



RPL – Pertemuan

SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

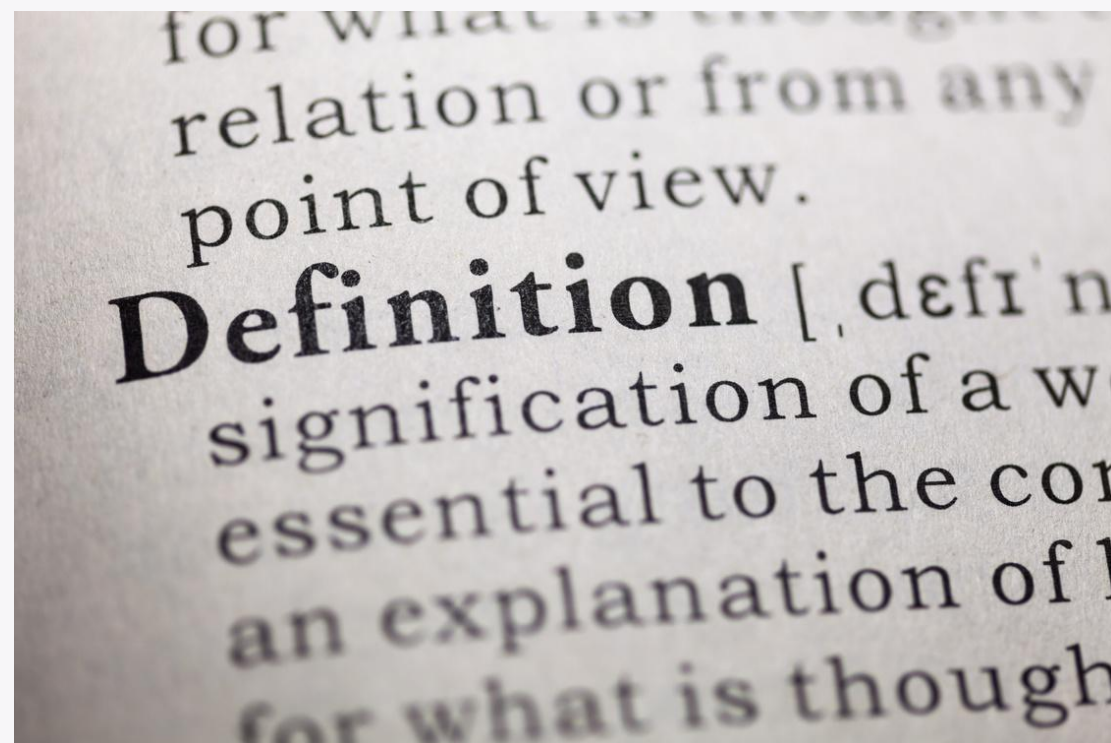
Pertemuan 2 • Sub-CPMK: Menjelaskan SDLC & mengaitkan model ke kasus nyata



Tujuan Pembelajaran

- Memahami konsep proses perangkat lunak dan SDLC.
- Membedakan pendekatan plan-driven vs. agile.
- Menjelaskan model proses: Waterfall, Iteratif/Incremental, V-Model, Spiral.
- Memetakan artefak/milestone SDLC pada studi kasus.

2 Definisi Proses Perangkat Lunak



Proses Perangkat Lunak: kumpulan aktivitas terstruktur untuk mengembangkan perangkat lunak.

Fase Utama (Sommerville):

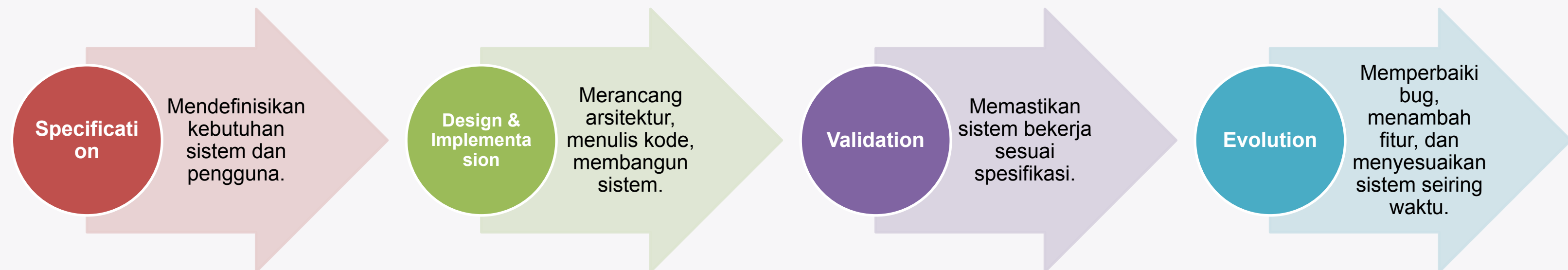
- Spesifikasi (Requirement)
- Desain & Implementasi
- Validasi (Testing)
- Evolusi / Pemeliharaan

