



**MODUL PENGEMBANGAN PERANGKAT
LUNAK (CPL 230)**

**MODUL 1
AGILE SOFTWARE DEVELOPMENT**

**DISUSUN OLEH
HANI DEWI ARIESSANTI, M.KOM**

**UNIVERSITAS ESA UNGGUL
2020**

PENGERTIAN AGILE SOFTWARE DEVELOPMENT

A. Pendahuluan

Kata Agile berarti bersifat cepat, ringan, bebas bergerak, waspada. Kata ini digunakan sebagai kata yang menggambarkan konsep model proses yang berbeda dari konsep model-model proses yang sudah ada. Konsep Agile Software Development dicetuskan oleh Kent Beck dan 16 rekannya dengan menyatakan bahwa Agile Software Development adalah cara membangun software dengan melakukannya dan membantu orang lain membangunnya sekaligus. Dalam Agile Software Development interaksi dan 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. Namun demikian, sama seperti model proses yang lain, Agile Software Development memiliki kelebihan dan tidak cocok untuk semua jenis proyek, produk, orang dan situasi. Agile Software Development memungkinkan model proses yang toleransi terhadap perubahan kebutuhan sehingga perubahan dapat cepat ditanggapi. Namun di sisi lain menyebabkan produktifitas menurun.

B. Kompetensi Dasar

Mengetahui, memahami dan menguasai ketrampilan dalam membangun pengetahuan tentang proses pengembangan perangkat lunak.

C. Kemampuan Akhir Yang Diharapkan

- 1) Mahasiswa mampu menjeleaskan definisi Agile Software Development
- 2) Mahasiswa mampu menjeleaskan Prinsip Agile Software Development
- 3) Mahasiswa mampu menjeleaskan Agile Manifesto

D. Kegiatan Belajar

Pengenalan Agile Software Development

Agile software development merupakan metode atau pendekatan khusus untuk manajemen proyek yang digunakan dalam pengembangan software. Metode ini memerlukan tim pengembang yang dapat beradaptasi dengan cepat terhadap perubahan. Agile development akan menggunakan urutan kerja inkremental dan berulang yang umumnya dikenal sebagai sprint.

Metode Agile juga akan berfokus pada kolaborasi tim yang baik antara tim pengembang dengan klien agar output yang dihasilkan sesuai dengan harapan. Agile software development ini dapat Anda gunakan jika software dan aplikasi yang Anda kembangkan membutuhkan pembaruan atau update secara rutin.

Ada beberapa langkah dalam Agile Development Methods, yaitu :

- Perencanaan, pada langkah ini pengembang dan klien membuat rencana tentang kebutuhan dari perangkat lunak yang akan dibuat.
- Implementasi, bagian dari proses dimana programmer melakukan pengkodean perangkat lunak.
- 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.
- Dokumentasi, setelah dilakukan tes perangkat lunak langkah selanjutnya yaitu proses dokumentasi perangkat lunak untuk mempermudah proses maintenance kedepannya.
- Deployment, yaitu proses yang dilakukan oleh penjamin kualitas untuk menguji kualitas sistem. Setelah sistem memenuhi syarat maka perangkat lunak siap dideployment.
- 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.

Ada juga beberapa model dalam Agile Development Methods, seperti: Acceptance Test Driven Development (ATDD), Agile Modeling, Adaptive Software Development (ASD), Agile Unified Process (AUP), Continuous integration (CI), Crystal Clear, Crystal Methods, Dynamic Systems Development Method (DSDM), Extreme Programming (XP), Feature Driven Development (FDD), Graphical System Design (GSD), Kanban, Lean software development, Rational Unified Process (RUP), Scrum, Scrum-ban, Story-driven modeling, Test-driven development (TDD), Velocity tracking, dan Software Development Rhythms.

Beberapa keuntungan dari Agile Development Methods :

- Menambah produktivitas tim
- Menambah kualitas dari perangkat lunak
- Menambah kepuasan dari klien
- Menghemat biaya

Dalam Agile Software Development interaksi dan 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.

Pada intinya, Agile Development Methods sangat membantu para pengembang perangkat lunak dalam melakukan penyerahan produk secara tepat waktu dari suatu tahap operasional perangkat lunak yaitu pada bagian analisa dan desain.

