

Agile Software Development

Agile software development adalah jenis metodologi pengembangan software yang membantu tim bekerja lebih cepat dengan cara membagi proyek menjadi bagian-bagian kecil dan bisa berjalan secara paralel.

Minggu 9

Agile Software Development

1.1 Pendahuluan

Agile software development adalah jenis metodologi pengembangan software yang membantu tim bekerja lebih cepat dengan cara membagi proyek menjadi bagian-bagian kecil dan bisa berjalan secara paralel.

Agile

Jenis metodologi pengembangan software yang membantu tim bekerja lebih cepat dengan membagi proyek menjadi bagian-bagian kecil dan bisa berjalan secara paralel.



Agile software development adalah jenis metodologi pengembangan *software* yang membantu tim bekerja lebih cepat dengan cara membagi proyek menjadi bagian-bagian kecil dan bisa berjalan secara paralel.

Dengan menggunakan *agile*, tim bekerja untuk mengevaluasi persyaratan dan hasil secara terus-menerus agar dapat mengimplementasikan perubahan yang efisien.

Agile berfokus pada kerja sama tim dan kepuasan pelanggan. Metode ini mengupayakan untuk terus menciptakan produk yang benar-benar diminati di pasar. *Roadmap* proyek pun menjadi lebih jelas dan bisa diperbarui sesuai kebutuhan konsumen. Itu sebabnya Agile selalu mengikuti tren pasar yang terus berubah dan mudah beradaptasi dengannya.

Agile merupakan metodologi yang lahir sebagai respons terhadap metodologi Waterfall yang cenderung kaku. Dalam metode *waterfall*, sistem bekerja secara linier dan mengharuskan tim menyelesaikan setiap fase proyek sebelum beralih ke fase selanjutnya. Hal ini tentu memakan waktu lebih lama untuk diselesaikan.

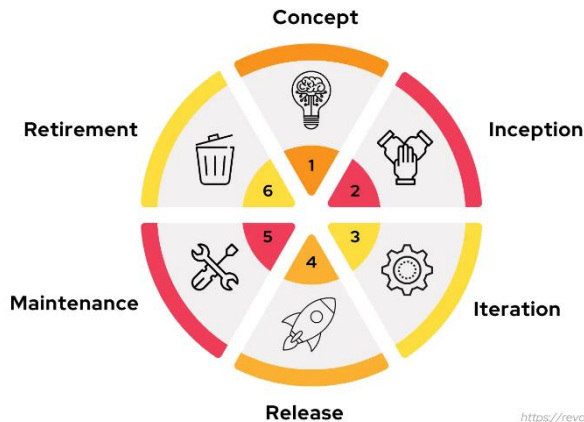
Sementara adanya Agile memungkinkan tim bekerja secara paralel atau bersamaan pada setiap fase proyek yang berbeda. Bayangkan kamu sedang membangun sebuah rumah. Dengan metode tradisional, kamu akan membuat desain yang sangat detail di awal, lalu baru mulai membangun. Masalahnya, seringkali selama proses pembangunan muncul kendala atau perubahan kebutuhan yang tidak terduga. Akibatnya, proyek bisa molor dan biaya membengkak.

Dengan Agile, pendekatannya berbeda. Kamu akan membuat desain awal yang lebih umum, lalu membangun rumah secara bertahap. Setiap tahap akan diuji dan dievaluasi, sehingga

kamu bisa langsung melakukan penyesuaian jika ada perubahan kebutuhan.

1.2 Tahapan Metode Agile Software Development

Dirangkum dari Spiceworks, berikut enam tahapan dalam metode *Agile software development*:



1. Concept

Pertama adalah fase pembuatan konsep. Di sini, *product owner* akan menentukan *scope* (ruang lingkup) dari proyek yang akan dibuat. Jika ada banyak proyek, mereka akan memprioritaskan pekerjaan yang paling penting.

Pada tahap konsep, *product owner* akan membahas persyaratan dengan klien serta memperkirakan waktu dan

biaya proyek. Analisis terperinci membantu mereka memutuskan apakah suatu proyek layak dieksekusi.

2. Inception

Setelah proyek dikonseptualisasikan, langkah selanjutnya yaitu membangun tim pengembangan *software*. Pada tahap ini, *product owner* memeriksa ketersediaan dan menetapkan siapa saja anggota tim yang cocok untuk proyek tersebut. Tak hanya itu, *product owner* juga bertanggung jawab untuk memfasilitasi anggota tim berbagai sumber daya yang diperlukan.

Setelah tim ditetapkan, proses desain akan dimulai dengan membuat *mockup user interface* maupun diagram UML. Arsitektur yang ada dalam proyek juga dibangun pada tahap ini.

Elemen-elemen yang sudah dirancang kemudian diberikan kepada *stakeholder* untuk mendapatkan saran dan masukan lebih lanjut.