Catatan: Proses dapat berbeda tergantung konteks; tidak ada satu model yang cocok untuk semua proyek.

Proses Aktivitas Generik (Sommerville)



Sebagian besar proses pengembangan perangkat lunak memiliki **empat aktivitas generik utama**:



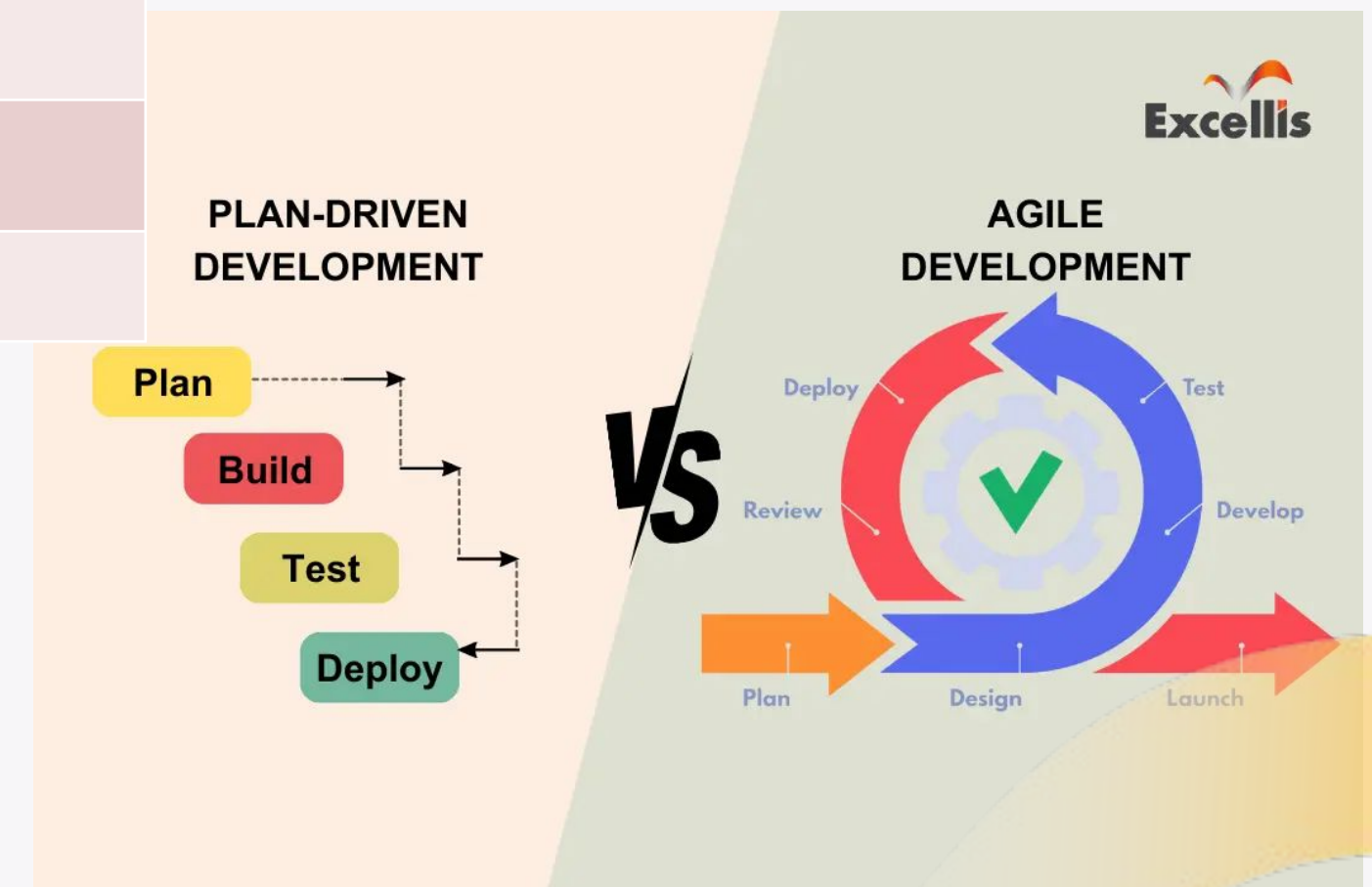
Plan-Driven vs Agile



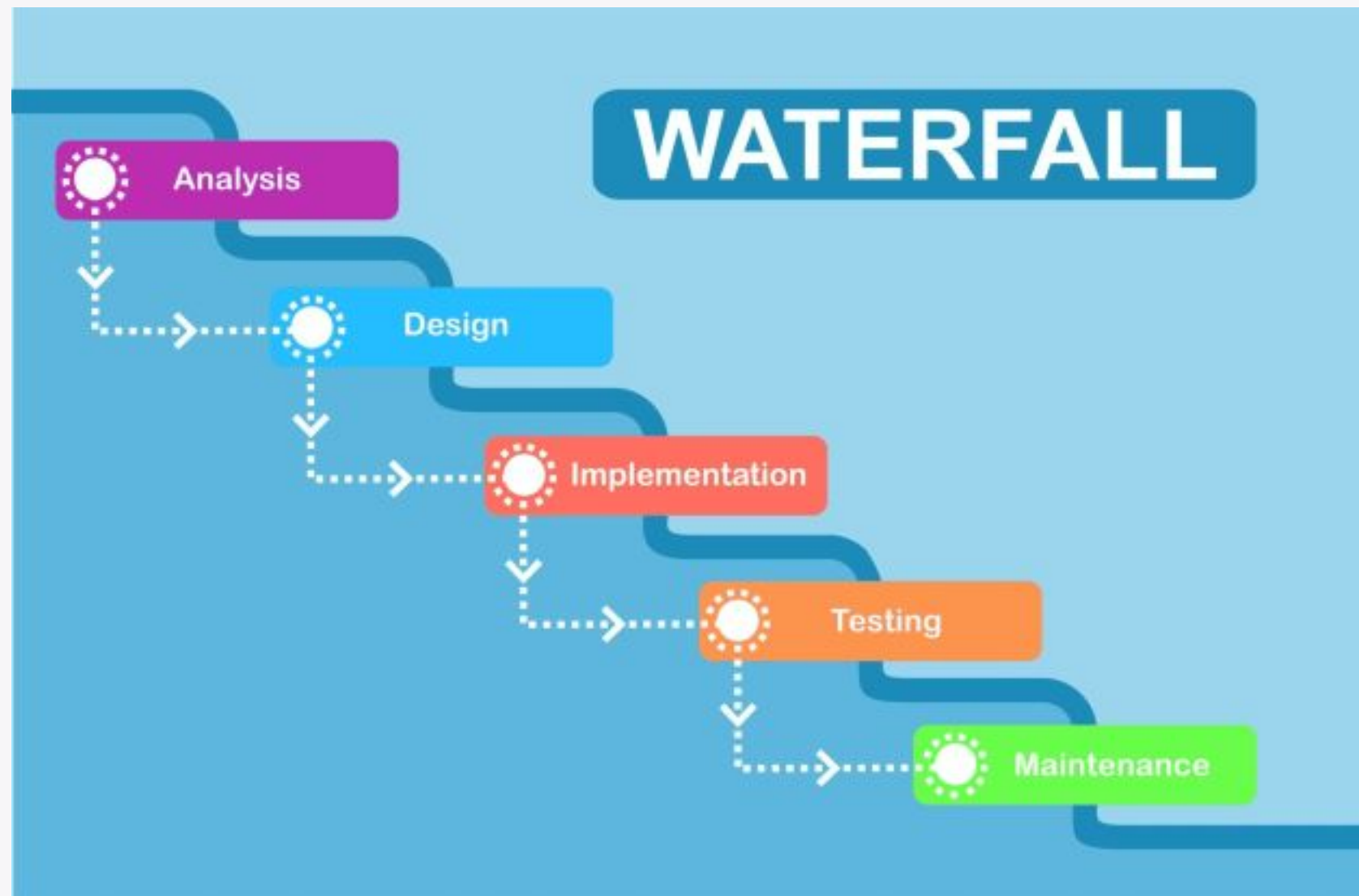
Aspek	Plan-Driven	Agile
Pendekatan	Proses linier & terstruktur	Iteratif dan fleksibel
Dokumentasi	Lengkap & formal	Minimal & cukup
Perubahan Kebutuhan	Sulit diubah, harus direncanakan	Mudah diadaptasi, responsif
Pengiriman	Sekali setelah semua fase selesai	Bertahap, tiap iterasi/fase
Keterlibatan Pengguna	Terbatas di awal & akhir	Terus-menerus sepanjang proyek
Contoh Model	Waterfall, V-Model	Scrum, XP, RAD

