

## Reading Material

### Mengimplementasikan Proses Testing - Scrum introduction



## READING

Dalam pengembangan produk, framework dipakai untuk menyusun, merencanakan, hingga mengontrol proses pengembangannya.

Framework paling populer yang biasa digunakan pada pengembangan sebuah aplikasi di antaranya ada Agile dan Waterfall.

Kelebihan utama dari Agile adalah fleksibel dan peka terhadap perubahan. Jadi, kalau ada kesalahan, bisa langsung diperbaiki.

Berbeda dengan Waterfall yang harus sistematis dan bertahap. Sekalipun ada kesalahan, harus diselesaikan dulu dari tahap awal sampai tahap akhir, baru bisa diperbaiki.

Oleh karena itu, Agile lebih disukai oleh developer untuk mengembangkan produk.

Waterfall adalah pendekatan tradisional dalam pengelolaan proyek dan pengembangan perangkat lunak. Ini adalah model pengembangan perangkat lunak yang berurutan, linear, dan sistematis. Pendekatan Waterfall mengikuti serangkaian tahapan yang terstruktur, dimulai dari fase perencanaan hingga implementasi dan pemeliharaan.

Metode Waterfall adalah pendekatan tradisional dalam pengelolaan proyek dan pengembangan perangkat lunak. Ini adalah model pengembangan perangkat lunak yang berurutan, linear, dan sistematis. Pendekatan Waterfall mengikuti serangkaian tahapan yang terstruktur, dimulai dari fase perencanaan hingga implementasi dan pemeliharaan.

Berikut adalah langkah-langkah yang umumnya ada dalam model Waterfall:

1. Analisis dan Perencanaan: Tahap ini melibatkan identifikasi kebutuhan, analisis persyaratan, dan perencanaan keseluruhan proyek. Tim menganalisis masalah yang harus dipecahkan, menentukan tujuan, mengidentifikasi batasan, dan merumuskan rencana proyek yang lengkap.

2. **Desain:** Setelah analisis dan perencanaan, tim mulai merancang solusi untuk memenuhi persyaratan proyek. Ini mencakup desain arsitektur perangkat lunak, desain antarmuka pengguna, dan desain detail komponen sistem.
3. **Implementasi:** Setelah desain selesai, tim mulai mengimplementasikan perangkat lunak berdasarkan desain yang telah dibuat. Kode program ditulis dan diuji untuk memastikan berfungsi dengan benar.
4. **Pengujian:** Setelah implementasi, tahap pengujian dilakukan untuk memastikan bahwa perangkat lunak berjalan sesuai dengan persyaratan dan tidak ada bug atau kesalahan yang kritis.
5. **Pengiriman:** Setelah berhasil melewati fase pengujian, perangkat lunak dianggap siap untuk dirilis dan dikirimkan ke pengguna atau pelanggan.
6. **Pemeliharaan:** Tahap ini melibatkan pemeliharaan dan dukungan perangkat lunak setelah dirilis. Perangkat lunak akan tetap dipantau, dan perbaikan atau peningkatan mungkin diperlukan untuk memenuhi kebutuhan yang berkembang.

Kelebihan Metode Waterfall meliputi struktur yang jelas, aliran kerja yang terprediksi, dan dokumentasi yang kuat pada setiap tahap. Ini cocok untuk proyek dengan persyaratan yang terdefinisi dengan baik dan sedikit perubahan yang diharapkan selama proses pengembangan.

Namun, ada beberapa keterbatasan dalam pendekatan Waterfall, termasuk:

1. Kurang fleksibilitas terhadap perubahan persyaratan, yang dapat menyebabkan kesulitan dalam menangani perubahan kebutuhan pelanggan.
2. Risiko tinggi terhadap ketidakcocokan antara persyaratan yang telah ditentukan dan hasil akhir produk.
3. Tidak memungkinkan untuk melihat hasil sebelum tahap implementasi, sehingga menghadapi risiko besar jika ada kesalahan yang terdeteksi terlambat.
4. Kurangnya keterlibatan pelanggan selama proses pengembangan.

Karena keterbatasan-keterbatasan ini, model Waterfall telah menjadi kurang populer dalam beberapa tahun terakhir, terutama di proyek-proyek yang lebih kompleks dan dinamis. Pendekatan Agile dan metodologi lain yang lebih adaptif telah menjadi pilihan yang lebih umum bagi banyak organisasi dalam menghadapi tantangan pengembangan perangkat lunak modern.

Agile adalah metodologi pengembangan software jangka pendek yang membutuhkan adaptasi cepat. Adaptasi ini terkait perubahan dalam bentuk apa pun.

Pengembangan produk menggunakan Agile ini dilakukan secara bertahap, berulang (iterasi), dan increment.

