

Reading Material

Dasar Quality Assurance - SDLC



Software Development Life Cycle

Output Pembelajaran

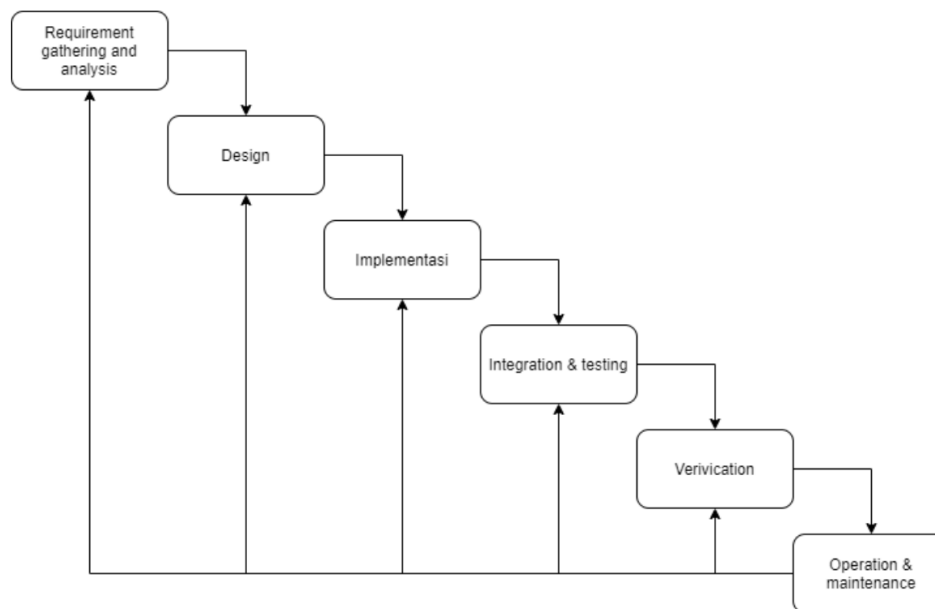
1. Memahami Apa itu SDLC
2. Mengetahui macam-macam metode pengembangan sistem

SDLC (*Software Development Life Cycle*) adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem rekayasa perangkat lunak. Metode SDLC hadir untuk membantu organisasi dalam pengembangan produk. Ada banyak metode SDLC, namun kali ini kita hanya akan membahas beberapa yang cukup sering digunakan dalam pengembangan produk.

Waterfall

Metode waterfall adalah metode kerja tertua yang menekankan fase-fase yang berurutan dan sistematis. Disebut waterfall karena proses mengalir satu arah “ke bawah” seperti air terjun. Metode waterfall ini harus dilakukan secara berurutan sesuai dengan tahap yang ada.

Berikut adalah tahap-tahap pengembangan dalam metode waterfall.



- **Requirement gathering and analysis**
Mengumpulkan kebutuhan secara lengkap untuk dianalisis dan mendefinisikan kebutuhan apa saja yang harus dicapai oleh program. Informasi dapat diperoleh melalui wawancara, diskusi, atau survey.
- **Design**
Melakukan perancangan desain perangkat lunak sebagai perkiraan sebelum dibuatnya kode. Desain sistem dapat dibuat menggunakan *Flowchart*, *Mind Map*, atau *Entity Relationship Diagram* (ERD).
- **Implementasi**
Implementasi ini adalah tahap dimana seluruh desain yang sebelumnya sudah dibuat diubah menjadi kode-kode program. Kode yang dihasilkan masih berbentuk modul-modul yang harus digabungkan di tahap selanjutnya.
- **Integration & testing**
Di tahap ini dilakukan penggabungan modul-modul yang sudah dibuat sebelumnya dan melakukan pengujian untuk mengetahui apakah perangkat lunak yang dibuat telah sesuai dengan desain dan fungsinya atau tidak.
- **Verification**
Di tahap ini, pengguna atau klien yang langsung melakukan pengujian pada sistem, apakah sistem telah sesuai dengan yang disetujui atau belum sesuai.
- **Operation & maintenance**
Tahap ini merupakan tahap terakhir dari model waterfall. Sistem yang sudah selesai dijalankan serta dilakukan pemeliharaan. Pemeliharaan berupa memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

