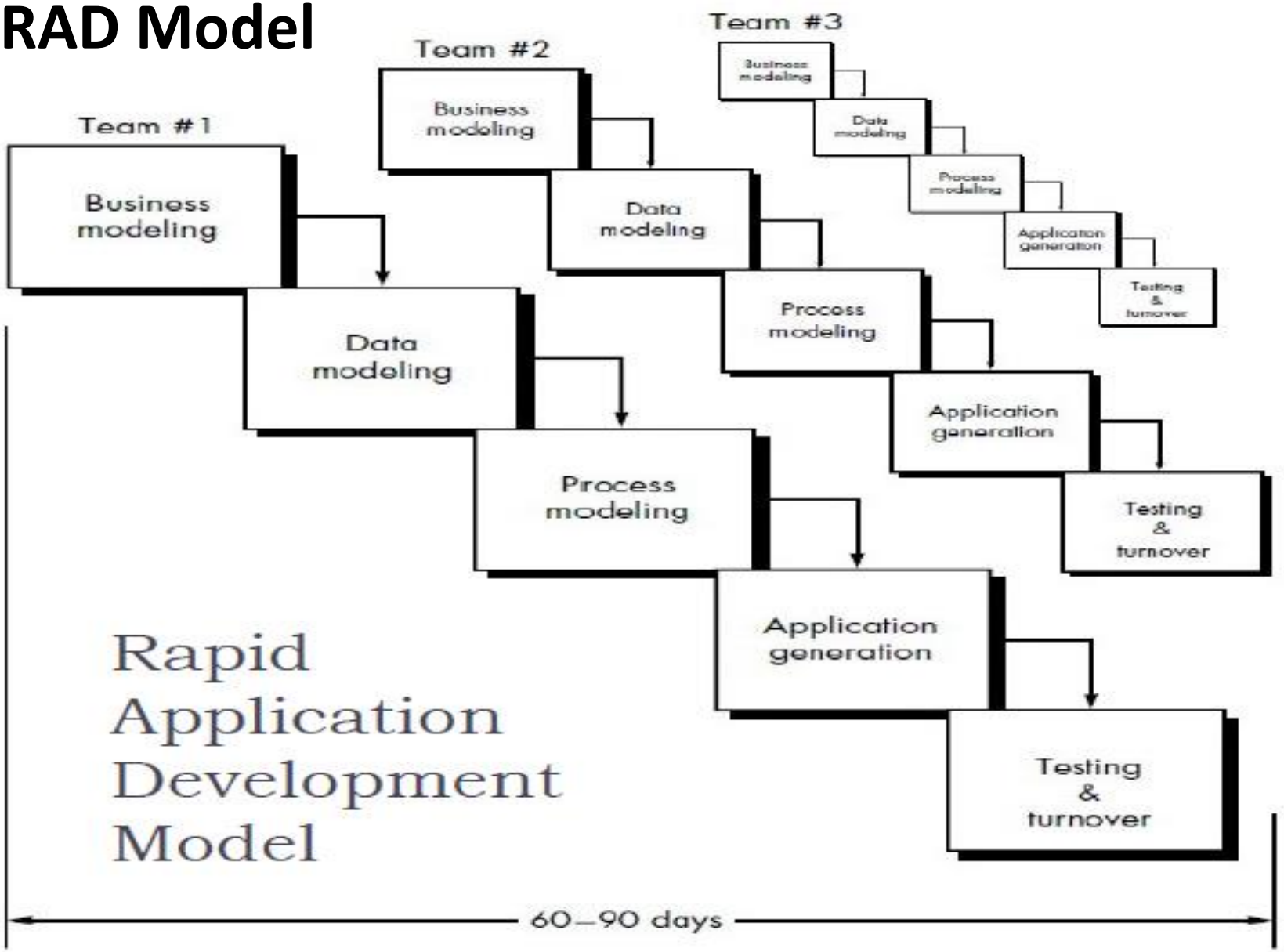
Software Development Life Cycle [3]