SCRUM adalah suatu kerangka kerja dalam pengembangan perangkat lunak yang berfokus pada pengelolaan proyek secara kolaboratif dan adaptif. Pendekatan ini pertama kali diperkenalkan oleh Jeff Sutherland dan Ken Schwaber pada awal 1990-an. SCRUM menekankan pada pengaturan waktu yang teratur (sprint) dan komunikasi tim yang efektif untuk mencapai tujuan proyek.

### Prinsip Utama SCRUM

1. Transparansi: Semua informasi tentang proyek harus jelas dan tersedia untuk seluruh anggota tim. Ini mencakup tujuan proyek, kemajuan, tantangan, dan kendala yang dihadapi.
2. Inspeksi: Tim secara teratur mengevaluasi hasil pekerjaan yang telah dilakukan untuk memastikan kualitasnya dan memperbaiki hal-hal yang perlu ditingkatkan.
3. Adaptasi: SCRUM menggabungkan umpan balik yang diberikan oleh hasil inspeksi ke dalam proses pengembangan selanjutnya, sehingga memungkinkan tim untuk beradaptasi dengan perubahan kebutuhan dan perubahan lingkungan.

## Struktur Tim dalam SCRUM

1. Product Owner: Bertanggung jawab atas visi produk, menentukan prioritas fitur, dan berkomunikasi dengan para pemangku kepentingan (stakeholder).
2. SCRUM Master: Memastikan tim mengikuti praktik SCRUM dengan benar, membantu menghilangkan hambatan (impediment), dan memastikan kelancaran jalannya sprint.
3. Tim Pengembang: Merupakan anggota tim yang secara aktif mengerjakan tugas dalam sprint untuk mencapai tujuan proyek.

## Artefak SCRUM

### **Product Backlog**

Product Backlog adalah daftar seluruh kebutuhan, fitur, perbaikan, dan pekerjaan yang harus dikerjakan untuk mengembangkan produk atau proyek.

Ini berfungsi sebagai dokumen dinamis yang terus diperbarui dan diprioritaskan oleh Product Owner, seorang anggota tim Scrum yang bertanggung jawab atas pemahaman kebutuhan pengguna dan pemangku kepentingan.

Product Backlog dapat berubah seiring waktu dengan adanya perubahan kebutuhan, pelajaran yang didapat dari iterasi sebelumnya, dan umpan balik dari pengguna.

### **Sprint Backlog**

Sprint Backlog adalah subset dari Product Backlog yang dipilih dan diambil oleh Development Team untuk dikerjakan selama Sprint tertentu. Sprint adalah periode waktu yang terbatas, biasanya antara 1 hingga 4 minggu, di mana tim berkomitmen untuk menyelesaikan pekerjaan berdasarkan Sprint Goal yang telah ditetapkan.

Tidak berubah selama Sprint: Selama Sprint berlangsung, isi dari Sprint Backlog sebaiknya tidak berubah agar tim dapat fokus pada pekerjaan yang telah dipilih.

Sprint Backlog membantu tim untuk memahami pekerjaan yang harus diselesaikan dalam Sprint tersebut, mengukur kemajuan, dan memastikan tim bergerak menuju mencapai Sprint Goal yang telah ditetapkan.

## **User Story**

User Story merupakan pendekatan untuk menggambarkan kebutuhan dari sudut pandang pengguna atau pelanggan. User Story biasanya berbentuk kalimat pendek yang berfokus pada keinginan pengguna terkait fitur atau fungsionalitas yang diharapkan dari sebuah produk atau proyek.

Contoh User Story:

Sebagai seorang pengguna, saya ingin dapat masuk ke aplikasi dengan menggunakan akun media sosial saya.

User Story membantu tim pengembang untuk memahami kebutuhan pelanggan dengan lebih baik dan memastikan bahwa fokus pengembangan berada pada nilai bisnis yang diinginkan oleh pengguna.

## **Story Points**

Story Points adalah satuan untuk mengukur kompleksitas atau usaha yang diperlukan untuk menyelesaikan sebuah User Story. Pengukuran ini bersifat relatif, bukan absolut, dan menggambarkan perkiraan kompleksitas berdasarkan pemahaman tim pengembang tentang User Story tersebut.

Penggunaan Story Points memungkinkan tim untuk menghindari perkiraan waktu yang pasti (seperti jam atau hari) yang seringkali bisa tidak akurat karena banyak faktor yang mempengaruhinya. Sebagai gantinya, tim akan memberi peringkat relatif terhadap tingkat kerumitan atau ukuran pekerjaan setiap User Story.

Hal ini juga memudahkan tim untuk membandingkan dan menilai kesulitan berbagai tugas dengan lebih mudah.