Prinsip Agile Software Development

Agile manifesto merupakan nilai-nilai yang digunakan sebagai landasan ketika menjalankan Agile software development. Agile manifesto ini diperkenalkan oleh 17 software developer, berikut 4 ciri-ciri utamanya:

1. Interaksi dan personel lebih penting daripada proses dan alat.

Di dalam metode ini, individu dan interaksinya akan dianggap lebih penting daripada proses dan tools yang digunakan. Proses dan tools yang hebat tidak akan berfungsi dengan baik jika personel tim tidak memiliki kemampuan untuk menjalankannya. Interaksi dan komunikasi antar tim juga diperlukan untuk menghindari konflik yang bisa membuat proses pengembangan berjalan tidak efektif.

2. Software yang berfungsi lebih penting daripada dokumentasi yang lengkap

Tujuan utama dari Agile development adalah untuk mengembangkan suatu software. Oleh karena itu, software yang berfungsi dengan baik dan berhasil di-deliver kepada user akan lebih penting nilainya daripada dokumen untuk mendeskripsikan produk.

3. Kolaborasi dengan klien lebih penting daripada negosiasi kontrak

Software akan dikembangkan sesuai dengan permintaan klien. Namun sayangnya, sebagian besar klien tidak memahami dunia IT sehingga mereka kesulitan untuk menjabarkan sistem yang mereka inginkan. Oleh karena itu, Anda harus melibatkan klien dalam proses pengembangan software. Hal ini untuk menghindari

pengembangan fitur atau sistem yang tidak sesuai dengan permintaan klien. Dengan terus berkomunikasi dengan klien, kesalahpahaman dapat diminimalisir.

4. **Respon terhadap perubahan lebih penting daripada mengikuti rencana.**

Dalam proses pengembangan software, perubahan dapat terjadi kapan saja. Permintaan perubahan bisa datang dari klien atau pihak developer sendiri. Selain itu, teknologi akan semakin berkembang sehingga perubahan dapat terus terjadi. Project plan juga penting, namun dalam Agile development, tim harus bisa beradaptasi dengan cepat ketika terjadi perubahan.

Ciri-ciri ini didukung oleh 12 prinsip yang ditetapkan oleh Agile Alliance. Menurut Agile Alliance, 12 prinsip ini adalah bagi mereka yang ingin berhasil dalam penerapan Agile Software Development:

- kepuasan klien adalah prioritas utama dengan menghasilkan produk lebih awal dan terus menerus
- menerima perubahan kebutuhan, sekalipun diakhir pengembangan.
- Penyerahan hasil/software dalam hitungan waktu dua minggu sampai dua bulan.
- Bagian bisnis dan pembangun kerja sama tiap hari selama proyek berlangsung
- Membangun proyek dilingkungan orang-orang yang bermotivasi tinggi yang bekerja dalam lingkungan yang mendukung dan yang dipercaya untuk dapat menyelesaikan proyek.
- Komunikasi dengan berhadapan langsung adalah komunikasi yang efektif dan efisien
- Software yang berfungsi adalah ukuran utama dari kemajuan proyek
- dukungan yang stabil dari sponsor, pembangun, dan pengguna diperlukan untuk menjaga perkembangan yang berkesinambungan
- perhatian kepada kehebatan teknis dan desain yang bagus meningkatkan sifat agile
- kesederhanaan penting
- arsitektur, kebutuhan dan desain yang bagus muncul dari tim yang mengatur dirinya sendiri

- secara periodik tim evaluasi diri dan mencari cara untuk lebih efektif dan segera melakukannya.

Kedua belas prinsip tersebut menjadi suatu dasar bagi model-model proses yang punya sifat agile. Dengan prinsip-prinsip tersebut Agile Process Model berusaha untuk menyasati 3 asumsi penting tentang proyek software pada umumnya:

1. kebutuhan software sulit diprediksi dari awal dan selalu akan berubah. Selain itu, prioritas klien juga sering berubah seiring berjalannya proyek.
2. Desain dan pembangunan sering tumpang tindih. Sulit diperkirakan seberapa jauh desain yang diperlukan sebelum pembangunan
3. Analisis, desain, pembangunan dan testing tidak dapat diperkirakan seperti yang diinginkan.

