

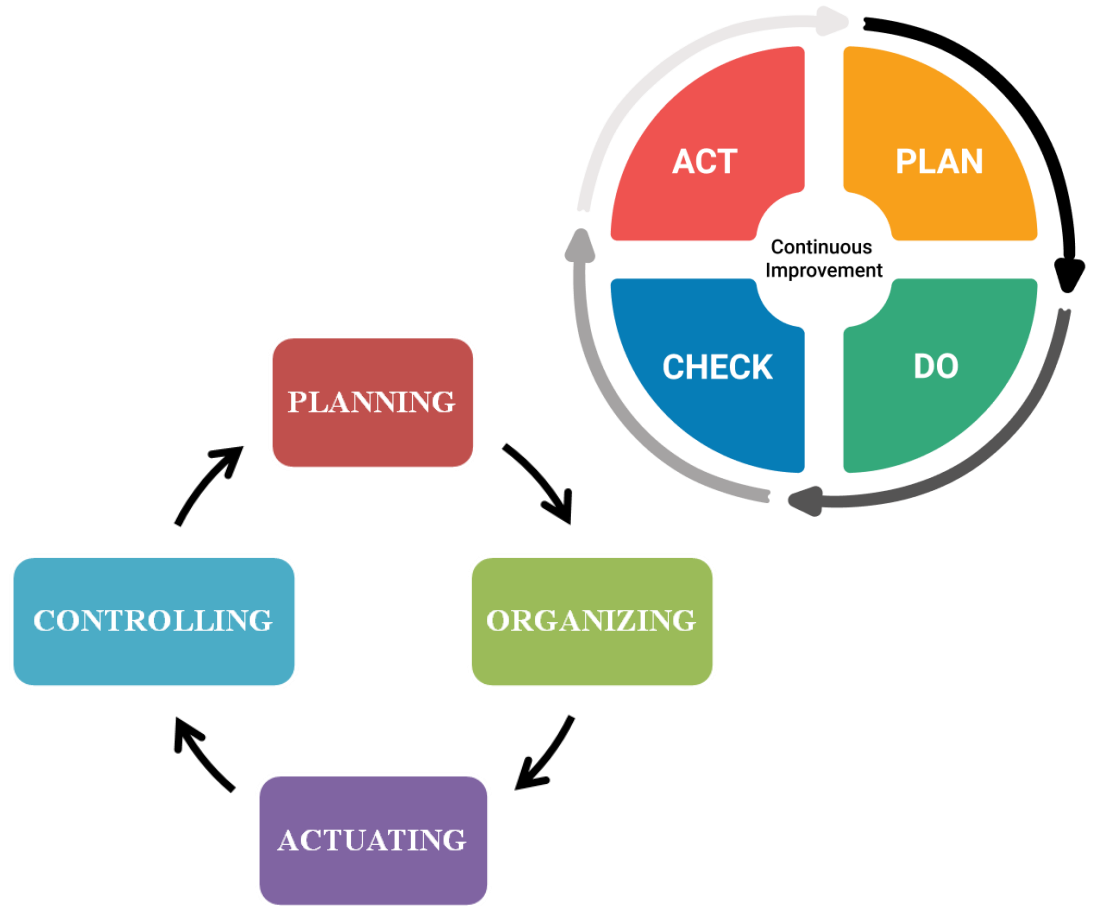
Daripada **Aaa Gile!**
Mending perbaiki dengan **Agile**

Aditya F. Ihsan

Manajemen?

Standard Classic Cycle.
Bisa PDCA, bisa POAC,
~~bisa PDKT~~

Intinya ada persiapan,
pelaksanaan, dan
timbalbalik (perbaikan)



Manajemen in Strategic Full Cycle



Dalam konteks audit dan penjaminan mutu, dikenal PPEPP, yang mana setiap tahap harus tercatat dan terdokumentasi.

Scale Problem

Masalah dari setiap siklus manajemen adalah di skala dan durasi.

Seberapa lama 1 siklus itu dilakukan, bagaimana bila ada masalah di tengah-tengah, apa yang terjadi jika berubah, dst



Misal ada perencanaan 1 proyek/kegiatan?

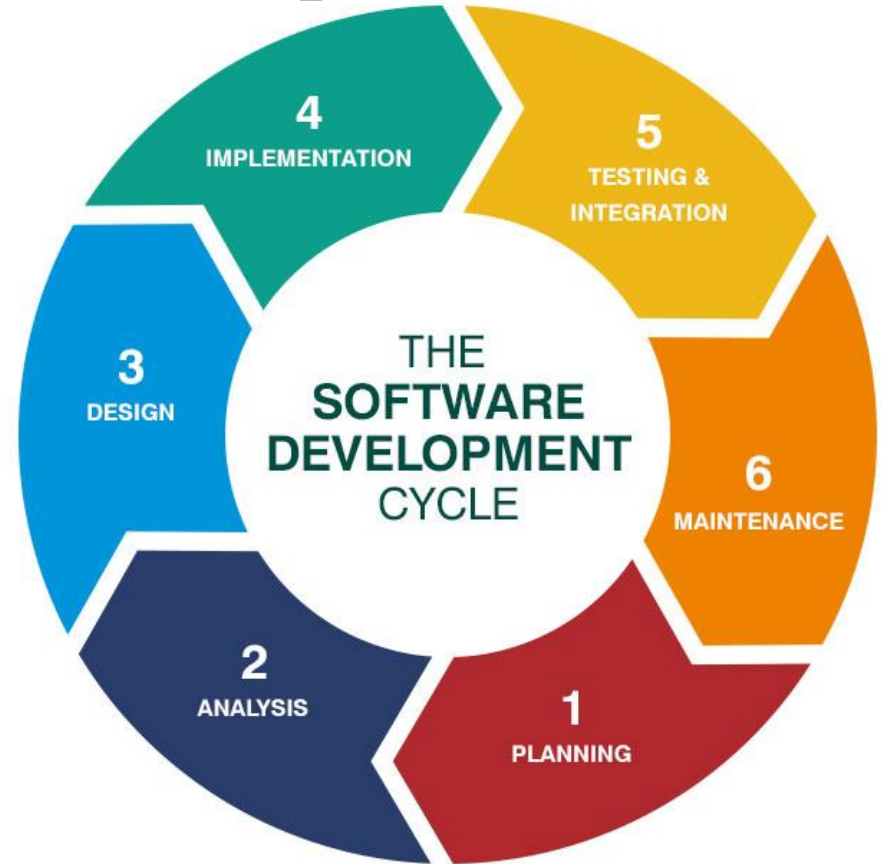
Dalam paradigma manajemen standar, itu setara 1 siklus atau 1 jalur.

Rencanakan proyeknya, sampai ke pelaksanaan, Evaluasi untuk proyek selanjutnya.



