

Nama : Monica Tifani Zahara
NRP 171111077
Tugas 2 : Rekayasa Perangkat Lunak

Agile Method

1. *Extreme Programming (XP)*

Extreme Programming (XP) merupakan salah satu *Agile Processes* yang cukup terkenal. Extreme programming menekankan pada kepuasan *client* karena mampu menghasilkan suatu perangkat lunak yang *client* butuhkan sesuai dengan tanggal yang telah disepakati. Metode ini merupakan metode yang biasa digunakan oleh *developer* yang profesional dan berpengalaman karena dalam pengerjaannya metode ini membutuhkan komitmen yang sangat tinggi serta kemampuan untuk beradaptasi dengan baik terhadap perubahan kebutuhan pelanggan atau klien. *Extreme programming* sangat menekankan kerja tim yang kolaboratif. Manajer, *client*, dan *developer* merupakan mitra yang setara.

Untuk meningkatkan mutu perangkat lunak yang dihasilkan, *Extreme Programming (XP)* menerapkan 5 nilai penting, diantaranya :

a. *Communication*

Setiap orang adalah bagian dari tim, dimana mereka akan bekerja sama dalam segala hal mulai dari *requirements* hingga *code*. Setiap orang yang terlibat akan memberikan solusi yang terbaik untuk permasalahan yang dihadapi.

b. *Simplicity*

Memberikan hal yang diminta dan dibutuhkan secara tidak berlebihan. Langkah-langkah kecil dan sederhana dipilih untuk mencapai tujuan sehingga dapat mengurangi resiko kegagalan yang terjadi. Maintain dalam jangka panjang serta biaya yang masuk akal juga tidak luput dari aspek yang diperhatikan dalam nilai ini.

c. *Feedback*

Perangkat lunak didemonstrasikan lebih awal untuk mengetahui setiap perubahan yang sekiranya diperlukan. Hal ini diperlukan untuk mengetahui kesesuaian perangkat lunak tersebut dengan apa yang diinginkan dan dibutuhkan oleh *client*.

d. *Respect*

Setiap orang memberi dan merasakan penghormatan yang layak mereka dapatkan sebagai anggota tim.

e. *Courage*

Nilai yang terkandung pada pokok bahasan ini yaitu “*We will tell the truth about progress and estimates. We don’t document excuses for failure because we plan to succeed. We don’t fear anything because no one ever works alone. We will adapt to changes when ever they happen.*” (Powell, 2017).

- **Kelebihan dan kekurangan *Extreme Programming (XP)***

- **Kelemahan**

- Pekerjaan tambahan
- Konsumen harus berpartisipasi dalam proses pengerjaannya
- Investasi waktu yang relatif lebih besar
- Biaya yang relatif tinggi
- Membutuhkan disiplin diri untuk berlatih

- **Kelebihan**

- Lebih dekat dengan konsumen
- Perangkat lunak stabil karena melalui pengujian yang berkelanjutan
- Meminimalisir kesalahan program
- Tepat waktu, tim bekerja dengan kecepatan mereka sendiri.
- Perubahan dapat dilakukan dengan pemberitahuan singkat
- Kode jelas dan mudah dipahami

2. *Adaptive Software Development (ASD)*

Teknik atau metode ini biasanya digunakan untuk membangun *software* dan sistem yang kompleks. Filosofi yang mendasari *adaptive software development* adalah kolaborasi manusia dan tim yang mengatur diri sendiri. Sistem kerja *adaptive software development* adalah *collaboration and learning*.

- **Kekurangan dan kelebihan *Adaptive software development (ASD)***

- **Kelebihan**

- Menambah produktivitas tim
- Menambah kualitas perangkat lunak

- Menambah kepuasan kepada klien
- Menghemat biaya produksi
- Mengurangi resiko kegagalan implementasi dari segi non-teknis

➤ **Kekurangan**

- Metode ini tidak akan berjalan dengan baik jika komitmen tim tersebut kurang
- Metode ini tidak cocok dengan skala tim yang besar(lebih dari 20 orang)
- Perkiraan waktu rilis dan harga perangkat lunak sulit ditentukan. Tergantung kesulitan pengerjaan.

