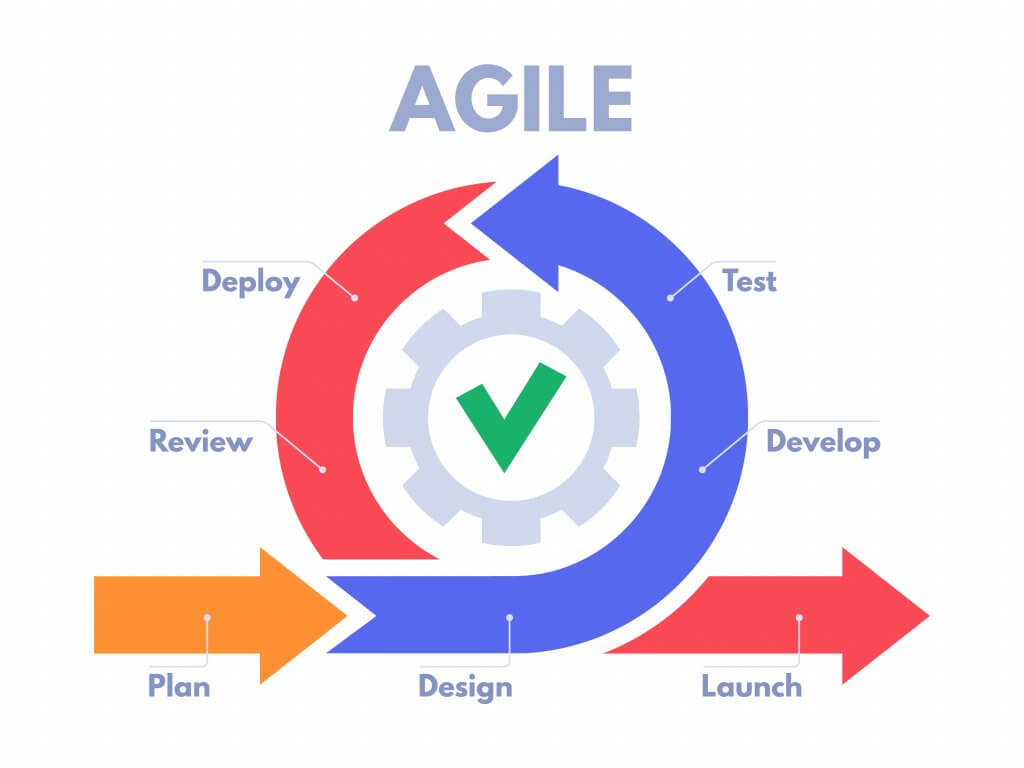
Nama : Dahlia Maya Sofa

Kelas : 4617

3 Model Software Engineering :

1. **Agile**
2. **Diagram**



Agile artinya cara pengembangan produk sebagai alternatif dari metode konvensional. Dengan kata lain, pengertian agile adalah dobrakan terhadap alur kerja lama yang tidak berkembang ketika melihat dinamika pasar.

Berawal dari kekakuan model waterfall, metode agile adalah cara efektif dalam memahami keinginan pasar. Fleksibilitas metode agile terlihat pada rangkaian proses yang dapat disesuaikan saat produksi tengah berjalan.

Pada metode waterfall, kita harus berkutat pada rencana awal tanpa improvisasi, sedangkan metode agile mengakomodasi komunikasi antara pemakai dan produsen bahkan setelah perancangan software telah terjadi. Sehingga pemakai merasa lebih puas dengan adanya perbaikan saat pengerjaan.

Urgensi pemanfaatan agile semakin besar terkait fakta bahwa banyak software dengan metode waterfall yang tidak lagi dibutuhkan karena kurang relevan.

**Tujuan Menerapkan Agile :**

1. Percepatan produk siap pakai

Membandingkan dengan metode waterfall, agile memungkinkan pemakai untuk mengetahui langkah-langkah produksi yang sedang dilakukan. Pada proses produksi, konsumen dapat meminta perubahan sehingga produk akhir lebih cepat terselesaikan. Hal tersebut pun akan membuat segalanya menjadi lebih efisien.

2. Efisien waktu

Agile adalah metode dengan pemanfaatan sumber daya secara maksimal. Tim-tim kecil yang telah dibentuk berdasarkan proyek tertentu tidak harus menunggu tim lain untuk menyelesaikan tugas mereka terlebih dahulu.

Sebab sangat berbeda dengan metode konvensional, agile adalah perbaikan dari keadaan kerja tidak efisien di mana tim satu dengan tim lain tidak dapat berjalan secara paralel.