Faktor Manusia Pada Agile Process Models

Kunci faktor manusia pada model ini adalah proses didasari pada kebutuhan orang dan tim bukan sebaliknya, Untuk dapat sukses menerapkan model proses ini, pada faktor manusia ada beberapa kunci penting:

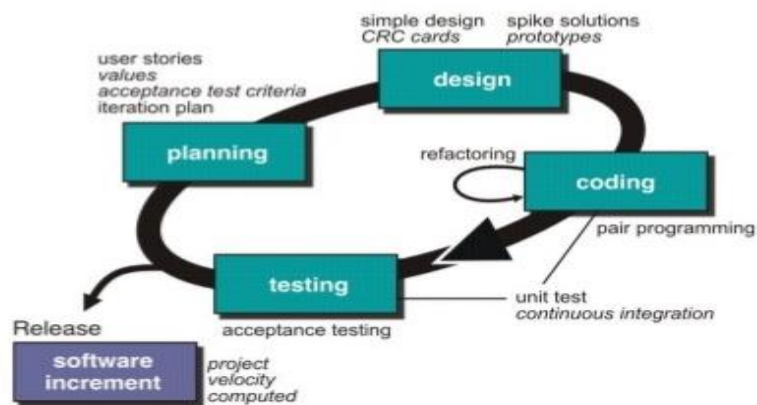
1. kompetensi: ketrampilan dalam membangun dan pengetahuan tentang proses membangun
2. Fokus: memiliki fokus yang sama sekalipun peran dalam tim berbeda
3. Kolaborasi : kerja sama dengan klien, anggota tim dan manajer.
4. Kemampuan ambil keputusan : tim pembangun memiliki otonomi dalam pengambilan keputusan terkait teknis dan proyek
5. Kemampuan fuzzy problem-solving: mampu menyelesaikan masalah yang penting untuk dipecahkan segera atau nanti.
6. Saling percaya dan hormat: kekompakan tim yang didukung oleh rasa percaya dan saling menghargai satu sama lain.
7. Manajemen diri: tim mengatur diri untuk selesaikan proyek, mengatur proses untuk disesuaikan dengan lingkungannya, tim menjadwalkan dirinya untuk menyerahkan hasil.

Agile Process Model

Berikut ini adalah model-model proses yang termasuk agile process model :

1. XP(Extreme Programming)

- dipublikasikan oleh Kent Beck tahun 1999
- Menggunakan pendekatan berorientasi objek
- Aktifitas Perencanaan : pengumpulan user stories dari klien yang klien tetapkan prioritasnya. Setiap story ditetapkan harga dan lama pembangunan, jika terlalu besar, story dapat dipecah menjadi beberapa story yang lebih kecil. Periksa dan pertimbangkan resiko
- Aktifitas Desain: berprinsip: sederhana.Memanfaatkan kartu CRC (Class-ResponsibilityCollaborator) untuk identifikasi dan mengatur class-class di konsep OO. Jika temui kesulitan, prototype dibangun [ini namanya spike solution]. Lakukan refactoring, yaitu mengembangkan desain dari program setelah ditulis
- Aktifitas Pengkodean: siapkan unit test sebelum pengkodean dipakai sebagai fokus pemrogram untuk membuat program. Pair programming dilakukan untuk real time program solving dan real time quality assurance
- Aktifitas Pengujian: menggunakan unit test yang dipersiapkan sebelum pengkodean.

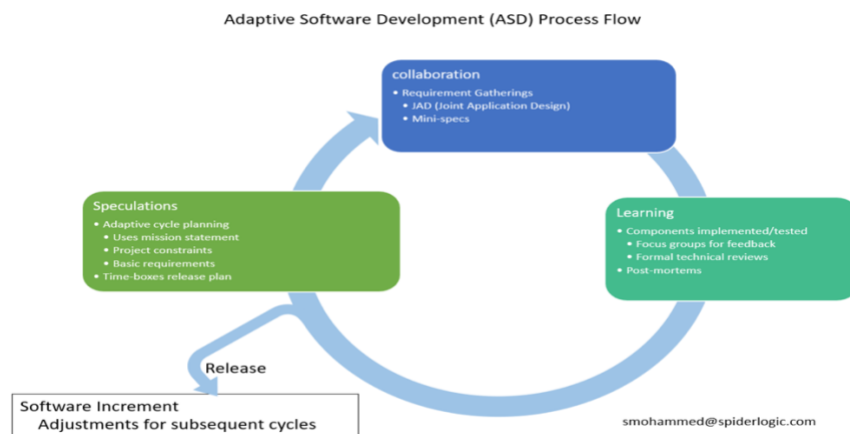


Gambar 1 : Xtreme Programming

2. ASD (Adaptive Software Development)

- Diajukan oleh Jim Highsmith sebagai teknik untuk membangun software dan sistem yang kompleks.
- Filosofi yang mendasari adalah kolaborasi manusia dan tim yang mengatur diri sendiri.
- Aktifitas: Speculation, Collaboration dan Learning