3. *Dynamic System Development Method (BSDM)*

Dynamic System Development Method (DSDM) adalah kerangka kerja *agile* yang membahas seluruh siklus hidup proyek dan dampaknya terhadap bisnis. Seperti filosofi *agile* yang lebih luas, DSDM adalah pendekatan berulang untuk pengembangan perangkat lunak, dan kerangka kerja ini secara eksplisit menyatakan "proyek apa pun harus diselaraskan dengan tujuan strategis yang jelas dan fokus pada pemberian awal manfaat nyata bagi bisnis." Kerangka kerja ini dibangun di atas empat prinsip: studi kelayakan dan bisnis, model fungsional dan iterasi prototipe, desain dan pembangunan iterasi, dan implementasi.

- **Kelebihan dan kekurangan *Dynamic System Development Method (DSDM)***

➤ **Kelebihan**

- Fungsi dasar produk dapat disampaikan dengan cepat
- Pengembang memiliki akses mudah ke pengguna akhir
- Proyek diselesaikan andal tepat waktu

➤ **Kekurangan**

- Mahal untuk diterapkan
- Tidak ideal untuk organisasi kecil

4. *Scrum*

Scrum adalah kerangka kerja di mana orang dapat mengatasi masalah adaptif yang kompleks, sementara secara produktif dan kreatif memberikan produk dengan nilai setinggi mungkin. Scrum sendiri adalah kerangka kerja sederhana untuk kolaborasi tim yang efektif pada produk yang kompleks. Rekan pencipta Scrum, Ken Schwaber dan Jeff Sutherland telah menulis *The Scrum Guide* untuk menjelaskan Scrum dengan jelas dan ringkas.

- **Kelebihan dan kekurangan Scrum**

- **Kelebihan**

- Scrum dapat membantu tim menyelesaikan hasil proyek dengan cepat dan efisien
- Scrum memastikan penggunaan waktu dan uang secara efektif
- Proyek-proyek besar dibagi menjadi sprint yang mudah dikelola
- Perkembangan diberi kode dan diuji selama tinjauan sprint
- Berfungsi dengan baik untuk proyek pembangunan yang bergerak cepat
- Tim mendapatkan visibilitas yang jelas melalui rapat scrum
- Scrum, gesit, mengadopsi umpan balik dari pelanggan dan pemangku kepentingan
- Sprint pendek memungkinkan perubahan berdasarkan umpan balik jauh lebih mudah
- Upaya individu dari setiap anggota tim terlihat selama pertemuan scrum harian

- **Kekurangan**

- Scrum sering mengarah ke ruang lingkup creep, karena kurangnya tanggal akhir yang pasti
- Kemungkinan kegagalan proyek tinggi jika individu tidak terlalu berkomitmen atau kooperatif
- Mengadopsi kerangka kerja Scrum di tim-tim besar itu sulit
- Kerangka kerja ini hanya dapat berhasil dengan anggota tim yang berpengalaman
- Rapat harian terkadang membuat frustrasi anggota tim
- Jika ada anggota tim pergi di tengah-tengah proyek, itu dapat memiliki dampak negatif yang sangat besar pada proyek
- Kualitas sulit diimplementasikan sampai tim melewati proses pengujian yang agresif

5. *Crystal*

Metodologi *Crystal* adalah salah satu pendekatan pengembangan perangkat lunak yang paling ringan dan mudah beradaptasi. Kristal sebenarnya terdiri dari keluarga metodologi tangkas seperti *Crystal Clear*, *Crystal Yellow*, *Crystal Orange* dan lainnya, yang karakteristik uniknya didorong oleh beberapa faktor seperti ukuran tim, kekritisitas sistem, dan prioritas proyek. Keluarga *Crystal* ini membahas realisasi bahwa setiap proyek mungkin memerlukan serangkaian kebijakan, praktik, dan proses yang sedikit disesuaikan untuk memenuhi karakteristik unik proyek. Beberapa prinsip utama *Crystal* termasuk kerja tim, komunikasi, dan kesederhanaan, serta refleksi untuk

sering menyesuaikan dan meningkatkan proses. Seperti metodologi proses gesit lainnya, *Crystal* mempromosikan pengiriman perangkat lunak yang berfungsi lebih awal, sering, keterlibatan pengguna yang tinggi, kemampuan beradaptasi, dan penghapusan birokrasi atau gangguan.

