

Reading Material

Memahami Testing Principles - Software Testing Life Cycle (STLC)



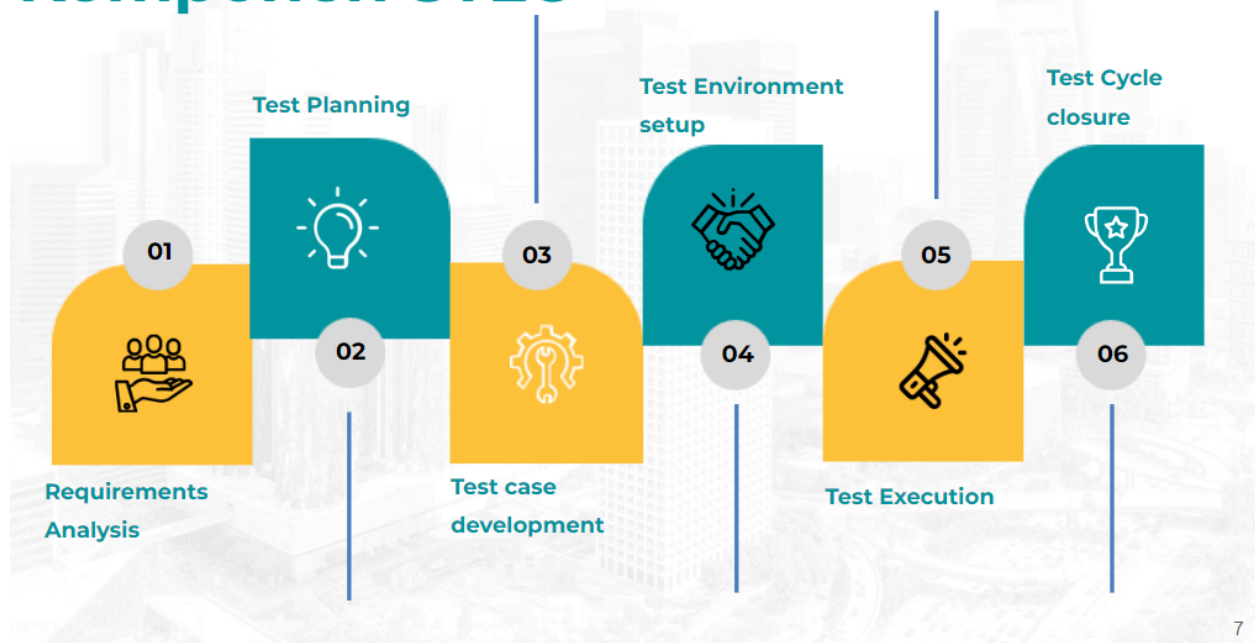
READING

Definisi STLC

STLC (Software Testing Life Cycle) merupakan urutan fase dalam proses pengujian sebuah software. STLC yang sudah ditentukan tahapannya ini dapat membantu proses pengujian lebih terstruktur dan mengurangi adanya bottleneck pada orang-orang yang terlibat. Setiap perusahaan bisa memiliki STLC mereka sendiri dan sudah disesuaikan dengan kebutuhan masing-masing. Secara umum berikut komponen tahapan STLC:

1. Requirement Analysis
2. Test Planning
3. Test case development
4. Test Environment setup
5. Test Execution
6. Test Cycle closure

Komponen STLC



Komponen Detail STLC

Life cycle pengujian perangkat lunak (STLC) yang menceritakan urutan atau proses berbagai aktivitas yang dilakukan oleh tim pengujian secara terencana untuk memastikan kualitas perangkat lunak atau produk. Ketika menerapkan STLC ada beberapa hal yang bisa diperhatikan yaitu:

1. Cost-effectiveness

Cost-effectiveness adalah ketika manfaat terbesar diperoleh dengan harga yang relatif rendah. Konsep ini umumnya digunakan ketika memilih dari berbagai pilihan investasi, sehingga pengembalian terbesar dihasilkan sebagai imbalan atas jumlah yang diinvestasikan.