- Aktifitas Speculation : Adaptive cycle planning yaitu menggunakan informasi awal seperti misi dari klien, batasan proyek dan kebutuhan dasar untuk mendefinisikan rangkaian software increment (produk software yang secara berkala diserahkan)
- Aktifitas Collaboration : orang-orang yang bermotivasi tinggi bekerja sama: saling melengkapi, rela membantu, kerja keras, trampil di bidangnya, dan mengkomunikasikan masalah untuk menghasilkan penyelesaian yang efektif.
- Aktifitas Learning: tim pembangun sering merasa sudah tahu semua hal tentang proyek, padahal tidak selamanya begitu. Karena itu proses ini membuat mereka belajar lebih tentang proyek melalui 3 cara:
 - ⇒ Focus group: klien dan pengguna memberi masukan terhadap software
 - ⇒ Formal Technique Reviews: Tim ASD lengkap melakukan review
 - ⇒ Postmortems: Tim ASD lakukan introspeksi pada kinerja dan proses.

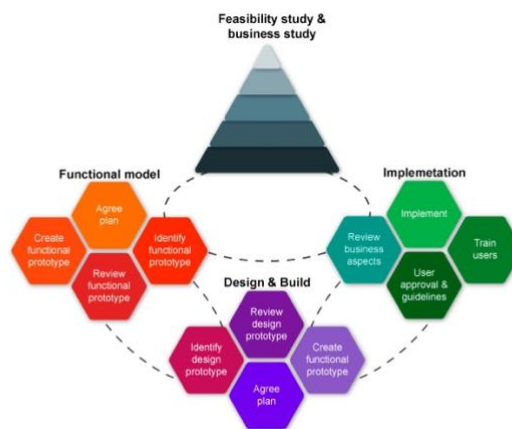


Gambar 2 : Adaptive Software Development

3. DSDM (Dynamic System Development Method)

- menyajikan kerangka kerja (framework) untuk membangun dan memelihara sistem dalam waktu yang terbatas melalui penggunaan prototyping yang incremental dalam lingkungan yang terkondisikan.
- Membangun software dengan cepat: 80% dari proyek diserahkan dalam 20% dari waktu total untuk menyerahkan proyek secara utuh.
- Aktifitas: feasibility study : siapkan requirement, dan batasan, lalu uji apakah sesuai gunakan proses DSDM
- Aktifitas: Business Study: susun kebutuhan fungsional dan informasi, tentukan arsitektur aplikasi dan identifikasi kebutuhan pemeliharaan untuk aplikasi

- Aktifitas: Functional model iteration : hasilkan incremental prototype yang perlihatkan fungsi software ke klien untuk dapatkan kebutuhan lebih jelas dan konfirmasi
- Aktifitas: Design and Build Iteration : cek ulang prototype yang dibangun untuk pastikan bahwa prototype dibangun dengan cara yang memungkinkan fungsi tersebut benar-benar bekerja
- Aktifitas: Implementation: menempatkan software pada lingkungan sebenar sekalipun belum lengkap, atau masih ada perubahan.
- DSDM dapat dikombinasikan dengan XP menghasilkan kombinasi model proses yang mengikuti DSDM dan praktek yang sejalan dengan XP.

