

<<TEST PLAN AIRBNB- F4>>

Test Plan

Document Change History

Version Number	Date	Contributor	Description

**** Note to Document Author** – Red and blue text (with the exception of the title and document name above) in this document is directed at the template user to describe processes, build standards and help build the document from the template. All such red and blue text should be removed before submitting any formal documentation, including both draft and/or final, deliverables. ****

Table of Contents

1	INTRODUCTION	3
1.1	SCOPE	
	Error! Bookmark not defined.	
1.1.1	<i>In Scope</i>	
	Error! Bookmark not defined.	
1.1.2	<i>Out of Scope</i>	
	Error! Bookmark not defined.	
1.2	QUALITY OBJECTIVE	
	Error! Bookmark not defined.	
1.2.1	<i>Primary Objective</i>	5
1.2.2	<i>Secondary Objective</i>	5
1.3	ROLES AND RESPONSIBILITIES	
	Error! Bookmark not defined.	
1.3.1	<i>Developer</i>	
	Error! Bookmark not defined.	
1.3.2	<i>Adopter</i>	
	Error! Bookmark not defined.	
1.3.3	<i>Testing Process Management Team</i>	
	Error! Bookmark not defined.	
1.4	ASSUMPTIONS FOR TEST EXECUTION	
	Error! Bookmark not defined.	
1.5	CONSTRAINTS FOR TEST EXECUTION	
	Error! Bookmark not defined.	
1.6	DEFINITIONS	7
2	TEST METHODOLOGY	8
2.1	PURPOSE	9
	<i>Overview</i>	9
2.1.1		
	<i>Usability Testing</i>	9
2.1.2		
	<i>Unit Testing (Multiple)</i>	9
2.1.3		
	<i>Iteration/Regression Testing</i>	10
2.1.4		
	<i>Final release Testing</i>	10
2.1.5		
	<i>Testing completeness Criteria</i>	10
2.1.6		
2.2	TEST LEVELS	10
	<i>Build Tests</i>	10
2.2.1		
	Level 1 - Build Acceptance Tests	11
2.2.1.1		
	Level 2 - Smoke Tests	11
2.2.1.2		

2.2.1.3	Level 2a - Bug Regression Testing	11
2.2.2	<i>Milestone Tests</i>	11
2.2.2.1	Level 3 - Critical Path Tests	11
2.2.3	<i>Release Tests</i>	11
2.2.3.1	Level 4 - Standard Tests	12
2.2.3.2	Level 5 - Suggested Test	12
2.3	BUG REGRESSION	12
2.4	BUG TRIAGE	
2.5	Error! Bookmark not defined.	
2.6	SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS	13
2.6.1	TEST COMPLETENESS	13
2.6.2	<i>Standard Conditions:</i>	13
	<i>Bug Reporting & Triage Conditions:</i>	13
3	TEST DELIVERABLES	
	Error! Bookmark not defined.	
3.1	DELIVERABLES MATRIX	14
3.2	DOCUMENTS	15
3.2.1	<i>Test Approach Document</i>	15
3.2.2	<i>Test Plan</i>	15
3.2.3	<i>Test Schedule</i>	15
3.2.4	<i>Test Specifications</i>	15
3.2.5	<i>Requirements Traceability Matrix</i>	16
3.3	DEFECT TRACKING & DEBUGGING	
3.3.1	Error! Bookmark not defined.	
3.3.2	<i>Testing Workflow</i>	
3.4	Error! Bookmark not defined.	
3.4.1	<i>Defect reporting using G FORGE</i>	
3.4.2	Error! Bookmark not defined.	
3.4.3	REPORTS	17
	<i>Testing status reports</i>	17
	<i>Phase Completion Reports</i>	18
	<i>Test Final Report - Sign-Off</i>	18

3.5	RESPONSIBILITY MATRIX	18
4	RESOURCE & ENVIRONMENT NEEDS	18
4.1	TESTING TOOLS	18
4.1.1	Tracking Tools	19
4.1.1.1	Configuration Management	19
4.2	TEST ENVIRONMENT	19
4.2.1	Hardware	19
4.2.2	Software	19
4.3	BUG SEVERITY AND PRIORITY DEFINITION	19
4.3.1	Severity List	20
4.3.2	Priority List	20
4.4	BUG REPORTING	21
5	TERMS/ACRONYMS	
	Error! Bookmark not defined.	

1 Introduction

Rencana pengujian ini menjelaskan strategi, proses, alur kerja, dan metodologi yang digunakan untuk merencanakan, mengorganisir, melaksanakan, dan mengelola pengujian untuk aplikasi Web Airbnb.

1.1 Scope

Jelaskan cakupan pendekatan pengujian saat ini berdasarkan peran dan tujuan proyek Anda.

1.1.1 In Scope

Rencana Pengujian Aplikasi Web Airbnb mendefinisikan pendekatan pengujian unit, integrasi, sistem, regresi, dan Penerimaan Klien (Client Acceptance testing). Lingkup pengujian meliputi hal-hal berikut:

- Pengujian semua persyaratan fungsional, kinerja aplikasi, keamanan, dan kasus penggunaan yang tercantum dalam dokumen Kasus Penggunaan untuk Aplikasi Web Airbnb.
- Persyaratan kualitas dan metrik kesesuaian untuk Aplikasi Web Airbnb.
- Pengujian end-to-end dan pengujian antarmuka dari semua sistem yang berinteraksi dengan Aplikasi Web Airbnb (misalnya, gateway pembayaran, layanan pemetaan).
- Pendaftaran dan Login Pengguna.
- Pencarian dan Filter Properti (Listing).
- Proses Pemesanan dan Reservasi.
- Manajemen Properti Host (membuat, mengedit, menghapus listing).
- Manajemen Profil Pengguna.
- Sistem Pesan antara Tamu dan Host.
- Sistem Peninjauan dan Peringkat.
- Proses Pembayaran dan Pengembalian Dana.
- Desain Responsif di berbagai perangkat (desktop, tablet, web seluler).
- Pengujian keamanan (misalnya, otentikasi, otorisasi, privasi data).
- Pengujian kinerja (misalnya, waktu muat halaman, waktu respons di bawah beban).

1.1.2 Out of Scope

Berikut ini dianggap di luar lingkup Rencana Pengujian sistem Airbnb

- Pengujian persyaratan fungsional untuk sistem di luar aplikasi Airbnb (misalnya, integrasi pihak ketiga yang tidak langsung dikelola oleh Airbnb).
- Pengujian SOP Bisnis, pemulihan bencana, dan Rencana Kelangsungan Bisnis.
- Aplikasi seluler native (iOS/Android).
- Pengujian API backend (kecuali jika terkait langsung dengan masalah fungsionalitas front-end).
- Pengujian kompatibilitas perangkat keras untuk perangkat pengguna tertentu di luar kompatibilitas browser umum.