## **Team Velocity**

Team Velocity mengacu pada jumlah Story Points yang selesai oleh tim pengembang dalam satu iterasi Sprint atau periode waktu tertentu. Dengan kata lain, ini adalah ukuran seberapa efisien tim dalam menyelesaikan pekerjaan dalam satu Sprint.

Team Velocity dapat membantu tim dan pemangku kepentingan untuk memperkirakan berapa banyak pekerjaan yang dapat diselesaikan dalam Sprint berikutnya atau dalam jangka waktu tertentu. Pengukuran ini memungkinkan tim

untuk mengelola ekspektasi dan merencanakan lebih efisien dalam menghadapi proyek atau iterasi berikutnya.

### **Burndown Chart**

Burndown Chart adalah grafik yang menggambarkan progres tim dalam menyelesaikan pekerjaan selama Sprint. Grafik ini memvisualisasikan jumlah pekerjaan yang masih harus diselesaikan (sumbu Y) dibandingkan dengan waktu yang berlalu (sumbu X).

Idealnya, garis burndown chart yang mengarah turun menandakan progres yang baik, karena tim semakin mendekati penyelesaian semua User Story yang ada dalam Sprint. Namun, jika garis burndown chart tetap datar atau bahkan mengarah ke atas, ini menunjukkan bahwa tim mengalami kesulitan atau ada hambatan yang menghambat progres.

## **Scrum Event**

### **Sprint Planning**

Sprint Planning Pertemuan di awal sprint di mana tim memilih item dari Product Backlog dan membuat rencana untuk mencapai tujuan sprint.

Ini adalah pertemuan kolaboratif antara Product Owner, Scrum Master, dan Development Team untuk merencanakan pekerjaan yang akan dikerjakan selama Sprint berikutnya.

#### **Penentuan Sprint Goal**

Product Owner menjelaskan tujuan umum dan keinginan bisnis untuk Sprint tersebut.

Development Team membantu menguraikan dan memahami kebutuhan lebih lanjut untuk mencapai Sprint Goal.

Sprint Goal ditetapkan sebagai target yang harus dicapai selama Sprint tersebut.

#### **Pemilihan dan Perkiraan Pekerjaan**

Development Team membantu memilih item-item dari Product Backlog yang akan dimasukkan ke dalam Sprint Backlog untuk dikerjakan selama Sprint.

Item-item ini didekomposisi menjadi tugas-tugas yang lebih detail dan diestimasi dengan Story Points.

Tim juga mempertimbangkan kapasitas dan keahlian anggota tim saat melakukan estimasi pekerjaan.

### **Sprint**

Sprint adalah periode waktu terbatas dan tetap di mana Development Team bekerja untuk mencapai Sprint Goal yang telah ditetapkan dalam Sprint Planning.

Durasi Sprint biasanya berkisar antara 1 hingga 4 minggu, dengan waktu yang lebih pendek lebih disukai untuk meminimalkan risiko perubahan kebutuhan atau arah proyek. Umumnya di Indonesia dengan waktu 2 minggu atau 10 hari kerja.

Waktu Tetap: Durasi Sprint harus konsisten dan tidak berubah selama Sprint berlangsung.

Tujuan Tetap: Sprint memiliki Sprint Goal yang jelas yang harus dicapai dalam waktu Sprint berakhir.

Bebas Gangguan: Selama Sprint berlangsung, perubahan kebutuhan atau arah proyek dihindari untuk memungkinkan fokus penuh pada tujuan yang ditetapkan.

### **Daily Standup Meeting**

Daily Standup Meeting, atau sering disebut juga Daily Scrum, adalah pertemuan harian selama Sprint yang berlangsung singkat dan terstruktur. Tujuannya adalah memungkinkan anggota Development Team untuk berkomunikasi secara singkat dan saling memahami tentang kemajuan mereka dalam mencapai Sprint Goal.

Waktu Terbatas: Pertemuan ini dibatasi hingga 15 menit agar tetap efisien dan fokus.

Tiga Pertanyaan:

Setiap anggota Development Team menjawab tiga pertanyaan:

- Apa yang telah saya lakukan sejak pertemuan terakhir?
- Apa yang akan saya lakukan berikutnya?
- Apa kendala yang saya temui?

### **Sprint Review**

Sprint Review adalah acara yang dilakukan setelah berakhirnya Sprint, di mana Development Team memperlihatkan kepada pemangku kepentingan hasil pekerjaan yang telah selesai dalam Sprint tersebut. Pertemuan ini berfokus pada demo produk yang telah dihasilkan.



Demo Produk: Development Team memperlihatkan hasil pekerjaan yang telah selesai kepada Product Owner dan pemangku kepentingan lain.