Catatan:

- **Plan-Driven** cocok untuk proyek stabil dan kritis.
- **Agile** cocok untuk proyek yang dinamis dan cepat berubah.



2 Model Waterfall



Ciri-ciri

1. Urutan **linier**: setiap fase selesai sebelum fase berikutnya dimulai.
2. **Dokumentasi kuat**: semua spesifikasi dan desain terdokumentasi lengkap.
3. **Perubahan sulit & mahal**: revisi setelah fase tertentu membutuhkan biaya tinggi.
4. Cocok untuk proyek dengan **kebutuhan stabil** dan **regulasi ketat**.

Waterfall:



Kelebihan



- ☐ Sederhana & mudah dipahami.
- ☐ Dokumentasi lengkap & terstruktur.
- ☐ Cocok untuk kebutuhan yang stabil.
- ☐ Memudahkan manajemen proyek besar & formal.

Kekurangan



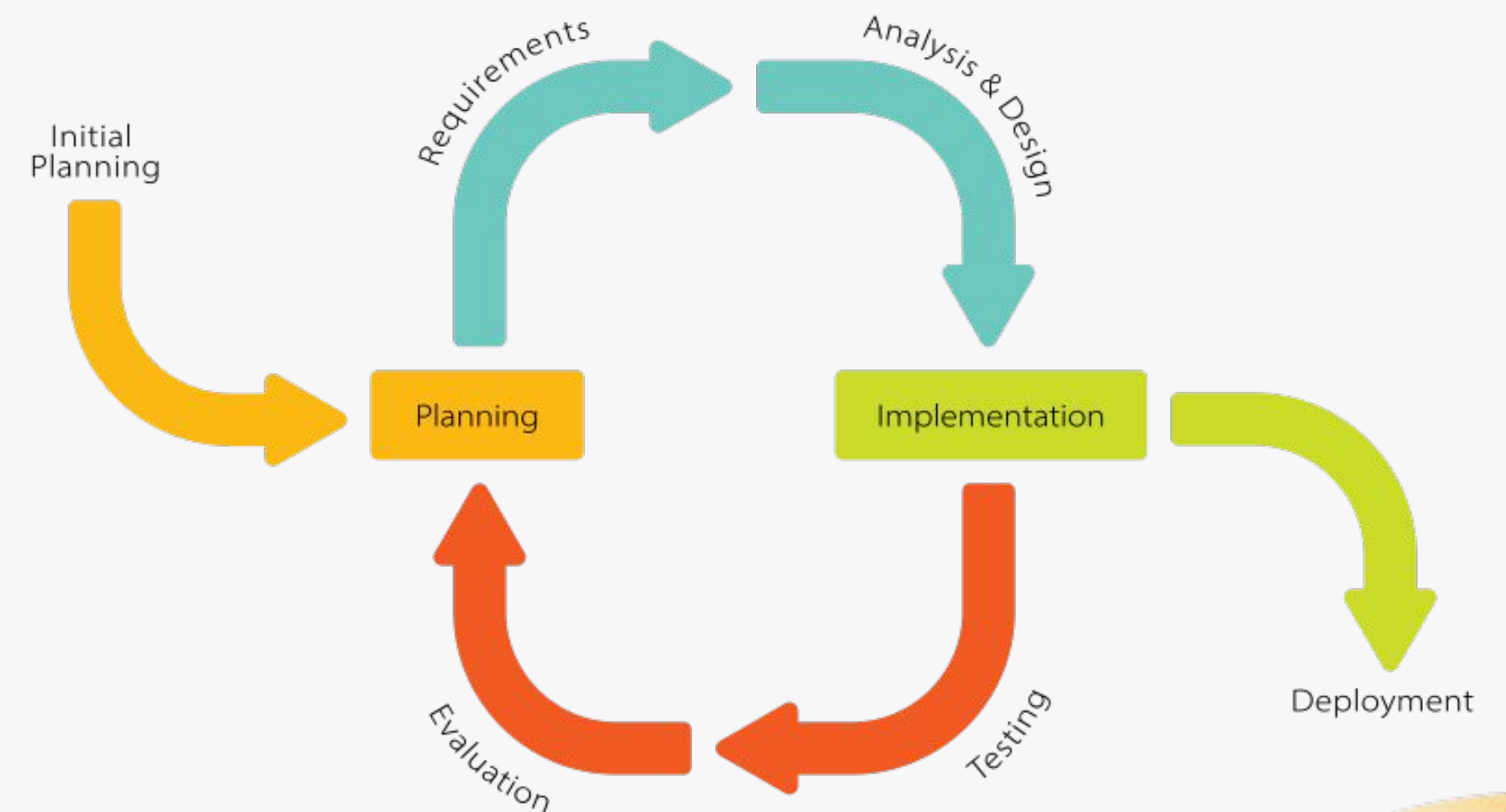
- ☐ Sulit beradaptasi dengan perubahan kebutuhan.
- ☐ Perubahan setelah fase berjalan mahal.
- ☐ Umpan balik pengguna terlambat (setelah selesai).
- ☐ Risiko tinggi bila terjadi kesalahan di awal.