2. Security

Security atau keamanan aplikasi ini mencakup semua siklus pengembangan perangkat lunak yang aman kepada tim pengembangan. Tujuan akhirnya adalah untuk meningkatkan praktik keamanan dan untuk menemukan, memperbaiki, dan mencegah masalah keamanan dalam aplikasi.

3. Product Quality

Kualitas produk mengacu pada seberapa baik suatu produk memenuhi kebutuhan pelanggan, memenuhi tujuannya dan memenuhi standar industri.

4. Customer satisfaction

Merupakan tolak ukur seberapa baik produk, layanan, dan pengalaman customer atau pelanggan secara keseluruhan memenuhi harapan customer. Ini mencerminkan kesehatan sebuah bisnis dengan menunjukkan seberapa baik produk atau layanan.

Requirements Analysis

Analisis kebutuhan sebuah aplikasi sangat diperlukan. Dalam tahap ini kita membutuhkan dokumen seperti SRS (software Requirement Specification) atau dokumen serupa yang berisi detail tentang aplikasi seperti hal-hal yang dapat dilakukan oleh user. Dari informasi itu penguji dapat menganalisa business logic, mengetahui fitur-fitur software yang harus diprioritaskan, menentukan tipe testing yang sesuai dan dapat memenuhi kebutuhan user.

Hal-hal yang dilakukan ketika melakukan requirement analysis:

- Siapkan daftar pertanyaan dan bekerja sama dengan Client, Technical Manager dan Technical Lead, dll.

- Menentukan fokus pengujian dan prioritas pengujian.
- Buat daftar rincian requirements yang dapat diuji.
- Siapkan tool automation yang akan digunakan
- Lalu siapkan laporan dari hasil requirements analysis.

Test Planning

Setelah requirement diketahui maka selanjutnya membuat rencana pengujian atau test planning. Test planning akan membantu proses pengujian lebih terstruktur, terukur dan terarah. Berikut hal-hal yang harus diperhatikan ketika menyusun test planning:

1. Menentukan berbagai tipe testing yang akan digunakan.
2. Menentukan teknik pendekatan testing yang sesuai. Ada dua teknik pendekatan testing:
 - a. Proactive: pendekatan testing dimana proses tes dimulai sedini mungkin untuk menemukan bug.
 - b. Reactive: pendekatan testing dimana proses tes akan dimulai ketika implementasi secara keseluruhan selesai.
3. Pemilihan test tool yang sesuai kebutuhan. Contohnya spreadsheet untuk management test cases.
4. Estimasi waktu dan cost yang diperlukan.
 - a. Menentukan sumberdaya yang dibutuhkan dan tugas-tugas ketika testing.
5. Membuat dokumen test planning.

Test Case Development

Test case development adalah proses pembuatan test case yang akan diterapkan pada proses pengujian software. Hal yang perlu diperhatikan dalam pembuatan test case antara lain:

1. Informasi tentang fitur yang diuji
2. Acceptance criteria
3. Steps
4. Data yang dibutuhkan atau dihasilkan dari test case

Perhatikan management test cases ketika membuat test case, hal ini akan sangat bermanfaat apabila dari proses inisiasi sudah memiliki format management yang baik. Contohnya buatlah database test cases di sebuah platform misalnya spreadsheet atau gunakan tool management

test cases yang ada di internet, mereka akan membantu sekali untuk proses pelacakan, dokumentasi dan versioning test case.

Test Environment Setup

Siapkan segala sesuatu yang dibutuhkan untuk menjalankan testing, misalnya hardware, koneksi internet, VPN, dan jangan lupa pastikan software sudah bisa dijalankan. Pada tahap ini penguji dapat melakukan smoke testing yang bertujuan untuk mengetahui apabila tidak ada blocker pada build dari aplikasi tersebut. Dokumentasikan hasil smoke test result untuk tujuan dokumentasi.