3. Iteration atau development

Fase iterasi atau pengembangan cenderung menjadi fase terpanjang karena sebagian besar pekerjaan dilakukan di sini. *Developer* akan bekerja dengan *UX designer* untuk menggabungkan semua persyaratan produk dan *feedback* pelanggan sekaligus mengubah desain menjadi kode.

4. Release

Setelah produk dikembangkan, produk hampir siap untuk dirilis. Namun, sebelum itu tim *quality assurance* perlu melakukan beberapa pengujian untuk memastikan *software* berfungsi sepenuhnya. Tim akan menguji sistem untuk memastikan kodenya bersih. Jika ada potensi *bug* terdeteksi, *developer* perlu segera menanganinya.

Dalam fase ini, tim juga akan memfasilitasi tutorial penggunaan aplikasi kepada *user*.

Setelah semua *bug* diperbaiki dan pelatihan pengguna *user*, iterasi akhir produk dapat diterapkan dan dirilis ke bagian produksi.

5. Maintenance

Setelah aplikasi berhasil dirilis dan sudah bisa digunakan *user*, tim beralih ke tahap pemeliharaan.

Selama fase ini, tim *software development* akan memberikan dukungan secara berkelanjutan untuk menjaga agar sistem berjalan lancar dan mengatasi jika sewaktu-waktu muncul *bug*. Tim juga akan siap menawarkan pelatihan tambahan kepada *user* dan memastikan mereka tahu cara penggunaan produk.

Di sisi lain, *developer* dapat menggunakan *feedback* yang dikumpulkan selama tahap *maintenance* untuk merencanakan fitur dan meng-*update* iterasi berikutnya.

6. Retirement

Ada dua alasan mengapa suatu produk akan memasuki fase *retirement* atau penghentian, yaitu diganti dengan versi terbaru atau sistem itu sendiri sudah usang dan tidak lagi digunakan.

Di tahap ini, tim *software development* akan memberi tahu *user* bahwa *software* akan dihentikan. Jika ada pengganti, pengguna akan dimigrasikan ke sistem baru. Terakhir, tim *developer* harus menyelesaikan semua aktivitas akhir masa pakai yang masih tertunda dan mulai menghentikan dukungan yang diberikan untuk aplikasi.

1.3 Metode Agile

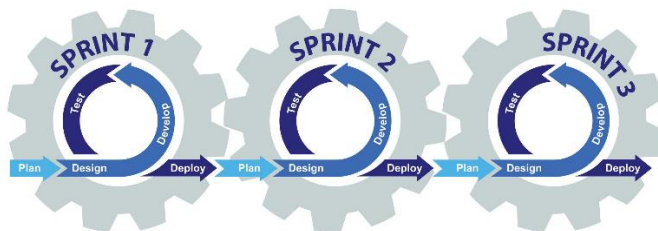
Pengembangan dalam perangkat lunak ada beberapa macam. Berikut adalah **metode agile** yang sering digunakan, yaitu:

1. Scrum Methodology

Scrum adalah salah satu metodologi dari Agile. Software developers biasanya menggunakan metodologi ini untuk mengembangkan produk yang bersifat kompleks dan mudah berubah-ubah. Scrum digunakan agar tim dapat

berkoordinasi dengan terstruktur dan komunikasi antara anggota tim terbentuk. Selain itu, kegunaan dari Scrum adalah untuk mempercepat rilis suatu produk dengan produktivitas dan kualitas yang tinggi. Metodologi ini juga memberikan cost yang lebih rendah dalam pengerjaan suatu proyek. Semua itu disesuaikan berdasarkan permintaan user.

Penggunaan scrum methodology pada umumnya untuk proyek sangat besar yang terkenal dengan istilah sprint. Sprint adalah jangka waktu dalam pengerjaan software untuk penyelesaian satu increment. Scrum dapat berjalan dengan baik jika terdiri dari beberapa tim yaitu product owner agar list produk tersusun dengan baik. Terdapat pula scrum master bertugas memastikan seluruh tim memahami proses dan development team untuk menjalankan tugas seperti IT, programmer dan yang lainnya.



Cara Kerja Scrum

Sebelum memulai sebuah proyek, project management harus menentukan anggota-anggota tim yang akan mengerjakan proyek tersebut. Pada tahapan awal dari Scrum, sebuah tim biasanya memiliki anggota tak lebih dari 5-10 orang. Pemilihan jumlah anggota juga harus teliti agar tim dapat bekerja lebih efektif dan efisien.

Kedua, project manager harus menentukan berapa lama waktu pengerjaan. Hal ini dikenal sebagai sprint. Sprint digunakan untuk menyelesaikan suatu masalah terutama dalam pembuatan produk baru dari sebuah rangkaian pekerjaan. Pada umumnya, sprint berlangsung selama 7 sampai 30 hari.

Langkah selanjutnya adalah menentukan peran dalam tim. Ini diperlukan agar tim dapat bekerja dengan maksimal dan juga tidak terjadi tumpang tindih tugas. Peran penting dalam team adalah Scrum Master, atau bisa disebut sebagai project manager. Scrum master harus memastikan bahwa proyek berjalan tanpa hambatan. Peran selanjutnya adalah product owner, mereka bertugas untuk memastikan kualitas dari produk sudah sesuai.