Learns from Software Development

Dalam pengembangan software dikenal juga SDLC, yang sebenarnya sama saja dengan siklus manajemen lainnya

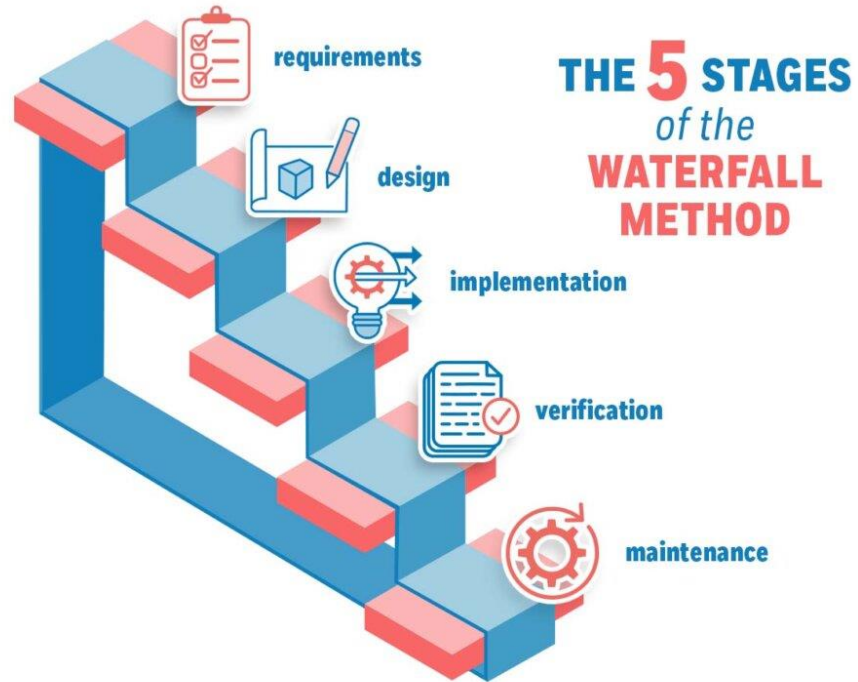


Learns from Software Development

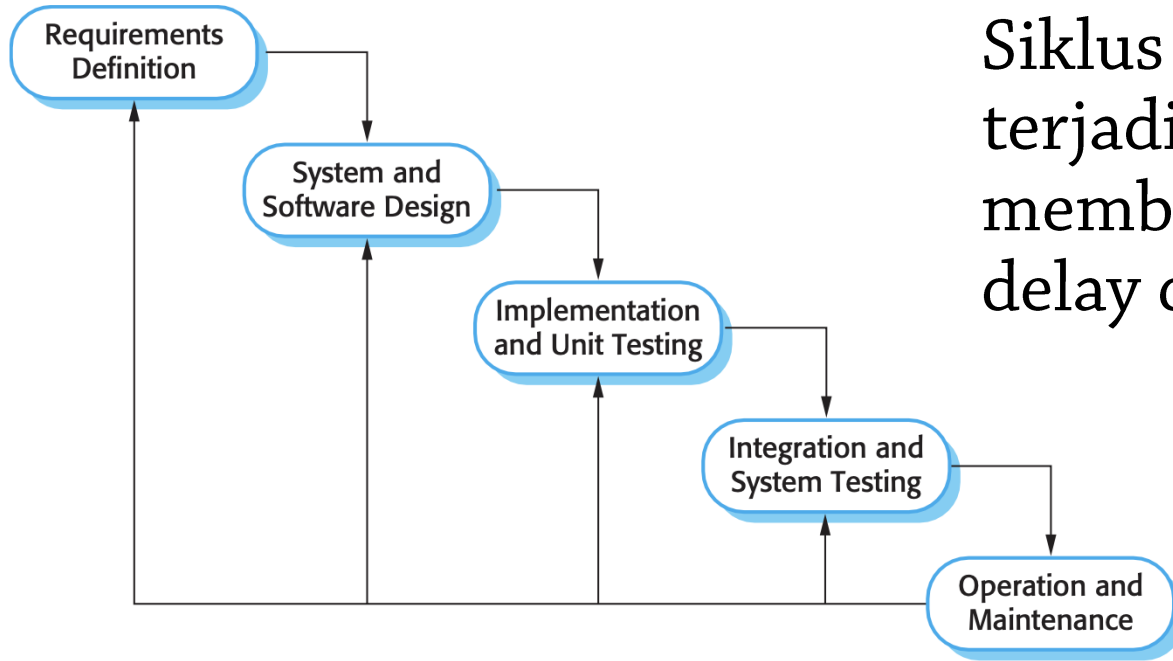
Dalam SDLC, dikenal Waterfall methodology.

Sifatnya mengalir dari hulu ke hilir, dari perencanaan sampai selesai dalam satu rangkaian.

Paradigma manajemen umum sering seperti ini



Learns from Software Development



Siklus timbalbaliknya terjadi **di ujung**, membuat selalu ada delay dalam evaluasi

Learns from Software Development

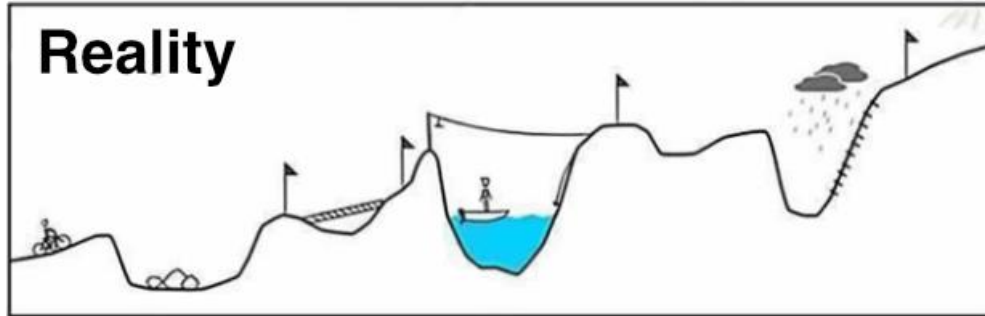
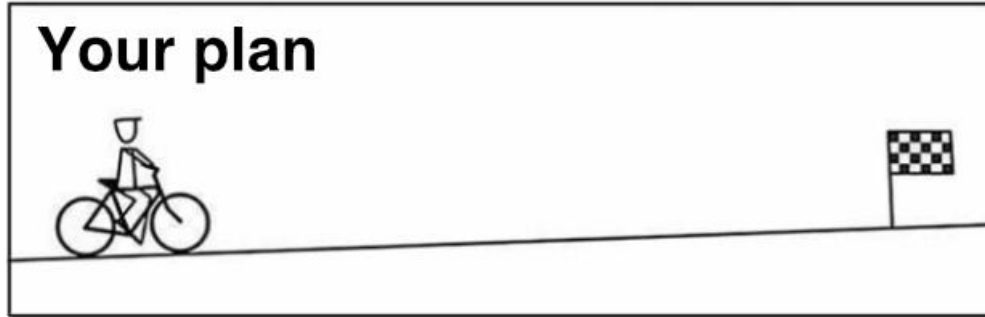
Model waterfall memang bisa memberi rencana yang lebih terstruktur. Segala hal bisa diestimasi di awal, dari kebutuhan dana, timeline, orang, dsb.

Namun, sangat tidak fleksibel. Isu di tengah-tengah akan jadi masalah besar. Butuh proses *feedback-loop* yang lebih dinamis.

Advantages

- Emphasizes Structured Organization
- Changes Can Be Implemented in Early Design Stages
- Suitable for Deadline and Milestone-Oriented Teams
- Accommodates Shifting Teams
- Allows for Accurate Cost Estimation

Learns from Software Development



Sama halnya juga manajemen (dan hidup) secara umum, realita selalu tidak pernah sesuai dengan rencana. Tidak mungkin menunggu sampai finish untuk evaluasi

Lantas bagaimana?

