**Jawaban Pertanyaan Singkat**

* **200 OK**

Deskripsi: Menunjukkan bahwa permintaan berhasil. Ini adalah respons standar untuk permintaan HTTP yang sukses.

* **404 Not Found**

Deskripsi: Menunjukkan bahwa sumber daya yang diminta tidak dapat ditemukan di server. Ini adalah respons umum untuk URL yang salah atau tidak valid.

* **403 Forbidden**

Deskripsi: Server memahami permintaan, tetapi server menolak untuk memprosesnya. Ini menunjukkan bahwa pengguna tidak memiliki izin yang sesuai untuk mengakses sumber daya tersebut.

* **500 Internal Server Error**

Deskripsi: Menunjukkan bahwa terjadi kesalahan server internal yang mencegah server untuk memproses permintaan. Ini adalah respons umum untuk kesalahan yang tidak diantisipasi atau tidak diatasi.

* **302 Found (atau 301 Moved Permanently)**

Deskripsi: Ini adalah jenis pengalihan. 302 menunjukkan bahwa sumber daya telah ditemukan sementara di lokasi lain, dan klien seharusnya mengikuti pengalihan tersebut. 301 menunjukkan bahwa pengalihan bersifat permanen.

* **Severity**

Tingkat keparahan bug dalam software, untuk mengukur sejauh mana dampak yang dihasilkan oleh bug tersebut terhadap sistem atau fungsionalitas.

* **Priority**

Tingkat kepentingan relatif dari suatu masalah atau pekerjaan, yang menentukan seberapa cepat masalah tersebut perlu diperbaiki atau pekerjaan tersebut perlu diselesaikan.

1. Product Requirement Document (PRD) adalah dokumen yang merinci persyaratan dan spesifikasi produk yang akan dikembangkan. PRD berisi informasi rinci tentang tujuan produk, fungsionalitas yang diinginkan, kebutuhan pengguna, batasan sistem, dan persyaratan teknis lainnya.

**Hubungannya dengan Peran QA (Quality Assurance):**

Peran QA dalam siklus pengembangan perangkat lunak sangat erat kaitannya dengan PRD.

* **Pemahaman Persyaratan:**

QA harus memiliki pemahaman mendalam tentang setiap persyaratan yang dijelaskan dalam PRD. Ini termasuk fungsionalitas produk, kebutuhan pengguna, dan skenario penggunaan yang diharapkan.

* **Pengujian Berbasis Persyaratan:**

QA menggunakan PRD sebagai dasar untuk merencanakan dan melakukan pengujian perangkat lunak. Setiap fitur atau fungsionalitas yang dijelaskan dalam PRD akan diuji untuk memastikan bahwa produk memenuhi persyaratan yang telah ditetapkan.

* **Pengembangan Kasus Uji (Test Case):**

PRD membantu QA dalam mengidentifikasi dan mengembangkan kasus uji yang komprehensif. Setiap skenario pengujian dirancang untuk memastikan bahwa fungsionalitas dan persyaratan utama dari PRD tercakup dalam pengujian.

* **Verifikasi dan Validasi:**

QA menggunakan PRD sebagai pedoman untuk melakukan verifikasi dan validasi. Verifikasi adalah proses memastikan bahwa produk sesuai dengan persyaratan PRD, sementara validasi adalah memastikan bahwa produk memenuhi kebutuhan pengguna dan memecahkan masalah bisnis.

* **Feedback ke Tim Developer:**

QA memberikan umpan balik kepada tim pengembangan berdasarkan hasil pengujian terhadap PRD. Jika ditemukan ketidaksesuaian atau masalah, QA bekerja sama dengan pengembang untuk memahami dan memperbaiki masalah tersebut.

1. Dalam konteks Product Requirement Document (PRD), istilah "User" atau "Pengguna" merujuk kepada individu yang akan menggunakan produk yang sedang dikembangkan. Hal ini penting dalam mengidentifikasi kebutuhan dan menentukan fitur serta fungsionalitas produk.

* **Reproduksi Bug:**

Usahakan untuk mereproduksi bug secara konsisten. Pastikan langkah-langkah yang diperlukan untuk memunculkan bug tersebut dapat diidentifikasi dan diulangi.