Keempat adalah mengumpulkan berbagai permasalahan. Langkah ini pada project management biasa disebut sebagai backlog. Berbagai permasalahan yang telah dikumpulkan

kemudian dibuat prioritasi pengerjaannya. Setelah semua tahapan sudah dikerjakan, hal selanjutnya adalah memulai sprint. Perlu diketahui, meskipun sprint sudah dimulai, backlog lain masih dapat ditemukan. Komunikasikan kepada product owner jika backlog tersebut dapat dikerjakan di dalam sprint atau sprint selanjutnya.

2. Extreme Programming (XP)

Extreme Programming disebut pula XP adalah metode yang lebih fokus ke teknis. Proses tim ekstra dalam segala hal untuk menghasilkan produk berkualitas tinggi. Proses dalam XP mencakup planning (perencanaan), designing (desain software), coding, testing dan listening (mendengarkan masukan pengguna).

Konsep ini berfokus pada pengembangan yang berkelanjutan dan juga bergantung kepada kepuasan pelanggan. Tim developer akan lebih sering berhubungan dengan user untuk mendapatkan feedback yang berguna. Melalui feedback pula tim akan mendengarkan dan menerima permintaan dari user, bahkan jika feedback diterima di akhir dari pengembangan software.

Alasan inilah yang menyebabkan metode ini disebut extreme, karena tim harus siap untuk bekerja lebih keras dan memberikan hasil produk terbaik sesuai kebutuhan user.

Tahapan Extreme Programming

Tahapan pertama dari XP adalah planning atau perencanaan. Tahap ini berguna untuk mengumpulkan berbagai kebutuhan dari perangkat yang akan dikembangkan. Selanjutnya tahapan design. Pada tahap ini, design yang dibuat haruslah sesimple dan sesederhana mungkin. Karena design simple selalu dipilih dibandingkan dengan yang kompleks. Developers tidak boleh asal menambahkan fungsi dan fitur kedalam produk meskipun fitur itu akan berguna tanpa permintaan dari user. Ketiga adalah coding. Setelah selesai tahap 1 dan 2, tim tidak langsung memulai coding, melainkan mengembangkan serangkaian tes yang akan dijalankan di setiap story dari planning. Ini berguna untuk melihat apa yang harus dibuat di dalam proyek tersebut, atau tim juga dapat melakukan pair programming dimana tes dijalankan bersamaan dengan pemrograman. Terakhir adalah tahapan testing. Ini diperlukan untuk melihat apakah sistem yang dikerjakan memiliki masalah atau tidak. Selain itu untuk memastikan apakah produk yang dibuat sudah sesuai dengan kebutuhan dari user.

3. Kanban

Kanban atau berasal dari kata Kanbanize adalah istilah dalam bahasa Jepang yang berarti sinyal visual. Dalam manajemen

produk, metode ini sangat berguna untuk memecahkan masalah dengan cara efektif dan efisien. Pada Awalnya, penggunaan metodologi Kanban ini hanya digunakan pada industri manufaktur, namun seiring perkembangan zaman dan era globalisasi dan teknologi seperti sekarang ini, banyak perusahaan menggunakan metodologi ini untuk keperluan pengembangan bisnis mereka.

Toyota adalah perusahaan yang mengembangkan dan mempopulerkan metode ini dan digunakan didalam proses manufaktur mereka. Metode ini bersifat visual yang memungkinkan tim yang sedang mengembangkan produk mereka dapat dengan mudah berkomunikasi pada pekerjaan yang akan dilakukan dan estimasi waktu pengerjaan. Kanban juga memberikan petunjuk dasar untuk mengurangi hal-hal yang tidak diperlukan.

Metode ini memanfaatkan board atau papan yang dibagi menjadi tiga kolom yang bertuliskan requested, in progress, dan done. Kanban board dianggap sangat efektif dalam pemecahan masalah dan dari ketiga kolom ini membuat tim yang bekerja dapat lebih fokus dan efektif dalam menyelesaikan pekerjaan yang ada.

Tahapan Menggunakan Kanban

Terdapat beberapa tahapan ketika ingin menjalankan metode Kanban ini. Hal pertama yang harus dilakukan

adalah memvisualisasikan pekerjaan. Ini berguna untuk mempermudah komunikasi dari setiap proses yang diterapkan. Pada dasarnya, setiap anggota tim akan membuat memvisualisasikan semua jenis pekerjaan yang akan dikerjakan dan alur dari pekerjaan dari setiap anggota.

Kedua, membatasi pekerjaan dalam setiap proses.

Pemberian batasan pada setiap penugasan, tim juga tidak perlu menyusun penugasan pekerjaan yang banyak di setiap proses. Ini berguna untuk mengurangi waktu pekerjaan dan hasil akhir yang diberikan akan lebih maksimal.

