



## Modul : 13

CPL 230-PENGEMBANGAN PERANGKAT LUNAK

Oleh :

5165 –Kundang K Juman

Prodi : Teknik Informatika

[www.esaunggul.ac.id](http://www.esaunggul.ac.id)

### **pengembangan perangkat lunak dengan metode agile**

#### **AGILE DEVELOPMENT METHODS**

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. Agile development methods merupakan salah satu dari **Metodologi pengembangan perangkat lunak** yang digunakan dalam pengembangan perangkat lunak.

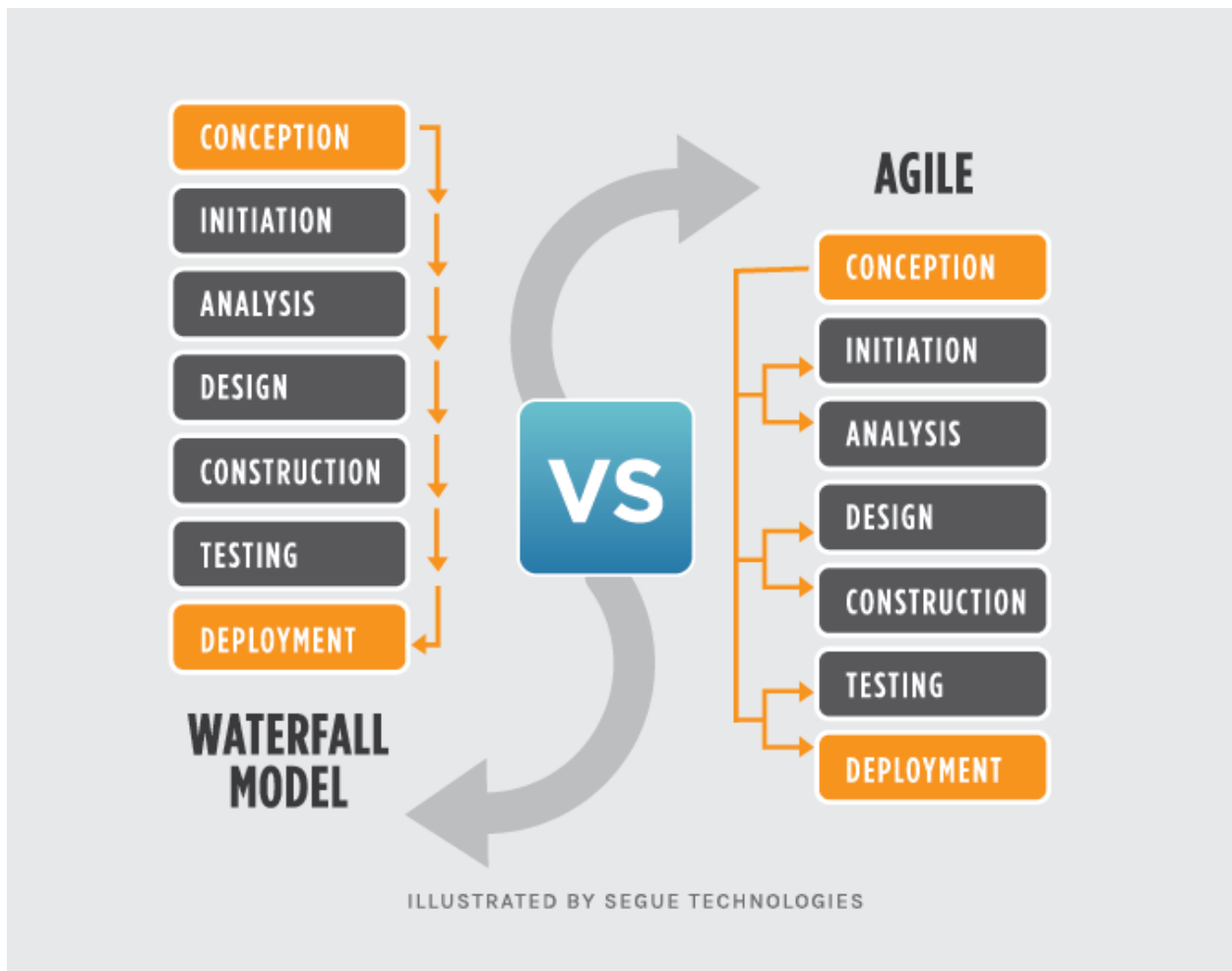
Ada beberapa langkah dalam Agile Development Methods, yaitu :

- 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.