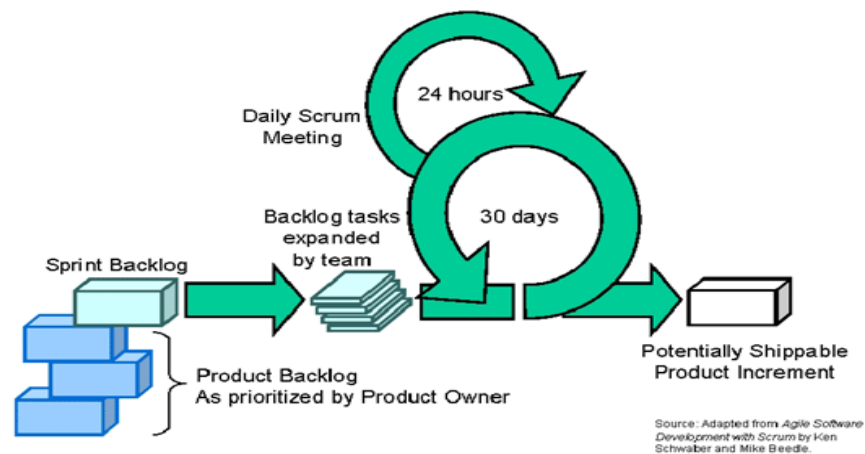


Gambar 3 : Dynamic System Development Method

4. SCRUM

- Diperkenalkan oleh Jeff Sutherland tahun awal tahun 1990an
- Pengembangan berikutnya dilakukan oleh Schwaber dan Beedle
- Scrum memiliki prinsip:
 - ⇒ ukuran tim yang kecil melancarkan komunikasi, mengurangi biaya, dan memberdayakan satu sama lain
 - ⇒ proses dapat beradaptasi terhadap perubahan teknis dan bisnis
 - ⇒ proses menghasilkan beberapa software increment • pembangunan dan orang yang membangun dibagi dalam tim yang kecil
 - ⇒ dokumentasi dan pengujian terus menerus dilakukan setelah software dibangun • proses scrum mampu menyatakan bahwa produk selesai kapanpun diperlukan
- Aktifitas Scrum : Backlog, Sprints, Scrum Meetings, Demo

- Aktifitas Backlog : Backlog adalah daftar kebutuhan yang jadi prioritas klien. Daftar dapat bertambah.
- Aktifitas Sprints: unit pekerjaan yang diperlukan untuk memenuhi kebutuhan yang ditetapkan dalam backlog sesuai dengan waktu yang ditetapkan dalam time-box (biasanya 30hari). Selama proses ini berlangsung backlog tidak ada penambahan.
- Aktifitas Scrum Meeting: pertemuan 15 menit perhari untuk evaluasi apa yang dikerjakan, hambatan yang ada, dan target penyelesaian untuk bahan meeting selanjutnya.
- Aktifitas Demo: penyerahan software increment ke klien didemonstrasikan dan dievaluasi oleh klien.

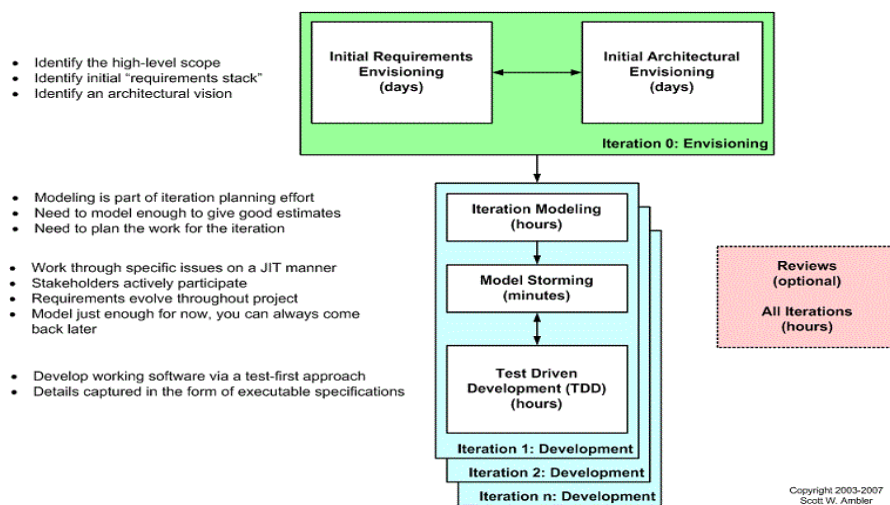


Gambar 4: Scrum

5. AGILE MODELING

- Banyak situasi pembangun software harus membangun sistem bisnis yang besar dan penting. Jangkauan dan kompleksitas sistem harus dimodelkan sehingga dapat dimengerti, masalah dapat dibagi menjadi lebih kecil dan kualitas dapat dijaga pada tiap langkah pembangunan software.
- AM adalah suatu metodologi yang praktis untuk dokumentasi dan pemodelan sistem software.
- AM adalah kumpulan nilai-nilai, prinsip dan praktek-praktek untuk memodelkan software agar dapat diaplikasikan pada software development proyek secara efektif.
- Prinsip dalam AM;
 - ⇒ Membuat model dengan tujuan: tentukan tujuan sebelum membuat model