Ketiga, Alur pekerjaan menjadi fokus utama karena dengan begitu metode kanban yang sedang dilakukan akan lebih baik. Anggota tim akan dapat dengan mudah untuk mengumpulkan metrik untuk analisa alur kerja, dan dapat mengetahui kemungkinan dari masalah yang akan muncul di masa depan dengan menganalisa proses tersebut.

Selanjutnya ketahui batasan. Ketika mengerjakan tugas, sering muncul masalah-masalah baru, kalian akan bertugas memecahkan masalah tersebut agar tidak mengganggu proses pekerjaan. Kalian harus memahami batasan kalian ketika sedang memecahkan masalah tersebut, agar tidak terjadi tumpang tindih dan mengganggu proses dari anggota lainnya.

Memberikan feedback ke sesama anggota itu juga menjadi poin yang penting. Karena dari feedback yang didapat, proses pengerjaan akan memiliki persepsi yang sama. Dan yang terakhir adalah terus-menerus melakukan perbaikan. Seluruh tim wajib untuk memantau workflow, throughput, lead time, dan indikator lainnya guna meningkatkan tingkat efektivitas yang baik.

4. Scaled Agile Framework (SAFe)

Metode Scaled Agile Framework atau SAFe biasanya untuk perusahaan besar dengan jumlah tim yang banyak. Dalam menjalankan kinerja antar divisi saling kerjasama dan mengadakan meeting rutin untuk hasil terbaik.

5. Lean Software Development (LSD)

Lean Software Development atau LSD adalah **metode agile** paling pas bagi yang ingin mengembangkan software hemat dana. LSD mengembangkan fitur MVP (Minimum Viable Product) sesuai kebutuhan pengguna dan jika tidak akan dialihkan ke yang lain.

6. Crystal Methodology

Metode Crystal lebih fokus untuk pengembangan tim dalam melakukan feedback, dokumentasi, komunikasi dan interaksi. Terdapat 7 prinsip utama dalam pengembangannya, yaitu:

- Frequent delivery dengan melakukan tes user untuk hasil software secara optimal.
- Reflection improvement bahwa setiap kualitas produk terbaik tetap ada yang masih harus ditingkatkan.
- Osmotic communication untuk memastikan semua pemahaman yang sama akan informasi.
- Personal safety agar semua dapat mengungkapkan ide tanpa ada rasa takut.
- Focus on work agar semua tim paham dengan tugas masing-masing.
- Easy access to expert users bahwa semua anggota boleh bertanya pada user.
- Technical tooling dengan menggunakan tools pendukung untuk mengetahui kesalahan.

7. Feature Driven Development (FDD)

Agile berikutnya adalah Feature Driven Development atau FDD yang fokus pada penyelesaian satu fitur. Menyelesaikan satu fitur membutuhkan waktu sekitar 2 hingga 10 hari dan lebih spesifik dalam proses software.

8. Rational Unified Process (RUP)

Berbeda dari agile methodology yang lain, Rational Unified Process atau (RUP) mengembangkan software sangat lengkap. Kelengkapan berupa panduan, simulai, contoh dan pengembangan sistem. Tahapan RUP meliputi insepction

(identifikasi sistem) dan elaboration (pembuatan desain lengkap). Selain itu ada construction (implementasi desain ke program) dan transition (penyerahan ke user).

9. Dynamic System Development Method (DSDM)

Jenis terakhir dari agile adalah DSDM dengan keterlibatan seluruh tim untuk hasil dalam bisnis secara lebih luas. Prinsip dari metode DSDM agar fokus dalam bisnis, penyelesaian ontime, kualitas hasil akhir optimal, desain software jelas dan pengembangan bertahap.

1.4 Kelebihan Metode Agile

Jika melihat macam-macam agile, sudah terlihat kelebihan dari proses pengembangan software, yaitu:

1. Hasil Software Berkualitas

Dengan tim unggulan dan metode bertahap yang efisien, hasil perangkat lunak tentunya lebih baik dengan kualitas unggulan.

2. Klien Puas dan Fleksibel

Klien akan puas dengan software sebab selama proses pembuatannya menjadi bagian dalam memberi feedback kekurangan sesuai yang dibutuhkan. Dengan begitu selama proses berlangsung, ada pembetulan jika dirasa kurang baik, fleksibel dan hasil tepat serta dapat efisien waktu.

3. Fokus Kebutuhan Konsumen

Tahapan proses project sesuai dengan keinginan dan kebutuhan konsumen sehingga fitur yang ada dalam software benar-benar yang paling dibutuhkan. Kelebihan dari agile methodology memberi dampak signifikan pada pengguna. Hasil software sesuai dengan kebutuhan sehingga akan terjadi peningkatan produktivitas.

1.5 Kekurangan Metode Agile

Meskipun begitu banyak kelebihan agile, tetap saja sebuah hasil software ada kekurangannya. Kekurangan dari agile methodology antara lain:

1. Hasil Akhir Kurang Jelas

Sebetulnya perencanaan dari metode ini kurang terencana sehingga hasil produk kurang jelas.