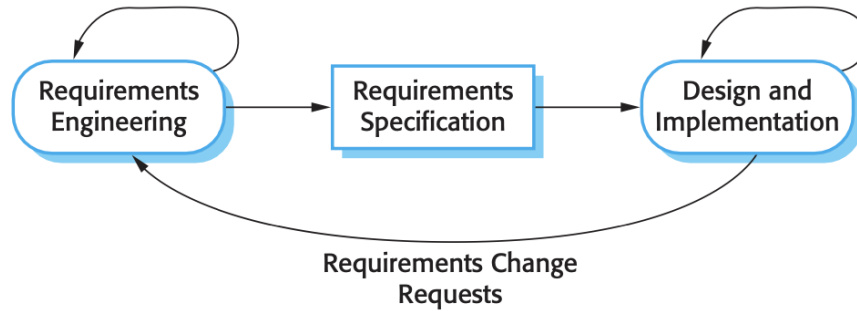
Do what is needed, not what was planned

Lantas bagaimana?
Agility is the key

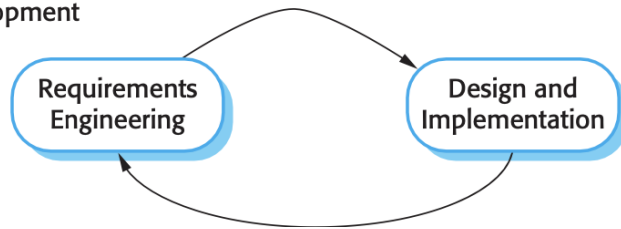
Do what is needed, not what was planned

*And that is, **Agile Methodology***

Plan-Based Development



Agile Development



Secara
mendasar, *agile*
fokus pada
agility, pada
fleksibilitas,
ketimbang
rencana yang
rigid

Agile is not a tool.
It is a way, a standard, a creed.

Agile Manifesto

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over **processes and tools**.
Working software over **comprehensive documentation**.
Customer collaboration over **contract negotiation**.
Responding to change over **following a plan**.

That is, while there is value in the items on the right, we value the items on the left more.”

(2001)

Customer satisfaction: through early and continuous delivery.

Welcoming change, even late in development.

Frequent delivery: of working software.

Close collaboration: between business and developers.

Motivated individuals: building projects around them.

Face-to-face conversation: as the most efficient communication.

Working software: as the primary measure of progress.

Sustainable development, maintaining pace indefinitely.

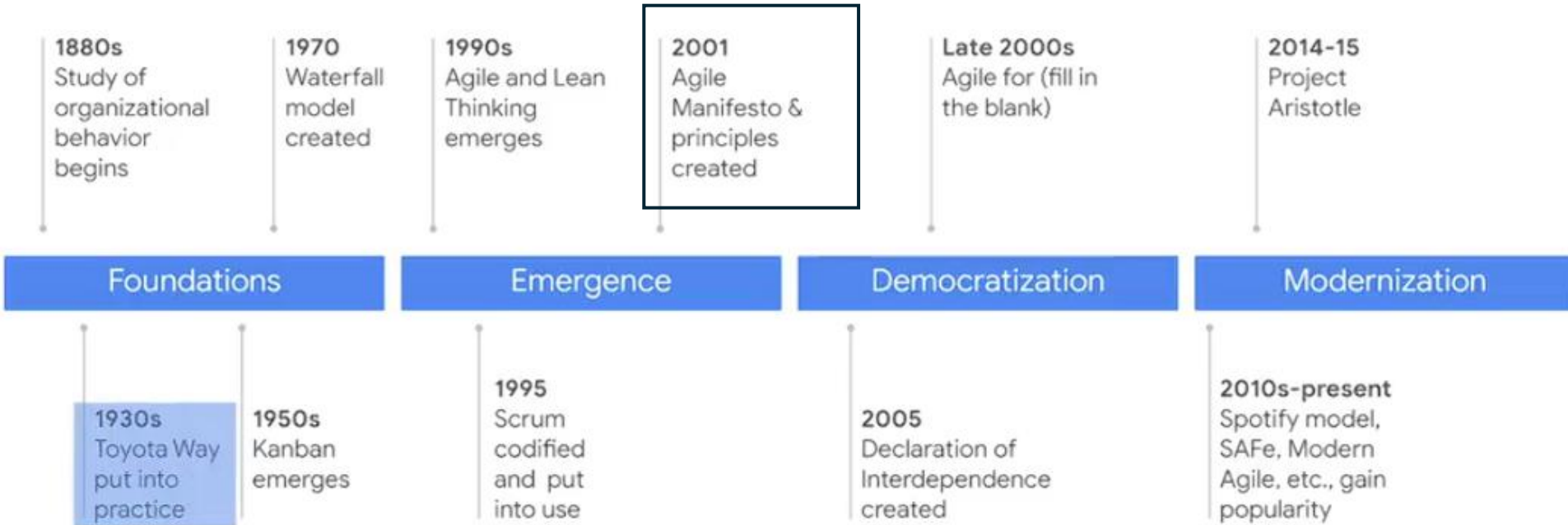
Continuous attention: to technical excellence and good design.

Simplicity, maximizing work not done.

Self-organizing teams: for best results.

Regular reflection: and adjustment for effectiveness.

Konteks Historis



Sedikit Perbandingan

Traditional:

- upfront planning and approval before work begins.
- Provides clear structure and predictability,
- can be inflexible if changes arise.
- Fit for well-defined requirements and **stable environments**, like manufacture

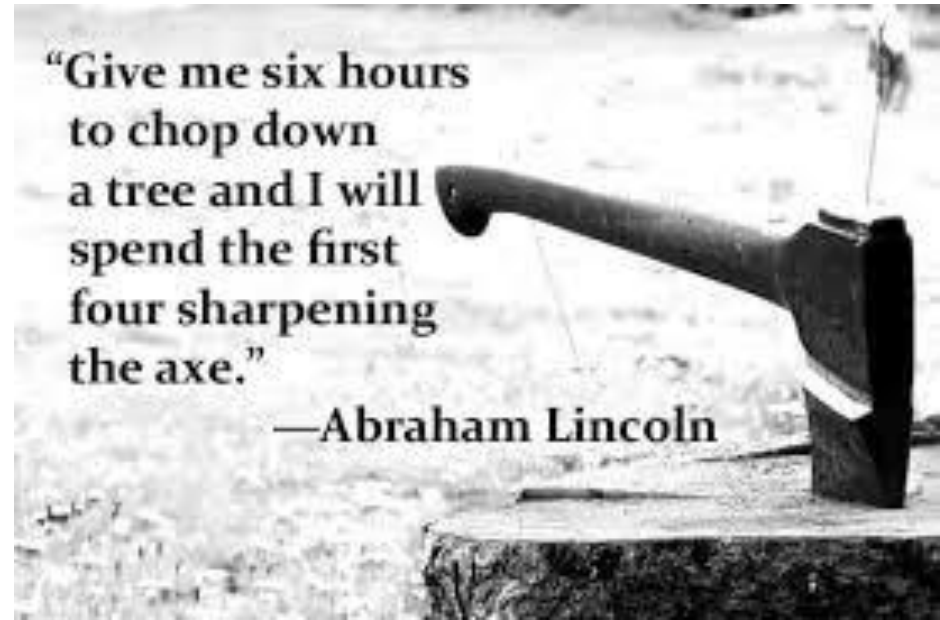
Agile:

- iterative, meaning teams work to continuously improve a product.
- allows for greater responsiveness as changes arise.
- encourages ongoing customer collaboration
- Fit for projects with a lot of uncertainty or complexity

Agile tidak mengatakan kita tidak butuh rencana, tapi ada skala dari setiap hal.

