Task

1.

As your first task, help the dev teams prioritise what features should be tested and why, between the following options:

- Authentication flows (sign-up, sign-in, password reset)
- Feeds (loading and pagination for both the user's feed and the global feed)
- Articles (creating, editing, reading, and deleting an article)
- Following authors (following a new author, feed getting updated, unfollowing)
- Comments (creating, reading, deleting)
- API Contracts

Respuesta

Para priorizar los test he considerado los siguientes factores: **importancia para la empresa, riesgo, complejidad, frecuencia de uso y nivel en la web (portadas, detalles)** . Por lo que mi orden sería el siguiente:

- **Authentication flows (sign-up, sign-in, password reset)**

  Esta función es fundamental para el sistema, sin que los usuarios puedan registrase o iniciar sesión, no pueden interactuar como es debido en la web además de la mala imagen para la empresa que ya al inicio del uso de la web tengas problemas de usabilidad. Ya no solo deben ser procesos que no fallen, también es muy importante que sean ágiles.

- **API Contracts**

  Esta función también es muy importante porque hay que comprobar que las API funcionen correctamente para el funcionamiento global de la web.

- **Articles (creating, editing, reading, and deleting an article)**

  El artículo es la entidad principal de la web, de él se extraen los feeds, detalles etc... sin que la parte de creación, edición, eliminación funcione bien, no sería viable la web y crearía frustración a los usuarios.

- **Feeds (loading and pagination for both the user's feed and the global feed)**

  A menos nivel de importancia pero también siendo parte fundamental la visualización correcta y la paginación de los feeds.

- **Comments (creating, reading, deleting)**

  La gestión de los comentarios es importante para que los usuarios interactuen entre sí y que la web vaya creciendo en popularidad. Los test aunque no son tan prioritarios como los anteriores, también son importantes.

- **Following authors (following a new author, feed getting updated, unfollowing)**

  Esta función para mí como usuario es importante pero para el sistema de testeo de esta web considero que es la que menos importancia tiene, pero no por ello se debería dejar sin testear.

2.

Once you have a prioritised list with your reasoning, prepare a test plan which will cover the features mentioned on it. This test plan should contain a standardised suite of tests with the usual fields (steps, expected results, etc.). It is paramount that test cases are described with the utmost clarity and detail, so as to allow any QA Engineer with no previous knowledge of the application's domain to properly follow and execute the cases. Every test case should have a criticality rating, as well as an impact rating. This will lead to the following task.

Respuesta

Para Conduit yo realizaría un documento con esta estructura:

Índice

Conduit Project

- **1. Introducción**
  - 1.1. Objetivo del Plan de Pruebas:

    Este documento, tiene como finalidad entregar las pautas y definir la estrategia que se seguirá para conseguir una web de Calidad. El objetivo general del plan es establecer la cronología y condiciones para la aplicación de las pruebas de manera de obtener, un sistema que pueda ser completado con una recepción total de los interesados y entrar en operación con la totalidad de las funcionalidades requeridas para su funcionamiento.

    En cada uno de los archicos de pruebas (xxx-spec.ts), usaremos un el patrón AAA (Arrange-Act-Assert) usado en diseños TDD.

- **2. Alcance de las Pruebas**
  - 2.1. Resumen de las Pruebas a realizar

    Para ello la mejor forma de representarlo es una tabla con todas las especificaciones desalladas, donde podemos ver Inicio de las pruebas, módulos a probar, orden, nivel de impacto etc.....

