

Tujuan

- Untuk menjelaskan Agile Development Methods yang merupakan salah satu metodologi pengembangan perangkat lunak
- Proses dan model-model pengembangan perangkat lunak Agile Development Methods

Agile Development Methods

Agile Model merupakan proses pengembangan software yang berkembang pada tahun 1990. Metodologi yang dikenal sebagai agile development methods ini mengutamakan fleksibilitas terhadap perubahan-perubahan yang terjadi selama pengembangan. Model-model dari agile diantaranya Rational Unified Process (1994), Scrum (1995), Crystal, Extreme Programming (1996), dan Adaptive Software Development, Feature Driven Development, and Dynamic Systems Development Method (DSDM) (1995). Dan pada akhirnya terbentuklah pada tahun 2001 proses pengembangan agile Modeling (AM).

Proses Pengembangan Agile Modeling (AM)

Dilakukan secara iterasi atau perulangan. Jika suatu proyek pengembangan software dikerjakan dengan menggunakan metode agile, maka selama waktu pengerjaannya akan selalu dijumpai proses pengembangan yang dilakukan berulang.



12 Prinsip-prinsip Agile Development Methods

- Prioritas utama adalah memuaskan klien dengan menghasilkan perangkat lunak yang bernilai secara cepat dan rutin
- Siap terhadap perubahan kebutuhan. Proses Agile memanfaatkan perubahan untuk keuntungan klien
- Menghasilkan perangkat lunak yang bekerja secara rutin, dari jangka waktu beberapa minggu sampai beberapa bulan, dengan mengutamakan jangka waktu yang pendek
- Rekan bisnis dan pengembang perangkat lunak harus bekerjasama sepanjang proyek
- Lingkungan pengembang proyek memiliki suasana yang motivatif. Berikan mereka lingkungan dan dukungan yang dibutuhkan, dan percayai mereka untuk dapat menyelesaikan pekerjaan dengan baik
- Metode yang paling efisien dan efektif untuk bertukar informasi dari dan dalam tim pengembang adalah dengan komunikasi secara langsung

- Perangkat lunak yang bekerja adalah ukuran utama kemajuan suatu tim
- Proses Agile mendukung pengembangan yang berkelanjutan dengan kecepatan pengembangan yang konsisten
- Perhatian terhadap detail-detail teknis dan desain akan meningkatkan agility
- Kesederhanaan (memaksimalkan jumlah pekerjaan yang belum dilakukan) adalah hal yang sangat penting
- Self-organizing team mendukung arsitektur, kebutuhan, dan rancangan perangkat lunak yang baik
- Secara berkala, tim pengembang berefleksi tentang bagaimana agar pengembangan lebih efektif, kemudian menyesuaikan cara bekerja mereka.

Manfaat Agile development methods

- High-value & working App system
- Iterative, incremental, evolutionary
- Cost control & value-driven development
- High-quality production
- Flexible & risk management
- Collaboration
- Self-organizing, self-managing teams

High-value & working App system



Dapat dihasilkan perangkat lunak yang mempunyai nilai jual yang tinggi, biaya pembuatan bisa ditekan dan perangkat lunak bisa berjalan dengan baik

Iterative, incremental, evolutionary



Tim harus bekerja dalam waktu yang singkat(biasanya 1-3 minggu) dan juga selalu menambah fungsionalitas dari perangkat lunak sesuai dengan kebutuhan klien

Cost control & value-driven development

pengembangan perangkat lunak disesuaikan dengan kebutuhan pengguna, tim bisa dengan cepat merespon kebutuhan yang diinginkan pengguna sehingga waktu dan biaya pembuatan perangkat lunak bisa dikontrol

High-quality production

Biaya pembuatan perangkat lunak bisa ditekan dan proses pembuatan bisa dipercepat , tetapi kualitas dari perangkat lunak yang dibuat harus tetap dijaga

Dengan melakukan tes setiap fungsionalitas perangkat lunak setelah selesai dibuat berarti agile juga mengakomodir kebutuhan ini

Flexible & risk management

Apabila menggunakan metode pembuatan yang biasanya dipakai dan apabila ingin mengubah fungsionalitas dari wireframe yang telah dibuat dibutuhkan proses yang rumit

Mulai dari pertemuan dengan sistem analis untuk mengubah sistem perangkat lunak, perubahan rencana rilis produk hingga perubahan biaya produksi sehingga fungsionalitas perangkat lunak mudah diubah dan akhirnya kegagalan perangkat lunak pun bisa diminimalisir

Collaboration

tim pengembang diharuskan sering bertemu untuk membahas perkembangan proyek dan feedback dari klien yang nantinya akan ditambahkan dalam perangkat lunak, sehingga tim bisa berkolaborasi dengan maksimal

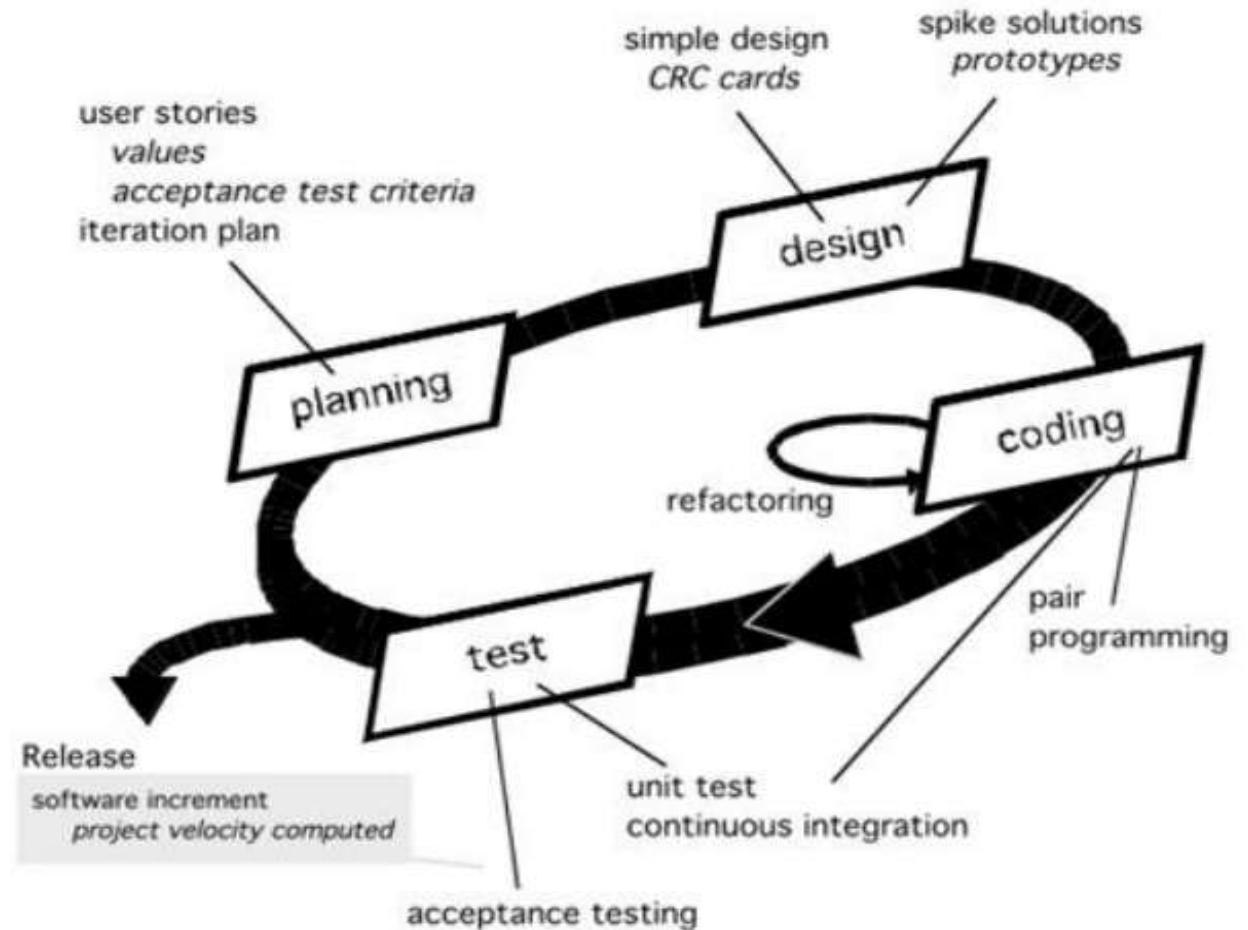
Self-organizing, self-managing teams

Rekrut orang terbaik, beri dan dukung kebutuhan mereka lalu biarkan mereka bekerja. Itulah perbedaan agile dan SDM lainnya. Dengan agile, developer dapat memanajemen dirinya sendiri, sedangkan manajer tim hanya bertugas mengkolaborasikan developer perangkat lunak dengan klien. Sehingga terciptalah tim yang solid

Model Agile XP (eXtreme Programming)



Model ini tidak hanya melakukan proses code dan testing saja, tetapi juga melakukan proses pemetaan arsitektur sistem agar proses coding dan testing bisa dilaksanakan tanpa mengganggu sistem yang ada sebelumnya. Proses analisis sistem yang telah berjalan dilaksanakan dalam rangka lebih memahami alur kerja sistem

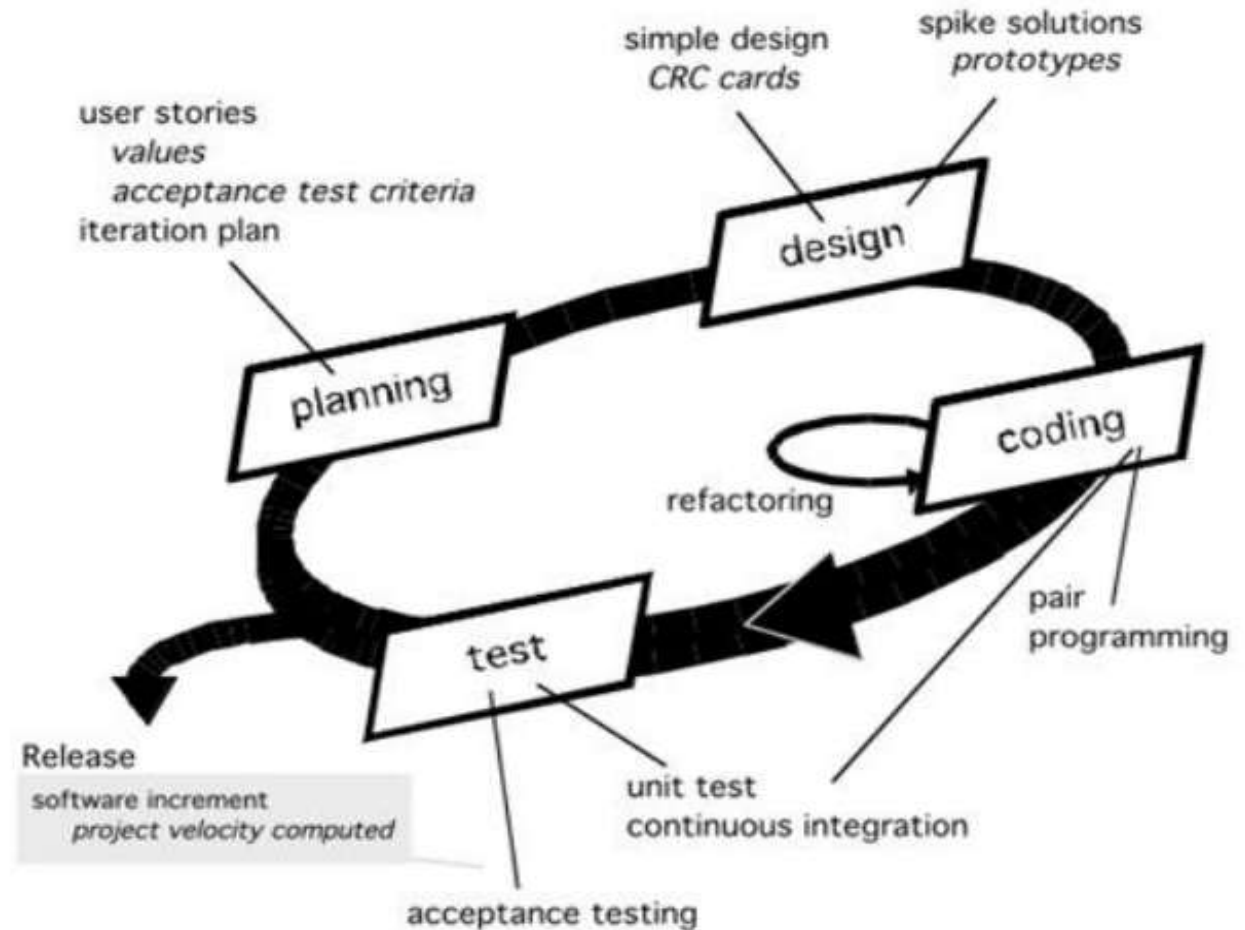


Tahapan - Model Agile

1. Planing

Perencanaan adalah proses metodis yang dirancang untuk mencapai tujuan tertentu dan pengambilan keputusan untuk mencapai hasil yang diinginkan. Kebutuhan yang dibutuhkan pada tahap ini teknik pengumpulan data:

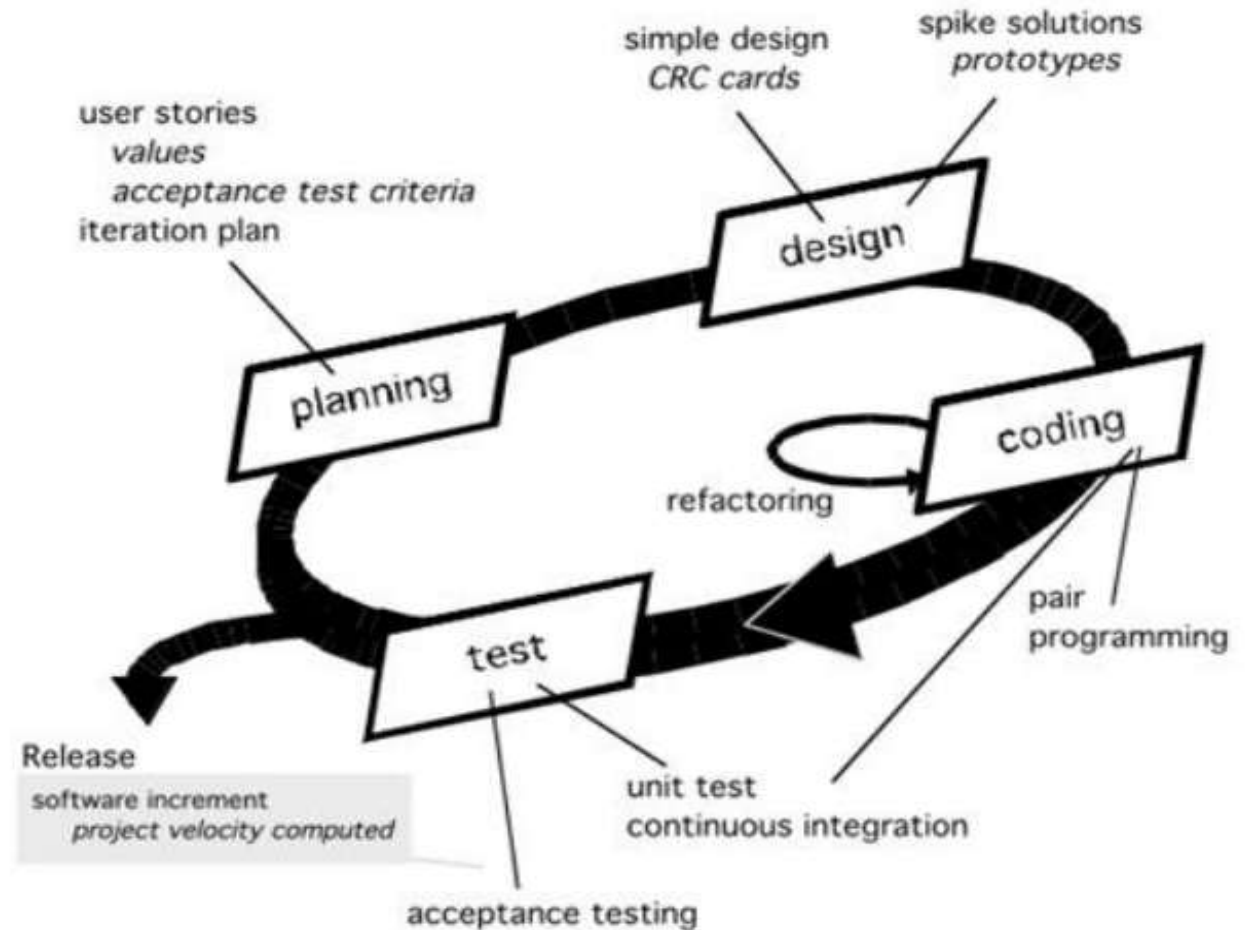
- Analisis kebutuhan sistem
- Identifikasi actor
- Identifikasi use case



Tahapan - Model Agile

2.Design

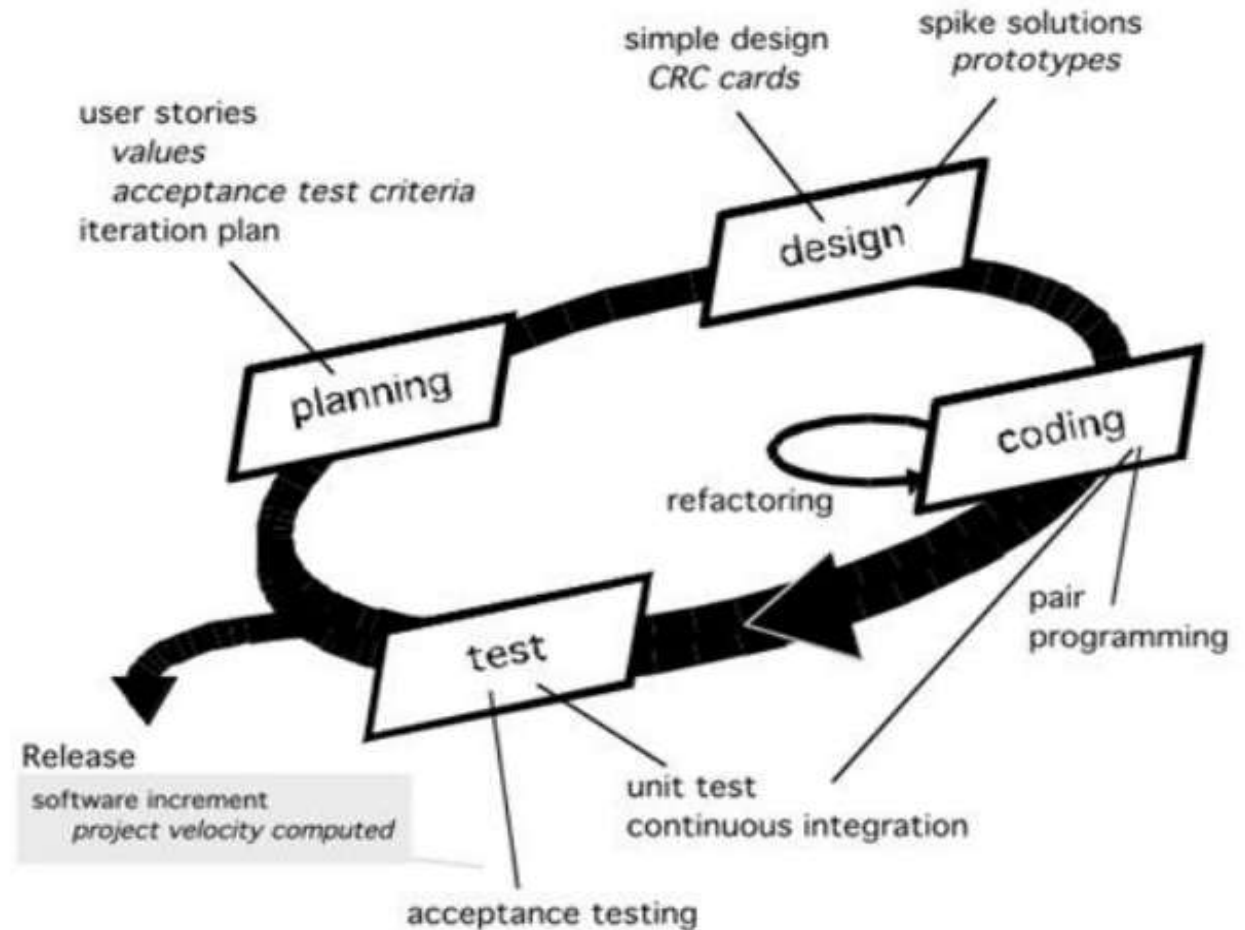
Aktivitas design dalam pengembangan aplikasi bertujuan untuk mengatur pola logika dalam sistem. Sebuah design yang baik, dapat mengurangi ketergantungan antar setiap proses pada sebuah sistem. Dengan begitu, jika salah satu fitur pada sistem mengalami kerusakan, tidak akan mempengaruhi sistem secara keseluruhan



Tahapan - Model Agile

3.Coding

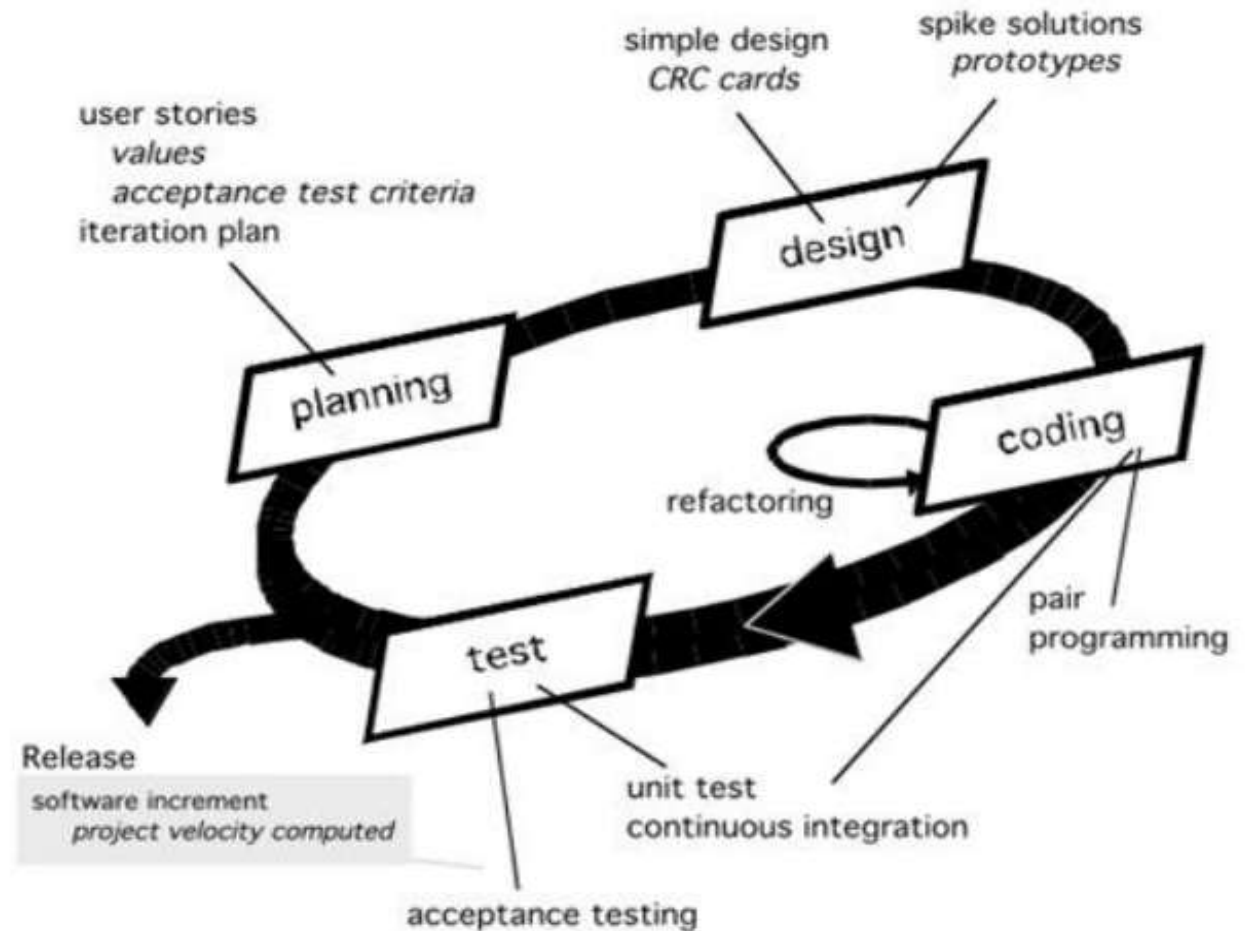
Setelah berbagai unit test selesai dibangun, tim barulah melanjutkan aktivitasnya ke penulisan coding aplikasi. XP menerapkan konsep pair programming dimana setiap tugas sebuah modul dikembangkan oleh 2 orang programmer