2 Model Incremental/Iterative



Fitur Utama

- **Rilis bertahap:** sistem dikembangkan dalam potongan (increment).
- **Umpan balik pengguna tiap iterasi:** evaluasi di setiap rilis.
- **Lebih adaptif terhadap perubahan:** kebutuhan baru dapat dimasukkan di iterasi berikutnya.
- **Pengiriman lebih cepat:** fitur dasar bisa digunakan sambil menunggu fitur lanjutan.



Iteratif/Incremental: Kelebihan & Kekurangan



Kelebihan

Sistem bisa digunakan lebih cepat karena rilis bertahap.

Umpan balik pengguna diterima di setiap iterasi → mudah adaptasi perubahan.

Risiko kesalahan berkurang karena perbaikan dilakukan tiap iterasi.

Pengembangan lebih fleksibel dan terkontrol.



Kekurangan

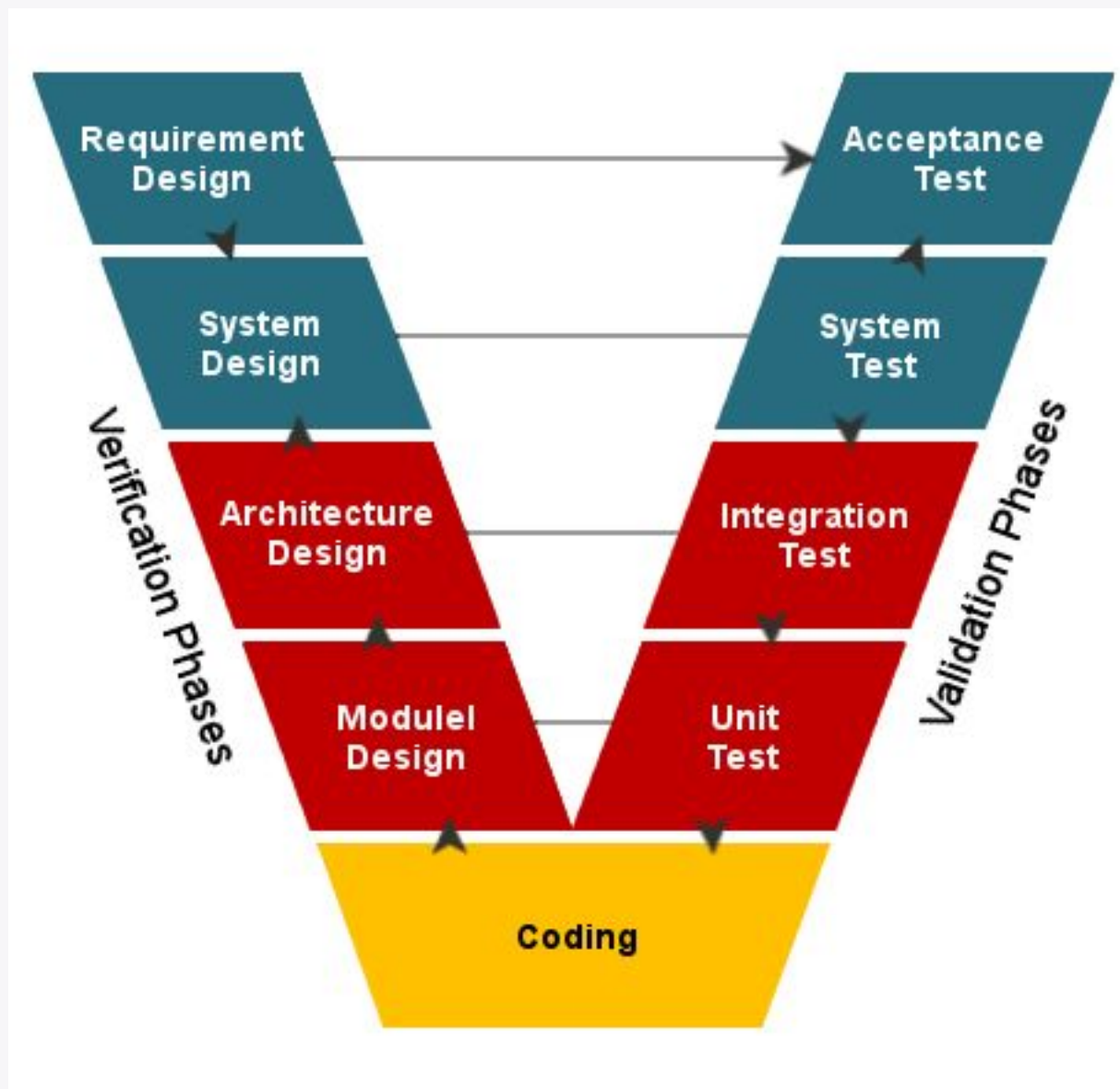
Perencanaan dan manajemen lebih kompleks.