- **Kelebihan dan kekurangan *Crystal***

- **Kelebihan**

- Mengizinkan tim bekerja dengan cara yang mereka anggap paling efektif
- Memfasilitasi komunikasi tim langsung, transparansi, dan akuntabilitas
- Pendekatan adaptif memungkinkan tim merespons dengan baik terhadap perubahan persyaratan

- **Kelemahan**

- Kurangnya rencana yang telah ditetapkan dapat menyebabkan cakupan *creep*
- Kurangnya dokumentasi dapat menyebabkan kebingungan

6. ***Feature Driven Development (FDD)***

Feature Driven Development (FDD) adalah *customer-centric*, *iterative*, dan *incremental*, dengan tujuan memberikan hasil perangkat lunak yang nyata dan efisien. FDD mendorong pelaporan status di semua tingkatan, yang membantu melacak kemajuan dan hasil.

FDD memungkinkan tim untuk memperbarui proyek secara teratur dan mengidentifikasi kesalahan dengan cepat. Plus, klien dapat diberikan informasi dan hasil substansial kapan saja. FDD adalah metode favorit di antara tim pengembangan karena membantu mengurangi dua pembunuh moral yang dikenal di dunia *development*: Kebingungan dan pengerjaan ulang.

- **Kelebihan dan kekurangan *Feature Driven Development (FDD)***

- **Kelebihan**

- Proses lima langkah sederhana memungkinkan untuk pengembangan yang lebih cepat
- Memungkinkan tim yang lebih besar untuk memajukan produk dengan kesuksesan yang berkelanjutan
- Memanfaatkan standar pengembangan yang telah ditentukan, sehingga tim dapat bergerak cepat

➤ **Kekurangan**

- Tidak bekerja secara efisien untuk proyek yang lebih kecil
- Dokumentasi yang kurang tertulis, yang dapat menyebabkan kebingungan
- Sangat tergantung pada pengembang atau pemrogram utama.

7. ***Lean Software Development (LSD)***

Lean Software Development (LSD) adalah kerangka kerja dalam agile yang didasarkan pada pengoptimalan waktu dan pengembangan sumber daya, menghilangkan pemborosan, dan hanya memberikan apa yang dibutuhkan produk. Pendekatan Lean juga sering disebut sebagai strategi *Minimum Viable Product (MVP)*, di mana tim merilis versi minimal-minimum produknya ke pasar, untuk kemudian belajar dari *feedback* pengguna apa yang mereka suka, tidak suka dan apa hal yang ingin ditambahkan, dan kemudian iterasi dibuat berdasarkan umpan balik ini.

- **Kelebihan dan kekurangan *Lean Software Development (LSD)***

➤ **Kelebihan**

- Pengiriman cepat. Berfokus hanya pada fitur yang paling penting membantu memberikan MVP lebih cepat dan menghindari membuang waktu pada fitur yang tidak penting.
- Biaya pengembangan rendah. Efisiensi model juga membantu mengurangi biaya yang akan terbuang sia-sia pada perubahan dan fitur yang tidak perlu.
- Kualitas pengembangan yang lebih baik. Lebih fokus pada lebih sedikit fitur menyiratkan kualitas pengembangan yang lebih baik.