1.2 Quality Objective

1.2.1 Primary Objective

Tujuan utama pengujian sistem aplikasi adalah untuk: memastikan bahwa sistem memenuhi semua persyaratan, termasuk persyaratan kualitas (AKA: persyaratan non-fungsional) dan metrik kesesuaian untuk setiap persyaratan kualitas dan memenuhi skenario kasus penggunaan serta menjaga kualitas produk. Pada akhir siklus pengembangan proyek, pengguna harus menemukan bahwa proyek telah memenuhi atau melampaui semua harapan mereka sebagaimana dirinci dalam persyaratan. Setiap perubahan, penambahan, atau penghapusan pada dokumen persyaratan, Spesifikasi Fungsional, atau Spesifikasi Desain akan didokumentasikan dan diuji pada tingkat kualitas tertinggi yang diizinkan dalam sisa waktu proyek dan dalam kemampuan tim pengujian.

Untuk Aplikasi Web Airbnb, ini berarti memastikan:

- Semua persyaratan fungsional yang ditentukan (misalnya, pencarian, pemesanan, manajemen listing) terpenuhi secara akurat dan efisien.
- Aplikasi berkinerja andal di bawah beban pengguna yang diharapkan.
- Antarmuka pengguna intuitif dan ramah pengguna.
- Kerentanan keamanan diidentifikasi dan ditangani.

1.2.2 Secondary Objective

Tujuan sekunder pengujian sistem aplikasi adalah untuk: mengidentifikasi dan mengungkapkan semua masalah dan risiko terkait, mengkomunikasikan semua masalah yang diketahui kepada tim proyek, dan memastikan bahwa semua masalah ditangani dengan tepat sebelum rilis. Sebagai tujuan, ini membutuhkan pengujian aplikasi yang cermat dan metodis untuk memastikan semua area sistem diteliti dan, sebagai hasilnya, semua masalah (bug) yang ditemukan ditangani dengan tepat.

1.3 Roles and Responsibilities

Peran dan tanggung jawab dapat berbeda berdasarkan SOW yang sebenarnya. Fungsi yang tercantum di bawah ini adalah untuk fase pengujian.

1.3.1 Developer

Seorang Pusat Kanker yang ditunjuk NCI yang dipilih dan didanai oleh NCICB untuk berpartisipasi dalam Ruang Kerja tertentu untuk melakukan kegiatan pengembangan perangkat lunak atau solusi. Catatan: Ini adalah placeholder dari template. Kita sesuaikan untuk Airbnb.

- Bertanggung jawab untuk:
- Mengembangkan aplikasi web Airbnb.
- Mengembangkan kasus penggunaan dan persyaratan bekerja sama dengan Pengguna (Pemilik Produk/Analisis Bisnis).
- Melakukan pengujian unit, sistem, regresi, dan integrasi untuk modul yang mereka kembangkan.
- Mendukung pengujian penerimaan pengguna.

- Melakukan debugging dan memperbaiki cacat yang dilaporkan.

1.3.2 Adopter

Sebuah Pusat Kanker yang ditunjuk NCI yang dipilih dan didanai oleh NCICB untuk melakukan adopsi formal, pengujian, validasi, dan penerapan produk atau solusi yang dikembangkan oleh Pengembang Ruang Kerja.

Bertanggung jawab untuk:

- Berkontribusi pada pengembangan kasus penggunaan, persyaratan melalui tinjauan.
- Berkontribusi untuk mengembangkan dan mengeksekusi skrip pengujian pengembangan melalui tinjauan.
- Melakukan Pengujian Penerimaan Pengguna Penuh, regresi, dan pengujian end-to-end; ini termasuk mengidentifikasi skenario pengujian, membangun skrip pengujian, mengeksekusi skrip, dan melaporkan hasil pengujian.
- Memberikan umpan balik tentang kegunaan dan fungsionalitas dari perspektif pengguna akhir.

1.3.3 Testing Process Management Team

Termasuk NCI, BAH dan Pemimpin Pusat Kanker yang dialokasikan

Grup ini bertanggung jawab untuk mengelola seluruh proses pengujian, alur kerja, dan manajemen kualitas dengan kegiatan dan tanggung jawab untuk:

- Memantau dan mengelola integritas pengujian dan mendukung kegiatan pengujian.
- Mengkoordinasikan kegiatan di berbagai pusat kanker.
- Memastikan kepatuhan terhadap rencana pengujian dan tujuan kualitas.
- Memfasilitasi rapat triage bug.
- Memberikan panduan yang tepat kepada Pengguna dan Pengembang untuk melakukan pengujian.

1.4 Assumptions for Test Execution

- Untuk pengujian Penerimaan Pengguna, tim Pengembang telah menyelesaikan pengujian unit, sistem, dan integrasi serta memenuhi semua Persyaratan (termasuk persyaratan kualitas) berdasarkan Matriks Ketertelusuran Persyaratan.
- Pengujian Penerimaan Pengguna akan dilakukan oleh Pengguna Akhir atau penguji QA khusus yang mensimulasikan perilaku pengguna akhir.
- Hasil pengujian akan dilaporkan setiap hari menggunakan [Selenium, mis. Jira, Asana, Trello, atau GForge sesuai template].
- Skrip yang gagal dan daftar cacat dari [Selenium] dengan bukti akan dikirim langsung ke Pengembang atau melalui alat yang ditunjuk.
- Kasus penggunaan telah dikembangkan oleh Pengguna untuk pengujian Penerimaan Pengguna. Kasus penggunaan disetujui oleh pemimpin pengujian.
- Skrip pengujian dikembangkan dan disetujui.
- Ketergantungan utama harus segera dilaporkan setelah rapat kickoff pengujian.

- Data pengujian yang diperlukan akan tersedia di lingkungan pengujian.
- Lingkungan pengujian akan stabil dan dapat diakses selama siklus pengujian.

1.5 Constraints for Test Execution

Di bawah ini adalah beberapa asumsi minimum (hitam) diikuti dengan contoh kendala (merah). Contoh apa pun dapat digunakan jika dianggap sesuai untuk proyek tertentu. Kendala baru juga dapat ditambahkan yang dianggap sesuai untuk proyek.