Bisa membutuhkan dokumentasi tambahan di tiap iterasi.

Risiko integrasi modul jika tidak terkoordinasi dengan baik.

Iterasi terlalu banyak → biaya dan waktu bisa meningkat.

2 V-Model (Verification & Validation Model)



Definisi

- ❖ V-Model adalah **varian** dari model **Waterfall** yang menekankan **verifikasi** dan **validasi**.
- ❖ Bentuknya seperti huruf “V”: **fase spesifikasi & desain di kiri, fase pengujian di kanan, dan implementasi di dasar V.**
- ❖ Cocok untuk proyek dengan **kebutuhan stabil dan kritis**.

Ciri-Ciri

- ❖ Aktivitas **pengujian dimulai sejak fase awal spesifikasi**.
- ❖ Setiap fase desain memiliki **pasangan pengujian** (corresponding test phase).
- ❖ Memastikan perangkat lunak sesuai **kebutuhan pengguna**.

2 V-Model: Kelebihan & Kekurangan



✓ Kelebihan

- **Verifikasi & Validasi di setiap fase:** kesalahan dapat dideteksi lebih awal.
- **Dokumentasi lengkap:** memudahkan pengelolaan proyek.
- **Cocok untuk proyek kritis dan kebutuhan stabil:** seperti sistem industri, militer, atau kesehatan.
- **Struktur jelas:** memudahkan perencanaan dan pengendalian proyek.

✗ Kekurangan

- **Kurang fleksibel terhadap perubahan:** sulit beradaptasi jika kebutuhan berubah.
- **Tidak cocok untuk proyek dinamis:** perubahan di tengah proyek mahal dan sulit.
- **Ketergantungan kuat antar fase:** kesalahan di fase awal berdampak besar di fase berikutnya.



²Spiral Model (Risk-Driven)