Rencana dibutuhkan untuk persiapan, terutama untuk tujuan besar dan target jangka panjang.

Namun, rencana bukan patokan utama untuk dilaksanakan.



*Agile emphasizes adaptive planning,
where work is planned in small
increments and adjusted based on
feedback.*

Oke, cukup dengan teorinya.
Let's see some practicalities

Pernah tahu tools seperti ini?





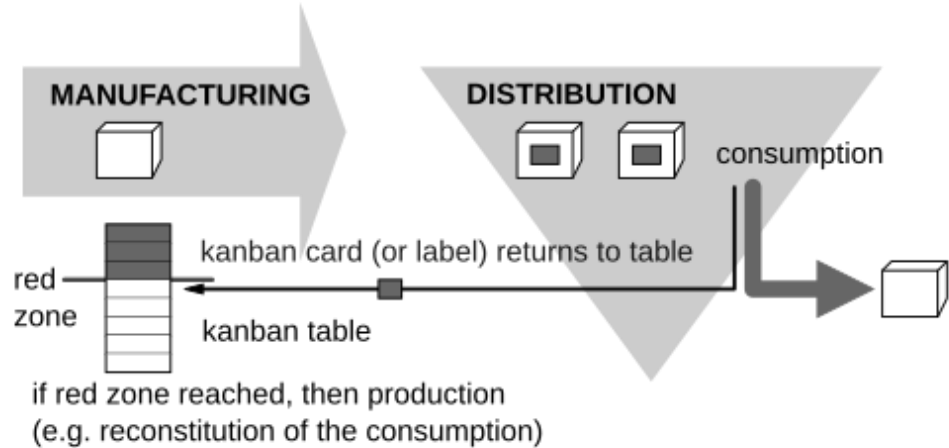
Semua ini tools untuk metodologi yang sama: Agile

Tools yang dapat dipakai:

(1) Kanban Board

Kanban (カンバン), berarti [sinyal](#).

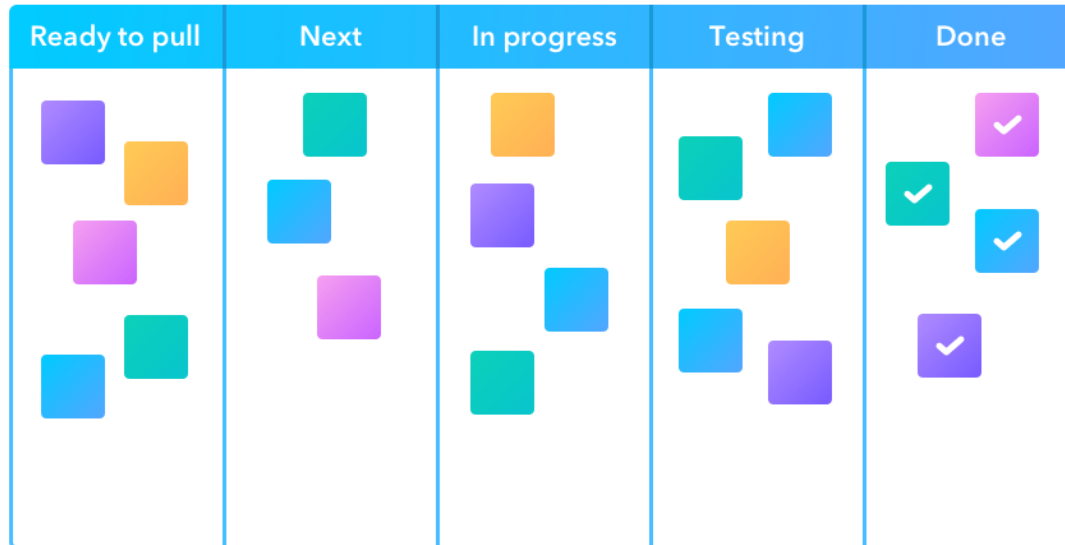
Dulunya dipakai dalam bentuk “kartu” yang ditempel untuk memberi status dalam proses manufaktur.



Tools yang dapat dipakai:

(1) Kanban Board

Sekarang menjadi salau satu visualization tool untuk Agile.

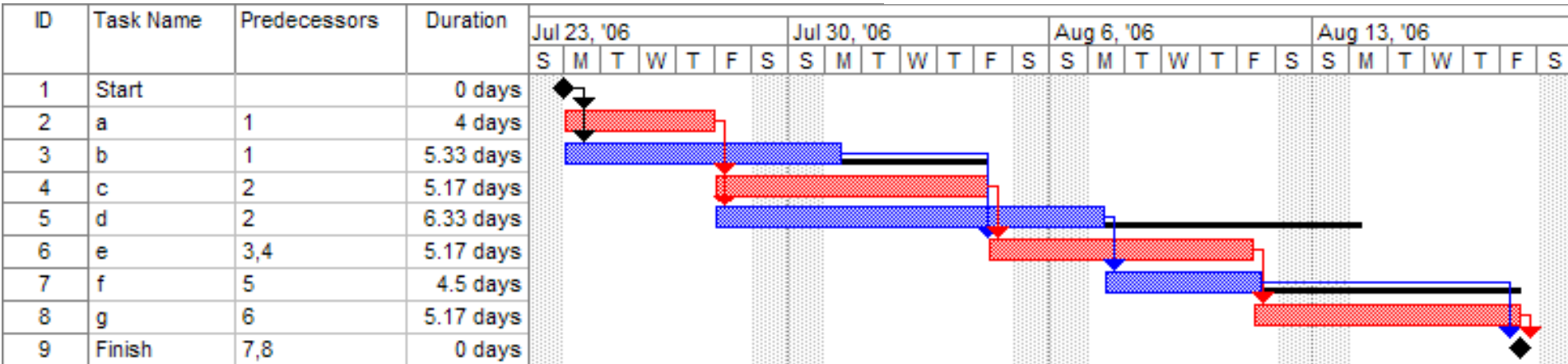


Tools yang dapat dipakai:

(2) Gantt Chart

Dipopulerkan oleh Henry Gantt
 Dipakai untuk melihat *dependency Relationship* antar aktivitas


| Activity | Predecessor | Time estimates (in days) | | | Expected time (T_E) |
|----------|-------------|--------------------------|------------|-----------|-------------------------|
| | | Opt. (O) | Normal (M) | Pess. (P) | |
| a | — | 2 | 4 | 6 | 4.00 |
| b | — | 3 | 5 | 9 | 5.33 |
| c | a | 4 | 5 | 7 | 5.17 |
| d | a | 4 | 6 | 10 | 6.33 |
| e | b, c | 4 | 5 | 7 | 5.17 |
| f | d | 3 | 4 | 8 | 4.50 |
| g | e | 3 | 5 | 8 | 5.17 |



Tools yang dapat dipakai: (3) Backlog

Catatan (*log*) yang digunakan di belakang (*back*) untuk melacak beragam hal yang terkait dengan project