- Pengguna harus memahami dengan jelas prosedur pengujian dan pencatatan cacat atau peningkatan.
- Tim Manajemen Proses Pengujian akan menjadwalkan telekonferensi dengan Pengembang dan Pengguna untuk melatih dan mengatasi masalah terkait pengujian.
- Pengembang akan menerima daftar konsolidasi permintaan untuk pengaturan lingkungan pengujian, pengaturan akun pengguna, set data (data aktual dan data tiruan), daftar cacat, dll. melalui Selenium setelah rapat kickoff pengujian Pengguna awal.
- Pengembang akan mendukung kegiatan pengujian yang sedang berlangsung berdasarkan prioritas.
- Skrip pengujian harus disetujui oleh Pemimpin Pengujian sebelum eksekusi pengujian.
- Skrip pengujian, lingkungan pengujian, dan ketergantungan harus ditangani selama rapat kickoff pengujian di hadapan SME dan daftar permintaan harus diajukan dalam waktu 3 hari setelah rapat kickoff.
- Pengembang tidak dapat mengeksekusi skrip pengujian Penerimaan Pengguna dan End-to-End. Setelah debugging, pengembang dapat melakukan pengujian internal mereka, tetapi tidak ada hasil dari pengujian tersebut yang dapat dicatat/dilaporkan.
- Pengguna bertanggung jawab untuk mengidentifikasi ketergantungan antara skrip pengujian dan mengajukan permintaan yang jelas untuk menyiapkan lingkungan pengujian.
- Ketersediaan terbatas dari lingkungan pengujian spesifik (misalnya, versi browser atau jenis perangkat tertentu).
- Kendala waktu untuk setiap fase pengujian, sebagaimana didefinisikan dalam jadwal pengujian.
- Ketergantungan pada layanan atau API pihak ketiga yang mungkin memiliki downtime atau batasan sendiri.

1.6 Definitions

- Bug: Setiap kesalahan atau cacat yang menyebabkan perangkat lunak/aplikasi atau perangkat keras tidak berfungsi. Hal tersebut juga termasuk dalam persyaratan dan tidak memenuhi alur kerja, proses, atau titik fungsi yang diperlukan.
- Peningkatan (Enhancement): 1) Setiap perubahan atau modifikasi pada sistem yang ada untuk alur kerja dan proses yang lebih baik. 2) Kesalahan atau cacat yang menyebabkan perangkat lunak/aplikasi atau perangkat keras tidak berfungsi. Di mana 1) dan 2) TIDAK

termasuk dalam persyaratan dapat dikategorikan sebagai peningkatan. Peningkatan dapat ditambahkan sebagai persyaratan baru setelah proses Manajemen Perubahan yang tepat.

- UAT (User Acceptance Testing): Pengujian formal oleh pengguna akhir atau perwakilan klien untuk memastikan sistem memenuhi persyaratan bisnis dan sesuai untuk tujuan.
- Pengujian Fungsional: Pengujian untuk memastikan perangkat lunak berfungsi sesuai dengan spesifikasinya.
- Pengujian Non-fungsional: Pengujian atribut sistem yang tidak terkait dengan fungsi tertentu, seperti keandalan, kegunaan, efisiensi, dan pemeliharaan.
- Pengujian Regresi: Pengujian ulang untuk memastikan bahwa perubahan atau penambahan pada perangkat lunak tidak memengaruhi fungsionalitas yang ada secara negatif.

2 Test Methodology

2.1 Purpose

2.1.1 Overview

Daftar di bawah ini tidak dimaksudkan untuk membatasi luasnya rencana pengujian dan dapat dimodifikasi agar sesuai untuk proyek tertentu. Tujuan dari Rencana Pengujian adalah untuk mencapai hal-hal berikut:

- Mendefinisikan strategi pengujian untuk setiap area dan sub-area untuk menyertakan semua persyaratan fungsional dan kualitas (non-fungsional).
- Membagi Spesifikasi Desain menjadi area dan sub-area yang dapat diuji (jangan bingung dengan spesifikasi pengujian yang lebih rinci). Pastikan juga untuk mengidentifikasi dan menyertakan area yang akan dihilangkan (tidak diuji).
- Mendefinisikan prosedur pelacakan bug.
- Mengidentifikasi risiko pengujian.
- Mengidentifikasi sumber daya yang diperlukan dan informasi terkait.
- Memberikan Jadwal Pengujian.

2.1.2 Usability Testing

Tujuan dari pengujian kegunaan adalah untuk memastikan bahwa komponen dan fitur baru akan berfungsi dengan cara yang dapat diterima oleh pelanggan. Pengembangan biasanya akan membuat prototipe komponen UI yang tidak berfungsi untuk mengevaluasi desain yang diusulkan. Pengujian kegunaan dapat dikoordinasikan dengan pengujian, tetapi pengujian aktual harus dilakukan oleh non-penguji (sedekat mungkin dengan pengguna akhir). Pengujian akan meninjau temuan dan memberikan tim proyek evaluasinya tentang dampak perubahan ini terhadap proses pengujian dan proyek secara keseluruhan.

Untuk Airbnb, ini akan melibatkan pengujian intuitivitas navigasi, kejelasan informasi, kemudahan pemesanan, dan pengalaman pengguna secara keseluruhan.

2.1.3 Unit Testing (Multiple)

Pengujian Unit dilakukan oleh Pengembang selama proses pengembangan kode untuk memastikan bahwa fungsionalitas dan cakupan kode yang tepat telah dicapai oleh setiap pengembang baik selama pengkodean maupun dalam persiapan untuk penerimaan ke pengujian iterasi. Berikut ini adalah contoh area proyek yang harus diuji unit dan ditandatangani sebelum diteruskan ke Pengujian Regresi:

- Database, Stored Procedures, Triggers, Tables, dan Indexes.
- Layanan NT.
- Konversi database.
- .OCX, .DLL, .EXE dan executable format biner lainnya.

Untuk Airbnb, ini termasuk menguji fungsi, komponen, dan modul individual seperti:

- Modul otentikasi pengguna.
- Algoritma pencarian listing.
- Integrasi pemrosesan pembayaran.

- Validasi data pada formulir.

2.1.4 Iteration/Regression Testing

Selama siklus berulang mengidentifikasi bug dan menerima build baru (mengandung perubahan kode perbaikan bug), ada beberapa proses yang umum untuk fase ini di semua proyek. Ini termasuk berbagai jenis pengujian: fungsionalitas, kinerja, stres, konfigurasi, dll. Ada juga proses mengkomunikasikan hasil dari pengujian dan memastikan bahwa rilis/iterasi baru mengandung perbaikan yang stabil (regresi). Proyek harus merencanakan minimal 2-3 siklus pengujian (rilis/iterasi build baru). Pada setiap iterasi, harus diadakan debriefing. Secara khusus, laporan harus menunjukkan bahwa sejauh yang dapat dicapai selama fase pengujian iterasi, semua bug severity 1 dan severity 2 yang teridentifikasi telah dikomunikasikan dan ditangani. Minimal, semua bug priority 1 dan priority 2 harus diselesaikan sebelum memasuki fase beta.