Incremental Model

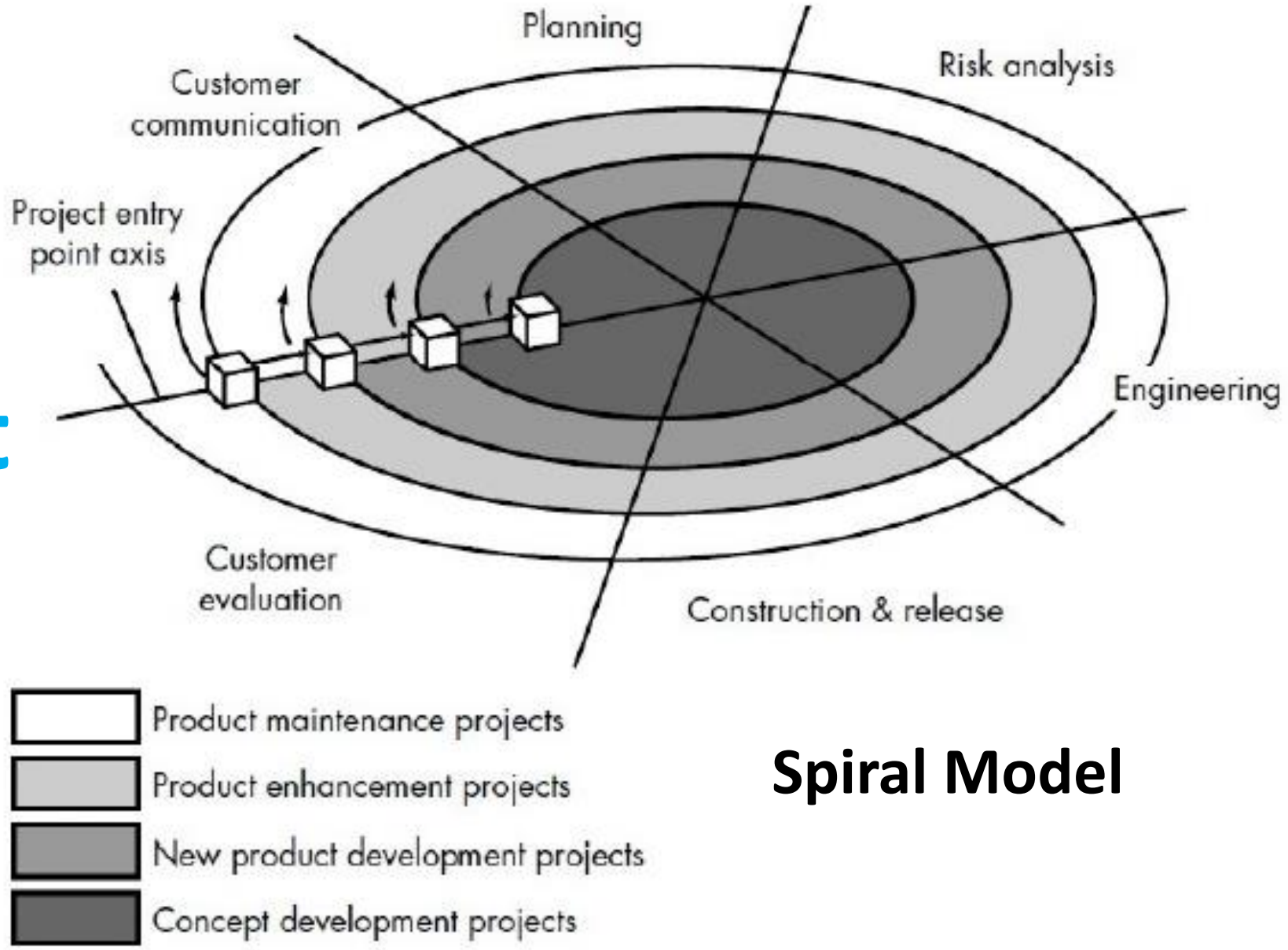


Software Development Life Cycle [4]