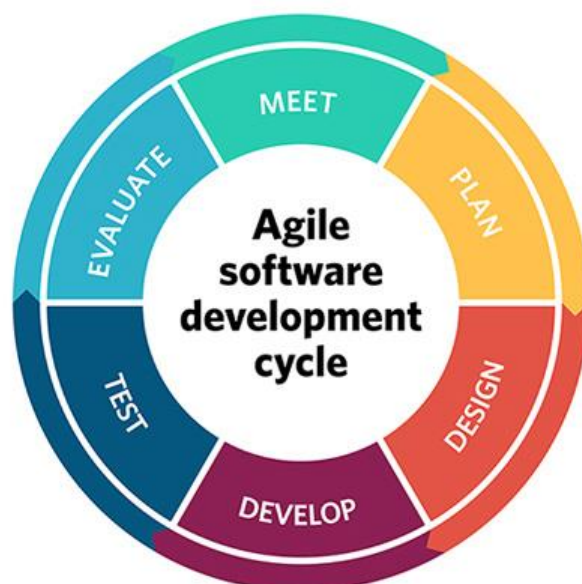
Beberapa kelebihan dari Agile Development Methods :

- 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.

Ada begitu banyak framework yang tersedia untuk menerapkan Agile Development. Diantaranya yaitu Scrum dan Kanban.



## **SCRUM**

Scrum merupakan framework agile yang cukup populer. Scrum memecah organisasi menjadi tim-tim kecil yang mengorganisasikan dirinya sendiri dan mengerjakan tugas-tugas kecil.

Framework ini baik untuk diterapkan dalam membangun kultur perusahaan/ organisasi. Framework ini bisa digunakan untuk membuat sebuah minimum viable product (MVP) maupun pengembangan produk yang kompleks.

Ia juga dapat diterapkan ketika organisasi menginginkan sebuah *culture transition*, bila telah muncul beberapa syarat:

- Tak ada lagi yang sekadar menunggu perintah
- Tim Anda sudah terlihat lebih proaktif
- Tim semakin produktif
- Ada perbaikan dalam hal pengujian maupun pengembangan
- Munculnya rasa tanggung jawab dalam tim

Scrum ini cocok untuk tipe pengembangan produk yang kompleks, yang membutuhkan requirement yang berubah-ubah. Dalam hal ini, klien memegang peranan kunci dalam pengembangan, sehingga menuntut kesiapan klien/ customer/ business owner untuk bisa berinteraksi dengan tim development setiap saat, untuk memastikan kelancaran pengembangan produk/ proyek. Adapun struktur tim pada framework ini mencakup role-role di Scrum (seperti *Scrum Master*, *Product Owner*, dan *Dev Team*), serta tim lintas fungsi (*cross functional*).

## **KANBAN**

Kanban adalah framework yang sangat baik bila digunakan untuk proses kontinyu atau rilis produk harian. Ia juga baik untuk diterapkan dalam proyek-proyek pemeliharaan/ maintenance. Kanban memiliki ciri-ciri alur kerja yang tervisualisasi (*visualized workflow*), tugas yang dipecah menjadi item-item diskrit, progres kerja yang dibatasi, dan pengaturan tugas-tugas dari *backlog*.

Kanban cocok untuk pengembangan produk/ proyek yang membutuhkan requirement-requirement yang sering berubah-ubah. Sama seperti Scrum, klien/ customer/ business owner memegang peran sentral. Bedanya dengan Scrum, klien hanya diperlukan pada periode-periode tertentu saja, untuk melakukan pendetailan. Sementara struktur tim pada Kanban adalah product owner dan tim yang tak harus permanen.