Untuk Airbnb, ini akan melibatkan siklus pengujian berulang setelah fitur baru ditambahkan atau bug diperbaiki, memastikan fungsionalitas yang ada tetap utuh.

2.1.5 Final release Testing

Tim penguji dengan pengguna akhir berpartisipasi dalam proses tonggak ini juga dengan memberikan umpan balik konfirmasi pada masalah baru yang ditemukan, dan masukan berdasarkan masalah yang identik atau serupa yang terdeteksi sebelumnya. Tujuannya adalah untuk memverifikasi bahwa produk siap untuk didistribusikan, dapat diterima oleh pelanggan dan menghilangkan masalah operasional potensial. Dengan asumsi bug kritis diselesaikan selama pengujian iterasi sebelumnya - Sepanjang siklus pengujian Rilis Final, perbaikan bug akan difokuskan pada bug minor dan trivial (severity 3 dan 4). Pengujian akan melanjutkan proses verifikasi stabilitas aplikasi melalui pengujian regresi (bug yang diketahui ada, serta kasus pengujian yang ada). Target tonggak dari fase ini adalah untuk menetapkan bahwa aplikasi yang diuji telah mencapai tingkat stabilitas, yang sesuai untuk penggunaannya (jumlah pengguna, dll.), sehingga dapat dirilis ke pengguna akhir dan komunitas caBIG.

Untuk Airbnb, fase ini akan memastikan aplikasi stabil dan siap untuk digunakan publik.

2.1.6 Testing completeness Criteria

Rilis untuk produksi hanya dapat terjadi setelah keberhasilan penyelesaian aplikasi yang diuji melalui semua fase dan tonggak yang dibahas sebelumnya. Target tonggak adalah menempatkan rilis/aplikasi (build) ke produksi setelah ditunjukkan bahwa aplikasi telah mencapai tingkat stabilitas yang memenuhi atau melampaui harapan klien sebagaimana didefinisikan dalam Persyaratan, Spesifikasi Fungsional, dan Standar Produksi caBIG.

2.2 Test Levels

Pengujian suatu aplikasi dapat dipecah menjadi tiga kategori utama dan beberapa sub-tingkat. Tiga kategori utama meliputi pengujian yang dilakukan setiap build (Pengujian Build), pengujian yang dilakukan setiap tonggak utama (Pengujian Tonggak), dan pengujian yang dilakukan

setidaknya sekali setiap siklus rilis proyek (Pengujian Rilis). Kategori pengujian dan tingkat pengujian didefinisikan di bawah ini:

2.2.1 Build Tests

2.2.1.1 Level 1 - Build Acceptance Tests

Pengujian Penerimaan Build harus membutuhkan waktu kurang dari 2-3 jam untuk diselesaikan (15 menit adalah tipikal). Kasus pengujian ini hanya memastikan bahwa aplikasi dapat dibangun dan diinstal dengan sukses. Kasus pengujian terkait lainnya memastikan bahwa pengguna menerima Dokumen Rilis Pengembangan yang tepat ditambah informasi terkait build lainnya (titik rilis, dll.). Tujuannya adalah untuk menentukan apakah pengujian lebih lanjut dimungkinkan. Jika ada kasus pengujian Tingkat 1 yang gagal, build dikembalikan ke pengembang tanpa diuji.

Untuk Airbnb, ini akan melibatkan verifikasi keberhasilan deployment ke lingkungan pengujian dan peluncuran aplikasi dasar.

2.2.1.2 Level 2 - Smoke Tests

Pengujian Asap harus diotomatisasi dan membutuhkan waktu kurang dari 2-3 jam (20 menit adalah tipikal). Kasus pengujian ini memverifikasi fungsionalitas utama pada tingkat tinggi. Tujuannya adalah untuk menentukan apakah pengujian lebih lanjut dimungkinkan. Kasus pengujian ini harus menekankan cakupan daripada kedalaman. Semua komponen harus disentuh, dan setiap fitur utama harus diuji secara singkat oleh Pengujian Asap. Jika ada kasus pengujian Tingkat 2 yang gagal, build dikembalikan ke pengembang tanpa diuji.

Untuk Airbnb, ini meliputi pemeriksaan cepat fungsionalitas kritis seperti login pengguna, pencarian properti, dan melihat listing.

2.2.1.3 Level 2a - Bug Regression Testing

Setiap bug yang Open selama build sebelumnya, tetapi ditandai sebagai Fixed, Needs Re-Testing untuk build yang sedang diuji saat ini, perlu diregresi, atau diuji ulang. Setelah pengujian asap selesai, semua bug yang diselesaikan perlu diregresi. Dibutuhkan antara 5 menit hingga 1 jam untuk meregresi sebagian besar bug.

2.2.2 Milestone Tests

2.2.2.1 Level 3 - Critical Path Tests

Kasus pengujian Jalur Kritis ditargetkan pada fitur dan fungsionalitas yang akan dilihat dan digunakan pengguna setiap hari. Kasus pengujian Jalur Kritis harus lulus pada akhir setiap 2-3 Siklus Pengujian Build. Mereka tidak perlu diuji setiap rilis, tetapi harus diuji setidaknya sekali per tonggak. Dengan demikian, semua kasus pengujian Jalur Kritis harus dieksekusi setidaknya sekali selama siklus Iterasi, dan sekali selama siklus Rilis Final.

Untuk Airbnb, jalur kritis meliputi: pendaftaran pengguna yang berhasil, mencari dan memesan properti, dan host berhasil mendaftarkan properti.

2.2.3 Release Tests

2.2.3.1 Level 4 - Standard Tests

Kasus Pengujian yang perlu dijalankan setidaknya sekali selama seluruh siklus pengujian untuk rilis ini. Kasus-kasus ini dijalankan sekali, tidak diulang seperti kasus pengujian di tingkat sebelumnya. Pengujian Fungsional dan Pengujian Desain Rinci (Kasus Pengujian Spesifikasi Fungsional dan Spesifikasi Desain, masing-masing). Ini dapat diuji beberapa kali untuk setiap Siklus Pengujian Tonggak (Iterasi, Rilis Final, dll.). Kasus pengujian standar biasanya meliputi Instalasi, Data, GUI, dan area pengujian lainnya.