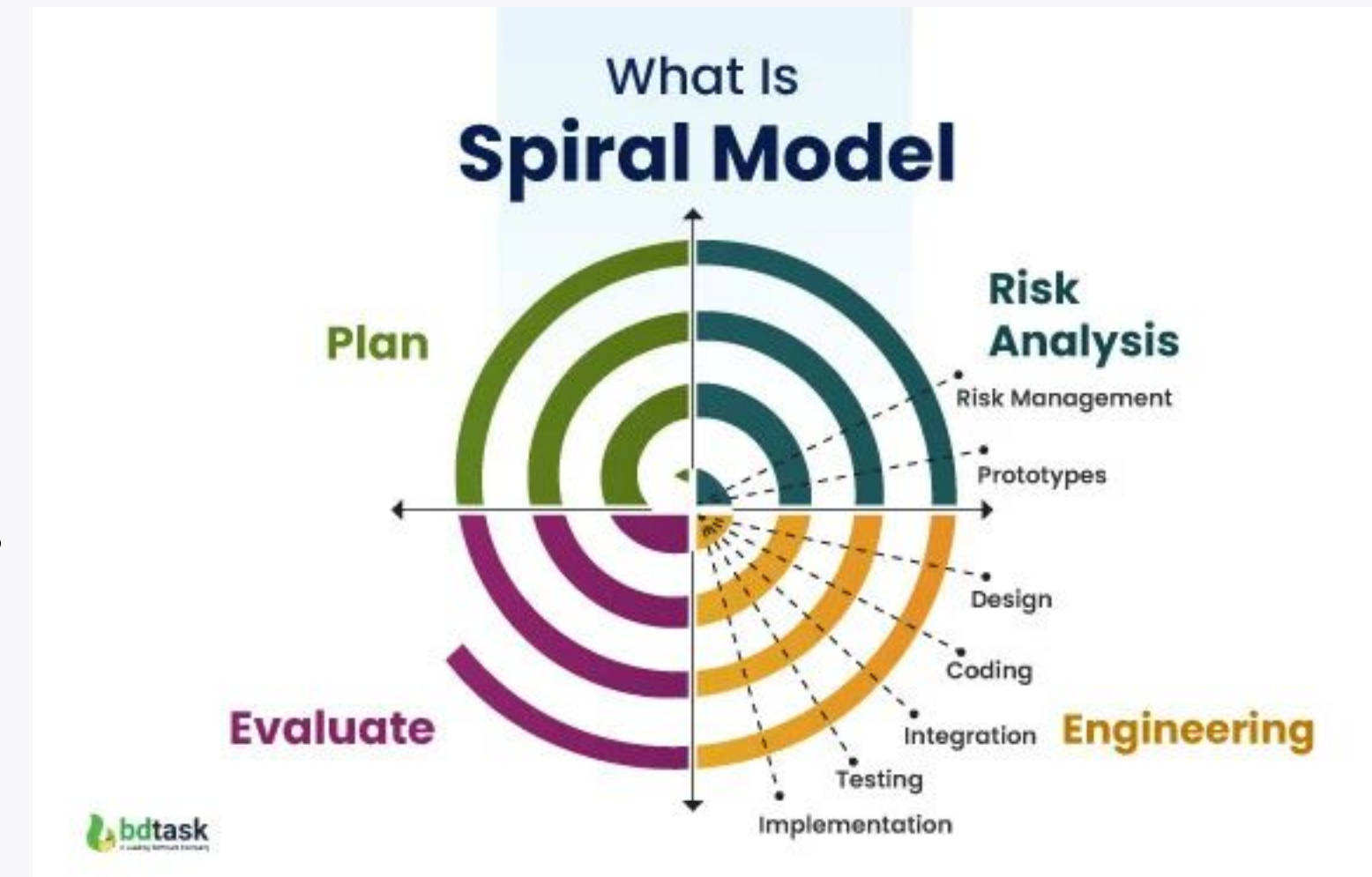


Definisi

- Model pengembangan perangkat lunak **berbasis risiko**.
- Dikembangkan oleh Barry Boehm; setiap iterasi disebut **spiral** / putaran.

Ciri-Ciri

- Setiap putaran menambahkan **level detail** setelah risiko dianalisis dan dimitigasi.
- Cocok untuk **proyek kompleks dan berisiko tinggi**.
- Menggabungkan unsur **Waterfall & Iteratif**, dengan fokus pada **identifikasi risiko**.