2. Tergantung Komitmen

Bagaimanapun juga pengerjaan membutuhkan tim solid. Jika ada satu saja yang tidak komit, hasilnya tentu kurang bagus untuk tim dalam pengembangan software.

3. Dokumentasi Kurang

Perencanaan yang kurang lengkap akan berdampak langsung dengan hal lain seperti dokumentasi. Hasilnya adalah dokumentasi kurang lengkap sehingga produk akhir benar-benar kurang maksimal. Meskipun demikian, kekurangan agile methodology masih bisa diminimalkan

dengan tim solid. Tim yang bagus dan memiliki komitmen tinggi akan menciptakan suatu produk unggulan dan berkualitas.

1.6 Prinsip-Prinsip Agile

Pengembangan *software* menggunakan metode Agile mempunyai 12 prinsip utama. Prinsip inilah yang dikenal sebagai Agile Manifesto dan inilah rinciannya.

1. Mengutamakan kepuasan klien dan menjadikannya sebagai prioritas utama.
2. Dalam proses development, semua bentuk perubahan diterima.
3. *Software* bisa diproduksi dalam jangka waktu pendek (14 – 60 hari) namun kualitasnya teruji.
4. Ada proses kerja sama antara developer dengan pebisnis selama proyek berjalan.
5. Menciptakan lingkungan dengan orang-orang bermotivasi tinggi agar proyek bisa selesai dengan efektif dan efisien.
6. Komunikasi langsung sangat diperlukan dalam proses pengembangan *software*.
7. Kemajuan proyek dinilai dari *software* yang mampu bekerja dengan baik sesuai harapan.

8. Pengembangan *software* secara kontinu bisa berlangsung apabila ada dukungan dari semua pihak termasuk developer, sponsor, dan user.
9. Keutamaan dalam metode Agile adalah keunggulan dari segi teknis.
10. Sumber daya atau *resource* yang ada harus dimaksimalkan dengan kesederhanaan.
11. Manajemen semua tim developer sangat mempengaruhi kebutuhan arsitektur dan software.
12. Untuk mengatur pola kerja sehingga lebih efektif, tim pengembang melakukan refleksi dan evaluasi secara berkala.

1.7 Tujuan Metode Agile

Setiap pengembangan software tentu memiliki tujuan khusus. **Metode agile** sendiri memiliki tujuan berbeda bagi owner, user hingga tim, antara lain:

1. Owner

Bagi owner atau pemilik dari produk dapat memberi penjelasan lengkap tentang software yang diciptakan. Owner juga mampu memberi gambaran detail tentang berbagai fitur unggulan software lengkap dengan segmen pasar yang menjadi target. Bahkan untuk hasil

terbaik owner biasanya terjun langsung dalam proses pembuatan software dari awal hingga meluncur ke pasaran.

2. User

Metode agile sifatnya dinamis sehingga user dapat memberi masukan positif saat tim pengembang sedang merencanakan dan membuat fitur terbaik. Apabila ada yang kurang bisa langsung diubah agar lebih baik dari sebelumnya. Dengan proses bertahap ini biasanya hasil lebih mumpuni dan berkualitas.

3. Development Team

Tujuan agile methodology bagi pengembang untuk pengembangan perangkat lunak lebih dinamis tanpa harus menunggu stakeholder selesai. Sistem yang cepat, efisien dan tangkas membuat pekerjaan tim selesai dalam waktu bersamaan sesuai tugasnya masing-masing.

1.8 Faktor Manusia dalam Metode Agile

Setelah mengetahui tujuan agile, faktor manusia dalam metode agile juga sangat penting untuk diketahui, berikut beberapa poinnya.

1. Kompetensi yang meliputi pengetahuan serta keterampilan dalam proses *software development*.

2. Fokus, artinya setiap anggota dalam tim harus fokus pada satu tujuan meski masing-masing memiliki pembagian tugas berbeda.
3. Kolaborasi antara anggota tim dengan manajer, serta dengan klien.
4. Kemampuan mengambil keputusan dengan cepat dan tepat untuk mengatasi permasalahan yang ada.
5. *Fuzzy problem solving* untuk menentukan prioritas masalah yang harus diselesaikan lebih dulu.
6. Saling percaya dan menghormati antar anggota sehingga menjadi tim menjadi solid.
7. Manajemen diri pada setiap anggota tim untuk menyelesaikan project, mengatur proses pengembangan sesuai dengan lingkungan, serta disiplin menyerahkan hasil sesuai dengan jadwal yang ditentukan.

1.9 Manfaat Menggunakan Metode Agile

Dilansir dari Decipher Zone, berikut beberapa manfaat menggunakan metode Agile:

1. Memiliki kontrol yang lebih baik

Agile memberikan kontrol proyek yang lebih besar karena memiliki fitur seperti transparansi, *quality control*, dan integrasi *feedback*. Sepanjang fase implementasi, para *stakeholder* bisa memastikan

kualitas *software* menggunakan *tools* dan teknik pelaporan harian.