Backlog









 +3

Epic ▾

Label ▾





Type ▾

▼ Sprint 3 7 Feb - 21 Feb (8 issues) 10 8 0 Complete sprint ⋮

| | | | | |
|--|---------------|---|---|---|
| ✓ NUC-344 Optimize experience for mobile ... | IN PROGRESS ▾ | 2 | ⋮ |  |
| 📌 NUC-360 Onboard workout options (OWO) | TO DO ▾ | 1 | ⋮ |  |
| ✓ NUC-337 Multi-dest search UI mobile web | TO DO ▾ | 5 | ⋮ |  |
| ✓ NUC-339 Billing system integration | TO DO ▾ | 3 | ⋮ |  |
| 📌 NUC-340 Account settings defaults | IN PROGRESS ▾ | 4 | ⋮ |  |
| ✓ NUC-341 Quick payment | IN PROGRESS ▾ | 2 | ⋮ |  |
| 🔴 NUC-342 Fast trip search | TO DO ▾ | 1 | ⋮ |  |
| 📌 NUC-335 Affiliate links integration - frontend | IN PROGRESS ▾ | 2 | ⋮ |  |

+ Create issue

▼ Sprint 4 22 Feb - 7 Mar (4 issues) 10 0 0 Complete sprint ⋮

| | | | | |
|--|---------|---|---|---|
| ✓ NUC-336 Quick booking for accommodations | TO DO ▾ | 1 | ⋮ |  |
| 🔴 NUC-346 Adapt web app payments provider | TO DO ▾ | 2 | ⋮ |  |
| ✓ NUC-343 Fluid booking on tablets | TO DO ▾ | 3 | ⋮ |  |
| 🔴 NUC-354 Shopping cart purchasing error | TO DO ▾ | 4 | ⋮ |  |

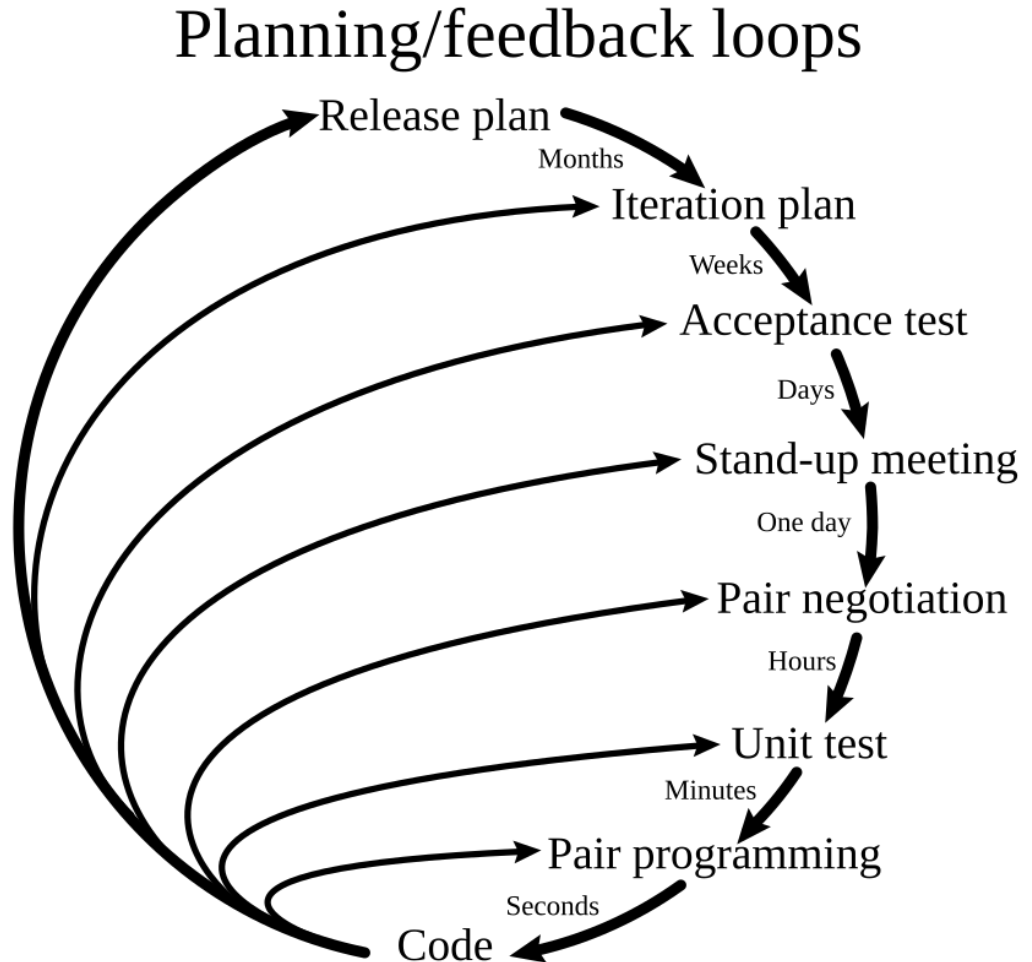
Kita semua sudah sering pakai alat-alat itu, tapi tanpa metodologi yang tepat, itu seperti memakai iPad sebagai talenan

Metode-metode Agile

- eXtreme Programming (XP)
- **Scrum**
- Crystal
- Scaled Agile Framework (SAFe)
- PRiSM (Projects Integrating Sustainable Methods)
- Feature-driven development (FDD)
- Adaptive Project Framework (ADF)
- dst

Extreme Programming

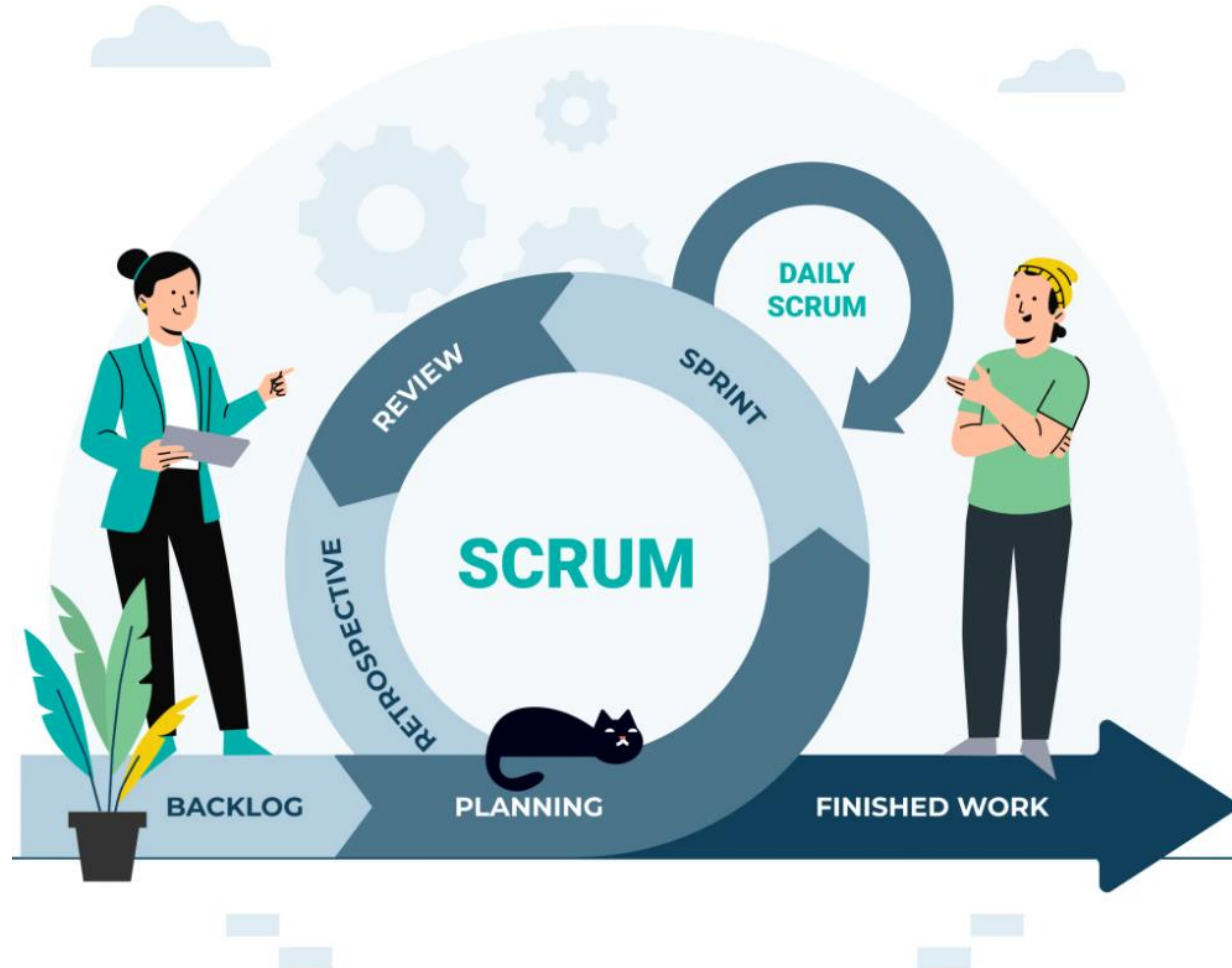
Fokus pada pemecahan task dan iterasi sampai ke semua skala