Inicio	Módulo	Ord. N.Imp.	Objetivo	Archivos	Detalle	Responsable
16/01/2024	Authentication	1 Alto	Comprobar que todas las funcionalidades de Registro, Login, Recuperación de Password funcionen correctamente.	-login-spec.ts -register-spec.ts -force-logout-spec.ts	Una vez que el departamento de desarrollo de por finalizadas las pruebas unitarias de cada una de las partes de este módulo, se realizan las pruebas end2end con cypres, para ello habrá que realizar los test de los archivos indicados.	Equipo QA
16/01/2024	API Contracts	2 Alto	Comprobar que la API funciona correctamente.		Para hacer las pruebas de la API se utilizará Postman, para comprobar que todas las llamadas API Rest funciones correctamente y reliazaremos también las pruebas de carga de peticiones masivas.	Equipo QA
16/01/2024	Articles	3 Alto	Realizar todas las pruebas necesarias para que se creen, modifique y elimine correctamente los artículos.	-new-post-spec.ts	Una vez finalizadas las pruesas unitarias, se realizarán las pruebas end2end con cypress, donde comprobaremos que todas las pruebas referentes a la creación, modificacion, eliminación de artículo funcionan correctamente.	Equipo QA
16/01/2024	Feeds	4 Medio	Se realizaran las pruebas necesarias para comprobar su funcionamiento correcto.	-feeds-spec.ts -pagination-spec.ts	Tras dar el visto bueno desde el departamento de desarrollo a las pruebas unitarias, se realizaran las pruebas end2end con cypress.	Equipo QA
16/01/2024	Comments	5 Medio	Realizar todas las pruebas necesarias para que se creen, modifique, elimine y validen correctamente los comentarios.	-comments-spec.ts	Al terminar las pruebas unitarias, realizaremos las pruebas end2end con cypress para comprobara que todos los procesos de creación, modificación y eliminación funcionan como debes.	Equipo QA
16/01/2024	Following authors	6 Bajo	Comprouraremos que la funcionalidad es correcta	-follow-user-spec.ts	Se realizaran las pruebas end2end con cypress una vez que hayan dado el visto bueno a las pruebas unitarias desde el departamento de desarrollo.	Equipo QA

  - 2.2. Casos de las Pruebas:

    En esta tabla visualizamos el número de pruebas que se realizarán agrupandolas por el módulo y el tipo. La finalidad de este cuadro es saber el número tal de pruebas que se van a realizar.

Casos Disponibles	Módulo	Tipo	Total Casos
120	Authentication	end2end	120
12	API Contracts	Postman	12
95	Articles	end2end	95
36	Feeds	end2end	36
72	Commenets	end2end	72
11	Following authors	end2end	11
Total			**346**

- **3. Entorno y configuración de las Pruebas**
  - 3.1. Descripción de entornos de Pruebas

    Para el proceso de pruebas del proyecto se requiere de la disponibilidad de los siguientes entornos y software:

    Servidor Windows Server 2022 con Internet Information Server 7.0.

    - Node.js 18x.+
    - Git
    - Jenkins
    - Java SE
    - npm
    - Cypress.io

  - 3.2. Criterios de Inicio

    Aceptación del plan de pruebas.

    Aceptación de paquetes.

    Revisión y aceptación de los paquetes de desarrollo, y que este cumpla con las condiciones de aceptación.

    Aceptación de ambiente.

    Revisión y aceptación del ambiente, y que este cumpla con las condiciones de aceptación

  - 3.3. Criterios de aprovación/rechazo.

    Errores Graves: información crítica presentada erróneamente, información mal registrada en la base de datos, caídas de programas, incumplimiento de objetivos en funciones principales, bloqueantes para el usuario, etc.

    Errores Medios (comunes): todos los errores similares a los anteriores pero que no sean bloqueantes para el usuario.

    Errores Leves: errores en presentación de datos secundarios, no adecuación a estándares, comportamientos correctos pero diferentes en situaciones similares, dificultades de operación, etc.

3.

Prepare an automated suite of tests for those scenarios which you consider "worthy" of being automated. You may use the tools you feel most comfortable with. Try and focus on scalability, reusability, and simplicity. You will be expected to reason about decisions regarding selector usage, error handling, SoC, CI/CD integration, efficiency, brittleness, performance, and readability.

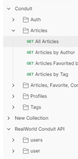
Respuesta

Como ya hemos mostrado en los cuadros anteriores de planificación de las prácticas, las pruebas se clasificarían en dos modalidades: **Pruebas API Rest** y **Pruebas end2end**, ambas modalidades deben ser automatizadas.

Para las **Pruebas API Rest** el software que se utilizará será **Postman**, actualmente es uno de los más utilizado por su sencillez y eficacia y da la posibilidad de programar los test.

Para las **Pruebas end2end** combinaremos el software de testing **Cypress.io** con el sistema de integración continua (CI/CD) **Jenkins**. Con jenkins podemos realizar las pruebas end2end de forma automática sin ningun problema además de realizar pasa a paso un integración continua.