2. Mengurangi risiko

Menggunakan Agile memungkinkan tim dalam mendapatkan *feedback* sesering mungkin yang bisa segera ditindaklanjuti. Hal ini memudahkan dalam mengidentifikasi *error* dan membangun proses mitigasi risiko yang efektif.

3. Meningkatkan fleksibilitas

Menerapkan Agile dalam tim pengembangan proyek memberi mereka fleksibilitas tinggi. Dalam metodologi manajemen proyek lainnya, perubahan biasanya memakan waktu yang lama dan mahal. Namun, Agile membagi proyek menjadi bagian-bagian kecil yang memungkinkan tim mengimplementasikan perubahan dalam waktu singkat.

4. Melakukan perbaikan secara berkelanjutan

Mengusahakan peningkatan berkelanjutan adalah salah satu prinsip inti dari Agile. Perbaikan berkelanjutan mendorong kolaborasi dan pertukaran ide anggota tim yang memungkinkan mereka untuk belajar serta meningkatkan pengalaman masing-masing.

5. Meningkatkan kualitas produk

Agile bekerja semaksimal mungkin dalam manajemen proyek dengan meningkatkan proses secara berulang

selama periode waktu tertentu. Kontrol terhadap kualitas juga terus dijaga untuk memastikan hasil akhir produk berkualitas tinggi.

6. Memberikan kepuasan bagi pelanggan

Dikarenakan banyak *stakeholder* yang terlibat dalam *project development life cycle*, perusahaan akan mendapatkan lebih banyak *feedback* yang akan membantu menilai apakah produk akhir sesuai dengan kebutuhan pelanggan. Dengan demikian, kepuasan pelanggan terhadap produk akhir juga cenderung meningkat.

1.10 Cara Menerapkan Metodologi Agile dalam Proyek

Jika Anda masih bingung bagaimana menerapkan teori Agile ke dalam pekerjaan tim secara nyata, berikut adalah panduan langkah demi langkah untuk mengimplementasikan proses metodologi Agile:

1. Pilih Kerangka Kerja Agile yang Tepat

Langkah pertama yang harus dilakukan adalah memilih kerangka kerja (framework) Agile yang paling sesuai dengan kebutuhan tim Anda.

Berikut beberapa pilihan kerangka kerja Agile yang paling populer:

- **Scrum:** Berbasis prinsip manajemen proyek secara terstruktur
- **Kanban:** Fokus pada visualisasi alur kerja
- **Scrumban:** Kombinasi antara Scrum dan Kanban
- **XP (Extreme Programming):** Fokus pada kualitas produk dan umpan balik pengguna
- **APF (Adaptive Project Framework):** Cocok untuk tim yang fleksibel
- **DSDM (Dynamic System Development Method):** Menyasar seluruh siklus hidup pengembangan perangkat lunak
- Saat memilih metode, pertimbangkan ukuran tim, karakteristik proyek, dan tingkat pengalaman tim terhadap masing-masing pendekatan tersebut.
- Semakin Anda memahami karakter tim dan jenis proyek yang sering Anda kerjakan, maka semakin mudah memilih metode Agile yang paling cocok.

2. Bentuk Tim Agile yang Efektif

Tim Agile berbeda dengan tim biasa.

Untuk sukses menjalankan metode Agile, Anda perlu membentuk tim dengan **pembagian peran yang jelas** dan budaya kerja yang kolaboratif.

Berikut beberapa karakteristik penting tim Agile:

- **Organisasi mandiri (self-organizing):** Setiap anggota tim mampu mengatur pekerjaannya sendiri secara mandiri untuk mencapai hasil terbaik.
- **Kolaborasi lintas fungsi (cross-functional collaboration):** Tim Agile sering bekerja lintas divisi dan harus mampu berkomunikasi serta berkoordinasi dengan baik.
- **Perencanaan iterasi (iteration planning):** Tim harus dapat menyusun rencana kerja bertahap (misalnya sprint) berdasarkan prioritas yang ada dalam product backlog.

3. Rencanakan Proyek dengan Matang

Setelah metode Agile dipilih dan tim terbentuk, langkah selanjutnya adalah **merancang proyek dengan rinci dan jelas**.

Hal-hal yang perlu dilakukan dalam tahap ini:

- Tentukan **tujuan dan ruang lingkup proyek** secara spesifik.
- Bagi pekerjaan menjadi tahapan kecil yang mudah dikelola, misalnya dalam bentuk **sprint** jika Anda menggunakan metode Scrum.
- Buat **product backlog**, yaitu daftar fitur atau tugas yang perlu dikerjakan, dilengkapi dengan urutan prioritas. Ini

akan membantu tim fokus pada hal yang paling penting terlebih dahulu.

4. Kelola Harapan Stakeholder

Sebelum proyek berjalan, penting untuk berkomunikasi dengan stakeholder utama agar umpan balik mereka dapat dipertimbangkan.