Untuk Airbnb, ini mencakup pengujian fungsional komprehensif dari semua fitur, integritas data, dan responsivitas GUI.

2.2.3.2 Level 5 - Suggested Test

Ini adalah Kasus Pengujian yang akan menyenangkan untuk dieksekusi, tetapi dapat dihilangkan karena kendala waktu. Sebagian besar Kasus Pengujian Kinerja dan Stres adalah contoh klasik dari kasus pengujian yang Disarankan (meskipun beberapa harus dianggap sebagai kasus pengujian standar). Contoh lain dari kasus pengujian yang disarankan meliputi WAN, LAN, Jaringan, dan pengujian Beban.

Untuk Airbnb, ini bisa mencakup pengujian kinerja ekstensif di bawah beban yang sangat tinggi, atau pengujian latensi jaringan untuk pengguna di berbagai lokasi geografis.

2.3 Bug Regression

Regresi Bug akan menjadi bagian utama di seluruh fase pengujian. Semua bug yang diselesaikan sebagai Fixed, Needs Re-Testing akan diregresi ketika tim Pengujian diberitahu tentang rilis baru yang berisi perbaikan. Ketika sebuah bug lulus regresi, itu akan dianggap Closed, Fixed. Jika sebuah bug gagal regresi, tim pengujian pengguna akan memberitahu tim pengembangan dengan memasukkan catatan ke [Selenium]. Ketika bug Severity 1 gagal regresi, tim Pengujian pengguna juga harus segera mengirim email ke pengembangan. Pemimpin Pengujian akan bertanggung jawab untuk melacak dan melaporkan kepada pengembangan dan manajemen produk status pengujian regresi.

2.4 Bug Triage

Triage Bug akan diadakan di seluruh fase siklus pengembangan. Triage Bug akan menjadi tanggung jawab Pemimpin Pengujian. Triage akan diadakan secara teratur dengan kerangka waktu yang ditentukan oleh tingkat penemuan bug dan jadwal proyek. Dengan demikian, akan menjadi tipikal untuk mengadakan beberapa triage selama fase Perencanaan, kemudian mungkin satu triage per minggu selama fase Desain, meningkat menjadi dua kali per minggu selama tahap akhir fase Pengembangan. Kemudian, fase Stabilisasi harus melihat pengurangan substansial dalam jumlah bug baru yang ditemukan, sehingga beberapa triage per minggu akan menjadi maksimum (untuk menangani status bug yang ada). Pemimpin Pengujian, Manajer Produk, dan Pemimpin Pengembangan harus terlibat dalam rapat triage ini. Pemimpin Pengujian akan menyediakan dokumentasi dan laporan yang diperlukan tentang bug untuk semua peserta. Tujuan dari triage adalah untuk menentukan jenis resolusi untuk setiap bug dan untuk memprioritaskan serta menentukan jadwal untuk semua Bug yang Akan Diperbaiki. Pengembangan kemudian akan menugaskan bug kepada orang yang tepat untuk perbaikan dan

melaporkan resolusi setiap bug kembali ke sistem pelacak bug [Selenium]. Pemimpin Pengujian akan bertanggung jawab untuk melacak dan melaporkan status semua resolusi bug.

2.5 Suspension Criteria and Resumption Requirements

Bagian ini harus didefinisikan untuk mencantumkan kriteria dan persyaratan lanjutan jika tingkat tertentu dan tingkat tujuan pengujian yang telah ditentukan tidak terpenuhi.

- Pengujian akan ditangguhkan pada modul perangkat lunak yang terpengaruh ketika bug kasus pengujian Asap (Tingkat 1) atau Jalur Kritis (Tingkat 2) ditemukan setelah iterasi ke-3.
- Pengujian akan ditangguhkan jika ada perubahan lingkup kritis yang memengaruhi Jalur Kritis.

Laporan bug harus diajukan oleh tim Pengembangan. Setelah memperbaiki bug, tim Pengembangan akan mengikuti kriteria rilis (dijelaskan di atas) untuk menyediakan rilis terbaru untuk Pengujian tambahan. Pada saat itu, pengguna akan meregresi bug, dan jika lulus, melanjutkan pengujian modul.

2.6 Test Completeness

Pengujian akan dianggap selesai ketika kondisi berikut telah terpenuhi:

2.6.1 Standard Conditions:

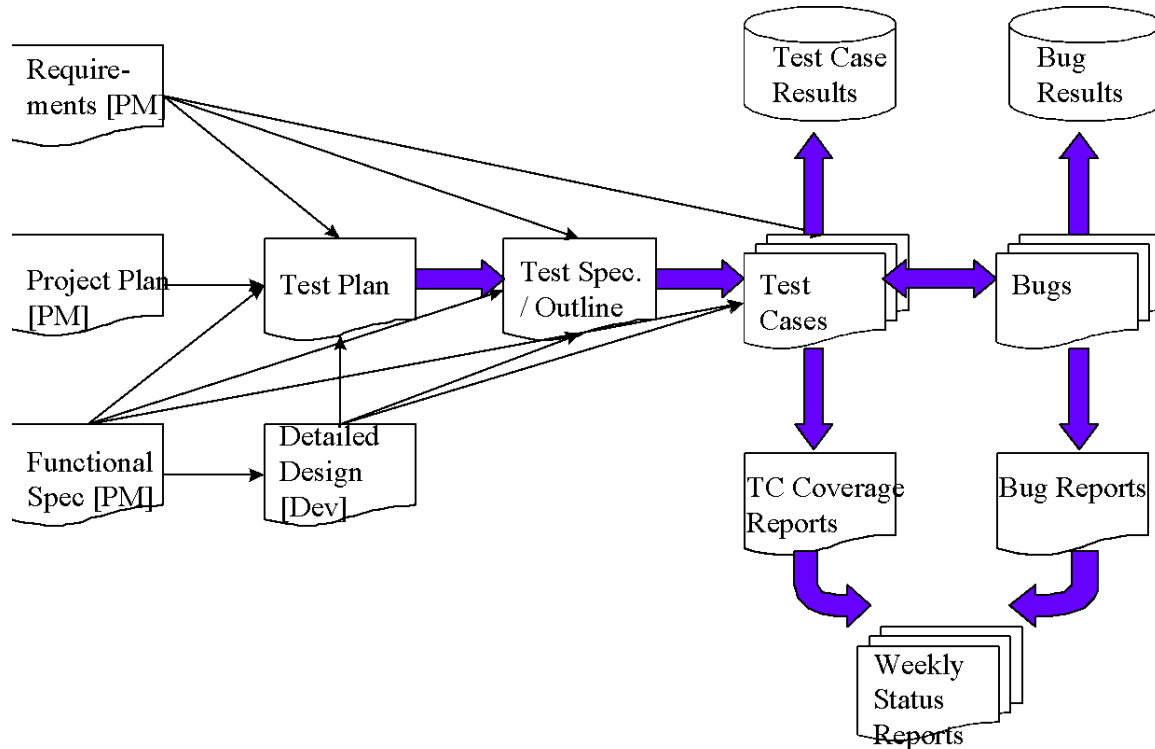
- Ketika Pengguna dan Pengembang, setuju bahwa pengujian telah selesai, aplikasi stabil, dan setuju bahwa aplikasi memenuhi persyaratan fungsional.
- Eksekusi skrip dari semua kasus pengujian di semua area telah lulus.
- Kasus pengujian otomatis di semua area telah lulus.
- Semua bug prioritas 1 dan 2 telah diselesaikan dan ditutup.
- [Nama Pemangku Kepentingan/Klien] menyetujui penyelesaian pengujian.
- Setiap area pengujian telah ditandatangani sebagai selesai oleh Pemimpin Pengujian.
- 50% dari semua bug severity 1 dan 2 yang diselesaikan telah berhasil diregresi ulang sebagai validasi akhir.
- Pengujian ad hoc di semua area telah selesai.