Testing

4. Testing

Pada tahapan pengujian ini aplikasi langsung diuji coba oleh pengguna atau klien dan mendapat tanggapan langsung mengenai penerapan cerita yang telah digambarkan sebelumnya



Kelebihan XP

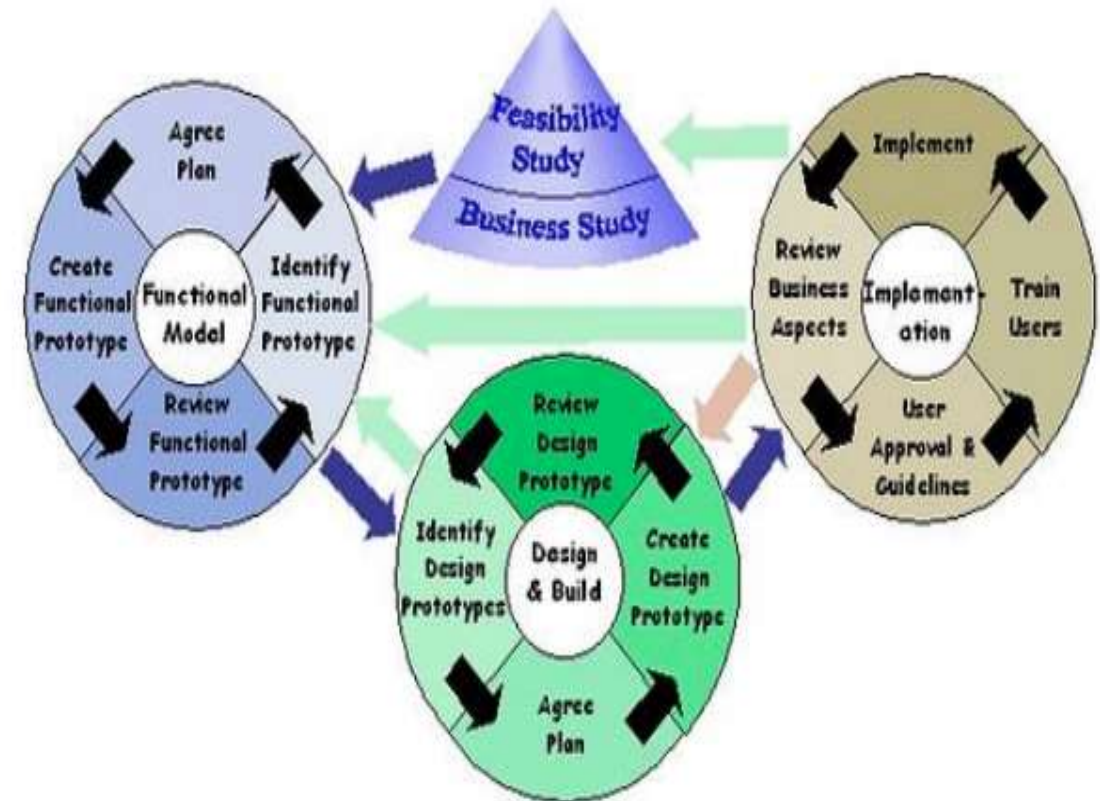
- Umpan balik yang segera setelah mendapat respon dari user
- Collective ownership yang tidak menggantungkan pekerjaan pada satu orang saja
- Perubahan dan penambahan requirements dapat direspon meskipun proses pengembangan sudah hampir selesai
- Proses pengembangan yang menyertakan satu orang dari pihak user menjadi on-site customer memudahkan komunikasi selama proses pengembangan
- Metode yang populer karena lebih santai dan non-restriktif
- Biaya lebih murah
- Mampu mengotomatiskan tes
- Setiap feedback ditanggapi dengan melakukan tes, unit tes atau system integration dan jangan menunda karena biaya akan membengkak (uang, tenaga, waktu)
- Banyak ide baru dan berani mencobanya, berani mengerjakan kembali dan setiap kali kesalahan ditemukan, langsung diperbaiki

Kekurangan

- Developer harus selalu siap dengan perubahan karena perubahan akan selalu diterima
- Tidak bisa membuat kode yang detail di awal (prinsip simplicity dan juga anjuran untuk melakukan apa yang diperlukan hari itu juga)
- XP tidak memiliki dokumentasi formal yang dibuat selama pengembangan. Satu-satunya dokumentasi adalah dokumentasi awal yang dilakukan oleh user
- Komunikasi yang selalu bersifat oral tanpa dokumentasi formal

Model Agile DSDM (Dynamic Systems Development Method)

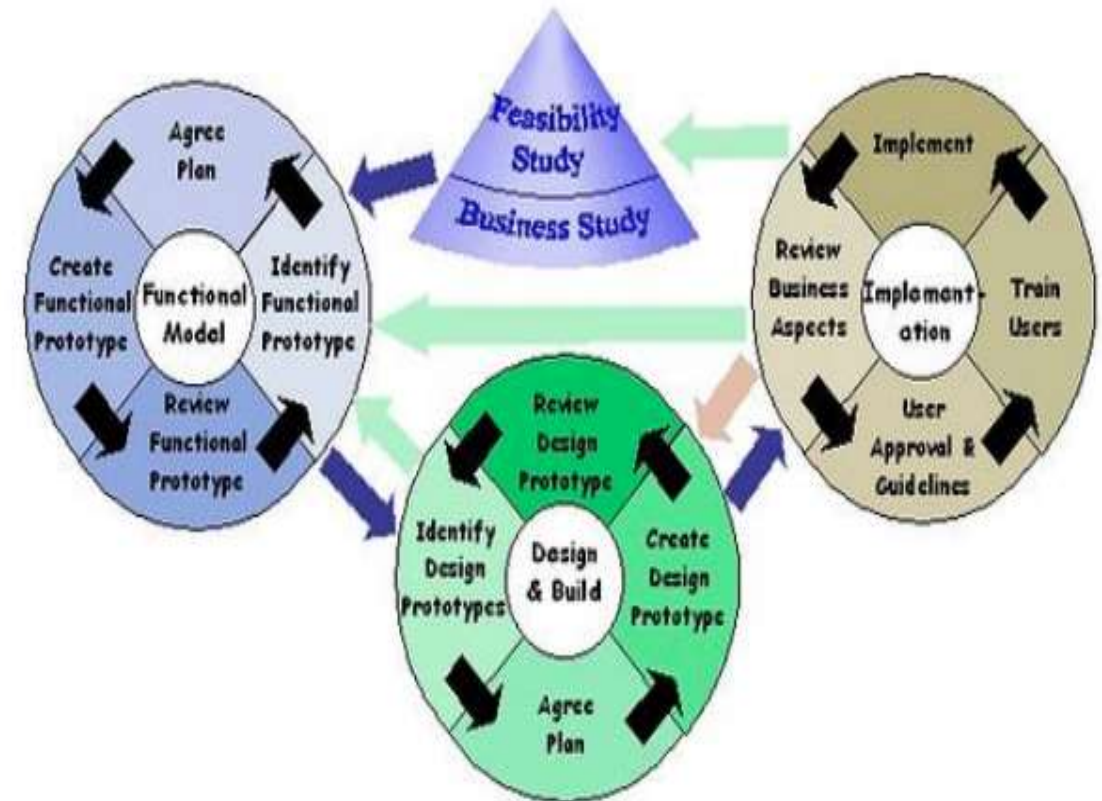
Dynamic System Development Method menyediakan kerangka kerja (framework) untuk membangun dan memelihara sistem dalam waktu yang terbatas melalui penggunaan prototyping yang incremental dalam lingkungan yang terkondisikan



Model Agile DSDM (Dynamic Systems Development Method)

1. Feasibility Study

Adalah laporan kelayakan tingkat tinggi yang memungkinkan komite pengarah proyek untuk memutuskan masa depan proyek, dan studi kelayakan lebih lanjut

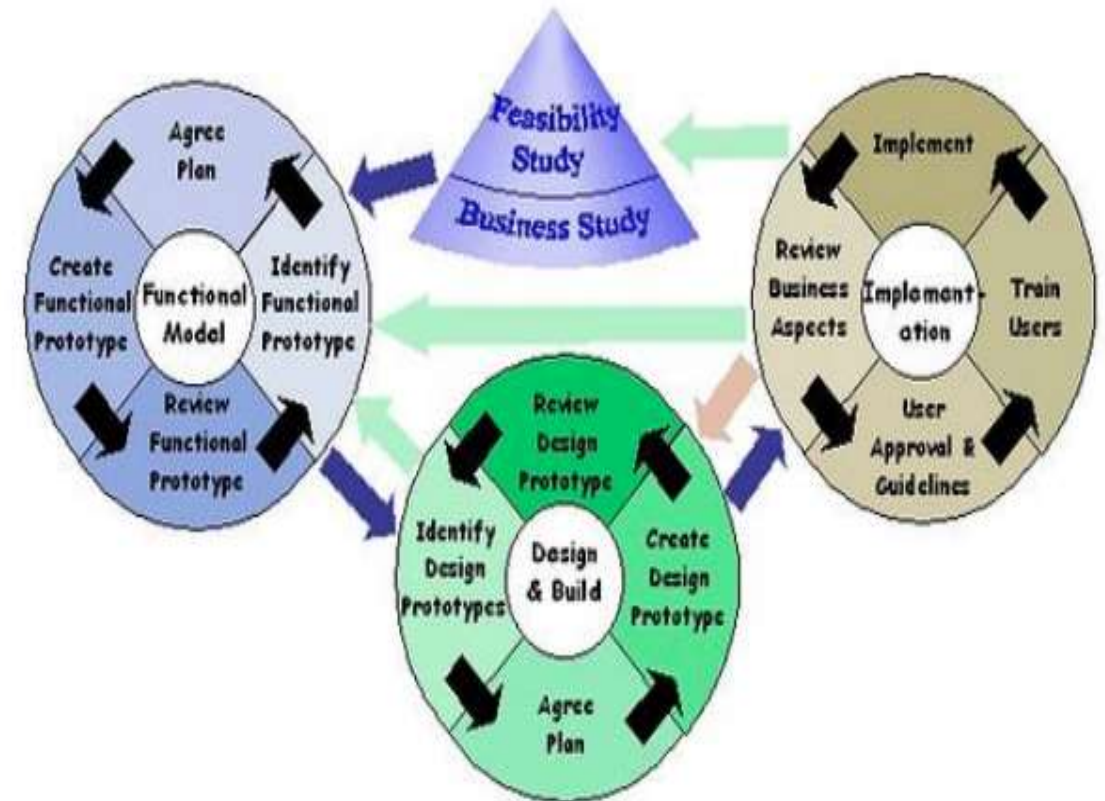


Model Agile DSDM (Dynamic Systems Development Method)