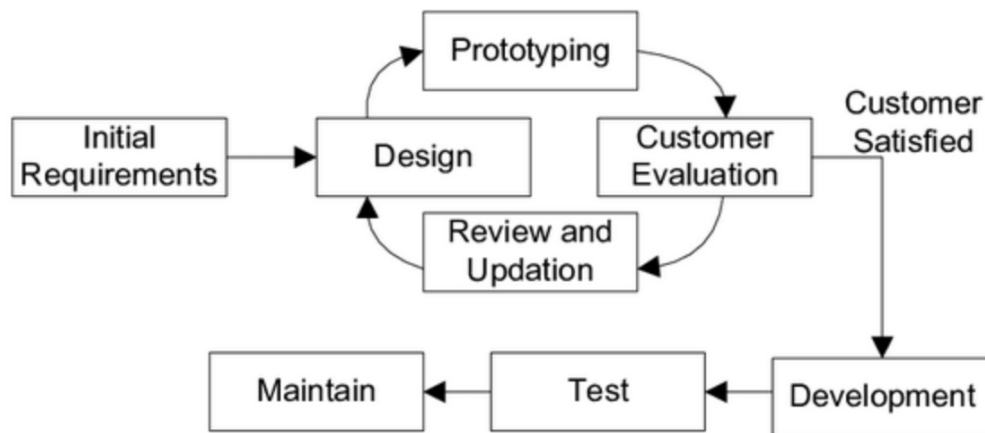
Berikut adalah kelebihan dari metode waterfall:

1. Memiliki proses yang terurut, sehingga pengerjaan dapat terjadwal dengan baik dan mudah.
2. Cocok untuk sistem dengan kompleksitas rendah (*predictable*).
3. Setiap proses yang dilakukan tidak dapat saling tumpah tindih.

Berikut adalah kekurangan dari metode waterfall:

1. Waktu pengerjaan relatif lebih lama, karena harus menunggu tahap sebelumnya selesai.
2. Biaya yang dibutuhkan lebih mahal karena waktu pengembangan yang dibutuhkan lebih lama.
3. Masalah tidak bisa diselesaikan sebelum mencapai tahap maintenance.

Prototype



Kelebihan dari metode prototype:

- Mempersingkat waktu pengembangan perangkat lunak
- Penerapan fitur menjadi lebih mudah, karena pengembang mengetahui apa yang diharapkan

Kekurangan dari metode prototype:

- Proses yang dilakukan untuk analisis dan perancangan terlalu singkat
- Kurang fleksibel jika terjadi perubahan

Agile

Dengan memecah produk menjadi beberapa siklus, model Agile dengan cepat membangun produk yang berfungsi dengan baik dan dianggap sebagai pendekatan pengembangan yang sangat realistis. Itulah mengapa metode Agile lebih sering digunakan untuk pengembangan software. Hal tersebut didukung pula dengan adanya Agile Manifesto.

Dicanangkan oleh Jon Kern, Kent Beck, Ward Cunningham, Arie van Bennekum, Alistair Cockburn, dan 12 orang lainnya, Agile Manifesto memiliki 4 nilai yang mendasarinya. Yakni *Individuals and interactions over processes and tools*, *Working software over comprehensive documentation*, *Customer collaboration over contract negotiation*, serta *Responding to change over following a plan*.



Nilai utama dalam Development Agile adalah memungkinkan tim untuk memberikan nilai lebih cepat, dengan kualitas dan prediksi yang lebih baik, dan bakat yang lebih besar untuk merespons perubahan. Jadi, Bisa disimpulkan jika Agile merupakan model development jangka pendek yang memerlukan adaptasi cepat dan pengembangan terhadap perubahan dalam bentuk apapun.

Keuntungan Agile Software Development

- Keterlibatan Stakeholder
- Transparansi
- Release Awal dan Mudah Diprediksi
- Biaya & Jadwal Bisa Ditentukan
- Memungkinkan Perubahan
- Berfokus Pada Nilai Bisnis
- Fokus pada Pengguna
- Meningkatkan Kualitas

Namun Agile juga memiliki beberapa hal yang perlu diperhatikan:

- Metode ini kurang sesuai dengan tim yang besar (lebih dari 20 orang).
- Tim harus selalu siap, karena perubahan dapat terjadi kapan saja.

Nah, setelah kita tahu bahwa Agile itu cocok untuk pengembangan software, ternyata pada prakteknya ada banyak jenis model turunan dari metode Agile, salah satunya adalah SCRUM.

Scrum adalah kerangka kerja yang digunakan untuk mengimplementasikan pengembangan Agile. Scrum lebih mengutamakan pada kekuatan kolaborasi tim dalam development software.