Agile Software Development Pengantar 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. Prinsip Agile Software Development Salah satu ciri dari Agile Software Development adalah tim yang tanggap terhadap perubahan. Mengapa? Karena perubahan adalah hal yang utama dalam pembangunan software: perubahan kebutuhan software, perubahan anggota tim, perubahan teknologi dll. Selain itu Agile Software Development juga melihat pentingnya komunikasi antara anggota tim, antara orang-orang teknis dan businessmen, antara developer dan managernya. Ciri lain adalah klien menjadi bagian dari tim pembangun software. 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: 1. kepuasan klien adalah prioritas utama dengan menghasilkan produk lebih awal dan terus menerus 2. menerima perubahan kebutuhan, sekalipun diakhir pengembangan. 3. Penyerahan hasil/software dalam hitungan waktu dua minggu sampai dua bulan. 4. Bagian bisnis dan pembangun kerja sama tiap hari selama proyek berlangsung 5. Membangun proyek dilingkungan orang-orang yang bermotivasi tinggi yang bekerja dalam lingkungan yang mendukung dan yang dipercaya untuk dapat menyelesaikan proyek. 6. Komunikasi dengan berhadapan langsung adalah komunikasi yang efektif dan efisien 7. Software yang berfungsi adalah ukuran utama dari kemajuan proyek 8. dukungan yang stabil dari sponsor, pembangun, dan pengguna diperlukan untuk menjaga perkembangan yang berkesinambungan 9. perhatian kepada kehebatan teknis dan desain yang bagus meningkatkan sifat agile 10. kesederhanaan penting 11. arsitektur,

kebutuhan dan desain yang bagus muncul dari tim yang mengatur dirinya sendiri 12. 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 meniasati 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 Model 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) a) dipublikasikan oleh Kent Beck tahun 1999 b) Menggunakan pendekatan berorientasi objek c) 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 d) Aktifitas Desain: berprinsip: sederhana.Memanfaatkan kartu CRC (Class-ResponsibilityCollaborator) untuk identifikasi dan mengatur class-class di konsep OO. Jika menemui kesulitan, prototype dibangun [ini namanya spike solution]. Lakukan refactoring, yaitu mengembangkan desain dari program setelah ditulis e) 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 f) Aktifitas Pengujian: menggunakan unit test yang dipersiapkan sebelum pengkodean 2. ADAPTIVE SOFTWARE DEVELOPMENT (ASD) a) Diajukan oleh Jim Highsmith sebagai teknik untuk membangun software dan sistem yang kompleks. b) Filosofi yang mendasari adalah kolaborasi manusia dan tim yang mengatur

diri sendiri. c) Aktifitas: Speculation, Collaboration dan Learning d) Aktifitas Speculation : Adaptive cycle planning yaitu menggunakan informasi awal seperti misi dari klien, batasan proyek dan kebutuhan dasar untuk definisikan rangkaian software increment (produk software yang secara berkala diserahkan) e) Aktifitas Collaboration : orang-orang yang bermotivasi tinggi bekerja sama: saling melengkapi, rela membantu, kerja keras, trampil di bidangnya, dan komunikasikan masalah untuk hasilkan penyelesaian yang efektif. f) 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.

3. DYNAMIC SYSTEMS DEVELOPMENT METHOD a) menyajikan kerangka kerja (framework) untuk membangun dan memelihara sistem dalam waktu yang terbatas melalui penggunaan prototyping yang incremental dalam lingkungan yang terkondisikan. b) Membangun software dengan cepat: 80% dari proyek diserahkan dalam 20% dari waktu total untuk menyerahkan proyek secara utuh. c) Aktifitas: feasibility study : siapkan requirement, dan batasan, lalu uji apakah sesuai gunakan proses DSDM d) Aktifitas: Business Study: susun kebutuhan fungsional dan informasi, tentukan arsitektur aplikasi dan identifikasi kebutuhan pemeliharaan untuk aplikasi e) Aktifitas: Functional model iteration : hasilkan incremental prototype yang perlihatkan fungsi software ke klien untuk dapatkan kebutuhan lebih jelas dan konfirmasi f) Aktifitas: Design and Build Iteration : cek ulang prototype yang dibangun untuk pastikan bahwa prototype dibangun dengan cara yang memungkinkan fungsi tersebut benar-benar bekerja g) Aktifitas: Implementation: menempatkan software pada lingkungan sebenar sekalipun belum lengkap, atau masih ada perubahan. h) DSDM dapat dikombinasikan dengan XP menghasilkan kombinasi model proses yang mengikuti DSDM dan praktek yang sejalan dengan XP.