Tingkat keterlibatan stakeholder bisa berbeda-beda.

Beberapa mungkin ingin menerima laporan rutin, sementara yang lain cukup diberi pembaruan saat-saat penting. **Buatlah jalur komunikasi dua arah** agar mereka merasa dilibatkan dan terbuka terhadap perubahan jika diperlukan.

5. Ukur Keberhasilan Proyek

Mengukur keberhasilan proyek adalah bagian penting dalam metodologi Agile.

Dengan menganalisis apa yang berhasil dan apa yang perlu ditingkatkan, Anda bisa mendapatkan pembelajaran berharga untuk proyek-proyek selanjutnya.

Berikut cara mengukur progres dalam proyek Agile:

- **Daily Standup:** Pertemuan singkat harian untuk menyampaikan hambatan dan solusi.
- **Sprint Review:** Evaluasi hasil sprint bersama tim dan stakeholder untuk mendapatkan masukan.

- **Retrospective:** Refleksi setelah sprint untuk memperbaiki proses ke depan.
Sebelum memulai proyek besar, tentukan juga **Key Performance Indicator (KPI)** atau indikator kinerja, agar Anda bisa menetapkan tolok ukur dan menilai kemajuan proyek secara objektif.

1.11 Langkah-langkah Penerapan Metode Agile

1. Membentuk Tim dan Menetapkan Peran: Bentuk tim yang terdiri dari anggota dengan berbagai keahlian dan tanggung jawab. Tetapkan peran seperti Product Owner, Scrum Master, dan tim pengembang.
2. Membuat Product Backlog: Buat daftar prioritas dari semua fitur dan kebutuhan yang ingin diwujudkan dalam perangkat lunak. Product Backlog harus selalu diperbarui dan diprioritaskan berdasarkan nilai bisnis dan feedback dari pengguna.
3. Melakukan Perencanaan Sprint: Adakan pertemuan perencanaan sprint untuk menentukan tujuan sprint, memilih item dari Product Backlog, dan memperkirakan waktu yang dibutuhkan untuk menyelesaikannya.

4. Melaksanakan Sprint: Tim bekerja secara kolaboratif untuk menyelesaikan tugas-tugas sprint dalam jangka waktu yang ditentukan (biasanya 1-4 minggu). Scrum Master membantu tim untuk fokus dan menyelesaikan pekerjaan sesuai target.
5. Melakukan Daily Scrum: Adakan pertemuan Daily Scrum setiap hari untuk membahas kemajuan sprint, mengidentifikasi hambatan, dan melakukan penyesuaian jika diperlukan.
6. Melakukan Sprint Review: Adakan pertemuan Sprint Review di akhir sprint untuk mendemonstrasikan hasil sprint kepada Product Owner dan stakeholders. Dapatkan feedback dan persetujuan untuk melanjutkan sprint berikutnya.
7. Melakukan Sprint Retrospective: Adakan pertemuan Sprint Retrospective untuk mengevaluasi sprint yang telah selesai, mengidentifikasi apa yang berjalan dengan baik dan apa yang perlu diperbaiki. Pelajari dari pengalaman dan lakukan penyesuaian pada proses Agile untuk sprint berikutnya.

1.12 Contoh Pertama Penerapan Metode Agile

Kasus:

Mengembangkan aplikasi mobile e-commerce untuk toko online.

Tim:

1. Product Owner: Mewakili kebutuhan pelanggan dan stakeholders. Bertanggung jawab atas Product Backlog dan memastikan bahwa aplikasi yang dikembangkan sesuai dengan kebutuhan pasar.
2. Scrum Master: Memfasilitasi proses Agile, membantu tim untuk menyelesaikan sprint dengan sukses, dan memastikan bahwa semua anggota tim memahami dan mengikuti prinsip-prinsip Agile.
3. Tim Pengembang: Terdiri dari programmer, tester, dan desainer yang bertanggung jawab untuk membangun aplikasi. Bekerja sama secara kolaboratif untuk menyelesaikan tugas-tugas sprint dan menghasilkan aplikasi yang berkualitas tinggi.

Product Backlog:

1. Mendaftarkan fitur-fitur penting untuk aplikasi mobile e-commerce, seperti browsing produk, menambahkan produk ke keranjang, checkout, dan pembayaran.
2. Prioritas berdasarkan nilai bisnis dan feedback dari pengguna. Dapat diubah dan diperbarui secara berkala selama proses pengembangan.

Sprint 1:

1. Tujuan: Menyelesaikan fitur browsing produk dan menambahkan produk ke keranjang.
2. Item Backlog:
 1. Menampilkan daftar produk dengan gambar, harga, dan deskripsi yang jelas.
 2. Memungkinkan pengguna untuk mencari produk berdasarkan kategori, nama, atau kata kunci.
 3. Memasukkan produk ke keranjang dan melihat isi keranjang.
4. Estimasi waktu: 2 minggu.