Scrum

Metode Agile paling popular.

Berisi siklus bersarang yang memastikan kontrol bertahap



Scrum

Scrum berpusat pada “sprint”, semacam target jangka pendek yang menjadi fokus.

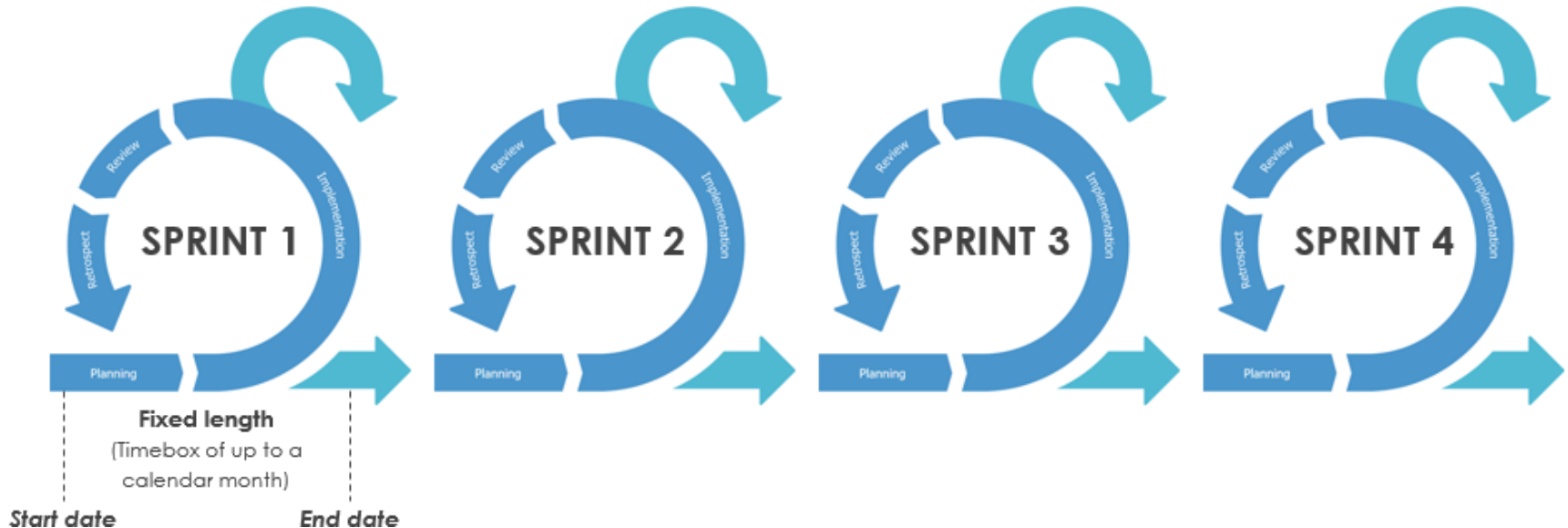
Satu sprint setara satu siklus manajemen sendiri

A Sprint in Agile



Scrum

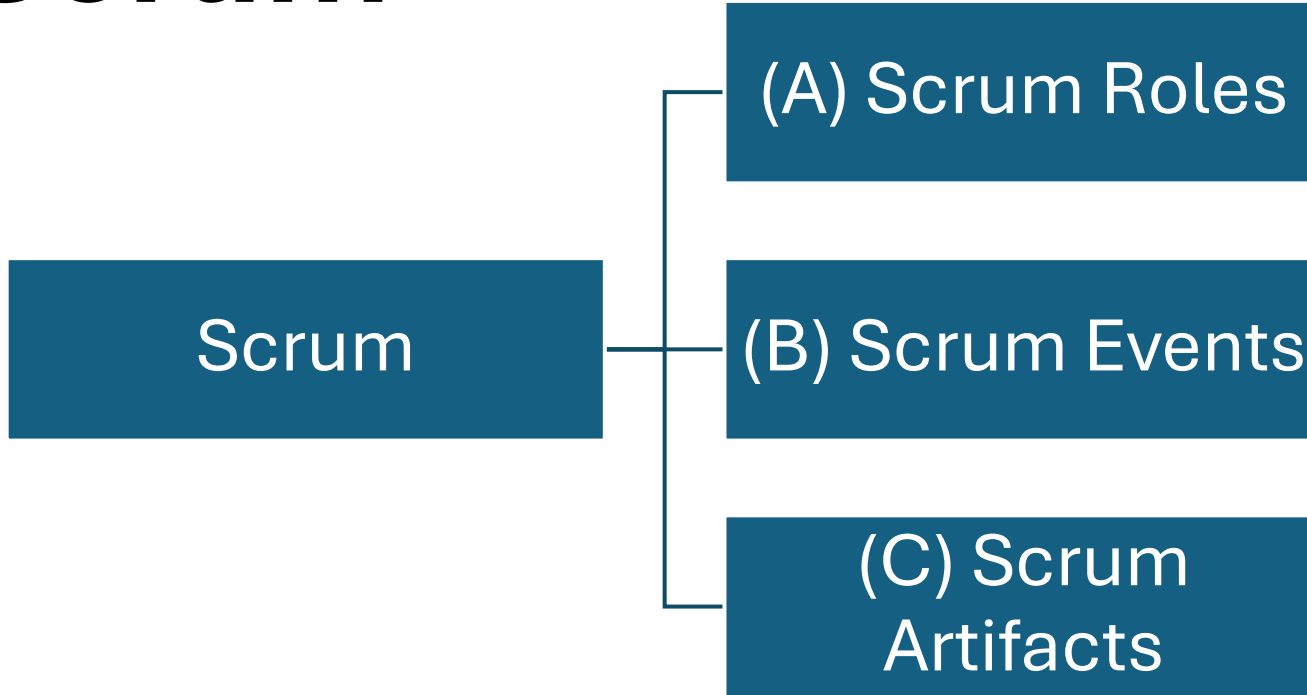
Sprint ini semacam fokus dalam satu waktu, dan akan diikuti sprint-sprint berikutnya. Suatu proyek besar dapat dipecah jadi beberapa sprint yang direncanakan di awal.