4. SCRUM a) Diperkenalkan oleh Jeff Sutherland tahun awal tahun 1990an b) Pengembangan berikutnya dilakukan oleh Schwaber dan Beedle c) 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 d) Aktifitas Scrum : Backlog, Sprints, Scrum Meetings, Demo

e) Aktifitas Backlog : Backlog adalah daftar kebutuhan yang jadi prioritas klien. Daftar dapat bertambah. f) 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. g) Aktifitas Scrum Meeting: pertemuan 15 menit perhari untuk evaluasi apa yang dikerjakan, hambatan yang ada, dan target penyelesaian untuk bahan meeting selanjutnya. h) Aktifitas Demo: penyerahan software increment ke klien didemonstrasikan dan dievaluasi oleh klien.

5. AGILE MODELING a) 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. b) AM adalah suatu metodologi yang praktis untuk dokumentasi dan pemodelan sistem software. c) AM adalah kumpulan nilai-nilai, prinsip dan praktek-praktek untuk memodelkan software agar dapat diaplikasikan pada software development proyek secara efektif. d) 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 digunakan untuk membuat software • adaptasi secara lokal

Dari model-model proses di atas dapat diambil beberapa poin penting: 1. komunikasi mempunyai peran penting dalam pembanguna software 2. kebutuhan software tidak mudah untuk diidentifikasi secara lengkap 3. kerja sama dalam tim menentukan kelancaran pembangunan software

Aktifitas yang terjadi di dalam Agile Model Process tetap mengandung aktifitas-aktifitas yang ada pada model proses generasi sebelumnya seperti waterfall, incremental, spiral dan RAD. Selain itu, modelmodel proses di atas tetaplah bukan model proses yang cocok untuk setiap jenis software. Referensi: Ambler, Scott W. "An Introduction to Agile Modeling." Agile Modeling Home Page. 2001-2008. Diakses di <http://www.agilemodeling.com/essays/introductionToAM.htm>. 07/09/08. Pressman, Roger S. Software Engineering : A Practitioner's Approach. 6th Ed. McGraw-Hall. NY. 2005.

Proyek adalah sebuah program kerja terencana dengan waktu dan budget yang ditentukan. Umumnya proyek memiliki tujuan dan sasaran yang harus dicapai. Agar kedua hal tersebut berhasil maka diperlukan usaha dan perencanaan yang tepat.

Tahun 2015 lalu, Maulidan Games mendeklarasikan sebuah resolusi besar yaitu memproduksi 100 game dalam 1 tahun. Dengan bekal sumber daya yang kompeten dan *engine* yang selalu berevolusi untuk mempercepat pengerjaan *game*, tentu tidak lengkap jika tidak diatur dengan manajemen proyek yang baik. Dari sekian banyak metode pengembangan software, Maulidan Games memilih untuk menerapkan metode Agile Project Management. Meskipun pada akhirnya produk yang selesai hanya 25 produk game (25%)



dari target resolusi, dengan adanya penerapan Agile, Maulidan Games telah berhasil meningkatkan produksi sebesar 800% daripada tahun sebelumnya (2014) yang hanya 3 produk game dalam 1 tahun.