2.6.2 Bug Reporting & Triage Conditions:

- Tingkat penemuan bug menunjukkan tren menurun sebelum Tingkat Bug Nol (tidak ada bug Sev. 1/2/3 baru yang ditemukan).
- Tingkat penemuan bug tetap 0 bug baru yang ditemukan (Severity 1/2/3) meskipun ada upaya pengujian konstan selama 3 hari atau lebih.
- Distribusi tingkat keparahan bug telah berubah menjadi penurunan yang stabil dalam bug Sev. 1 dan 2 yang ditemukan.
- Tidak ada bug Must Fix yang tersisa meskipun pengujian berkelanjutan.

3 Test Deliverables

Pengujian akan memberikan hasil spesifik selama proyek. Hasil ini terbagi dalam tiga kategori dasar: Dokumen, Kasus Pengujian / Penulisan Bug, dan Laporan



3.1 Deliverables Matrix

Di bawah ini adalah daftar artefak yang digerakkan oleh proses dan harus diproduksi selama siklus hidup pengujian. Hasil tertentu harus disampaikan sebagai bagian dari validasi pengujian, Anda dapat menambahkan ke daftar hasil di bawah ini yang mendukung tujuan keseluruhan dan untuk menjaga kualitas. Matriks ini harus diperbarui secara rutin selama siklus pengembangan proyek dalam Rencana Pengujian spesifik proyek

Deliverable
Documents
-> Test Approach
□ Test Plan
□ Test Schedule
□ RTM
Test Case / Bug Write-Ups
Test Cases / Results
Test Coverage Reports
GForge Bug tracker for bug reporting

Reports
Test results report
Test Final Report - Sign-Off

3.2 Documents

3.2.1 Test Approach Document

Dokumen Pendekatan Pengujian berasal dari Dokumen Rencana Proyek, Persyaratan, dan Spesifikasi Fungsional. Dokumen ini mendefinisikan pendekatan pengujian keseluruhan yang akan diambil untuk proyek tersebut. Dokumen Pendekatan Pengujian Standar yang sedang Anda baca ini adalah boilerplate dari mana dokumen Pendekatan Pengujian proyek yang lebih spesifik dapat diambil.

Setelah dokumen ini selesai, Pemimpin Pengujian akan mendistribusikannya kepada Manajer Produk, Pemimpin Pengembangan, Perwakilan Pengguna, Manajer Program, dan pihak lain yang diperlukan untuk peninjauan dan penandatanganan.

3.2.2 Test Plan

Rencana Pengujian berasal dari Pendekatan Pengujian, Persyaratan, Spesifikasi Fungsional, dan Spesifikasi Desain terperinci. Rencana Pengujian mengidentifikasi detail pendekatan pengujian, mengidentifikasi area kasus pengujian terkait dalam produk spesifik untuk siklus rilis ini. Tujuan dari dokumen Rencana Pengujian adalah untuk:

- Menentukan pendekatan yang akan digunakan Pengujian untuk menguji produk, dan hasil (diekstrak dari Pendekatan Pengujian).
- Membagi produk menjadi area-area yang berbeda dan mengidentifikasi fitur-fitur produk yang akan diuji.
- Menentukan prosedur yang akan digunakan untuk penandatanganan pengujian dan rilis produk.
- Menunjukkan alat yang digunakan untuk menguji produk.
- Mencantumkan rencana sumber daya dan penjadwalan.
- Menunjukkan orang yang dapat dihubungi yang bertanggung jawab atas berbagai area proyek.
- Mengidentifikasi risiko dan rencana kontingensi yang mungkin memengaruhi pengujian produk.
- Menentukan prosedur manajemen bug untuk proyek.
- Menentukan kriteria untuk penerimaan rilis pengembangan ke pengujian (build).

3.2.3 Test Schedule

Bagian ini tidak vital untuk dokumen secara keseluruhan dan dapat dimodifikasi atau dihapus jika diperlukan oleh penulis. Jadwal Pengujian adalah tanggung jawab Pemimpin Pengujian (atau Penjadwal Departemen, jika ada) dan akan didasarkan pada informasi dari Penjadwal Proyek (dilakukan oleh Manajer Produk). Jadwal Pengujian spesifik proyek dapat dilakukan di MS Project atau alat manajemen proyek lainnya seperti Jira/Asana.

3.2.4 Test Specifications

Dokumen Spesifikasi Pengujian berasal dari Rencana Pengujian serta dokumen Persyaratan, Spesifikasi Fungsional, dan Spesifikasi Desain. Dokumen ini menyediakan spesifikasi untuk konstruksi Kasus Pengujian dan mencakup daftar area kasus pengujian dan tujuan pengujian untuk setiap komponen yang akan diuji sebagaimana diidentifikasi dalam Rencana Pengujian proyek.