3. Memastikan produk tetap relevan

Relevansi produk harus tetap dijaga melalui umpan balik yang diberikan oleh user. Sempat disinggung di awal, software dengan metode waterfall berkurang relevansinya ketika dipasarkan sebab pada tahapan produksi, tim tidak melakukan komunikasi dengan pihak luar.

Kekurangan dapat diatasi dengan mendengarkan pasar dan bersikap fleksibel terhadap perubahan yang datang darinya. Keluwesan membaca pasar menjadi hal pokok jika perusahaan ingin produknya tetap sesuai dengan kebutuhan user.

1. **Deskripsi fase**

Pengembangan perangkat lunak Agile memiliki beberapa tahapan dalam mengembangkan suatu proyek/perangkat lunak. Tahapan-tahapan tersebut adalah sebagai berikut.

1. Perencanaan, pada langkah ini pengembang dan klien membuat rencana tentang kebutuhan dari perangkat lunak yang akan dibuat.
2. Implementasi, bagian dari proses dimana programmer melakukan pengkodean perangkat lunak.
3. Tes perangkat lunak, disini perangkat lunak yang telah dibuat di tes oleh bagian kontrol kualitas agar bug yang ditemukan bisa segera diperbaiki dan kualitas perangkat lunak terjaga.
4. Dokumentasi, setelah dilakukan tes perangkat lunak langkah selanjutnya yaitu proses dokumentasi perangkat lunak untuk mempermudah proses maintenanance kedepannya.
5. Deployment, yaitu proses yang dilakukan oleh penjamin kualitas untuk menguji kualitas sistem. Setelah sistem memenuhi syarat maka perangkat lunak siap dideployment.
6. Pemeliharaan, langkah terakhir yaitu pemeliharaan. Tidak ada perangkat lunak yang 100% bebas dari bug, oleh karena itu sangatlah penting agar perangkat lunak dipelihara secara berkala.
7. **Kelebihan dan Kekurangan**

~Kelebihan Agile :

1. Kualitas Software Lebih Baik

Dengan Agile Methodology, Anda bisa dengan cepat menerapkan setiap feedback dari konsumen di iterasi selanjutnya. Baik itu tentang penambahan fitur atau memperbaiki bug. Alhasil, kualitas software Anda akan membaik karena lebih sesuai dengan keinginan konsumen.

2. Konsumen Puas dan Lebih Dihargai

Software dengan kualitas yang baik akan disukai konsumen. Apalagi kalau fitur software di dalamnya sesuai dengan keinginan konsumen berkat feedback yang diberikan. Dengan begitu, mereka akan merasa lebih dihargai sebagai konsumen.

3. Fleksibilitas Tinggi

Metode Agile memungkinkan Anda melakukan perubahan rutin sesuai feedback konsumen pada software yang dikerjakan. Bila ada banyak feedback, Anda bisa memilih akan melakukan perubahan apa di setiap iterasinya. Jadi, Anda tidak harus saklek mengikuti sebuah rencana dari awal sampai akhir.

4. Software Cepat Selesai

Metode Agile berfokus mengembangkan software yang benar-benar dibutuhkan konsumen. Jadi, software akan cepat selesai karena Anda tak menghabiskan waktu menambahkan banyak fitur yang belum tentu dibutuhkan konsumen.

5. Pengembangan yang Lebih Terprediksi

Setiap Agile methodology mempunyai iterasi rutin yang fokus pada pengembangan bertahap. Jadi, pengembangan software Anda akan lebih terprediksi, sehingga Anda bisa tahu berbagai pengeluaran dengan baik. Efeknya, risiko bisnis akan berkurang.

~Kekurangan Agile :

1. Produk Akhir yang Kurang Jelas

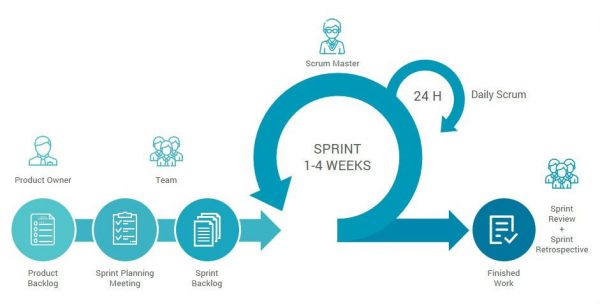
Agile tidak membutuhkan banyak perencanaan, sehingga tim Anda akan kesulitan menentukan bentuk akhir software-nya. Apalagi, fitur baru juga selalu ditambahkan dan membuat software Anda semakin kompleks. Efeknya, selama kebutuhan konsumen berubah-ubah, Anda juga akan semakin sulit menentukan bentuk akhirnya.