Agile Project Management adalah metodologi manajemen proyek yang mempunyai adaptabilitas tinggi terhadap perubahan yang terjadi pada setiap elemen-elemennya. Metode ini memecah sebuah proyek besar menjadi beberapa bagian kecil yang akan di-*deliver* secara *incremental* kepada *publisher / customer / end user*. *Delivery* selalu dilakukan tiap 1-4 minggu sampai proyek tersebut selesai. Proses ini disebut iterasi (*iteration*) atau *sprint*. Cara ini dapat memberikan kepuasan kepada *customer* karena developer selalu memberikan *delivery* tepat waktu secara kontinyu. Pada tiap iterasi, *developer* akan memberikan *delivery* kepada *customer* bagian *software* atau *game* yang telah dibagi sebelumnya. Namun, *developer* harus memastikan bagian tersebut dapat bekerja dengan baik. Oleh karena itu, *review* dan *testing* harus selalu dilakukan secara kontinyu untuk menjamin kualitas produk. Pada tahap ini, *customer* selalu memberikan *feedback* berupa informasi maupun arahan yang menjadi petunjuk penting bagi developer dalam pengerjaan proyek. *Feedback* juga bisa berupa perubahan terhadap beberapa elemen fitur dalam *game*. Developer agile harus mampu beradaptasi dengan hal-hal tersebut, bahkan bila perlu melakukan perencanaan ulang jika rencana awal sudah tidak relevan. Proses ini akan terus dilakukan hingga proyek sampai pada tahap produk siap di-*deploy*.

Ada beberapa hal yang membedakan Agile Project Management dengan yang lainnya, diantaranya :

- Agile menekankan bahwa seluruh anggota **tim harus bersatu menjadi sebuah tim yang solid** dan mampu menjalin kerjasama yang kompak. Tim tersebut terdiri dari *developer, quality assurance, project management, dan customer*
- Komunikasi secara frekuentif adalah salah satu faktor kunci yang bisa membuat *teamwork* menjadi solid. Oleh karena itu, tim agile selalu melakukan **rapat harian** yang membahas progress dan strategi proyek saat ini serta mencari solusi jika ada permasalahan
- Melakukan ***delivery* dalam waktu singkat** yaitu 1 - 4 minggu. Proses ini disebut *sprint* atau *iteration*

- Teknik **komunikasi terbuka** yang memungkinkan tiap anggota tim (termasuk *customer*) untuk memberikan masukan berupa *feedback* dan pandangannya terhadap proyek tersebut. Masukan tersebut akan menjadi pertimbangan dalam pengerjaan. Dengan tensi yang tinggi dalam proses *development*-nya, maka bisa dikatakan bergabung dalam sebuah tim Agile sama seperti bekerja dalam sebuah *mini startup*. Proyek selalu dinamis dengan iklim kerja cepat dan mendukung tiap anggota tim untuk terus berkembang. Anggota tim ini adalah individu-individu dengan semangat tinggi, *passionate*, dan selalu termotivasi. Setiap anggota tim harus mempunyai rasa kepemilikan terhadap proyek, sehingga mereka akan selalu berusaha maksimal untuk menghasilkan produk dengan kualitas terbaik.

Kekurangan dan kelebihan agile

Dalam situasi pembangunan 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. Agile Modeling adalah suatu metodologi yang praktis untuk dokumentasi dan pemodelan system software. Agile Modeling adalah kumpulan nilai-nilai, prinsip dan praktek-praktek untuk memodelkan software agar dapat diaplikasikan pada software development proyek secara efektif.

#### **Prinsip dalam Agile Modeling :**

- 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 digunakan untuk membuat software
- Adaptasi secara local

#### **Kelebihan**

- Meningkatkan kepuasan kepada klien.
- Dapat melakukan review pelanggan mengenai software yang dibuat lebih awal.
- Pembangunan system dibuat lebih cepat.

- Mengurangi resiko kegagalan implementasi software dari segi non-teknis.
- Jika pada saat pembangunan system terjadi kegagalan kerugian dari

materi relatif kecil.

### **Kekurangan**

- Developer harus selalu siap dengan perubahan karena perubahan akan selalu diterima.
- Agile tidak akan berjalan dengan baik jika komitmen tim kurang.
- Tidak cocok dalam skala tim yang besar (>20 orang).
- Perkiraan waktu release dan harga perangkat lunak sulit ditentukan.

### **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 di deployment.
- **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

**Metode Scrum** merupakan framework untuk manajemen pengembangan software dengan karakteristik cekatan dan bersifat iteratif dan incremental. Scrum mendefinisikan dirinya fleksibel, strategi pengembangan yang menyeluruh dimana seluruh tim bekerja sebagai satu unit dalam mencapai sebuah gol yang sama.