2 Perbandingan Model Proses Perangkat Lunak

Aspek	Waterfall	V-Model	Incremental / Iterative	Spiral (Risk-Driven)
Pendekatan	Linier, sekuensial	Linier + Verifikasi & Validasi	Iteratif & bertahap	Iteratif & berbasis risiko
Fleksibilitas	Rendah	Rendah	Tinggi	Tinggi
Dokumentasi	Lengkap	Lengkap	Minimal–cukup	Lengkap, fokus risiko
Pengiriman Sistem	Sekali di akhir	Sekali di akhir	Bertahap / tiap iterasi	Bertahap / tiap putaran
Cocok untuk proyek	Stabil & formal	Stabil & kritis	Dinamis & cepat berubah	Kompleks & berisiko tinggi
Manajemen Risiko	Rendah	Sedang	Sedang	Tinggi



Artefak & Milestone SDLC

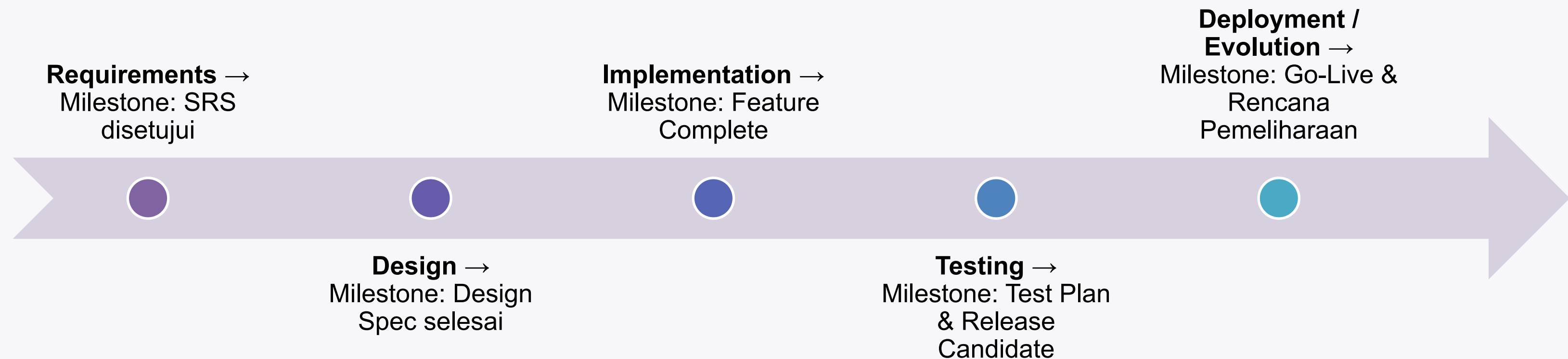
Fase	Artefak Kunci	Milestone
Requirements	SRS, Use Case	Baseline kebutuhan
Design	Arsitektur, Model UML	Design review
Implementation	Kode, Build	Feature complete
Testing	Test Plan, Test Case, Report	Release candidate
Deployment / Evolution	Manual, Change Log	Go-live & rencana pemeliharaan

Keterangan:

- ❖ **Artefak Kunci** → hasil utama dari tiap fase.
- ❖ **Milestone** → pencapaian penting yang menandai fase selesai.



Contoh Timeline SDLC (Plan-Driven)



Ciri-Ciri

- ❖ **Sekuensial:** tiap fase dilakukan berurutan.
- ❖ **Overlap kecil:** transisi antar fase bisa bersifat sedikit paralel.
- ❖ **Milestone jelas**

Pemilihan Model Proses Berdasarkan Kriteria



Kriteria	Kondisi	Model Proses yang Cocok
Stabilitas kebutuhan & domain regulasi	Kebutuhan stabil, domain regulasi ketat (mis. kesehatan, penerbangan)	Waterfall / V-Model (plan-driven)
	Kebutuhan sering berubah, regulasi longgar	Agile (Scrum, XP, RAD)
Skala tim & kompleksitas teknis	Tim besar, sistem kompleks, terdistribusi	Plan-driven (Waterfall, Spiral, V-Model)
	Tim kecil–menengah, sistem modular	Agile (Scrum, Kanban, RAD)
Risiko & ketidakpastian teknologi	Tinggi, teknologi baru, banyak ketidakpastian	Spiral Model (risk-driven)
	Rendah, teknologi mapan	Waterfall atau Agile
Kebutuhan audit, kepatuhan, & dokumentasi	Perlu dokumentasi lengkap, audit, compliance	Plan-driven (V-Model, Waterfall)
	Fokus kecepatan & adaptasi, dokumentasi minimal	Agile (Scrum, Kanban, RAD)



Diskusi & Tugas

Ringkasan :

SDLC mencakup aktivitas generik; model yang dipilih harus sesuai konteks.

Tugas :

Petakan proses untuk studi kasus pendaftaran kampus—pilih model, jelaskan alasan, buat daftar artefak & milestone.

Bacaan :

Sommerville (Ch.2).