* **Pencatatan Detail Bug:**

Buat catatan detail tentang bug, termasuk langkah-langkah untuk mereproduksinya, deskripsi yang jelas tentang bug tersebut, serta informasi tambahan seperti lingkungan pengujian dan data masukan yang digunakan.

* **Pemberian prioritas dan Severity:**

Tetapkan tingkat prioritas dan severity pada bug. Prioritas menentukan seberapa cepat bug tersebut perlu diperbaiki, sedangkan severity menunjukkan seberapa parah dampaknya terhadap fungsionalitas atau sistem.

* **Pemberitahuan ke Tim Pengembangan:**

Beri tahu tim pengembangan secepat mungkin. Gunakan alat pelacakan bug atau sistem manajemen proyek yang digunakan oleh tim untuk membuat laporan bug.

* **Pengujian Regresi:**

Jika perlu, lakukan pengujian regresi untuk memastikan bahwa perbaikan bug tidak mempengaruhi fungsionalitas yang ada secara negatif.

* **Kolaborasi dengan Tim Pengembangan:**

Bekerjasama dengan tim pengembangan untuk memberikan informasi tambahan, menjelaskan konteks, atau memberikan dukungan apapun yang diperlukan untuk membantu mereka memahami dan memperbaiki bug.

* **Verifikasi Bug yang Diperbaiki:**

Setelah pengembang mengklaim bahwa bug telah diperbaiki, verifikasi bahwa perbaikan tersebut berhasil. Lakukan pengujian untuk memastikan bahwa bug tidak lagi muncul atau tidak ada dampak negatif yang timbul dari perbaikan.

* **Dokumentasi:**

Perbarui dokumen pengujian, termasuk catatan uji dan skenario yang terkait dengan bug tersebut.

* **Analisis Akar Penyebab (Root Cause Analysis):**

Melakukan analisis akar penyebab untuk memahami mengapa bug tersebut muncul. Ini dapat membantu mencegah munculnya bug serupa di masa depan.

* **Keterlibatan Pemangku Kepentingan (Stakeholders):**

Beri tahu pemangku kepentingan tentang bug dan perbaikannya, terutama jika bug tersebut memiliki dampak signifikan pada pengguna atau proyek secara keseluruhan.

**Dokumentasi:**

Catat semua detail yang diberikan oleh tim pihak luar, termasuk langkah-langkah untuk mereproduksi bug, deskripsi masalah, dan informasi lingkungan yang relevan.

**Verifikasi Bug:**

Cobalah untuk mereproduksi bug dengan mengikuti langkah-langkah yang telah diberikan. Pastikan bahwa masalah tersebut memang ada dan dapat diidentifikasi.

**Penetapan Prioritas dan Severity:**

Tetapkan tingkat prioritas dan severity pada bug berdasarkan informasi yang diberikan dan pemahaman saya tentang dampak bug tersebut terhadap pengguna atau bisnis.

**Pemberian Tanggapan Cepat:**

Berikan tanggapan cepat kepada pihak yang melaporkan bug untuk memberi tahu mereka bahwa laporan mereka diterima dan dalam proses penanganan.

**Kolaborasi dengan Tim Pengembangan:**

Bekerjasama dengan tim pengembangan untuk mendiskusikan laporan bug, memberikan informasi tambahan yang mungkin diperlukan, dan mendukung upaya perbaikan.

**Pembaruan Status:**

Berikan pembaruan status secara teratur kepada pihak yang melaporkan bug. Ini dapat mencakup informasi tentang langkah-langkah yang telah diambil, perkiraan waktu penyelesaian, dan status perbaikan.

**Verifikasi Perbaikan:**

Setelah pengembang mengklaim bahwa bug telah diperbaiki, lakukan verifikasi untuk memastikan bahwa perbaikan tersebut berhasil dan tidak menimbulkan dampak negatif pada fungsionalitas lain.

**Pendokumentasian dan Pelaporan Keberhasilan:**

Catat semua langkah yang diambil, informasi yang diberikan, dan hasil dari verifikasi perbaikan. Ini penting untuk dokumentasi dan pelaporan keberhasilan penanganan bug.

**Feedback dan Pembelajaran:**

Jika diperlukan, berikan umpan balik kepada tim pihak luar tentang status dan hasil penanganan bug. Gunakan kesempatan ini untuk pembelajaran bersama dan meningkatkan komunikasi di masa depan.