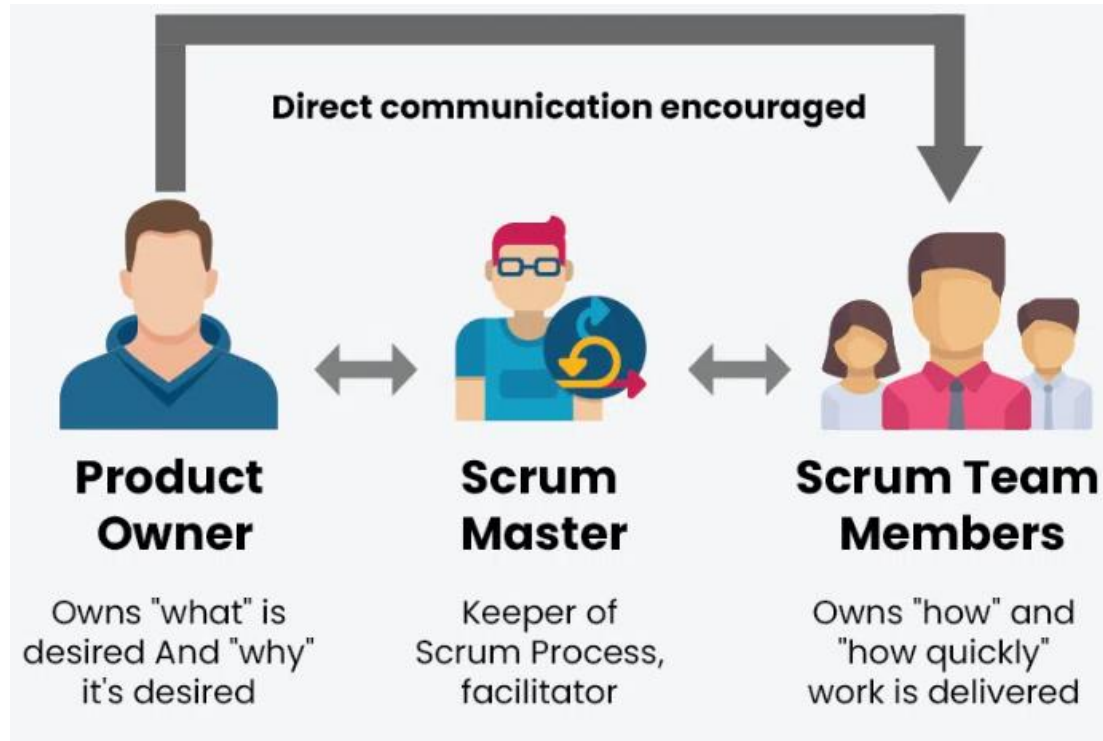
Scrum



Scrum

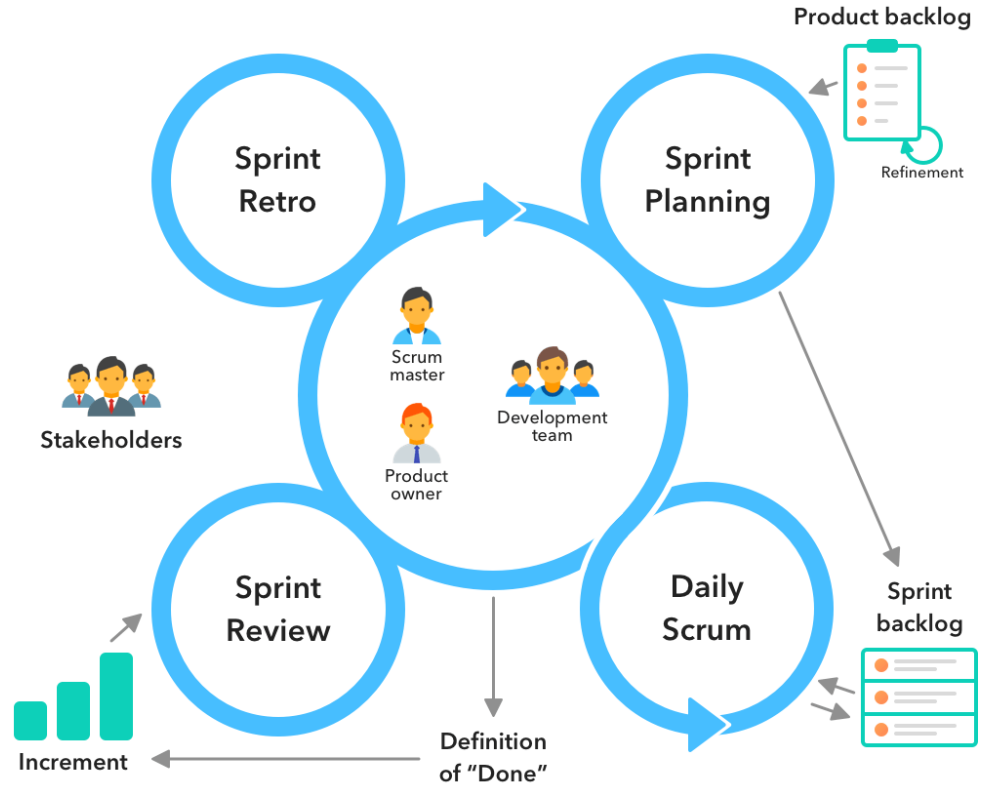


A. Scrum Roles



B. Scrum Events

Setiap sprint paling tidak terdiri dari 4 aktivitas



B. Scrum Events

- **Sprint planning** → rapat di awal sprint untuk memecah target sprint ke beberapa task/target dan menetapkan PIC
- **Daily Standup** → rapat singkat (10-15 menit) tiap hari, biasanya di pagi hari sebelum mulai kerja, untuk control harian: evaluasi hari sebelumnya dan hari itu mau ngapain.
- **Sprint Review** → rapat di akhir sprint untuk memeriksa ketercapaian target sprint dan merencanakan ulang sprint berikutnya.
- **Sprint retrospective** → rapat setelah review untuk evaluasi internal. Fokus pada Evaluasi proses, terkait apa yang bisa di-improve secara internal.

B. Scrum Events: Sprint Review vs Sprint Retro

Retro dan review mirip, bedanya Cuma di fokus bahasan dan peserta.