2. Functional Model

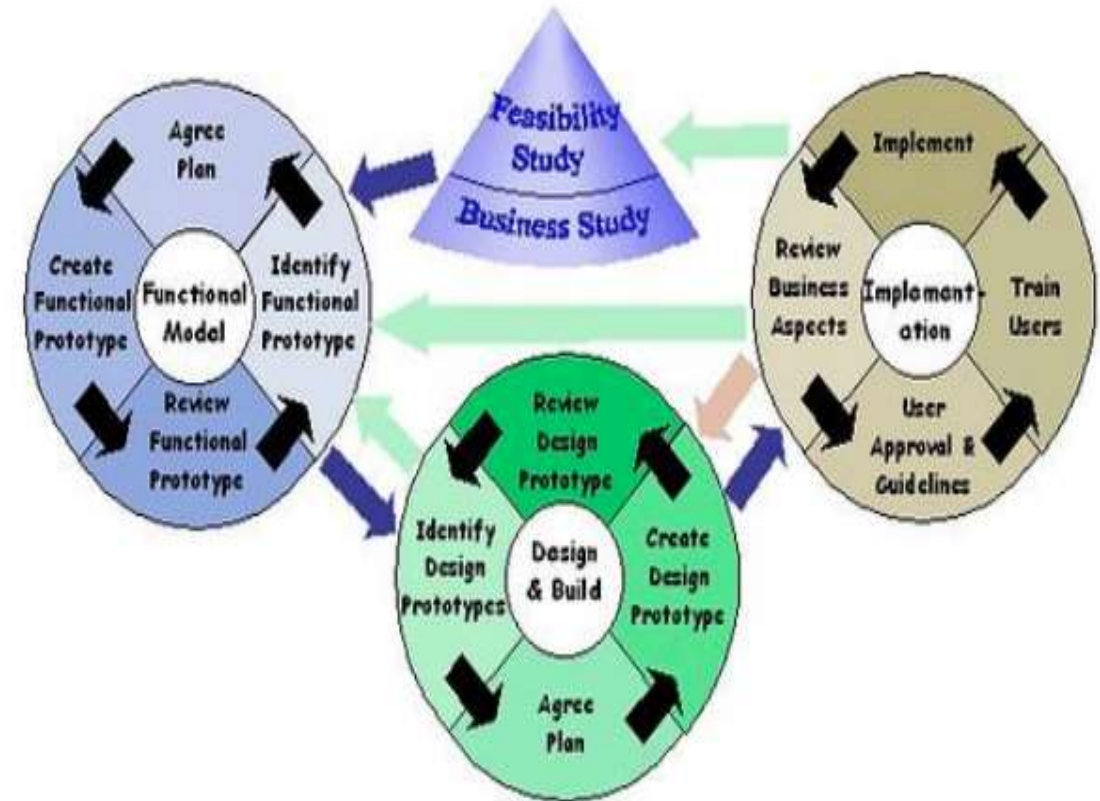
Model fungsional yang terdiri dari kedua prototipe perangkat lunak yang bekerja dan model statis. Fase ini menghasilkan pengolahan informasi yang diperoleh dalam penelitian bisnis



Model Agile DSDM (Dynamic Systems Development Method)

3. Design and Build Iteration

Fase ini menyempurnakan prototype fungsional yang dikembangkan untuk memenuhi kebutuhan fungsional (Modul) yaitu: Mengidentifikasi, Merencanakan, mengembangkan, validasi fungsi

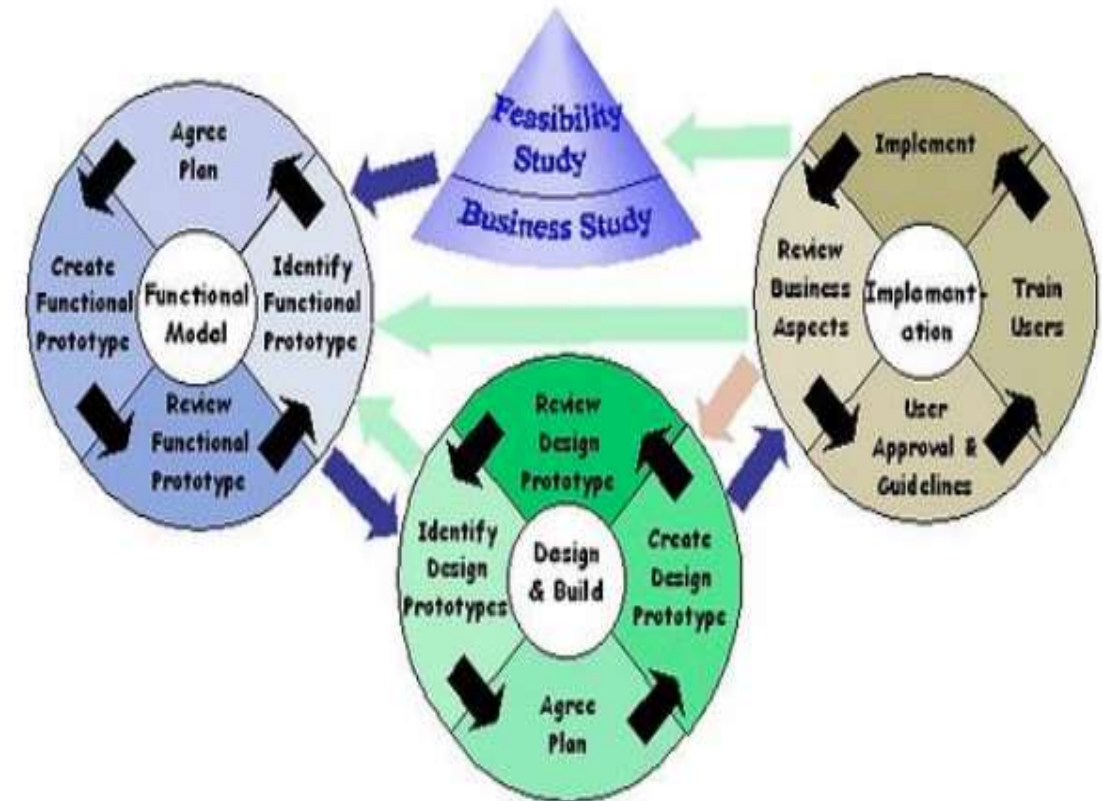


Model Agile DSDM (Dynamic Systems Development Method)



4. Implementation Phase

Setelah berbagai unit test selesai dibangun, tim barulah melanjutkan aktivitasnya ke penulisan coding aplikasi. XP menerapkan konsep pair programming dimana setiap tugas sebuah modul dikembangkan oleh 2 orang programmer



Kelebihan **DSDM** (Dynamic Systems Development Method)



- Metode ini akan membangun software dengan cepat, 80% dari proyek diserahkan dalam 20% dari waktu total untuk menyerahkan proyek secara utuh
- Dynamic System Development Method dapat dikombinasikan dengan eXtreme Programming menghasilkan kombinasi model proses yang mengikuti Dynamic System Development Method dan praktek yang sejalan dengan eXtreme Programming

Kekurangan **DSDM** (Dynamic Systems Development Method)



- Studi Bisnis memberikan dasar untuk semua pengembangn prodak berikutnya
- Fase ini mengarah pada proses bisnis yang terkena dampak secara rinci dan informasi-informasi yang mereka butuhkan



Kesimpulan

Agile memiliki pengertian bersifat cepat, ringan, bebas bergerak. Sehingga saat membuat perangkat lunak dengan menggunakan agile development methods diperlukan inovasi dan responsibility yang baik antara tim pengembang dan klien agar kualitas dari perangkat lunak yang dihasilkan bagus dan kelincahan dari tim seimbang

Agile Manifesto merupakan nilai-nilai yang digunakan dalam mendasari berlangsungnya Agile Software Development

Agile Software Development, berinteraksi dengan personel lebih penting dari pada proses dan alat, software yang berfungsi lebih penting daripada dokumentasi yang lengkap, kolaborasi dengan klien lebih penting dari pada negosiasi kontrak, dan sikap tanggap terhadap perubahan lebih penting daripada mengikuti rencana

Agile Development Methods dikembangkan karena pada metodologi tradisional terdapat banyak hal yang membuat proses pengembangan tidak dapat berhasil dengan baik sesuai tuntutan user.

Metodologi ini sudah cukup banyak berkembang, di antaranya adalah Extreme Programming (XP), Adaptive Software Development (ASD), Dynamic Systems Development Method (DSDM), Scrum Methodology, Crystal, Feature Driven Development (FDD), Agile Modeling (AM)

SCRUM

Tujuan

- Mahasiswa mampu menjelaskan konsep kerangka kerja Scrum yang terdiri dari
 - Peran (Roles)
 - Event
 - Artifacts

Scrum in 100 words

- Scrum adalah sebuah proses Agile yang memungkinkan kita untuk fokus kepada *delivery* dengan nilai bisnis tertinggi dalam waktu yang paling singkat.
- Scrum memungkinkan kita untuk melakukan inspeksi terhadap software yang dibuat secara cepat dan berulang-ulang (setiap 2 minggu – 1 bulan sekali)
- Personel dari sisi bisnis menentukan prioritas. Tim secara mandiri menentukan jalan yang terbaik untuk men-*deliver* fitur-fitur dengan prioritas tertinggi.
- Setiap 2 minggu – 1 bulan sekali, setiap orang bisa melihat hasil kerja berupa software yang bisa digunakan, yang kemudian bisa diputuskan untuk dirilis atau dilanjutkan pengembangannya pada sprint berikutnya.

Scrum Telah Digunakan Oleh:

- Microsoft
- Yahoo
- Google
- Electronic Arts
- IBM
- Lockheed Martin
- Philips
- Siemens
- Nokia
- Capital One
- BBC
- Intuit
- Nielsen Media
- First American Real Estate
- BMC Software
- Ipswitch
- John Deere
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting
- Océ

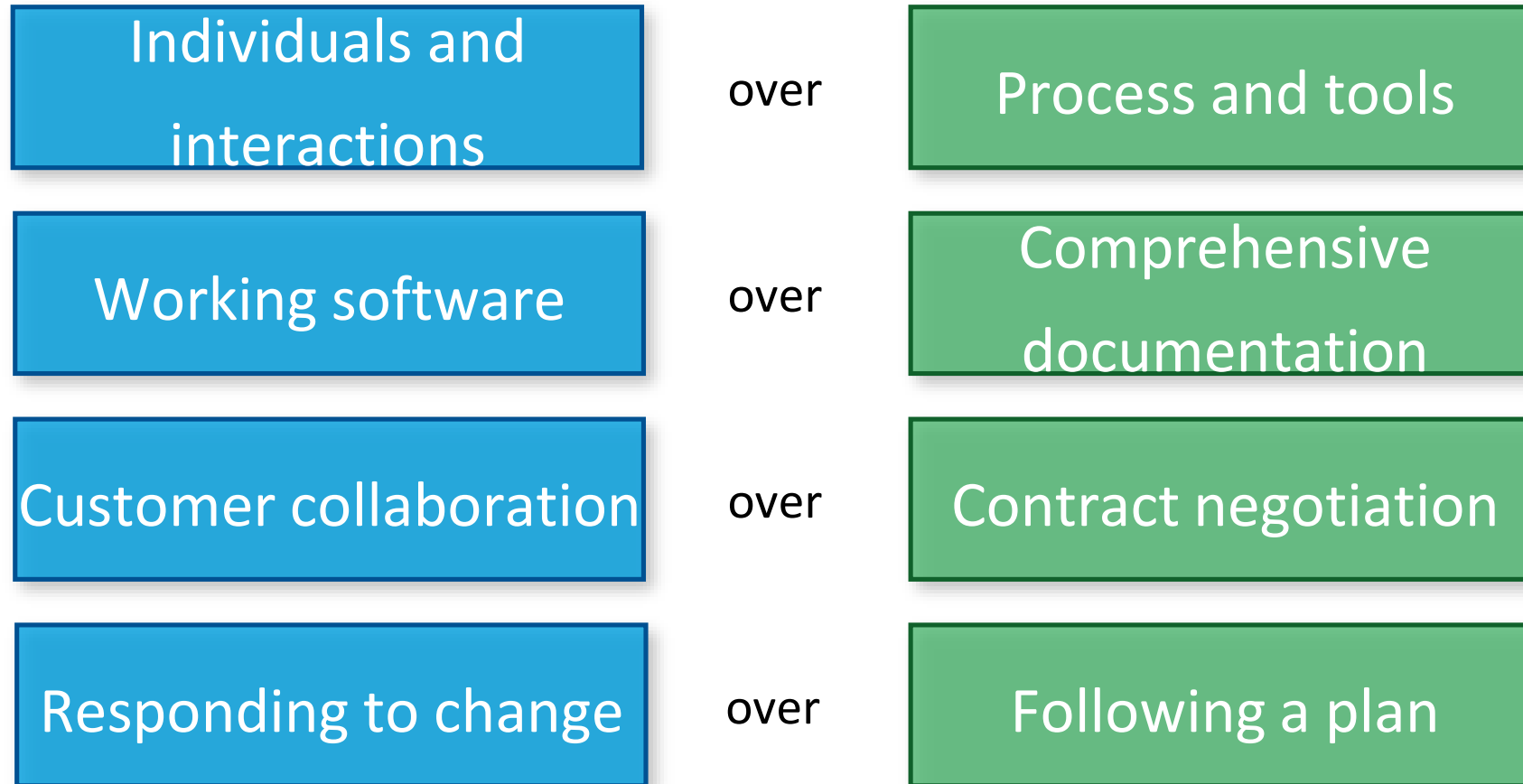
Scrum Telah Digunakan Untuk:

- Software komersial
- In-house development
- Proyek kontrak
- Proyek *fixed-price*
- Aplikasi finansial
- Aplikasi tersertifikasi ISO 9001
- Sistem benam
- Sistem 24x7 dengan 99.999% *uptime*
- Proyek *Joint Strike Fighter*
- Pembuatan *video game*
- FDA-approved, life-critical systems
- Perangkat lunak kendali satelit
- Website
- Handheld software
- Mobile phones
- Network switching applications
- ISV applications
- Beberapa aplikasi-aplikasi besar yang kita kenal

Karakteristik

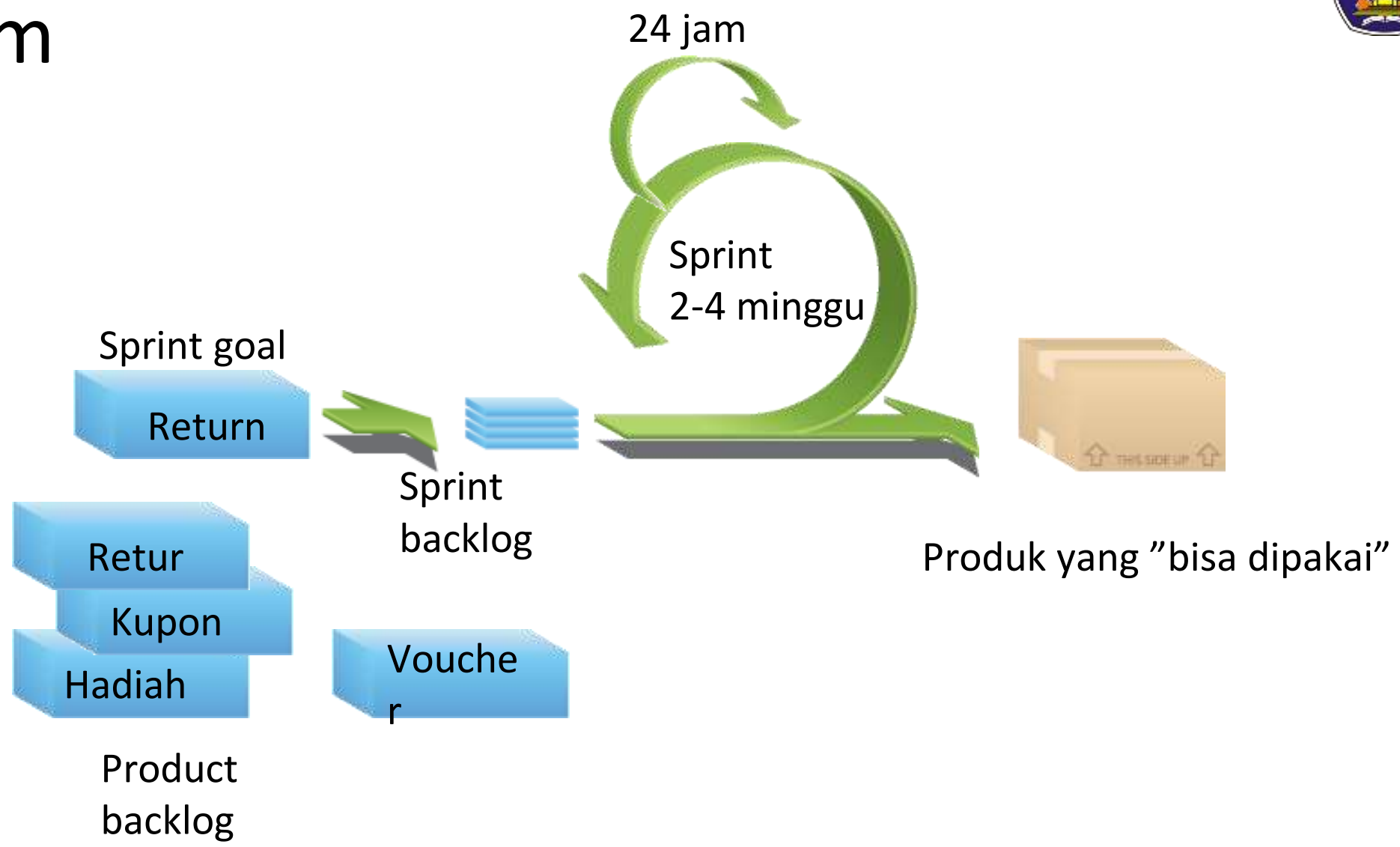
- Tim yang mandiri, *self-organized*
- Produk berproses dalam serangkaian kegiatan berdurasi 1-4 minggu yang disebut dengan “Sprint”.
- Persyaratan sistem dihimpun sebagai item-item dalam sebuah daftar yang disebut sebagai “*Product Backlog*”.
- Sengaja dibuat tidak terlalu rigid, detail dan teknis.
- Menggunakan aturan-aturan generative tertentu untuk menciptakan “lingkungan agile” dalam penyelesaian proyek.
- Merupakan salah satu dari “agile processes”