- ⇒ Menggunakan multiple models: tiap model mewakili aspek yang berbeda dari model lain.
- ⇒ Travel light: simpan model-model yang bersifat jangka panjang saja
- ⇒ Isi lebih penting dari pada penampilan: modeling menyajikan informasi kepada audiens yang tepat. • memahami model dan alat yang yang digunakan untuk membuat software
- ⇒ Adaptasi secara lokal



Gambar 5: Agile Modelling

Agile Manifesto

Agile Manifesto merupakan nilai-nilai yang digunakan dalam mendasari berlangsungnya Agile Software Development. Agile Manifesto muncul dilatarbelakangi oleh frustrasi para pengembang software akibat metode yang digunakan pada saat itu, yaitu Waterfall Model. Salah satunya jeda waktu yang panjang antara penentuan requirement dan product delivery yang dapat berujung pada pembatalan project karena prosesnya yang lama. Kebutuhan klien juga berubah-ubah pada jangka waktu tersebut sehingga produk akhir tidak memenuhi requirement.

Pada tahun 2000, Jon Kern, Kent Beck, Ward Cunningham, Arie van Bennekum, Alistair Cockburn, dan 12 orang lainnya mengadakan pertemuan di Oregon dan kemudian pada tahun 2001 di The Lodge, Utah. Pertemuan tersebut menghasilkan Agile Manifesto yang dituliskan secara formal. Agile Manifesto dapat dilihat melalui [link ini](#). Agile Manifesto terdiri dari 4 nilai utama. Berikut adalah 4 nilai Agile Manifesto yang dikutip dari website tersebut.

1. Individuals and Interactions Over Processes and Tools

Individu dan interaksi antar individu pelaku development lebih diutamakan dibandingkan prosesnya, karena yang merespon terhadap kebutuhan klien adalah individu, bukan proses, sehingga individu lah yang menyetir development yang berlangsung. Dengan demikian, development akan lebih responsif terhadap kebutuhan klien. Selain itu, jika individu sebagai penyetir development, komunikasi antar individu dapat berlangsung dengan fleksibel, misalnya ketika dibutuhkan sesuatu dan perlu dikomunikasikan. Sedangkan jika disetir oleh proses, bisa jadi komunikasi antar anggota harus terstruktur, terjadwal, dan kontennya spesifik. Selain individu itu sendiri, interaksi antar individu juga penting. Mungkin perkataan seperti *“loh, ini codenya kok udah berubah? sejak kapan? percuma dong gue udah bikin code sampe begadang!”* tidak akan terjadi ketika terdapat interaksi yang baik antar individu. Pentingnya interaksi juga menandakan bahwa developer tidak boleh “masa bodoh” dengan pekerjaan developer lain.

2. Working Software Over Comprehensive Documentation

Dulu, banyak waktu yang dialokasikan untuk membuat dokumentasi produk untuk melakukan development maupun delivery. Misalnya penjabaran berbagai spesifikasi produk, requirement, desain interface, dan sebagainya yang ditulis secara detail sehingga menghasilkan dokumen yang cukup banyak. Kemudian dokumen-dokumen tersebut juga memerlukan persetujuan pihak terkait agar development dapat dilaksanakan. Hal ini menyebabkan waktu development semakin panjang (menambah delay untuk delivery). Pada agile, lebih diutamakan working produk yang terdeliver ke klien dengan cepat, sehingga hal-hal terkait dokumentasi tersebut dikurangi. Pada Agile, dokumentasi-dokumentasi requirement diwujudkan sebagai user stories, dan itu sudah cukup untuk developer untuk memulai mengimplementasikan fungsi yang diinginkan. Namun, agile tetap memperhatikan dokumentasi meskipun working software lebih diutamakan.