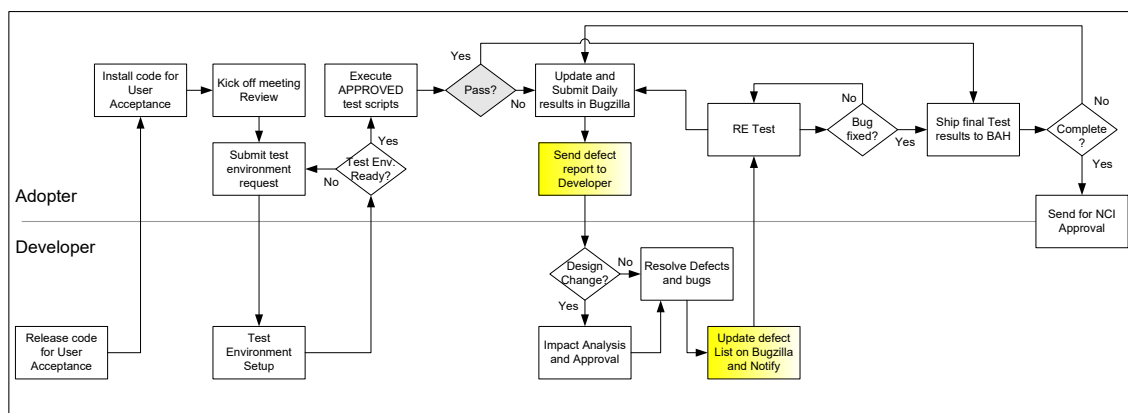
3.2.5 Requirements Traceability Matrix

Matriks Ketertelusuran Persyaratan (RTM) yang digunakan untuk menghubungkan skenario pengujian dengan persyaratan dan kasus penggunaan adalah bagian yang wajib dari dokumentasi Rencana Pengujian untuk semua proyek. Ketertelusuran persyaratan didefinisikan sebagai kemampuan untuk menjelaskan dan mengikuti kehidupan suatu persyaratan, baik dalam arah maju maupun mundur (yaitu dari asalnya, melalui pengembangan dan spesifikasinya, hingga penyebaran dan penggunaannya, dan melalui periode penyempurnaan dan iterasi berkelanjutan dalam fase mana pun dari ini). Penting untuk memilih template atau dasar dokumen yang mencapai ketertelusuran menyeluruh sepanjang masa proyek.

3.3 Defect Tracking & Debugging

3.3.1 Testing Workflow

Alur kerja di bawah ini menggambarkan proses alur kerja pengujian untuk Pengembang dan Pengguna untuk pengujian Penerimaan Pengguna dan End-to-End. Harap dicatat proses yang disorot kuning di mana Pengguna diminta untuk langsung mengirim daftar cacat dengan bukti kepada Pengembang. Demikian pula, Pengembang diminta untuk mengkonfirmasi langsung dengan Pengguna setelah perbaikan bug bersama dengan pembaruan di Bugzilla.



3.3.2 Defect reporting using Selenium

SEMUA cacat harus dicatat menggunakan [Selenium], untuk menangani dan melakukan debugging cacat. Pengguna juga diminta untuk mengirim laporan cacat harian kepada

pengembang. Pengembang akan memperbarui daftar cacat di [Selenium] dan memberitahu pemohon setelah cacat telah diselesaikan. Pengembang dan Pengguna diminta untuk meminta akun di [Selenium] untuk ruang kerja relatif.

Debugging harus didasarkan pada Prioritas Tinggi > Sedang > Rendah, prioritas ini ditetapkan oleh Pengguna dan didasarkan pada seberapa kritis skrip pengujian dalam hal ketergantungan dan terutama didasarkan pada skenario kasus penggunaan.

- Semua cacat prioritas Tinggi harus ditangani dalam 1 hari setelah permintaan dan diselesaikan/ditutup dalam 2 hari setelah permintaan awal.
- Semua cacat prioritas Sedang harus ditangani dalam 2 hari setelah permintaan dan diselesaikan/ditutup dalam 4 hari setelah permintaan awal.
- Semua cacat prioritas Rendah harus diselesaikan/ditutup paling lambat 5 hari setelah permintaan awal.

3.4 Reports

Pemimpin Pengujian akan bertanggung jawab untuk menulis dan mendistribusikan laporan berikut kepada personel proyek yang sesuai sesuai kebutuhan.

3.4.1 Testing status reports

Laporan status mingguan atau dua mingguan akan diberikan oleh Pemimpin Pengujian kepada personel proyek. Laporan ini akan merangkum kegiatan pengujian mingguan, masalah, risiko, jumlah bug, cakupan kasus pengujian, dan metrik relevan lainnya.

3.4.2 Phase Completion Reports

Ketika setiap fase pengujian selesai, Pemimpin Pengujian akan mendistribusikan Laporan Penyelesaian Tahap kepada manajer Produk, Pemimpin Pengembangan, dan Manajer Program untuk peninjauan dan penandatanganan. Poin-poin di bawah ini mengilustrasikan contoh apa yang mungkin termasuk dalam dokumen tersebut. Dokumen tersebut harus berisi metrik berikut:

- Total Kasus Pengujian, Jumlah yang Dieksekusi, Jumlah Lulus / Gagal, Jumlah yang Belum Dieksekusi.
- Jumlah Bug yang Ditemukan Hingga Saat Ini, Jumlah yang Diselesaikan, dan Jumlah yang Masih Terbuka.
- Rincian Bug berdasarkan Matriks Tingkat Keparahan / Prioritas.
- Diskusi Risiko yang Belum Terselesaikan.
- Diskusi Kemajuan Jadwal (apakah kita berada di tempat yang seharusnya?).

3.4.3 Test Final Report - Sign-Off

Laporan Pengujian Akhir akan dikeluarkan oleh Pemimpin Pengujian. Ini akan menyatakan sejauh mana pengujian sebenarnya telah selesai (disarankan laporan cakupan kasus pengujian), dan penilaian kesiapan produk untuk Dirilis ke Produksi.

3.5 Responsibility Matrix

Insert testing resource matrix – who is involved and what is the role.

4 Resource & Environment Needs

4.1 Testing Tools

4.1.1 Tracking Tools

Pelacak bug seperti Selenium digunakan oleh F4 untuk memasukkan dan melacak semua bug dan masalah proyek. Pemimpin Pengujian bertanggung jawab untuk memelihara database Selenium

- Pelacakan Bug: Jira, Asana, Trello, atau yang serupa.
- Manajemen Kasus Pengujian: TestRail, Zephyr, qTest, atau terintegrasi dengan alat pelacakan bug.
- Alat Otomatisasi: Selenium WebDriver, Cypress, Playwright (untuk otomatisasi UI web).
- Alat Pengujian Kinerja: JMeter, LoadRunner, k6.
- Alat Pengujian Keamanan: OWASP ZAP, Burp Suite.
- Alat Komunikasi: Slack, Microsoft Teams.