Manifesto Agile

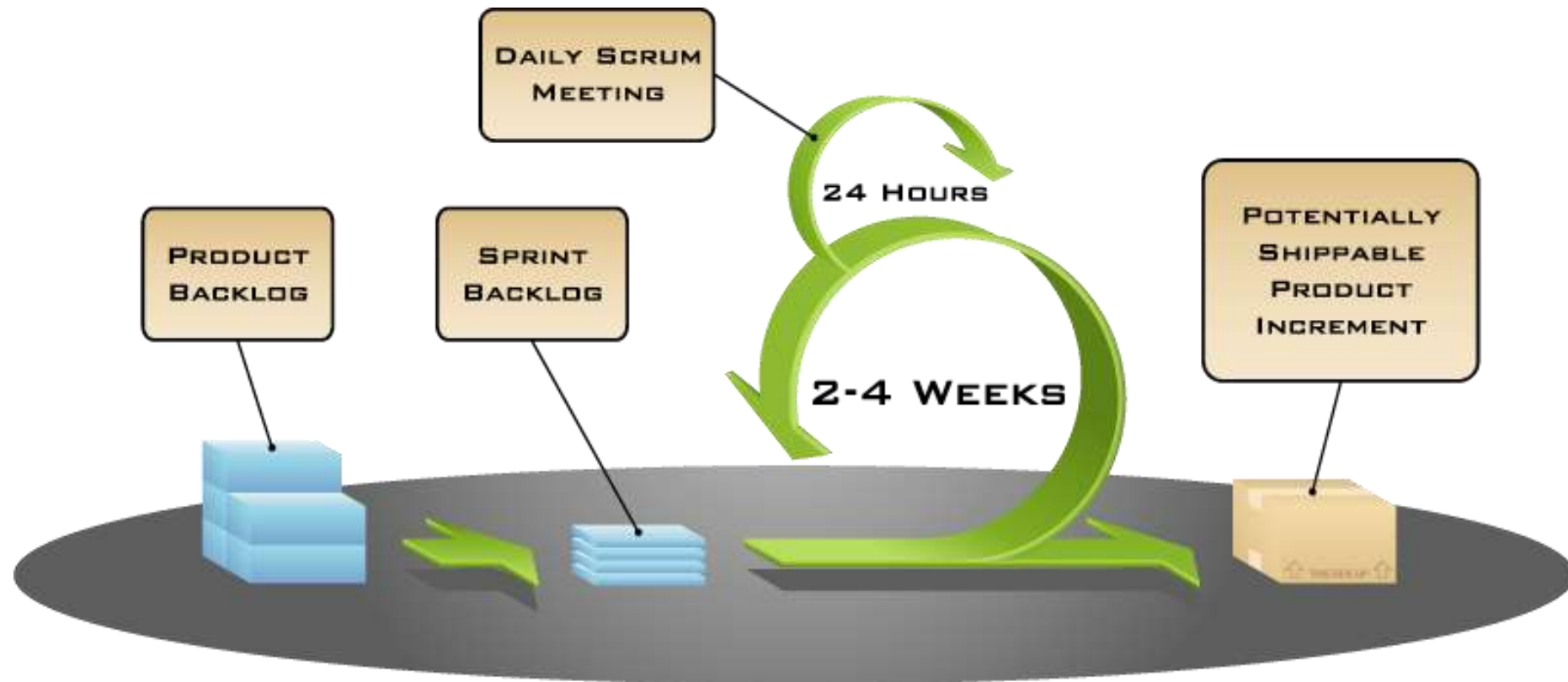


Source: www.agilemanifesto.org

Scrum



Scrum



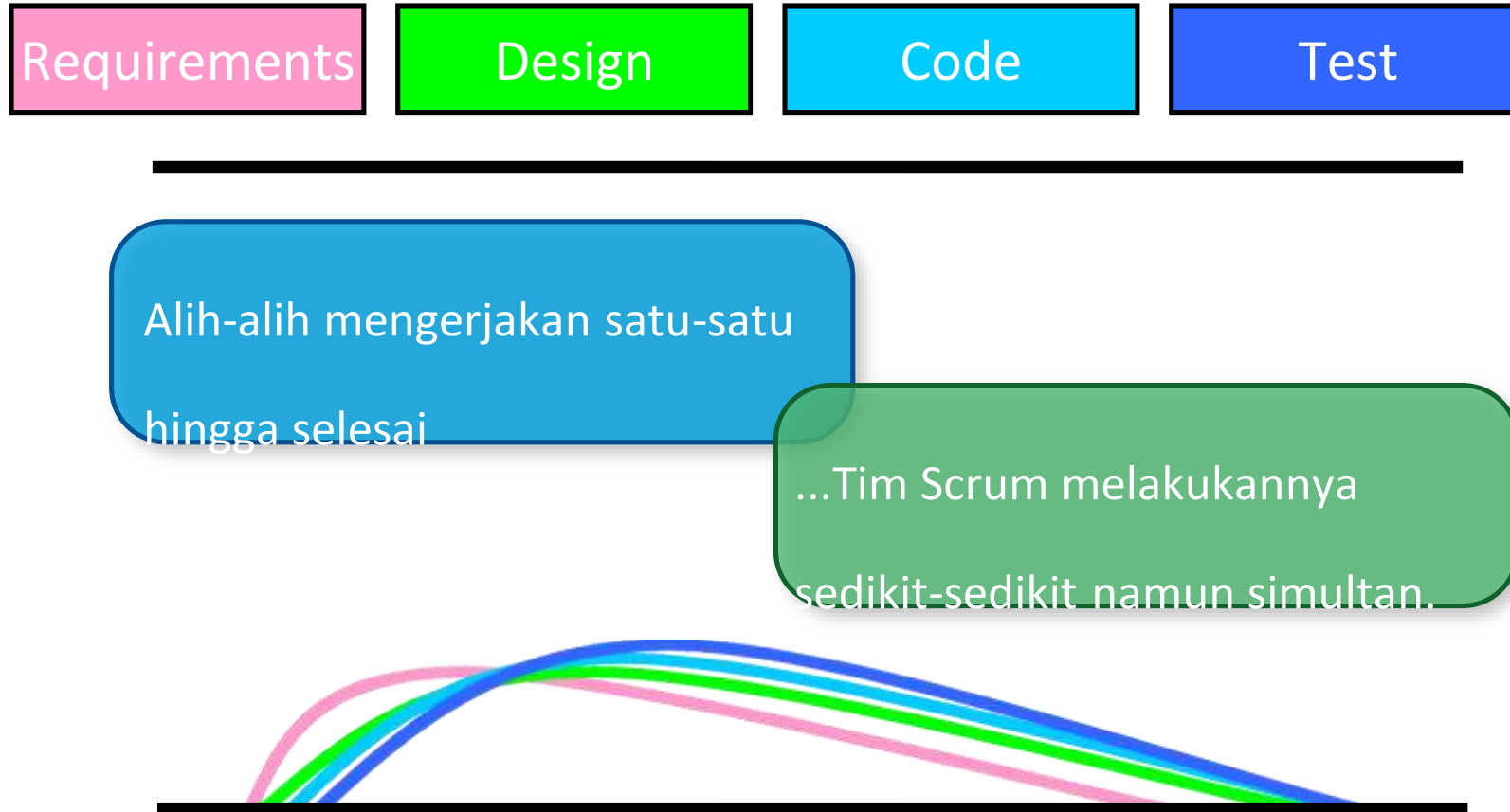
COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Sumber: www.mountaingoatsoftware.com/scrum

Sprints

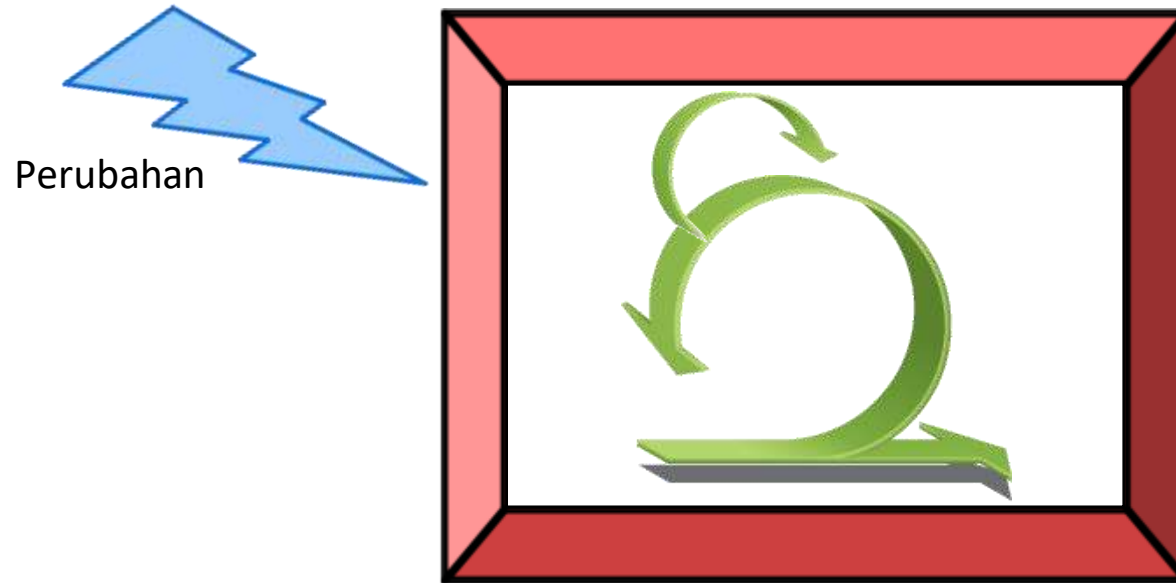
- Progres pada proyek Scrum dilakukan dalam “Sprint”.
 - Atau dikenal dengan “iterasi” jika dalam metode Extreme Programming.
- Durasi pada umumnya adalah 2-4 minggu atau paling lama 1 bulan.
- Durasi yang konsisten akan menghasilkan ritme kerja tim yang baik.
- Perangkat lunak dirancang, dikode, dan dites dalam setiap Sprint.

Sequential vs. overlapping development



Sumber: "The New New Product Development Game" by Takeuchi and Nonaka. *Harvard Business Review*, January 1986.

Tidak boleh ada perubahan ketika Sprint sedang berjalan..



- Durasi Sprint harus direncanakan sesuai dengan kemampuan tim untuk mencegah terjadinya perubahan dalam satu Sprint tersebut.

Kerangka Kerja (Framework) Scrum

Roles

- Product owner
- ScrumMaster
- Team

Events

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Kerangka Kerja (Framework) Scrum

Roles

- Product owner
- ScrumMaster
- Anggota Tim

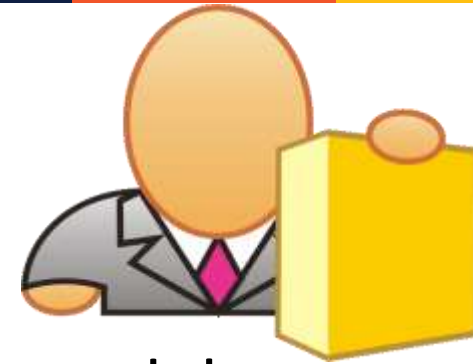
Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

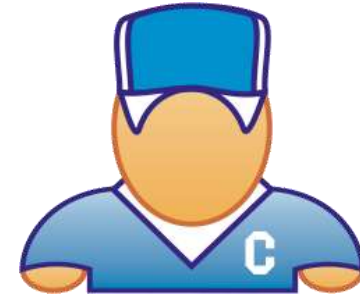
- Product backlog
- Sprint backlog
- Burndown charts

Product owner



- Menentukan fitur apa saja yang harus ada dalam produk yang dikembangkan.
- Memutuskan konten dan waktu rilis.
- Bertanggung jawab terhadap keuntungan dari product (ROI).
- Menentukan prioritas fitur berdasarkan kondisi pasar.
- Mengatur ulang fitur-fitur dan prioritasnya pada setiap iterasi, sesuai kebutuhan.
- Menerima dan/atau menolak hasil kerja tim.

The ScrumMaster



- Berperan sebagai “manajemen” dalam proyek.
- Bertanggung jawab dalam menegakkan nilai-nilai serta praktik-praktik Scrum.
- Menyingkirkan penghalang/hambatan.
- Memastikan tim berfungsi penuh dan produktif.
- Memastikan Kerjasama dan kedekatan pada semua *roles* dan fungsi dalam tim.
- Melindungi tim dari gangguan eksternal.

Anggota Tim

- Umumnya terdiri dari 5-9 orang.
- *Cross-functional*:
 - Programmer, Q.A. Engineer, Pengguna, Udesainer UI/UX, dst.
- Semua anggota harus bekerja penuh waktu.
 - Walau terkadang ada pengecualian (Contoh: database administrator)
- Tim bersifat mandiri, *self-organizing*
 - Idealnya tidak ada jabatan/titel, tapi kadang (sangat jarang) bisa ada.
- Keanggotaan tidak boleh berubah selama Sprint masih berlangsung.
 - Bisa diganti ketika ada jeda diantara dua Sprint.



Kerangka Kerja (Framework) Scrum

Roles

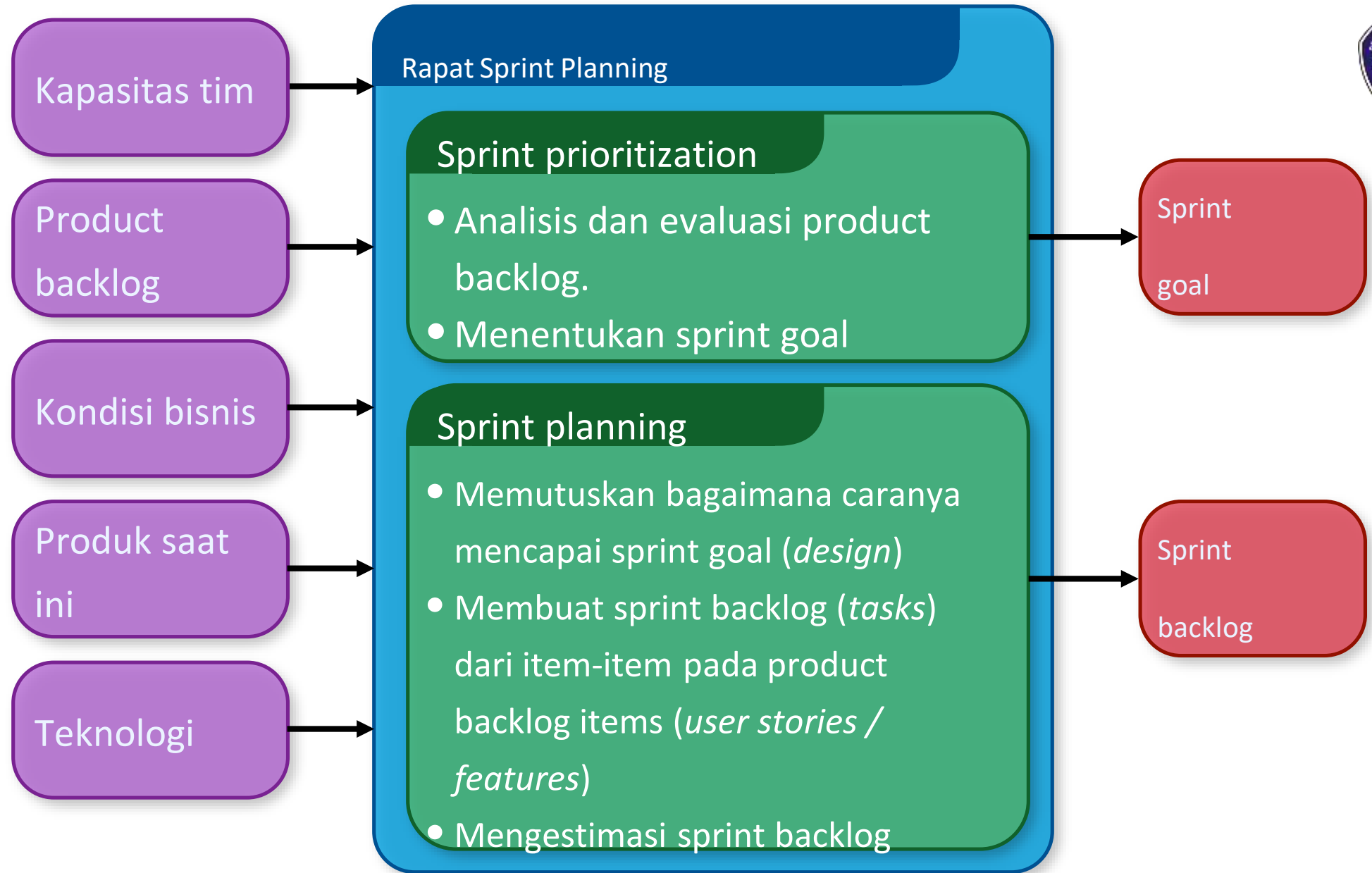
- Product owner
- ScrumMaster
- Team

Events

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts



dalam jam kerja.