3. Customer Collaboration Over Contract Negotiation

Pada agile, iterasi yang dilengkapi dengan kolaborasi dengan klien akan lebih efektif untuk development. Klien dapat memberikan feedback terhadap hasil yang telah dibuat pada setiap iterasi sehingga dapat diperbaiki maupun ditambahkan. Ini

menguntungkan kedua pihak, developer dan klien. Developer diuntungkan karena spesifikasi produk menjadi lebih jelas dan mengeliminasi adanya fitur yang kurang feasible dan klien juga diuntungkan karena mendapatkan produk yang lebih sesuai dengan keinginannya. Sedangkan pada Waterfall, klien bernegosiasi dengan development terkait requirement produk secara detail sebelum development dimulai. Klien hanya terlibat sebelum proses dimulai, bukan ketika proses berjalan. Ini akan lebih menyulitkan developer untuk melakukan implementasi sesuai kebutuhan klien. Agile berusaha memastikan produk sesuai atau sedekat mungkin dengan apa yang klien inginkan sebenarnya.

4. Responding to Change Over Following a Plan

Model development tradisional seperti Waterfall tidak fleksibel terhadap perubahan, sehingga sekarang banyak dihindari. Model tersebut menekankan proses development yang sesuai dengan perencanaan dan secara detail serta sesuai urutan yang telah ditetapkan dengan fitur-fitur yang sudah pasti, tidak dapat berubah-ubah. Pada Agile, development lebih fleksibel terhadap perubahan. Agile menggunakan iterasi yang pendek dan fitur baru dapat ditambahkan pada iterasi berikutnya jika diperlukan fitur baru. Menurut Agile, fleksibilitas terhadap perubahan dapat meningkatkan kualitas project sehingga diperoleh nilai lebih pada hasil akhirnya.

Dari keempat nilai tersebut, dapat dijabarkan ke dalam 12 prinsip sebagai Agile Principles. Berikut adalah prinsip-prinsip tersebut yang dikutip dari website agile manifesto.

- 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.

E. Latihan

1. Which of the following is delivered at the end of the Sprint?
2. Product Backlog should be ordered on the basis of?
3. In an Agile environment, what is the main responsibility of a tester?
4. When is a Sprint Retrospective ceremony performed?
5. When can a Sprint be canceled?
6. What should a Development Team do during a Sprint Planning meeting when they have realized that they have selected more than the items they can complete in a Sprint?
7. Who is responsible to measure the Project's performance?
8. What are the main responsibilities of a self-organizing development team?
9. What does a BurnDown Chart display? What are the main responsibilities of a Scrum Master?

F. Rangkuman

Agile Development Methods adalah sekelompok metodologi pengembangan perangkat lunak yang didasarkan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun.

G. Tes Formatif

1. Product Backlog should be ordered on the basis of?
 - a. Value of the items being delivered
 - b. The complexity of the items being delivered
 - c. Size of the items being delivered
 - d. The risk associated with the items
 - e. Based on the Scrum Team choice

 2. An an Agile environment, what is the main responsibility of a tester?
 - a. a. Create test scenarios and test cases
 - b. b. Finding bugs
 - c. c. Create automation scripts
 - d. d. Send test execution reports to the stakeholders
 - e. e. There is no role as a Tester in Scrum

 3. Product Backlog should be ordered on the basis of?
 - a. Value of the items being delivered
 - b. The complexity of the items being delivered
 - c. Size of the items being delivered
 - d. The risk associated with the items
 - e. Based on the Scrum Team choice

 4. An an Agile environment, what is the main responsibility of a tester?
 - a. Create test scenarios and test cases
 - b. Finding bugs
 - c. Create automation scripts
 - d. Send test execution reports to the stakeholders
 - e. There is no role as a Tester in Scrum

 5. When is a Sprint Retrospective ceremony performed?
 - a. Whenever the team suggests
-

- b. At the end of each Sprint
 - c. Whenever needed
 - d. Whenever the Product Owner suggests
 - e. Whenever the Scrum Master suggests
6. When can a Sprint be canceled?
- a. The Sprint items are no longer needed
 - b. Sprint can never be canceled
 - c. When Development is unable to complete the work
 - d. Information required to start the development is not available
 - e. Whenever the Product Owner say

H. Daftar Pustaka

Abrahamsson P, Salo O and Ronkainen J. Agile software development methods (Review and analysis).

Scott W Ambler. Agile model driven development.

Cohen D, Lindvall M, Costa P. Agile software development.

http://en.wikipedia.org/wiki/Agile_Modeling.

http://en.wikipedia.org/wiki/Extreme_Programming.

http://en.wikipedia.org/wiki/Agile_Unified_process.

http://en.wikipedia.org/wiki/Scrum_28development29.