### **Proses penting dalam Scrum antara lain:**

#### **Cara Implementasi Scrum**

Untuk mencoba metode Scrum, Anda bisa memulai dengan mengerjakan satu proyek yang sederhana dan singkat. Dengan begitu, Anda bisa membantu tim Anda untuk memiliki pola

pikir yang benar mengenai metode Scrum. Perlu diingat bahwa sama dengan kerja tim pada umumnya, beberapa orang dalam tim akan menyukainya dan beberapa orang akan membencinya. Ini sangat normal dan Anda harus mendorong orang-orang di tim untuk mencobanya dengan benar sebelum mereka menyerah. Jika seseorang akhirnya menyerah, lepaskan mereka dari tim dan biarkan orang lain masuk. Untuk memulainya, Anda bisa mengikuti langkah-langkah berikut:

### **1. Tentukan Tim Scrum pertama Anda**

Tim terdiri dari 5-9 anggota. Semua anggota ini memiliki kombinasi kompetensi dan dapat mencakup untuk menjadi developer, penguji, pendukung, perancang, analisis bisnis, dan lain-lain. Semua anggota terus bekerja sama. Tim itu sendiri bertanggung jawab untuk memastikan bahwa mereka akan memproduksi produk yang bisa dipresentasikan di akhir setiap sprint.

### **2. Tentukan panjang atau lamanya Sprint Anda**

Sprint adalah tenggang waktu yang berlangsung antara 7 dan 30 hari, dan biasanya tetap sama panjang selama proyek berlangsung. Tentukan tenggang waktunya dalam sebuah meeting perencanaan dan tim tersebut harus berkomitmen untuk menyelesaikan pekerjaan ini. Pada akhir sprint, Anda sebaiknya mengadakan sebuah review atau ulasan (bisa berupa meeting) dengan demonstrasi hasil kerja. Disini perbaikannya ditinjau dan dikerjakan agar sprint berikutnya bisa direncanakan. Jika Anda masih tidak yakin dengan tenggang waktu, Anda bisa mulai dengan 2 minggu terlebih dahulu.

### **3. Tunjuk Seorang Master Scrum**

Scrum Master adalah seorang katalisator untuk sebuah kelompok scrum. Mereka memastikan bahwa kelompok scrum bekerja secara efektif dan progresif. Jika terjadi hambatan, master Scrum akan menindaklanjuti dan menyelesaikan masalah untuk tim tersebut.

Master Scrum memang bisa dianggap sebagai manajer proyek untuk tim tersebut, kecuali orang tersebut tidak boleh mendikte apa yang tim kerjakan dan seharusnya tidak terlibat dalam pengelolaan mikro sekalipun. Master Scrum akan membantu tim dalam merencanakan pekerjaan sprint yang akan datang.

#### **4. Tunjuk Pemilik Produk (Product Owner)**

Pemilik Produk harus merupakan seseorang yang dapat bertanggung jawab memastikan tim menghasilkan sebuah produk yang bisa dipresentasikan dan pasarkan ke bisnis, klien atau siapapun yang menginginkan hasil proyek tersebut (pembeli akhir). Pemilik Produk biasanya menuliskan persyaratan sehubungan dengan apa yang diinginkan dari produk tersebut dalam bentuk sebuah cerita, kemudian memprioritaskan item-item dalam proses pembuatan, dan memasukkannya ke backlog.

#### **5. Buat Backlog Produk Awal**

Produk backlog adalah daftar keinginan yang berupa semua cerita pengguna yang diharapkan akan dibuat dan diselesaikan dalam proyek tersebut. Cerita yang paling penting harus berada di urutan teratas daftar, jadi keseluruhan simpanan tersusun secara teratur berdasarkan urutan

Jika sebuah epic meningkat dalam prioritas, sebaiknya masukkan detail yang lebih rinci agar tim dapat mulai mengerjakannya. Pemilik Produk bebas untuk memprioritaskan kembali backlog sesuai dengan keinginan dia.