RAD Model



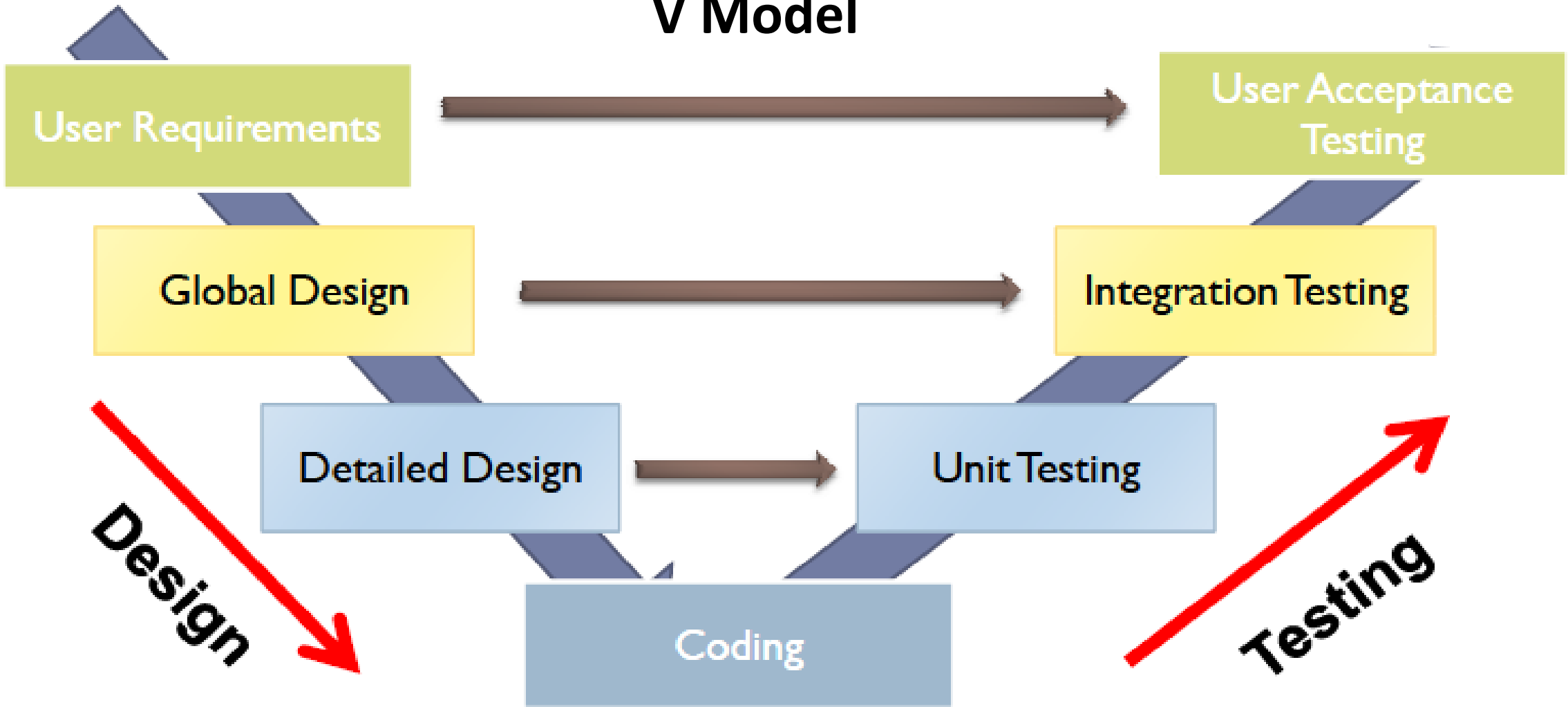
Software Development Life Cycle [5]