Feedback: Pemangku kepentingan memberikan umpan balik tentang hasil pekerjaan dan mungkin memberikan saran untuk perubahan atau peningkatan di masa mendatang.

Potensi Perubahan Product Backlog: Hasil dari Sprint Review dapat mempengaruhi perubahan dalam Product Backlog untuk Sprint selanjutnya.

### **Sprint Retrospective**

Sprint Retrospective adalah acara refleksi yang berlangsung setelah Sprint Review. Tujuannya adalah untuk memungkinkan tim untuk memeriksa proses kerja selama Sprint dan mengevaluasi cara mereka bekerja.

Evaluasi Proses: Tim membahas apa yang telah berjalan dengan baik selama Sprint dan apa yang dapat ditingkatkan dalam proses kerja mereka.

Tindakan Perbaikan: Tim mencatat tindakan perbaikan yang akan diimplementasikan dalam Sprint berikutnya untuk meningkatkan efisiensi dan efektivitas tim.

## **Peranan QA Pada Methode Scrum**

### **QA di Sprint Planning:**

- Menanyakan tentang Goals dalam sprint tersebut
- Menanyakan tentang integrasi yang terkait
- Menanyakan tentang kesiapan dokumen seperti Flow Chart, PRD, ERD dll
- Merencanakan Testing Strategi dan Test Plan
- Meminta card / task pembuatan test scenario dan test case

### **QA di Daily Standup**

- Melaporkan update pembuatan test case dan test scenario
- Melaporkan hasil testing dari yang sudah DONE dari dev
- Melaporkan temuan atau progress aplikasi yang sedang dites

### **QA di Sprint Review**

- Supporting PO untuk demonstrasi produk atau fitur yang sudah dijalankan dalam 1 sprint

- Membantu dalam mengevaluasi kualitas produk dan memastikan bahwa produk memenuhi standar kualitas yang ditetapkan sebelum dianggap "selesai" atau "siap untuk dirilis".

### **QA di Sprint Retrospektif**

- Melakukan review bersama seluruh anggota tim agar sprint berikutnya lebih baik dan kolaboratif

## **Kelebihan & Kekurangan SCRUM**

### **Kelebihan Scrum:**

1. **Transparansi:** Scrum mendorong transparansi dalam setiap aspek proyek. Melalui artefak seperti Product Backlog, Sprint Backlog, dan Burndown Chart, semua anggota tim dan pemangku kepentingan dapat memahami status proyek dengan jelas.
2. **Kolaborasi:** Scrum mempromosikan kerja tim yang kuat dan kolaboratif. Tim Scrum terdiri dari anggota yang berbeda dengan peran dan keterampilan yang berbeda, yang saling bekerja sama untuk mencapai tujuan yang ditetapkan.
3. **Adaptabilitas:** Pendekatan iteratif dalam Scrum memungkinkan tim untuk beradaptasi dengan cepat terhadap perubahan kebutuhan atau perubahan prioritas dalam proyek.
4. **Peningkatan Terus Menerus:** Dengan Sprint Review dan Sprint Retrospective, tim Scrum secara aktif mencari umpan balik dan kesempatan untuk perbaikan berkelanjutan, sehingga meningkatkan kualitas produk secara keseluruhan.
5. **Peningkatan Kecepatan Pengiriman:** Dengan pembagian pekerjaan menjadi iterasi Sprints yang teratur, tim dapat mencapai kecepatan pengiriman yang lebih baik dan menghasilkan increment produk yang bernilai secara teratur.

### **Kekurangan Scrum:**

1. **Struktur Formal:** Beberapa organisasi mungkin merasa bahwa Scrum terlalu formal atau berat untuk proyek-proyek kecil atau tim kecil.

2. Kurangnya Rincian: Dalam beberapa kasus, deskripsi User Story dalam Product Backlog mungkin tidak cukup rinci, menyebabkan kesulitan bagi tim untuk memahami kebutuhan yang tepat.
3. Kesulitan dalam Estimasi: Estimasi pekerjaan dalam Story Points atau unit lainnya dapat menjadi tantangan bagi tim yang belum berpengalaman dalam menggunakan pendekatan ini.
4. Fokus Terlalu Internal: Terkadang, fokus tim Scrum dapat terlalu terpusat pada upaya pengembangan sendiri, dan dapat mengabaikan aspek bisnis yang lebih luas atau masalah integrasi.
5. Perlu Komitmen Tinggi: Scrum mengharuskan keterlibatan dan komitmen tinggi dari semua anggota tim, termasuk Product Owner dan pemangku kepentingan, yang tidak selalu mudah diimplementasikan dalam beberapa situasi organisasi.