Sprint planning

- Tim memilih item dari product backlog yang mereka yakin bisa menyelesaikan dalam rentang waktu 1 Sprint.
- Membuat Sprint Backlog:
 - Setiap task yang ada diidentifikasi dan diperkirakan akan membutuhkan waktu berapa lama dalam jam (1-16 jam)
 - Dilakukan secara kolaboratif, tidak sendirian oleh Scrum Master saja.
- Pertimbangan yang digunakan adalah "high level design" atau "user story".

Sebagai seorang yang akan pergi liburan, saya ingin melihat foto-foto dari hotel yang tersedia.



Koding middle tier (8 jam)

Mengerjakan user interface (4 jam)

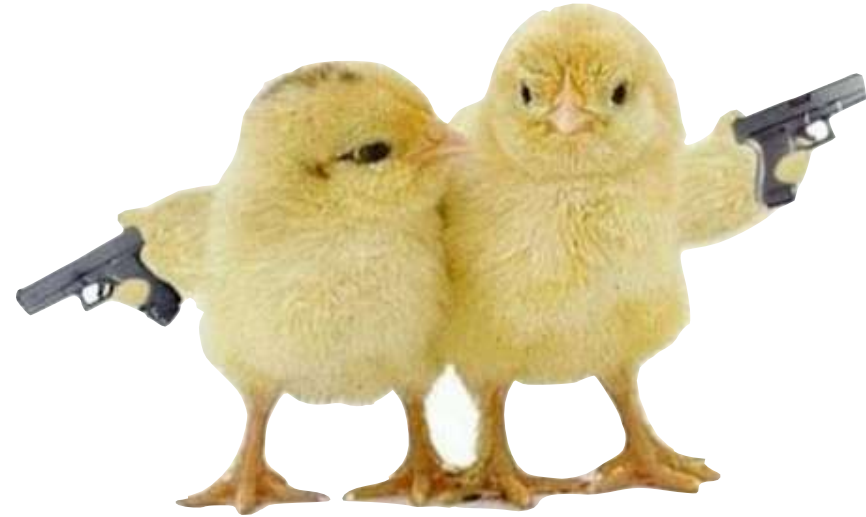
Membuat test fixtures (4 jam)

Membuat the class-class utama (6 jam)

Memperbarui performance tests (4 jam)

Event #1: Daily scrum

- Sifat:
 - Harian
 - Maksimal 15-menit harus selesai
 - Berdiri (No, seriously. Yang giliran ngomong ga boleh sambil duduk)
- Bukan untuk “menyelesaikan masalah”
 - Semuanya diundang.
 - Tapi yang boleh berbicara hanya: Anggota Tim, ScrumMaster, dan Product Owner
- Membantu agar terhindar dari rapat-rapat yang berkepanjangan dan kesana-kemari.





Dalam Daily Scrums, setiap orang menceritakan:

1
Kemarin ngapain aja?

2
Hari ini mau ngapain?

3
Apa ada *blockers*?

- Di sini semua anggota tim harus jujur. Bukan cuma pencitraan kalau seolah-olah kemarin kerja padahal cuma *youtube*-an.
 - Setiap orang berkomitmen terhadap semua rekan-rekannya.

Event #2: Sprint Review

- Tim mempresentasikan apa yang telah dicapai di akhir satu Sprint yang baru saja selesai.
- Pada umumnya dalam bentuk mendemokan fitur baru aplikasi yang dibuat atau menjelaskan arsitektur di dalam aplikasi tersebut.
- Informal
 - Maskimum 4 jam harus selesai.
 - Tidak pakai PPT.
- Semua tim ikut.
- Undang semua orang dan/atau pihak terkait.



Event #4: Sprint retrospective

- Secara periodik meninjau apa yang bekerja dan apa yang tidak bekerja dengan baik.
- Maskimal 3 jam jika sprint-nya 1 bulan, atau lebih pendek.
- Dilakukan setiap kali selesai Sprint.
- Semua orang ikut serta:
 - ScrumMaster
 - Product owner
 - Anggota Tim
 - Bisa jadi klien kita dan yang lainnya yang perlu/berkepentingan.



Dalam Sprint Retrospective, dibahas tentang:

Start / Stop / Continue

- Seluruh tim berkumpul dan mendiskusikan apa yang ingin:

Mulai dilakukan

Berhenti dilakukan

Tidak hanya terkait teknis, tapi bisa apa saja. Asalkan semua sepakat. Contoh: "Bu, kami butuh cemilan"

Terus dilakukan

Kerangka Kerja (Framework) Scrum

Roles

- Product owner
- ScrumMaster
- Team

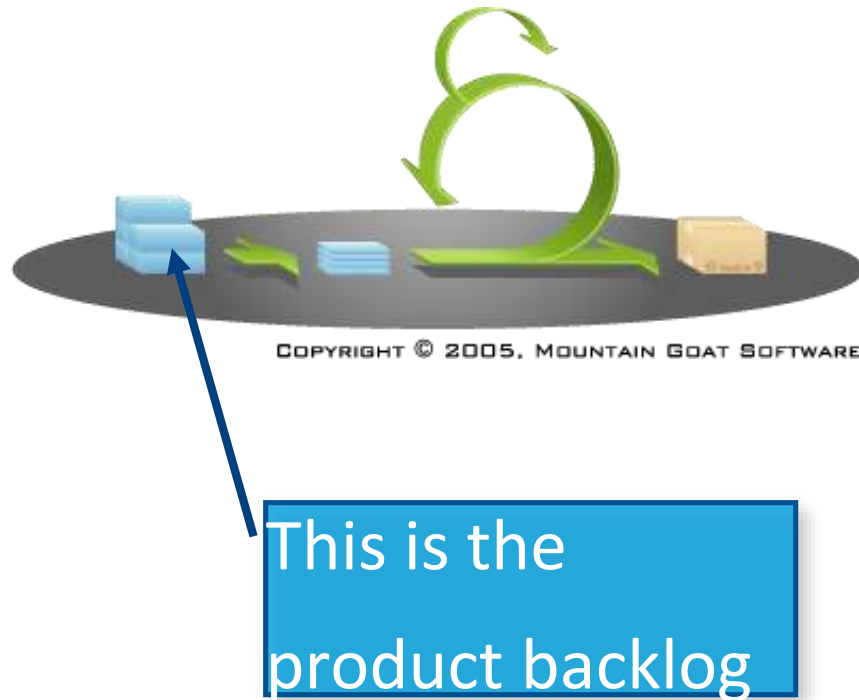
Event

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Product backlog



- Persyaratan Sistem.
- Daftar pekerjaan yang ingin dicapai pada suatu proyek.
- Secara ideal diekspresikan sedemikian sehingga setiap item memiliki *value* yang nyata bagi *end-user*.
- Prioritas diatur oleh Product Owner.
- Bisa diatur ulang prioritasnya di awal tiap sprint.

Contoh Product Backlog

Item Backlog	Estimasi
Allow a guest to make a reservation. Fitur untuk tamu agar bisa membuat reservasi.	3
Sebagai seorang tamu, saya ingin membatalkan pemesanan.	5
Sebagai seorang tamu, saya ingin mengubah tanggal pemesanan.	3
Sebagai seorang karyawan hotel saya ingin bisa membuat laporan pendapatan per kamar.	8
Memperbaiki <i>error</i> dan <i>exception</i> .	8
...	30
...	50



Sprint Goal

- Sebuah pernyataan pendek yang menegaskan pekerjaan kita akan fokus kemana pada saat berlangsungnya suatu Sprint.

Aplikasi Database

Membuat aplikasi kita bisa berjalan di SQL Server dan Oracle

Life Sciences

Menyediakan fitur-fitur yang dibutuhkan untuk studi populasi genetic.

Aplikasi Fintek

Membantu menyediakan indikator-indikator teknis secara real-time dengan streaming data.

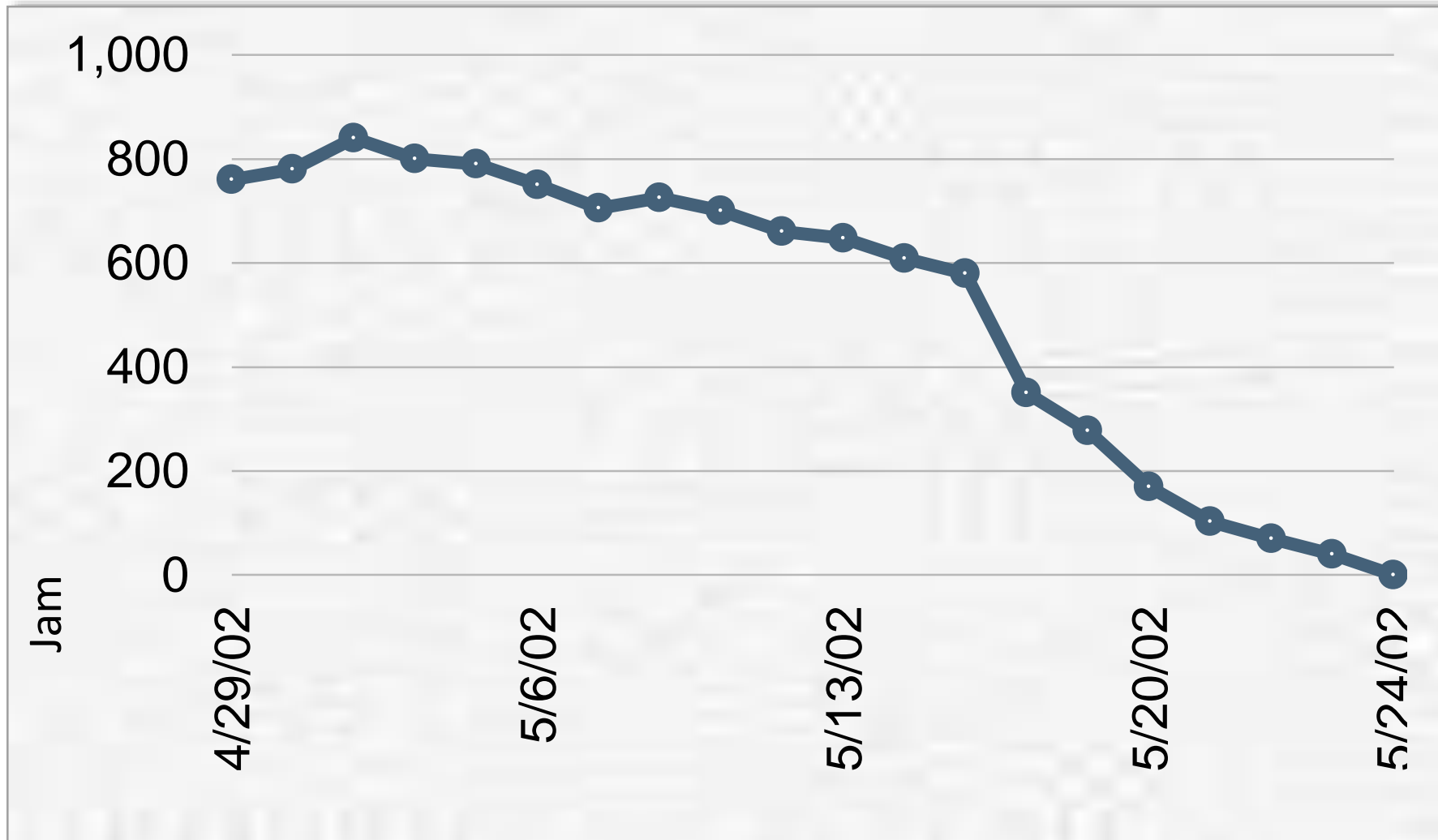
Mengelola Sprint Backlog

- Setiap orang memilih sendiri backlog mana yang ingin dia kerjakan.
 - Tidak ada penugasan.
- Estimasi pekerjaan yang belum selesai diperbarui setiap hari.
- Setiap anggota tim boleh menambah, menghapus, atau mengubah Sprint Backlog.
- Mulai dan tetap bekerja selama waktu Sprint berjalan.
- Jika ada pekerjaan yang kurang jelas, buatlah menjadi satu item Sprint Backlog dengan perkiraan waktu yang dilebihkan untuk di-breakdown nanti.
- Perbarui sisa pekerjaan yang belum selesai bebarengan dengan bertambah jelasnya keseluruhan pekerjaan.