Test Execution Phase

Proses eksekusi rencana pengujian yang sudah dilakukan sebelumnya, dari hal ini penguji akan mendapatkan bug untuk dilaporkan kemudian. Bug yang sudah dilaporkan akan diperbaiki oleh tim pengembang dan dikembalikan lagi ke tim penguji untuk melakukan proses pengujian bug ulang, apabila bug tidak terjadi kembali dan dipastikan tidak ada bug baru dari proses perbaikan tersebut maka bug dinyatakan selesai. Hal ini dilakukan hingga sebuah software dinyatakan layak untuk pengguna.

Test Cycle Closure

Tahap terakhir dari STLC merupakan pembuatan laporan, pengumpulan matriks dan hasil pengujian. Pada proses ini dapat dilakukan analisa antara tim pengembang dengan tim penguji, hal ini bermanfaat untuk retrospective, mengetahui masalah-masalah yang terjadi selama proses pengujian kemudian bisa dilakukan improvement untuk proses pengujian kedepan. Proses ini seharusnya menghasilkan dokumen pengujian yang bisa dijadikan acuan klien ataupun pengujian di masa mendatang.

Exit - Entry Criteria

Entry criteria dalam STLC, menentukan kumpulan dokumen, informasi, dan data yang harus tersedia sebelum memulai fase STLC. Pada beberapa tahapan, kelengkapan dokumen, laporan, dan data dari tahapan sebelumnya akan menjadi entry criteria. Misalnya, dalam tahap

test development test cases, dokumen requirements yang telah dilengkapi, dan laporan test planning dari tahap sebelumnya akan menjadi dokumen entry kriteria.

Di sisi lain, exit criteria menentukan serangkaian tugas, tindakan, penutupan laporan, dan dokumentasi yang harus diselesaikan sebelum menyelesaikan sebuah fase. Ini seperti ekspektasi yang harus dicapai atau dipenuhi sebelum menyelesaikan fase Software Testing Life Cycle. Misalnya, dalam fase development test cases, harapan yang harus dipenuhi adalah penyelesaian penulisan test cases, pendefinisian data test, dan script automation, dan lain-lain.

Berikut entry dan exit criteria dimasing-masing fase STLC:

Fase	Entry Criteria	Exit Criteria
Requirement Analysis	<ul style="list-style-type: none"> Requirements specification document Acceptance criteria document Technical Specification 	<ul style="list-style-type: none"> Requirement traceability matrix (RTM) Automation Feasibility Document
Test Planning	<ul style="list-style-type: none"> Requirement Documents Requirement Traceability Matrix Document 	<ul style="list-style-type: none"> Test Plan document Testing Techniques Timeline Cost Risks Resource allocation
Test Case Development	<ul style="list-style-type: none"> Test Plan document Timeline Cost Risks Resource allocation 	<ul style="list-style-type: none"> Test Case Documentation/Test Script Identified Test Data
Test Environment Setup	<ul style="list-style-type: none"> Test strategy and test plan document Test case document Testing data 	<ul style="list-style-type: none"> Environment ready with test data set up Smoke Test Report
Test Execution Phase	<ul style="list-style-type: none"> Test strategy and test plan document. Test case document. Testing data. Testing Environment Testing Tool 	<ul style="list-style-type: none"> Test Case -Execution Result RTM with execution status Defect Report
Test Cycle Closure	<ul style="list-style-type: none"> Test Execution Report, Defect Report 	<ul style="list-style-type: none"> Test Closure Report Summary Report

Manfaat Menggunakan STLC

Beberapa manfaat yang dapat kita rasakan ketika menggunakan STLC pada project development:

1. Meningkatkan konsistensi dan efektivitas karena project requirement teranalisis.
2. Tujuan yang jelas dan terdefinisi untuk test dan proses pengujian dapat terdokumentasi dengan baik.
3. Keyakinan pada setiap fitur yang berhasil melewati uji coba. Hal ini juga membantu rasa yakin ketika akan menambah fitur baru.
4. Spesifikasi yang tertulis dengan jelas akan membantu tim bekerja lebih terarah.
5. Pendekatan yang sistematis dan terstruktur akan dengan cepat menyelesaikan masalah atau issue pada produk.

References

- <https://www.guru99.com/software-testing-life-cycle.html>