Rapat Review Bersama PO, sedangkan rapat retro hanya internal tim

OPENING

Review

informing everyone about the status quo and the results



Retrospective

creating an atmosphere for honest exchange

MIDDLE

Review

discussion about the work done and the backlog



Retrospective

sharing facts and feelings, clustering and prioritizing topics, in-depth discussion

CLOSING

Review

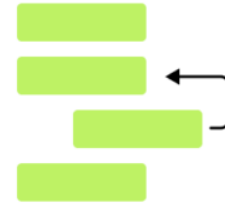
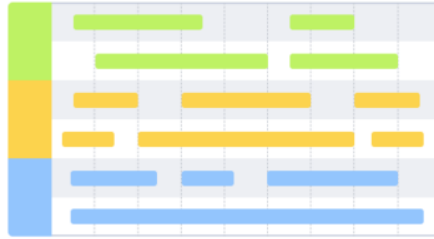
brief wrap-up, kudos, potential door opener for the drafting session



Retrospective

check-out (e.g. feedback round)

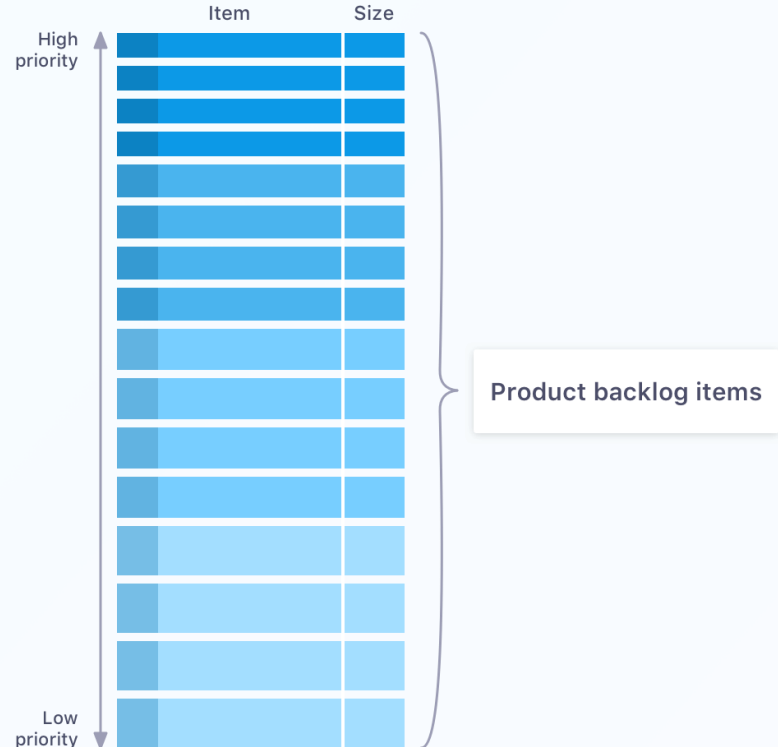
C. Scrum Artifacts: Product roadmap and backlog



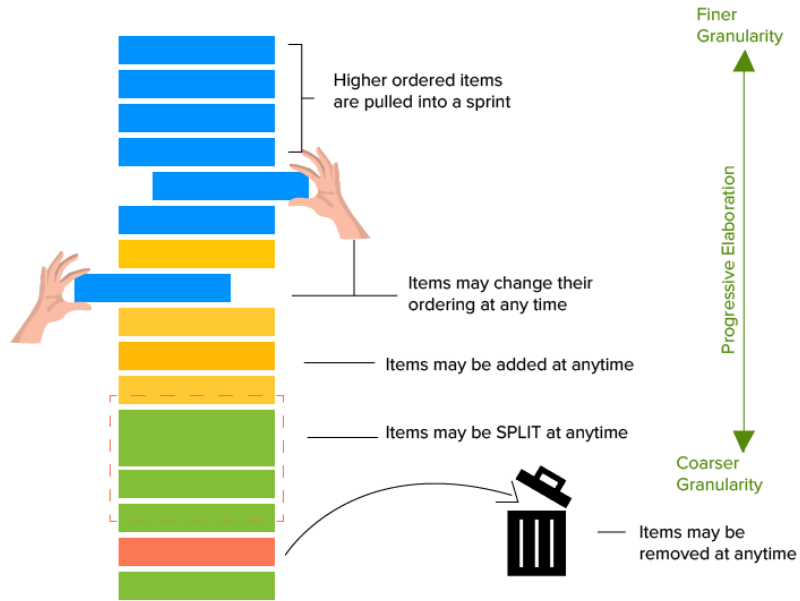
| | Product Roadmap | Product Backlog |
|------------|---|---------------------------------|
| Content | High-level: themes, outcomes, and goals | Task-level |
| Audience | Executive Team (and other stakeholders) | Primarily for development teams |
| Intent | Conveys a strategy | Conveys tactical step |
| Time Frame | Varies, typically 3-12 months | 1-4 sprints |

C. Scrum Artifacts: Product backlog

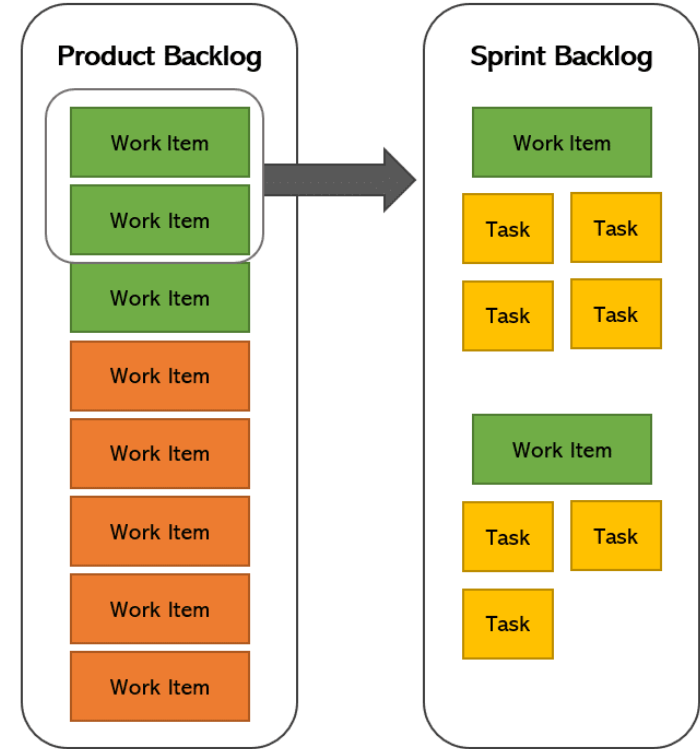
- Product backlog berisi work item yang dapat dikerjakan, bisa berisi:
 - Aktivitas yang memang direncanakan
 - Ide-ide dan usul baru
 - Isu atau masalah
 - Dll
- Dapat terupdate setiap saat, namun selalu dipilah berdasarkan prioritas.



C. Scrum Artifacts: Product backlog and sprint backlog



Source : Visual Paradigm



Practicalities?

Individuals and interactions over processes and tools.

-> banyak-banyak ngobrol

Working software over comprehensive documentation.

-> yang penting jalan dulu

Customer collaboration over contract negotiation.

-> yang penting bareng

Responding to change over following a plan.

-> selalu siap ngerombak setiap saat

Adaptasi ke Manajemen

- Define goals, kemudian susun *product roadmap* (1 tahun).
- Rincikan roadmap ke dalam *product backlog* (tampung semua ide dan kemungkinan, urutkan berdasarkan prioritas, kelompokkan berdasarkan kegiatan)
- Rencanakan *sprints* untuk setiap kegiatan. Susun *sprint backlog* untuk setiap kegiatan (ambil dari product backlog)
- Jalankan *sprints*, lakukan *daily scrum*, *sprint review & retro*, ulangi.
- Semua proses di atas dapat dilakukan di level Bidang dan level program.
- Di level bidang, manajer adalah PO sekaligus *scrum master*. Di level program, *scrum master*-nya adalah ketua program (PO tetap manajer). Sesuaikan level di masing-masing tempat.

Selamat ber-Agile!