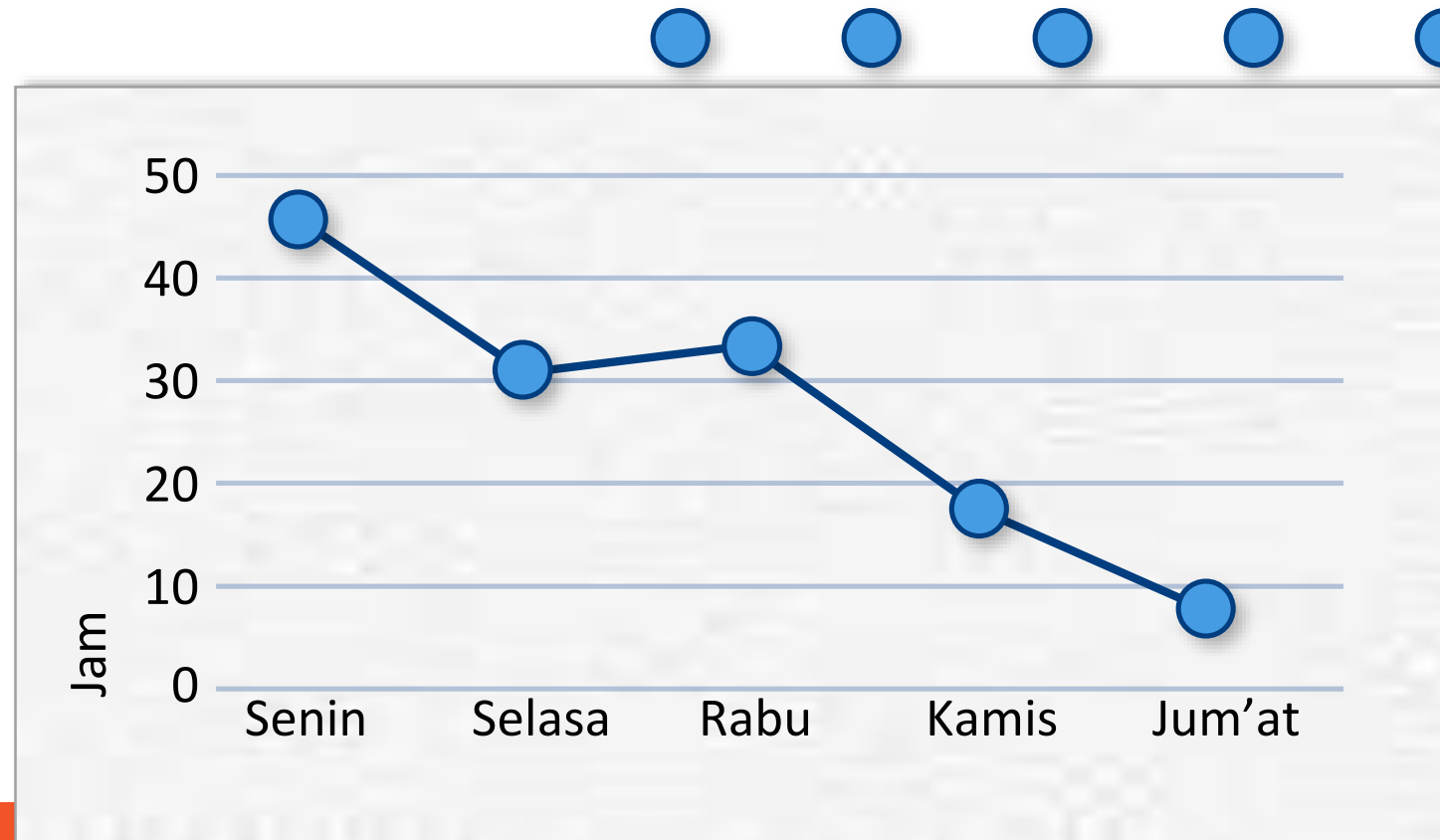
Contoh Sprint Backlog

Pekerjaan	Senin	Selasa	Rabu	Kamis	Jumat
Koding UI	8	4	8		
Koding middleware	16	12	10	4	
Tes middleware	8	16	16	11	8
Bikin menu help	12				
Buat class-class utama	8	8	8	8	8
Tambahin log error			8	4	

Contoh Diagram "Sprint Burndown"



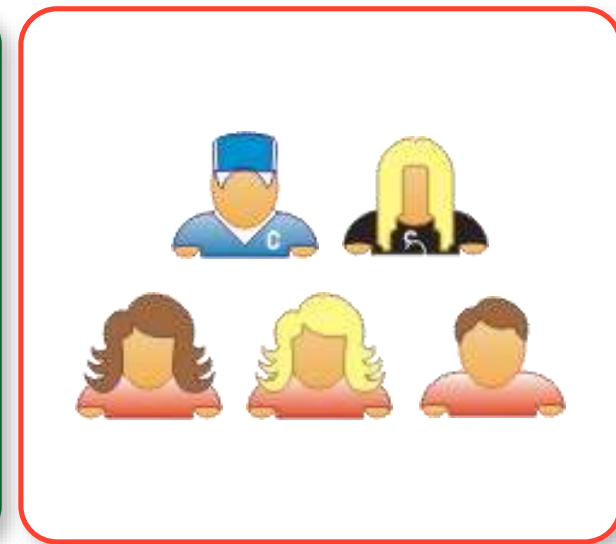
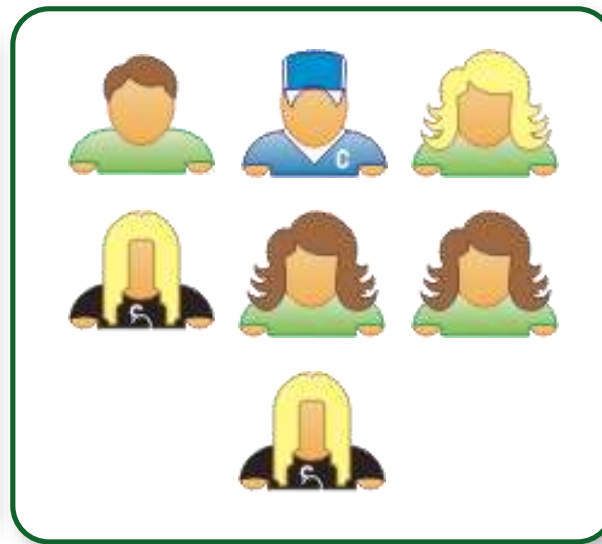
Pekerjaan	Sen	Sel	Rab	Kam	Jum
Koding UI	8	4	8		
Koding middleware	16	12	10	7	
Ngetes middleware	8	16	16	11	8
Bikin menu Help	12				



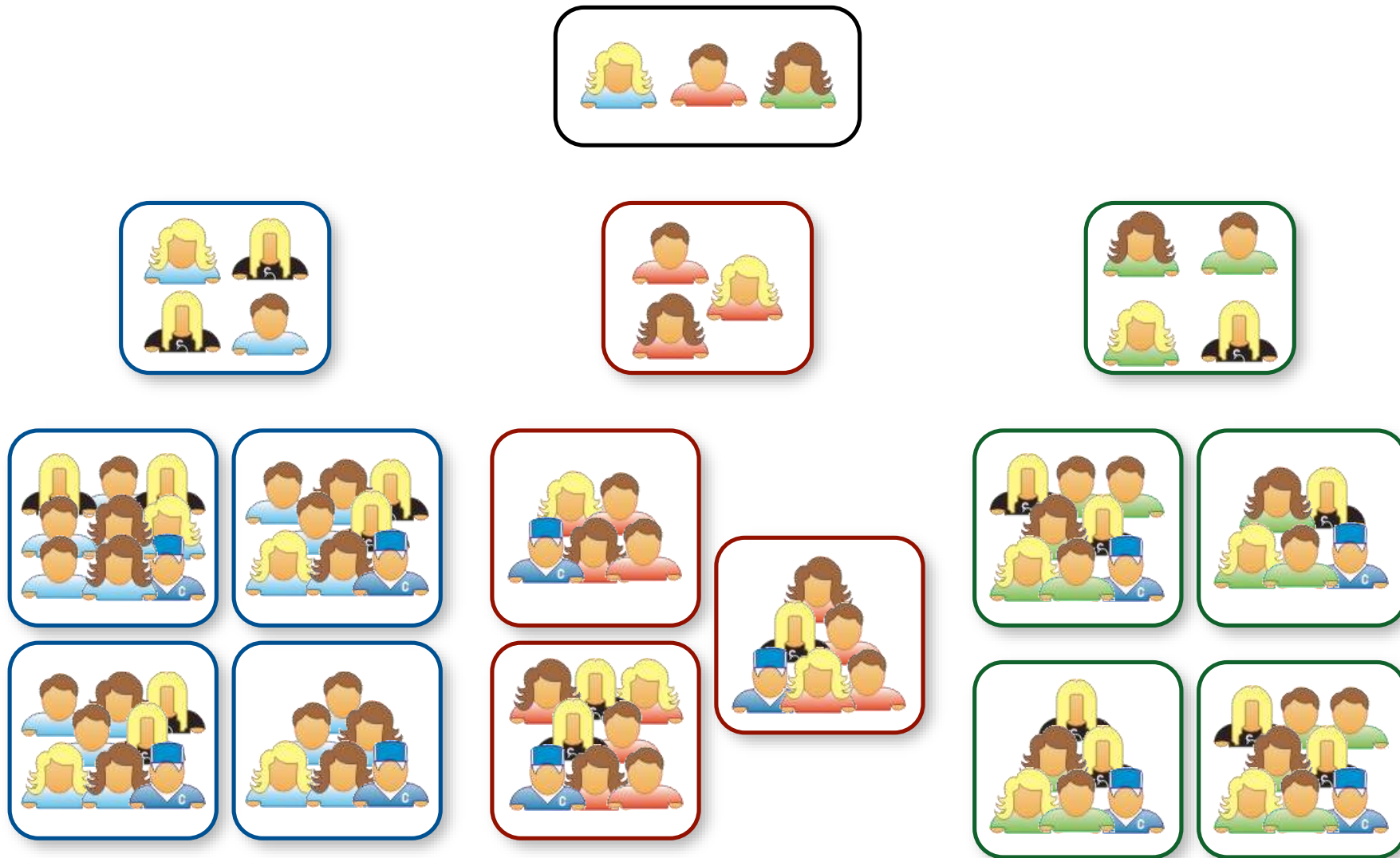
Skalabilitas

- Umumnya setiap tim beranggotakan 7 ± 2 orang
 - Skalabilitas bisa didapatkan dengan pengaturan “tim dalam tim”
- Faktor skalabilitas
 - Jenis aplikasi yang dikerjakan
 - Ukuran tim
 - Persebaran tim
 - Durasi proyek
- Scrum telah terbukti, dan bisa digunakan dalam proyek-proyek beranggotakan 500+ orang.

Scaling through the Scrum of scrums



Scrum of scrums of scrums





Pertanyaan?

Terima Kasih.. :)

Referensi

- Agile Estimating and Planning by Mike Cohn
- Agile Product Management: Creating Products that Customers Love by Roman Pichler
- Agile Project Management with Scrum by Ken Schwaber
- Agile Software Development Ecosystems by Jim Highsmith
- Essential Scrum: A Practical Guide to the Most Popular Agile Process by Kenneth Rubin
- Scrum and XP from the Trenches by Henrik Kniberg
- Succeeding with Agile: Software Development using Scrum by Mike Cohn
- The Scrum Guide at www.ScrumGuides.org
- User Stories Applied for Agile Software Development by Mike Cohn



Any questions?