2. Bergantung Pada Komitmen Tinggi Tim

Agile mengharuskan semua pihak untuk selalu berinteraksi secara rutin agar software sesuai kebutuhan konsumen. Mulai dari memberikan feedback, menerapkan perubahan, melakukan test, dan sebagainya. Ini tentu saja membutuhkan komitmen tinggi dari setiap individu, sehingga akan menguras banyak energi dan waktu Anda.

3. Dokumentasi yang Kurang Lengkap

Dalam agile, dokumentasi dibuat dalam waktu yang singkat sehingga berujung pada hasil yang kurang lengkap. Artinya, saat ada anggota baru di tim, ia akan kesulitan memahami berbagai hal tentang project yang berjalan. Efeknya, akan timbul miskomunikasi yang bisa memperlambat project Anda.

1. **Scrum**
2. **Diagram**

**Scrum** adalah salah satu kerangka kerja dari metode Agile yang berfokus pada pengembangan produk bersifat kompleks dan dapat berubah sesuai kebutuhan customer.

Selama project berjalan, customer atau calon pengguna produk juga terlibat untuk memberikan feedback dan koreksi. Oleh sebab itu, proses agile scrum adalah bersifat iteratif. Artinya, di tengah proses, Anda bisa melakukan testing terlebih dulu dan menyesuaikan ulang fokus project.

Pengerjaan project dalam scrum biasanya dibagi menjadi beberapa proses kecil sesuai fungsi produk. Misalnya, pembuatan fitur transfer pada dompet digital atau lainnya.

1. **Deskripsi fase**

1. Product backlog

Tahapan yang pertama ini menjadi tanggung jawab product owner atau manajer. Secara sederhana, tahapan ini berisi daftar apa saja yang harus tim lakukan sesuai dengan skala prioritas perusahaan.

Perwakilan perusahaan bersama representasi stakeholder bertugas untuk terus menata ulang product backlog. Hal ini karena kondisi pasar akan selalu berubah, sehingga apa yang tidak esensial dalam backlog juga harus dihilangkan.

2. Sprint planning

Dalam tahapan sprint planning, produk atau proyek teratas dalam product backlog kemudian disusun kembali menjadi sprint backlog. Tugas lain yang dilakukan tim scrum adalah menentukan bagaimana mereka akan menyelesaikan apa yang ada dalam sprint backlog.

3. Sprint

Setelah cara dan batas waktu pengerjaan ditentukan dalam sprint planning, tahapan metode scrum selanjutnya adalah melakukan sprint. Proses ini dilakukan melalui kegiatan daily scrum.

Daily scrum adalah saat dimana tim berkumpul dan bekerja untuk memastikan perkembangan produk terus berjalan. Dalam proses ini juga dikenal adanya peran scrum master.

Umumnya, tim yang terlibat dalam proses scrum adalah sekelompok orang. Di antara orang-orang tersebut, ada perwakilan yang berperan sebagai scrum master. Tugas scrum master adalah untuk membantu tim agar tetap fokus.

4. Sprint Review

Sprint review adalah tahapan sprint yang terakhir. Dalam proses ini, produk atau projek yang dikerjakan harus sudah selesai dan siap digunakan. Kemudian produk tersebut akan direview Kembali.

5. Retrospective Process

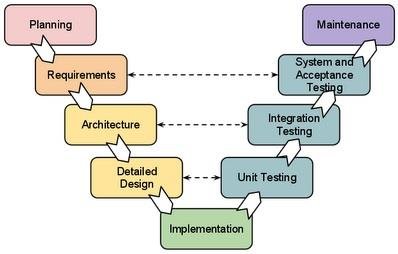
Scrum adalah metode yang sifatnya berulang. Proses yang dilakukan perusahaan mulai dari pembuatan backlog hingga review diingat dan dilakukan kembali dalam proses scrum projek selanjutnya.