➤ **Kekurangan**

- Sangat tergantung pada tim yang terlibat, membuatnya tidak scalable seperti kerangka kerja lainnya
- Bergantung pada dokumentasi yang kuat, dan kegagalan untuk melakukannya dapat menghasilkan kesalahan pengembangan

8. ***Agile Modelling***

Agile Modeling (AM) adalah metodologi berbasis praktik untuk pemodelan dan dokumentasi sistem berbasis perangkat lunak. Ini dimaksudkan sebagai kumpulan nilai, prinsip, dan praktik untuk pemodelan perangkat lunak yang dapat diterapkan pada proyek pengembangan perangkat lunak dengan cara yang lebih fleksibel daripada metode pemodelan tradisional.

- **Kelebihan dan kekurangan *Agile Modeling* (AM)**

- **Kelebihan**

- Kepuasan pelanggan dengan pengiriman perangkat lunak yang cepat dan berkelanjutan.
- Orang dan interaksi lebih ditekankan daripada proses dan alat.
- Pelanggan, pengembang, dan penguji terus berinteraksi satu sama lain.
- Perangkat lunak yang berfungsi sering dikirimkan (berminggu-minggu bukannya berbulan-bulan).
- Percakapan tatap muka adalah bentuk komunikasi terbaik.
- Akrab, kerja sama harian antara pelaku bisnis dan pengembang.
- Perhatian terus menerus terhadap keunggulan teknis dan desain yang baik.
- Adaptasi reguler untuk perubahan keadaan.
- Bahkan perubahan terlambat dalam persyaratan disambut.

- **Kekurangan**

- Dalam hal beberapa pengiriman perangkat lunak, terutama yang besar, sulit untuk menilai upaya yang diperlukan pada awal siklus hidup pengembangan perangkat lunak.
- Ada kurangnya penekanan pada perancangan dan dokumentasi yang diperlukan.
- Proyek dapat dengan mudah diambil keluar jalur jika perwakilan pelanggan tidak jelas apa hasil akhir yang mereka inginkan.
- Hanya programmer senior yang mampu mengambil keputusan yang diperlukan selama proses pengembangan. Oleh karena itu tidak ada tempat untuk programmer pemula, kecuali jika dikombinasikan dengan sumber daya yang berpengalaman.

9. ***Agile Unified Process* (AUP)**

Agile Unified Process (AUP) adalah versi sederhana dari *Rational Unified Process* (RUP) yang dikembangkan oleh Scott Ambler. *Rational unified process*, adalah suatu kerangka pengembangan perangkat lunak iteratif yang dibuat oleh *Rational Software*, suatu divisi dari IBM sejak 2003. *Rational unified process* bukanlah suatu proses dengan aturan yang konkrit, melainkan suatu kerangka proses yang dapat diadaptasi dan dimaksudkan untuk disesuaikan oleh tim pengembang perangkat lunak yang akan memilih elemen proses disesuaikan dengan kebutuhan mereka. Model ini membagi suatu sistem aplikasi menjadi beberapa komponen sistem dan memungkinkan para

developer aplikasi untuk menerapkan metoda iterative (analisis, disain, implementasi dan pengujian) pada tiap komponen.

- **Kelebihan dan kekurangan *Agile Unified Process* (AUP)**

- **Kelebihan**

- Menyediakan akses yang mudah terhadap pengetahuan dasar bagi anggota tim.
 - Menyediakan petunjuk bagaimana menggunakan UML secara efektif.
 - Mendukung proses pengulangan dalam pengembangan software.
 - Memungkinkan adanya penambahan-penambahan pada proses.
 - Memungkinkan untuk secara sistematis mengontrol perubahan- perubahan yang terjadi pada software selama proses pengembangannya.

- **Kekurangan**

- Metodologi ini hanya dapat digunakan pada pengembangan perangkat lunak yang berorientasi objek dengan berfokus pada UML (Unified Modeling Language).
 - Membutuhkan waktu yang cukup lama dibandingkan XP dan Scrum.

Daftar Pustaka

Powell, Andrew. 2017. *Extreme Programming: What Is It And How Do You Use It?*.
<https://airbrake.io/blog/sdlc/extreme-programming>. Online (Diakses pada 11 Maret 2020)