Contoh stories yang bisa dikembangkan menjadi produk:

- Sebagai power user, saya bisa menentukan file atau folder yang akan di backup berdasarkan ukuran file, tanggal yang dibuat dan tanggal dimodifikasi.
- Sebagai pembelanja buku, saya bisa membaca ulasan sebuah buku pilihan untuk membantu saya memutuskan apakah akan membelinya.
- Seorang nasabah bank bisa mengganti PIN-nya

#### **6. Rencanakan dan Mulailah Sprint Pertama Anda**

Berdasarkan prioritas backlog, tim sekarang mengambil item dari daftar (biasanya dari paling atas). Tim lalu melakukan brainstorming dan memutuskan apa dan berapa banyak yang bisa mereka selesaikan dalam sprint mendatang. Ini disebut rapat perencanaan sprint.

Begitu tim setuju, sprint dimulai dan tim dapat mulai mengerjakan proyeknya.

#### **7. Tutup Arus dan Mulai Sprint Berikutnya**

Bila akhir tenggat waktu sudah tercapai dan semua pekerjaan yang direncanakan sudah lengkap dilakukan, tergantung pada tim untuk memutuskan apakah pekerjaan yang tersisa harus ditransfer ke sprint berikutnya atau dimasukkan kembali ke dalam backlog. Setiap sebuah tim menyelesaikan tugasnya, tim disarankan untuk melakukan retrospektif di mana mereka mendiskusikan apa yang berjalan dengan baik dan apa yang bisa diperbaiki untuk sprint berikutnya. Setelah itu, pertemuan perencanaan sprint untuk sprint berikutnya dimulai dan prosesnya diulang. Tidak ada batasan untuk jumlah sprint kecuali jika ditetapkan oleh deadline (berdasarkan budget atau waktu) atau seluruh backlog selesai. Jika tidak satu pun kriteria ini terpenuhi, sprint terus berlanjut tanpa batas waktu.

## **Kegiatan yang harus dilakukan sebagai bagian dari Scrum**

### **1. Rapat Perencanaan**

Pertemuan Perencanaan Sprint adalah titik awal Scrum. Ini adalah pertemuan dimana seluruh tim Scrum berkumpul; Bekerja sama dengan Product Owner dan Master Scrum, tim memilih story dari backlog dan brainstorm secara bersama-sama. Berdasarkan percakapan, kelompok Scrum memutuskan kompleksitas story dan memutuskan mana yang harus masuk ke sprint.

### **2. Melengkapi Pekerjaan**

Seperti judulnya, perlu bagi tim untuk menyelesaikan pekerjaan untuk kemajuan proyek. Orang-orang di tim bekerja pada epics atau story sampai mereka selesai dan kemudian mereka beralih ke yang berikutnya. Biasanya story disusun di papan dengan langkah-langkah terpisah, jadi mudah untuk merasakan bagaimana sprint tersebut berkembang.

### **3. Pertemuan Scrum Harian**

Melalui siklus sprint, setiap hari tim scrum bertemu maksimal lima belas menit (biasanya di pagi hari). Ada 3 hal yang akan dibahas oleh setiap member tim, yaitu:

- Apa yang saya lakukan kemarin?
- Apa yang ingin saya lakukan hari ini?
- Jelaskan jika orang tersebut memiliki sesuatu yang menghalangi mereka menyelesaikan pekerjaan mereka pada hari itu.

#### **4. Pertemuan Review Sprint**

Setelah setiap sprint, tim mengadakan Sprint Review Meeting untuk menunjukkan progress kerja kepada Product Owner dan orang lain yang tertarik. Pertemuan tersebut harus menampilkan demonstrasi langsung, bukan sebuah laporan. Pemilik Produk dapat memverifikasi story sesuai dengan kriteria penerimaan mereka.

Refrensi :

1. <https://slideplayer.info/slide/12295162/>
2. [https://www.researchgate.net/publication/329116904\\_I\\_PROSES\\_IMPLEMENTASI\\_SISTEM\\_INFO\\_RMASI](https://www.researchgate.net/publication/329116904_I_PROSES_IMPLEMENTASI_SISTEM_INFO_RMASI)
3. Roger Presman, Software engineering