1. **Kelebihan dan Kekurangan**

~Kelebihan :

1. SCRUM dapat membantu perusahaan Anda dalam menghemat waktu dan biaya (dalam hal ini uang). Biaya overhead dari proses dan manajemen sangat minim sehingga dapat mengarahkan kita kepada hasil yang lebih cepat dan lebih murah.
2. Dengan menggunakan metode SCRUM, Anda dapat mentransformasikan bisnis yang sulit untuk diukur menjadi mudah untuk dikembangkan.
3. Pada metode SCRUM, pergerakan pengembangan cutting edge dapat dengan cepat dikodekan dan diuji menggunakan metode ini. Bagaikan kesalahan yang mudah untuk diperbaiki.
4. Dengan menggunakan SCRUM, Anda dapat mengontrol dan memonitoring aktivitas peningkatan dan penurunan beban pekerjaan yang bisa terjadi kapan saja.
5. Seperti metodologi agile pada umumnya, SCRUM merupakan metode iterative yang membutuhkan feedback secara berkelanjutan dari user atau pengguna.
6. Dengan adanya short sprint dan constant feedback, SCRUM dapat dengan mudah mengatasi setiap perubahan yang terjadi.
7. Dengan adanya daily scrum meeting, memungkinan SCRUM untuk mengukur produktvitas individu, hal ini mengarah pada peningkatan produktivitas dari setiap anggota tim.
8. Dengan SCRUM, setiap ada masalah yang timbul dapat di identifikasi dengan baik pada pertemuan harian dan oleh karena itu setiap masalah dapat di selesaikan dengan cepat.
9. Dengan menggunakan metode SCRUM, Anda dapat dengan mudah untuk mengirim produk berkualitas sesuai dengan waktunya.
10. SCRUM dapat bekerja dengan berbagai teknologi dan bahasa pemrograman. Namun secara khusus berguna untuk pengembangan proyek dengan teknologi web 2.0 ataupun media proyek baru lainnya.

~Kekurangan :

1. SCRUM bisa menjadi salah satu penyebab utama terjadinya scope creep, kecuali ada tanggal akhir tertentu. Stakeholder proyek atau manajemen akan terus menuntut fungsi dan fitur baru untuk disampaikan.
2. Setiap tugas harus didefinisikan dengan baik, karena hal ini dapat mempengaruhi perkiraan biaya dan waktu pengerjaan proyek. Jika tidak didefinisikan dengan baik maka semua hal tersebut tidak akan akurat. Dalam kasus seperti ini, biasanya tugas dapat tersebar di beberapa sprint.
3. Jika anggota tim Anda tidak berkomitmen dengan baik, maka proyek Anda tidak akan selesai atau bahkan bisa gagal.
4. Metode SCRUM ini hanya membutuhkan anggota tim yang sudah berpengalaman, jika tim Anda berisi orang-orang yang masih pemula maka proyek tidak dapat selesai sesuai dengan waktunya.
5. SCRUM dapat bekerja dengan baik jika seorang Scrum Master dapat mempercayai tim yang mereka kelola. Jika Scrum Master terlalu mengontrol secara ketat, hal ini dapat menyebabkan tim menjadi tertekan dan stress, sehingga mengakibatkan demoralisasi dan kegagalan dari proyek tersebut.
6. Jika sering terjadi pergantian anggota tim saat pengembangan proyek berlangsung, hal ini dapat menyebabkan efek yang kurang baik bagi perkembangan proyek tersebut, proyek akan semakin lama selesai dari waktunya.
7. **V-Model**
8. **Diagram**

V-Model adalah sebuah model SDLC yang eksekusi per fasenya dilakukan secara sekuensial dalam bentuk V. V-Model adalah sebuah perpanjangan dari model waterfall dan didesain berdasarkan hubungan antar tahap pengujian antar fase. Ini berarti bahwa untuk setiap fase dalam satu siklus, ada tahap pengujian yang terkait langsung. Seperti halnya model waterfall, model ini merupakan model yang menerapkan tingkat kedisiplinan yang sangat tinggi di mana fase selanjutnya tidak dapat dimulai sebelum fase sebelumnya selesai.

Di dalam V-Model, tahapan pengujian di dalam satu siklus dilakukan secara paralel. Jadi di dalam V-Model terdapat fase Verifikasi di satu sisi dan fase Validasi di sisi lainnya. Fase Koding (Implementation) menggabungkan kedua sisi dari V-Model ini.

1. **Deskripsi Fase**
2. Requirement Analysis

Seperti halnya pada Model Firewall, ini adalah fase pertama dalam siklus di mana semua informasi kebutuhan pengembangan informasi digali dari customer. Di dalam fase ini melibatkan proses komunikasi yang intensif dengan customer untuk memahami kebutuhan sistem dengan tepat. Ini adalah fase yang sangat penting dan perlu dikelola dengan baik, karena sebagian besar customer tidak yakin tentang apa yang sebenarnya mereka butuhkan. Perencanaan Acceptance Test Design dilakukan pada tahap ini karena requirement dapat digunakan sebagai input untuk pengujian accepatence.