Para las pruebas **end2end** cogeremos cada uno de los archivos de los módulos los clasificaremos y agruparemos por cada módulo y crearemos pipelines **sencillos y escalables**.



Los archivos **spec** tiene que estar bien creados y estructurados definiendo claramente con mensajes cada funcíon que se está comprobando para que cuando se lancen las pruebas y haya algun error podamos ver de forma detalla qué es lo que ha ocurrido.

```
describe("Conduit Login", () => {
  before(() => cy.registerUserIfNeeded());
  beforeEach(() => {
    cy.visit("/");
    // we are not logged in
  });

  it("Does not work with wrong credentials", () => {
    cy.contains("a.nav-link", "Sign in").click();

    cy.get('input[type="email"]').type("wrong@email.com");
    cy.get('input[type="password"]').type("no-such-user");
    cy.get('button[type="submit"]').click();

    // error message is shown and we remain on the login page
    cy.contains(".error-messages li", "User Not Found");
    cy.url().should("contain", "/login");
  });
```

Una vez comprobados los archivos specs los agruparemos segun su módulo y crearemos en Jenkins un **pipeline** por cada uno de los módulos que queremos probar, así conseguiremos que sea **sencillo y estructurado**.

S	W	Nombre	Último Éxito	Último Fallo ↓	Última Duración
✕	☔	Test_Authentication	N/D	3 Hor 6 Min #8	25 Seg
✕	☔	Test_Article	N/D	2 Hor 33 Min #3	12 Seg
✕	☔	Test_Feeds	N/D	2 Hor 20 Min #2	11 Seg
✕	☔	Test_Comments	N/D	2 Hor 14 Min #2	11 Seg
✕	☔	Test_Following_authors	N/D	2 Hor 11 Min #2	11 Seg

Estos pipelines son fáciles de crear y reutilizar, además de poder ser programados y ser lanzados de forma **automática** cuando nosotros queramos. De igual forma podríamos agruparlos o dividirlos segun la sencillez o complejidad del proyecto.

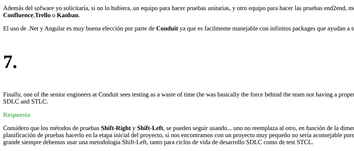
4.

The current Conduit infrastructure does not boast powerful resources. This means that the CI/CD pipelines where the tests will be executed can't handle multiple agents with multiple threads. With the criticality and impact ratings from the previous tasks, decide what would be an ideal regression testing landscape, which tests should be included in which execu-tion pipelines, and why. If you consider there are any particular algorithms or tech-niques to be used to reduce the amount of regression tests without compromising coverage, briefly discuss them. Also, mark the tests (if any) that you consider worthy of being tested in a multi-browser environment.

Respuesta

Tanto las pruebas iniciales como las de regresión están incluidas dentro de un proceso de integración continua en nuestro caso con Jenkins.

Stage View



Los diferentes tipos de pruebas de regresión son: **Pruebas de regresión correctivas**, **Pruebas de regresión progresivas**, **Pruebas de regresión retrospectivas**

En el caso de **Conduit** solo sería necesario hacer las **Pruebas de regresión correctivas**, ya que de lo que se quejan los usuarios es de los errores que se encuentran a la hora de usar la web, no piden actualizaciones para que tuviéramos que realizar **Pruebas de regresión progresivas** ni tampoco se va a realizar ningun cambio de plataforma para tener que hacer **Pruebas de regresión retrospectivas**.

Una vez decidido que se van a hacer este tipo de pruebas de regresión debemos hacer una selección de dichas pruebas:

- **Selección de pruebas de regresión**: Consiste en seleccionar un subconjunto de casos de prueba que sean relevantes para los cambios realizados en el software, y descartar los que no lo sean. Esto reduce el tiempo y el esfuerzo de ejecutar pruebas innecesarias. Existen diferentes criterios y métodos para realizar la selección de pruebas de regresión, como el análisis de impacto, la trazabilidad de requisitos, la priorización de casos de prueba, etc.
- **Minimización de pruebas de regresión**: Consiste en eliminar los casos de prueba redundantes o duplicados que no aportan valor a la cobertura. Esto reduce el tamaño del conjunto de pruebas de regresión y mejora la eficiencia. Existen diferentes técnicas para realizar la minimización de pruebas de regresión, como el análisis de dependencias, la agrupación de casos de prueba, el uso de heurísticas, etc.
- **Automatización de pruebas de regresión**: Consiste en utilizar herramientas y scripts que permitan ejecutar las pruebas de regresión de forma automática, sin intervención humana. Esto reduce el error humano, aumenta la velocidad y la consistencia de las pruebas, y permite ejecutarlas en diferentes entornos y configuraciones. Existen diferentes herramientas y frameworks para realizar la automatización de pruebas de regresión, como Selenium, TestNG, Cucumber, etc.

Para hacer una buena selección de las pruebas de regresión hay que conocer bien el producto y eliminnar ya pruebas pasadas que sabemos perfectamente que funcionan con toda seguridad, en definitiva no realizar pruebas nuevamente que sabesmos que no fallan.

Para que las pruebas de regresión sean mínimas lo ideal es que inicialmente se haga un estudio exhaustivo de todos los casos de pruebas, para evitar posibles fallos cuando se realicen cambios en el software.

A nivel de multi-browser haría pruebas de edicion de contenido simultaneos

5.

Conduit does have a suite of tests for their API (one of the developers had worked with Postman before, and generated a collection of tests). The spec and collection are located here: Conduit API Spec Go through the collection briefly, and see if there's anything you would add, change, or remove. In case some of the tests should be included in any of the regression pipe-lines, do state why.

Respuesta

Cuando se pregunta qué se puede mejorar en algo, es seguro que habrá cosas...pero yo sinceramente veo que es una collection muy completa, no se me ocurre cómo ampliarla.

6.

After all your hard work, the Conduit's tests feel more than confident that you could be a very good asset to the team and the company. That's why they wish to know more about you and your approach to QA. Please navigate to the QA Roadmap, and mark as "Done" (right click) each of the topics you consider yourself to be proficient with. Then you can "Share Progress" obtaining a personalised URL. Once you've established your knowledge, they mention that they have some spare budget to consider switching stack for the newly created QA department. This means you have total freedom of choice in terms of what tools you'd like to use for your QA needs. Frameworks; Assertion Libraries, Test Management, Bug Tracking... Anything that you require would be considered by the financial department and acquired. What would be your ideal tool stack, and why? Bear in mind that the current architec-ture of Conduit is a .NET micro-service backend, with an Angular frontend SPA.

Respuesta

Este sería mi QA Roadmap, me he basado en las tecnología y metodologías que considero más apropiadas para este 2024 y también las que he usado preferentemente yo en mi trabajo día a día.

Además del software yo solicitaría, si no lo hubiera, un equipo para hacer pruebas unitarias, y otro equipo para hacer las pruebas end2end, montaje del servidor de Jenkins etc... y respecto al software principalmente lo que ya hemos mencionado anteriormente, **cypress.io, Jenkins**. **postman, Jira, Confluence, Trello** y **Kanban**.

El uso de .Net y Angular es muy buena elección por parte de **Conduit** ya que es facilmente manejable con infinitos packages que ayudan a testear, generar reports etc... por lo que los costes de montaje de un sistema de Pruebas no es un gasto muy grande para la empresa.

7.

Finally, one of the senior engineers at Conduit sees testing as a waste of time (he was basically the force behind the team not having a proper testing landscape). He wants to know your approach towards the "new fad" of Shift-Left, and if you consider it wiser than a Shift-Right approach to the SDLC and STLC.

Respuesta

Considero que los métodos de pruebas **Shift-Right** y **Shift-Left**, se pueden seguir usando... uno no reemplaza al otro, en función de la dimensión del proyecto usaremos una metodología u otra. Por ejemplo aunque la tendencia actual sea realizar un Shift-Left, es decir todo el estudio de casos y planificación de pruebas hacerlo en la etapa inicial del proyecto, si nos encontramos con un proyecto muy pequeño no sería aconsejable porque podemos aumentar el plazo de entrega para un proyecto que a priori es sencillo, por el contrario siempre que el proyecto sea de una dimensión mediana o grande siempre debemos usar una metodología Shift-Left, tanto para ciclos de vida de desarrollo SDLC como de test STCL.