Proses Pengembangan:

1. Tim pengembang bekerja sama untuk merancang, mengembangkan, dan menguji fitur-fitur yang telah ditentukan dalam backlog sprint.
2. Scrum Master membantu tim untuk fokus pada tugas-tugas sprint, menyelesaikan hambatan, dan memastikan bahwa semua anggota tim bekerja secara efektif.
3. Daily Scrum diadakan setiap hari untuk membahas kemajuan sprint, mengidentifikasi masalah yang dihadapi, dan melakukan penyesuaian jika diperlukan.

Sprint Review:

1. Di akhir sprint, tim mendemonstrasikan fitur yang telah diselesaikan kepada Product Owner dan stakeholders.

2. Product Owner memberikan feedback dan persetujuan untuk melanjutkan sprint berikutnya.
3. Tim pengembang mengumpulkan feedback dari stakeholders untuk meningkatkan dan menyempurnakan aplikasi.

Sprint Retrospective:

1. Setelah sprint selesai, tim mengadakan pertemuan Sprint Retrospective untuk mengevaluasi sprint yang telah dilakukan.
2. Tim mengidentifikasi apa yang berjalan dengan baik dan apa yang perlu diperbaiki dalam sprint berikutnya.
3. Pelajaran yang didapat dari sprint sebelumnya digunakan untuk meningkatkan proses Agile dan mencapai hasil yang lebih baik di masa depan.

1.13 Contoh Kedua Penerapan Metode Agile

Untuk memahami lebih jauh tentang keunggulan Agile, berikut contoh perbandingan penggunaan Agile dengan Waterfall.

1. Contoh durasi pengerjaan software menggunakan metode Waterfall

Sebuah tim di perusahaan X akan membuat suatu *software* menggunakan Waterfall.

- Pertama, tim mendedikasikan 5 minggu untuk mengumpulkan dan menganalisis persyaratan produk.
- Kedua, tim mulai merancang *wireframe* dasar dari semua fitur aplikasi dan membutuhkan waktu sekitar 6,5 minggu.
- Ketiga, tim akan menerjemahkan desain ke dalam kode lalu mengujinya. Proses ini memakan waktu sekitar 13 minggu.
- Keempat, tim mulai menguji sistem sekitar 6,5 minggu.
- Terakhir, *developer* akan menghabiskan waktu yang tersisa untuk menguji penerimaan *user* melalui tim pemasaran.

Total waktu yang dihabiskan untuk pembuatan *software* ini adalah 31 minggu atau sekitar 8 bulan.

2. Contoh durasi pengerjaan software menggunakan metode Agile

Sekarang, bandingkan dengan waktu pembuatan *software* menggunakan metode Agile berikut.

- Di Agile, proyek dipecah menjadi banyak iterasi. Masing-masing iterasi memiliki durasi yang sama (antara 2 sampai 8 minggu) lalu setiap tim akan mengirimkan hasil akhirnya.

- Untuk contoh ini, misalnya proyek dibagi menjadi delapan bagian yang masing-masing berdurasi 4 minggu.
- Tim bisnis dan *developer* bekerja sama untuk menentukan fitur penting yang diperlukan.
- Setelah itu, tim akan mengirimkan aplikasi dengan fitur yang telah ditentukan.
- Setelah aplikasi siap, tim secara kolaboratif menentukan apakah aplikasi tersebut sesuai dengan tujuan awal. Mereka juga akan memutuskan perubahan apa yang dapat dilakukan dan fitur mana yang dapat ditambahkan pada iterasi berikutnya berdasarkan prioritas.
- Semua proses tersebut berjalan secara paralel sampai menunjukkan produk akhir yang bisa digunakan *user* hanya dalam waktu empat minggu.

1.14 FAQ (Frequently Asked Question)

Mengapa banyak perusahaan mengembangkan perangkat lunak menggunakan konsep Agile?

Metode Agile memberi perusahaan *framework* yang dapat digunakan untuk menyelesaikan proyek secara efektif. Semua kebutuhan dan persyaratan proyek akan tercantum

dalam *framework* tersebut sehingga proyek bisa berjalan dengan lancar.

Ini adalah cara yang jauh lebih efisien untuk mengelola proyek jika dibandingkan dengan metode tradisional (Waterfall). Metode Waterfall tidak memperhitungkan perubahan pasar seiring dengan perubahan permintaan konsumen. Ini akan berdampak pada kerugian besar bagi *developer* dan risiko kegagalan proyek juga tinggi.

Ketika perusahaan mulai beralih ke metode Agile, mereka dapat mengikuti tren pasar yang berubah dan membuat produk sesuai keinginan pasar.

Lebih lanjut, berikut alasan banyak perusahaan mengembangkan *software* menggunakan Agile:

- membantu meningkatkan proses manajemen tim proyek menggunakan *tools management*.
- meningkatkan kerja sama dalam tim karena bisa menjalin komunikasi yang lebih efektif.
- Membantu memecah tugas kompleks menjadi beberapa bagian tugas yang dapat dikelola dan dikerjakan oleh tim dengan lebih mudah.