1. System Design

Setelah Anda memiliki requirement produk yang jelas dan terperinci, sekarang saatnya untuk merancang sistem dengan lengkap. Desain sistem yang lengkap akan memiliki perincian perangkat keras dan pengaturan komunikasi yang lengkap untuk sistem yang sedang dikembangkan. Rencana system test dikembangkan berdasarkan desain sistem.

1. Architectural Design

Spesifikasi arsitektur dipahami dan dirancang dalam fase ini. Biasanya, lebih dari satu pendekatan teknis diusulkan dan keputusan akhir diambil berdasarkan kelayakan teknis dan finansial. Desain sistem dipecah menjadi lebih kecil untuk menjadi modul yang memiliki fungsi yang berbeda – beda.

Transfer data dan komunikasi antara modul secara internal dan secara eksternal (dengan sistem lain) dipahami dengan jelas dan didefinisikan dalam fase ini. Berdasarkan informasi tersebut, integration test dapat dirancang dan didokumentasikan selama tahap ini.

1. Module Design

Dalam fase ini, desain internal terperinci untuk semua modul sistem dispesifikasikan. Penting bahwa desain harus kompatibel dengan modul lain dalam arsitektur sistem dan sistem eksternal lainnya. Unit Test merupakan bagian penting dari setiap proses pengembangan dan membantu menghilangkan kesalahan dan kesalahan pada tahap yang sangat awal. Unit test ini dapat dirancang pada fase ini berdasarkan pada desain modul internal.

**V-Model > Proses Coding**

Proses coding untuk membuat modul sistem yang telah dirancang pada tahap desain dilakukan pada fase coding. Bahasa pemrograman yang paling cocok diputuskan berdasarkan kebutuhan sistem dan arsitektur.

Proses coding dilakukan berdasarkan pedoman dan standar coding. Baris kode yang ditulis melewati banyak code review dan dioptimalkan untuk kinerja terbaik sebelum final build.

**V-Model > Proses Validation**

Pada proses validation terdapat beberapa fase yang antara lain sebagai berikut.

1. Unit Testing

Unit Test yang dirancang dalam fase module design dijalankan pada kode yang sudah ditulis selama fase validasi ini. Unit testing adalah pengujian pada tingkat kode dan membantu menghilangkan bug pada tahap awal, meskipun semua bug tidak dapat diungkap oleh unit testing.

Integration Testing

Integration Testing dikaitkan dengan fase architectural design. Integration Testing dilakukan untuk menguji koeksistensi dan komunikasi modul secara internal dalam sistem.

1. System Testing

System testing secara langsung terkait dengan fase system design. System testing memeriksa seluruh fungsionalitas sistem dan komunikasi sistem yang sedang dikembangkan dengan sistem eksternal. Sebagian besar masalah kompatibilitas perangkat lunak dan perangkat keras dapat diungkap selama pelaksanaan system testing ini.

1. Acceptance Testing

Acceptance testing dikaitkan dengan fase requirement analysis dan melibatkan pengujian produk di lingkungan pengguna. Acceptance testing mengungkap masalah kompatibilitas dengan sistem lain yang tersedia di lingkungan pengguna. Selain mengungkap masalah kompatibilitas dengan sistem lain, pada fase ini juga mampu mengungkap masalah non-fungsional seperti load dan performance kinerja sistem di lingkungan pengguna yang sebenarnya.

1. **Kelebihan dan Kekurangan**

~Kelebihan :

1. Model yang menerapkan disiplin tingkat tinggi dan fase – fase dalam model diselesaikan satu persatu
2. Dapat bekerja dengan sangat baik untuk proyek kecil dimana requirement sangat mudah dipahami
3. Sederhana dan mudah untuk dipahami serta digunakan
4. Mudah untuk me-manage karena sifat kekakuan dari model dan setiap fase dalam siklus memiliki tujuan yang spesifik.

~Kekurangan :

1. Memiliki risiko dan ketidakpastian yang tinggi apabila diaplikasikan untuk proyek proyek besar
2. Model yang tidak bagus untuk proyek yang kompleks dan berorientasi obyek
3. Model yang buruk untuk proyek yang panjang dan berkelanjutan
4. Tidak cocok untuk proyek yang di mana requirement-nya mudah sekali berubah
5. Sekali sistem di dalam proses testing, sangat sulit sekali untuk mundur ke fase sebelumnya
6. Tidak ada sistem yang dapat digunakan sampai satu siklus selesai