Tim scrum idealnya terdiri dari 5-9 individu yang kolaboratif. Setiap individu dalam tim bisa memiliki peran-peran seperti berikut ini:

Product Owner	Scrum Master	Scrum Team
<ul style="list-style-type: none">• Mendefinisikan fitur produk.• Pemilik Produk memutuskan tanggal rilis dan fitur yang sesuai.	<ul style="list-style-type: none">• Mengelola tim dan menjaga produktivitas tim.• Mempertahankan daftar blokir dan menghilangkan hambatan dalam pengembangan.	<ul style="list-style-type: none">• Tim biasanya sekitar 5-9 anggota.• Pengembang, perancang dan terkadang penguji, dll.
<ul style="list-style-type: none">• Memprioritaskan fitur sesuai dengan nilai pasar dan profitabilitas produk.• Bertanggung jawab atas profitabilitas produk.• Dapat menerima atau menolak hasil item pekerjaan.	<ul style="list-style-type: none">• Berkoordinasi dengan semua peran dan fungsi.• Melindungi tim dari gangguan eksternal.• Membuat undangan rapat harian scrum, tinjauan sprint, dan perencanaan.	<ul style="list-style-type: none">• Mengatur dan menjadwalkan pekerjaan mereka sendiri.• Memiliki hak untuk melakukan segala sesuatu dalam batas-batas proyek untuk memenuhi tujuan sprint.• Berpartisipasi aktif dalam rapat harian scrum.

Scrum Artifacts

Dalam model scrum, ada beberapa proses yang biasa disebut dengan artifact yang secara berkelanjutan dilakukan di setiap prosesnya.

- User stories: Penjelasan singkat tentang fungsionalitas fitur yang akan dikembangkan.
- Product Backlog: Kumpulan fitur yang diambil untuk produk scrum. Product owner menyiapkan dan memelihara backlog produk, scrum team dapat menambahkan atau melengkapi dengan persetujuan dari product owner.
- Release Backlog: Kerangka waktu di mana jumlah iterasi selesai. Product owner berkoordinasi dengan scrum master untuk memutuskan fitur mana yang harus ditargetkan untuk rilis lebih dulu.

- Sprint: Periode waktu yang ditentukan untuk menyelesaikan fitur, diputuskan oleh product owner dan scrum team, biasanya 1-4 minggu.
- Sprint Backlog: Kumpulan fitur atau daftar tugas yang harus diselesaikan dalam sprint. Backlog dikelola oleh tim namun penentu prioritasnya adalah product owner.
- Burndown chart: Bagan burn-down mewakili kemajuan keseluruhan dari pekerjaan yang sedang berlangsung dan pekerjaan yang diselesaikan selama proses berlangsung.

Proses di dalam Scrum

Sprint Planning

Sebuah sprint dimulai dengan tim menganalisa user story dari release backlog ke dalam sprint backlog. Idealnya sprint planning dipimpin oleh scrum master.

Daily Scrum

Idealnya daily scrum dipimpin oleh scrum master, berlangsung sekitar 15 menit. Selama Daily Scrum, para anggota akan membahas pekerjaan yang diselesaikan pada hari sebelumnya, pekerjaan yang direncanakan untuk hari berikutnya dan masalah yang dihadapi selama sprint. Daily Scrum juga berguna untuk memonitor progress dari kemajuan tim setiap harinya.

Sprint Review/ Retrospective

Kegiatan ini juga dipandu oleh scrum master, berlangsung sekitar 2-4 jam dan membahas apa yang telah dicapai tim dalam sprint terakhir dan pelajaran apa yang didapat.

Quality Assurance di Scrum

Sebagai bagian dari scrum team, ada beberapa hal yang perlu dilakukan oleh seorang Quality Assurance.

- | | |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Sprint Planning | <ul style="list-style-type: none">• Memahami dokumen requirement• Menganalisa MockUp / Design• Bertanya jika ada yang kurang jelas• Memberikan feedback terhadap design/requirement• Estimasi waktu pengujian |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- | | |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Daily Scrum | <ul style="list-style-type: none">• Membuat test scenario dan test case• Melakukan testing berdasarkan skenario yang telah dibuat• Update progress anda pada Daily Stand Up• Laporkan dependensi anda pada Daily Stand Up |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Sprint
Review**

- Memberikan feedback anda kepada tim sebagai modal perbaikan di sprint selanjutnya

**Sprint
Retro**

- Memberikan tanggapan pribadi tentang pengalaman selama sprint berjalan dan apa harapan pada sprint selanjutnya