4.2 Test Environment

4.2.1 Hardware

Sertakan persyaratan perangkat keras minimum yang akan digunakan untuk menguji Aplikasi. Pengujian akan memiliki kontrol akses ke satu atau lebih server aplikasi/database yang terpisah dari yang digunakan oleh anggota tim proyek non-pengujian. Pengujian juga akan memiliki kontrol akses ke sejumlah workstation PC yang dikonfigurasi secara bervariasi untuk memastikan pengujian dari berbagai konfigurasi perangkat keras klien minimum hingga yang direkomendasikan yang tercantum dalam dokumen Persyaratan, Spesifikasi Fungsional, dan Spesifikasi Desain proyek.

Untuk Airbnb, ini meliputi:

- Server Pengujian: Server aplikasi dan database khusus untuk pengujian, terpisah dari lingkungan pengembangan dan produksi.
- Workstation Klien: Berbagai konfigurasi PC, Mac, dan perangkat seluler (aktual atau emulasi) untuk menguji responsivitas dan kompatibilitas.

4.2.2 Software

Selain aplikasi dan perangkat lunak lain yang ditentukan pelanggan, daftar perangkat lunak berikut harus dianggap sebagai minimum:

- Sistem Operasi: Windows 10, macOS Ventura/Sonoma, berbagai distribusi Linux.
- Peramban Web (Web Browsers): Versi terbaru Chrome, Firefox, Edge, Safari (di macOS/iOS).
- Alat Pengembangan (untuk Pengembang): IDEs, klien kontrol versi.
- Lainnya: Klien VPN (jika lingkungan pengujian dibatasi), plugin/ekstensi browser spesifik jika diperlukan untuk pengujian.

4.3 Bug Severity and Priority Definition

Bidang Tingkat Keparahan dan Prioritas Bug keduanya sangat penting untuk mengkategorikan bug dan memprioritaskan apakah dan kapan bug akan diperbaiki. Tingkat Tingkat Keparahan dan Prioritas bug akan didefinisikan sebagaimana diuraikan dalam tabel berikut di bawah ini. Pengujian akan menetapkan tingkat keparahan untuk semua bug. Pemimpin Pengujian akan bertanggung jawab untuk memastikan bahwa tingkat keparahan yang benar ditetapkan untuk setiap bug.

Pemimpin Pengujian, Pemimpin Pengembangan, dan Manajer Program akan berpartisipasi dalam rapat tinjauan bug untuk menetapkan prioritas semua bug yang saat ini aktif. Rapat ini akan dikenal sebagai Rapat Triage Bug. Pemimpin Pengujian bertanggung jawab untuk mengatur rapat-rapat ini secara rutin untuk menangani kumpulan bug baru dan yang ada tetapi belum terselesaikan saat ini.

4.3.1 Severity List

Penguji yang memasukkan bug ke Selenium] juga bertanggung jawab untuk memasukkan Tingkat Keparahan bug.

Severity ID	Severity Level	Severity Description
1	Critical	Modul/produk crash atau bug menyebabkan kondisi yang tidak dapat dipulihkan. Sistem crash, GP Faults, atau korupsi database atau file, atau potensi kehilangan data, program hang yang membutuhkan reboot adalah semua contoh bug Sev. 1. Untuk Airbnb: Aplikasi tidak responsif, ketidakmampuan untuk menyelesaikan pemesanan, kehilangan data pengguna/listing.
2	High	Komponen sistem utama tidak dapat digunakan karena kegagalan atau fungsionalitas yang salah. Bug Sev. 2 menyebabkan masalah serius seperti kurangnya fungsionalitas, atau pesan kesalahan yang tidak memadai atau tidak jelas yang dapat berdampak besar pada pengguna, mencegah area lain dari aplikasi untuk diuji, dll. Bug Sev. 2 dapat memiliki workaround, tetapi workaround tersebut tidak nyaman atau sulit. Untuk Airbnb: Pencarian tidak mengembalikan hasil, kesalahan gateway pembayaran yang mencegah transaksi, host tidak dapat membuat listing.
3	Medium	Fungsionalitas komponen atau proses yang salah. Ada workaround sederhana untuk bug jika itu Sev. 3. Untuk Airbnb: Masalah tampilan minor, pesan kesalahan yang salah yang tidak menghalangi fungsionalitas, masalah paginasi.
4	Minor	Kesalahan dokumentasi atau bug severity 3 yang ditandatangani. Untuk Airbnb: Kesalahan ketik, ketidaksejajaran UI minor, masalah estetika yang tidak memengaruhi fungsionalitas.

4.3.2 Priority List

Priority ID	Priority Level	Priority Description
-------------	----------------	----------------------

5	Must Fix	Bug ini harus segera diperbaiki; produk tidak dapat dirilis dengan bug ini.
4	Should Fix	Ini adalah masalah penting yang harus diperbaiki sesegera mungkin. Akan memalukan bagi perusahaan jika bug ini dirilis.
3	Fix When Have Time	Masalah harus diperbaiki dalam waktu yang tersedia. Jika bug tidak menunda tanggal rilis, maka perbaiki.
2	Low Priority	Tidak penting (saat ini) bahwa bug ini ditangani. Perbaiki bug ini setelah semua bug lain diperbaiki.
1	Trivial	Peningkatan/Fitur yang bagus untuk dimiliki yang hanya di luar lingkup saat ini.

4.4 Bug Reporting

Tim pengujian mengakui bahwa proses pelaporan bug adalah alat komunikasi kritis dalam proses pengujian. Tanpa komunikasi informasi bug dan masalah lainnya yang efektif, proses pengembangan dan rilis akan terpengaruh secara negatif. Pemimpin Pengujian akan bertanggung jawab untuk mengelola proses pelaporan bug. Alat dan proses pelaporan bug

Standar pengujian akan digunakan. [Selenium] adalah alat Pencatat / Pelacak Bug standar perusahaan. Pengujian dan pengembangan akan memasukkan data mereka ke dalam database Selenium mengikuti definisi entri bidang yang didefinisikan di bawah ini.

5 Terms/Acronyms

The below terms are used as examples, please add/remove any terms relevant to the document.

TERM/ACRONYM	DEFINITION
API	Application Program Interface
BAH	Booz Allen Hamilton
BCP	Business Continuity Plan
CAT	Client Acceptance Testing
End-to End Testing	Tests user scenarios and various path conditions by verifying that the system runs and performs tasks accurately with the same set of data from beginning to end, as intended.
ER;ES	Electronic Records; Electronic Signatures
N/A	Not Applicable
NCI	National Cancer Institute
QA	Quality Assurance
RTM	Requirements Traceability Matrix
SME	Subject Matter Expert
SOP	Standard Operating Procedure

TERM/ACRONYM	DEFINITION
TBPT	Tissue Bank and Pathology Tools
TBD	To Be Determined
TSR	Test Summary Report