Spiral Model

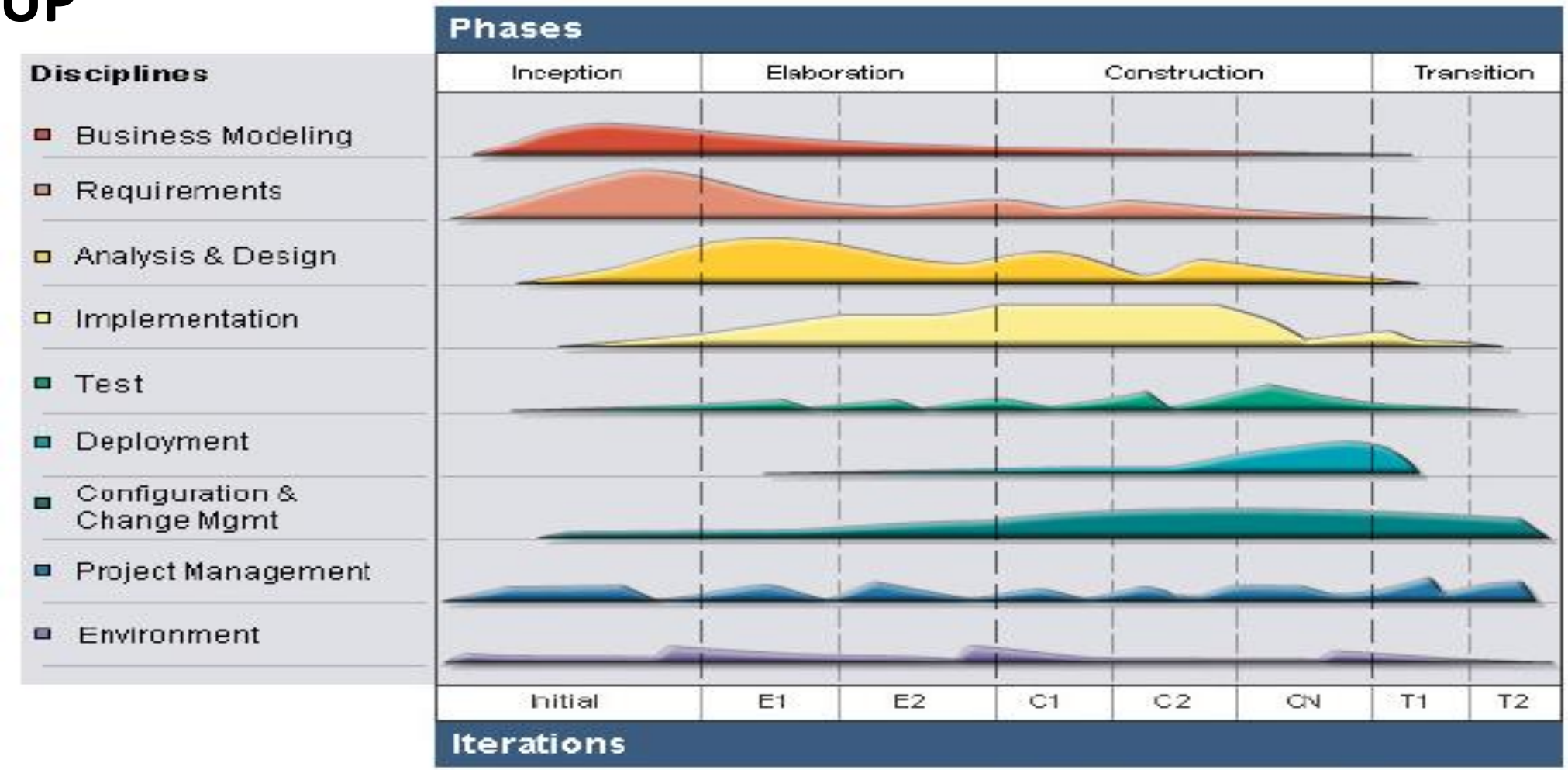
Software Development Life Cycle [6]

V Model



Software Development Life Cycle [6]

RUP



IBM Software Group-Rational Unified Process

Software Development Life Cycle [8]

Agile

Pendekatan yang iteratif dan bertambah (evolusioner) yang dilakukan dengan cara kolaborasi dengan jumlah pertemuan yang tepat untuk menghasilkan PL berkualitas tinggi dalam waktu dan biaya yang efektif sehingga dapat memenuhi perubahan kebutuhan dari tiap stakeholder.

Prinsip utama:

- "Fits just right" process
- Continuous testing and validation
- Consistent team collaboration
- Rapid response to change
- Ongoing customer involvement
- Frequent delivery of working software

Referensi

- Roger S. Pressman: “Software Engineering: A Practitioner’s Approaches”, McGraw Hill, ? Edition.
- Ian Sommerville: “Software Engineering”, Addision-Wesley, 9th Edition.
- Bernd Bruegge, Allen Dutoit: “Object-Oriented Software Engineering: Using UML, Patterns, and Java”, Prentice Hall, 2003.
- Stephen R Schach: “Object-Oriented and Classical Software Engineering”, McGraw Hill , 2002
- Thomas Erl. “SOA Principles of Service Design“. Prentice Hall, 2003

THANKS

ANY QUESTIONS?