

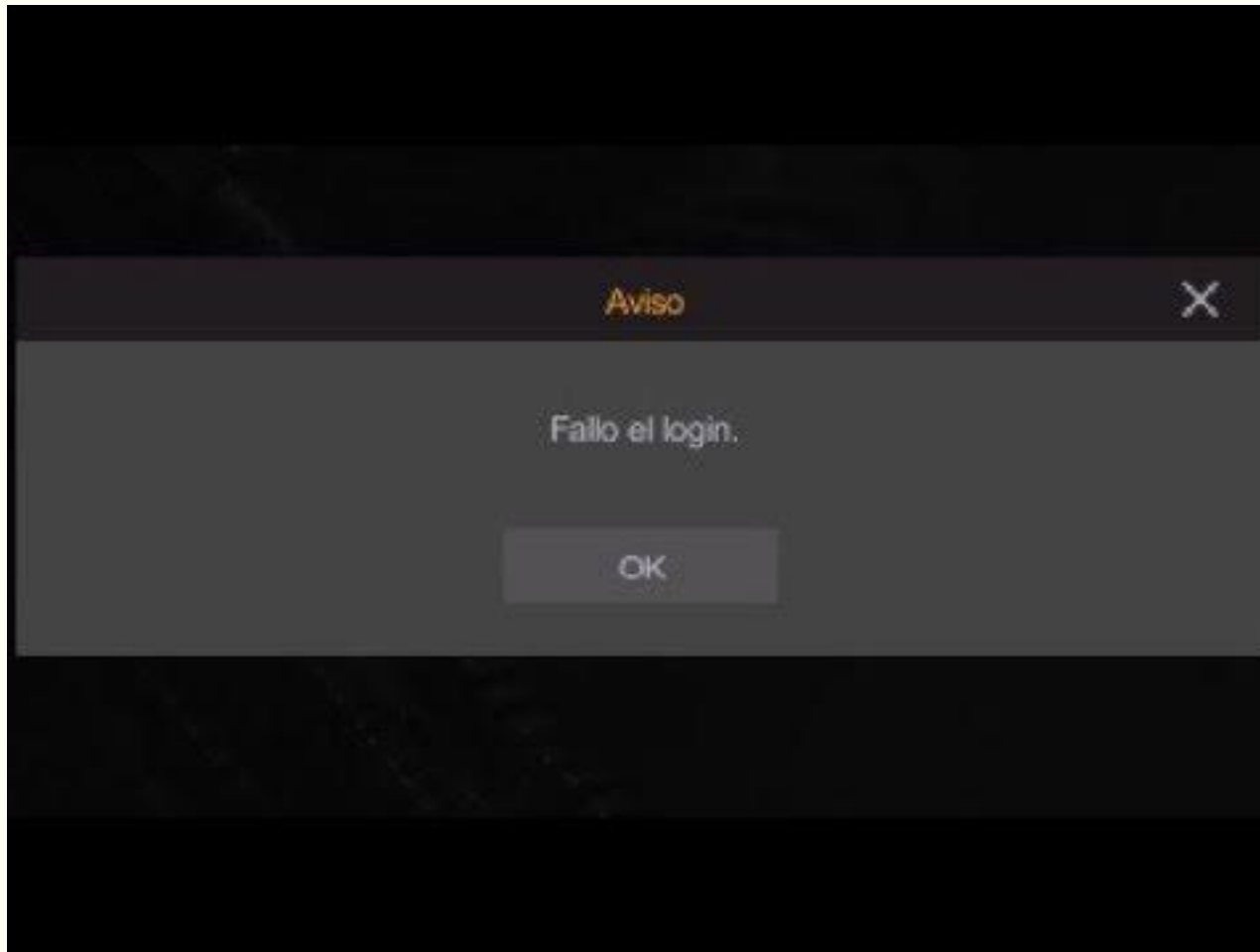


III. ERRORES

“It works on my machine.”

III – Errores

3.1 Reporte de Defectos



III – Errores



III – Errores

3.1 Reporte de Defectos

- Se entiende por defecto, cualquier error o carencia en los productos (software y documentación) en su fase de desarrollo.
- La formalización de los defectos incluye el conjunto de actividades encaminadas a registrar, clasificar y asignar los defectos al responsable correspondiente para su posterior resolución, evitando situaciones de bloqueo que impidan la ejecución de los servicios acordados

El **Bug** es el nombre informal para los defectos, lo que significa que una aplicación o software no está funcionando según los requisitos.

También se conoce como: issue, error, fault, o failure.



Antes de reportar un informe de un fallo:

- Volver a reproducir
- Reproducir en otra máquina
- Pair Testing con un compañero
- Comunicación clara y concisa



III – Errores

3.1 Reporte de Defectos

1. Steps to Reproduce

2. Expected Result

3. Current Result

4. Details

1. Environments
2. Browsers/OS
3. Resolution

5. Attachments

6. Others:

1. Component Affected
2. Affected version
3. Priority
- 4

Rider-34864 Created by Scott Leckie 2 weeks ago Updated by Ilya Usov 3 days ago Visible to All Users

☆ Edit and Continue - produces error when used with Razor pages

What steps will reproduce the issue? **1**

1. Project with Razor pages, and a component file
2. Break in C# code
3. Make code change, apply and continue

What is the expected result? **2**

Should apply the change and continue.

What happens instead? **3**

Get multiple errors in Edit and Continue Build Results window, all of the same form:

```
EnvironmentsComponent.razor.g.cs(16, 2): [CS1024] Preprocessor directive expected
```

Double clicking one of these messages takes me to a "razor.g.cs" version of one of my files ("EnvironmentComponent" in this example). The full .g.cs file is attached, but here's the offending line according to the above erro (error takes you to line 1 of this subset):

```
csharp
1 #nullable restore
2 #line 1 "C:\data\devstudio\usermgmt-backend\user-management\UserManagementAPI.Management\Component
3 using System.Web;
```

The only resolution is to close the debug session and start a new one. **4**

Attachments

- EnvironmentsComponent.razor: Original component file
- EnvironmentsComponent.razor.g.cs: Generated cs file that seems to upset the Edit and Continue process
- 2019-11-05-15-37-10.gif: Animated gif of problem

Attachments **3** +

EnvironmentsComponent.razor.g.cs EnvironmentsComponent.razor

Project: Rider **6**

Type: Bug

User priority: Annoying

Technology: .Net Core

OS: Windows

Subsystem: Debug - Edit & Continue

State: To Reproduce

Assignee: Tatiana Kapustina

Fixed in build: No Fixed in build

Planned for version: 2019.3

Available in: No available in

Affected versions: 2019.3 EAP1 (193.4778.12), 2019.3 EAP2 (193.4932.13)

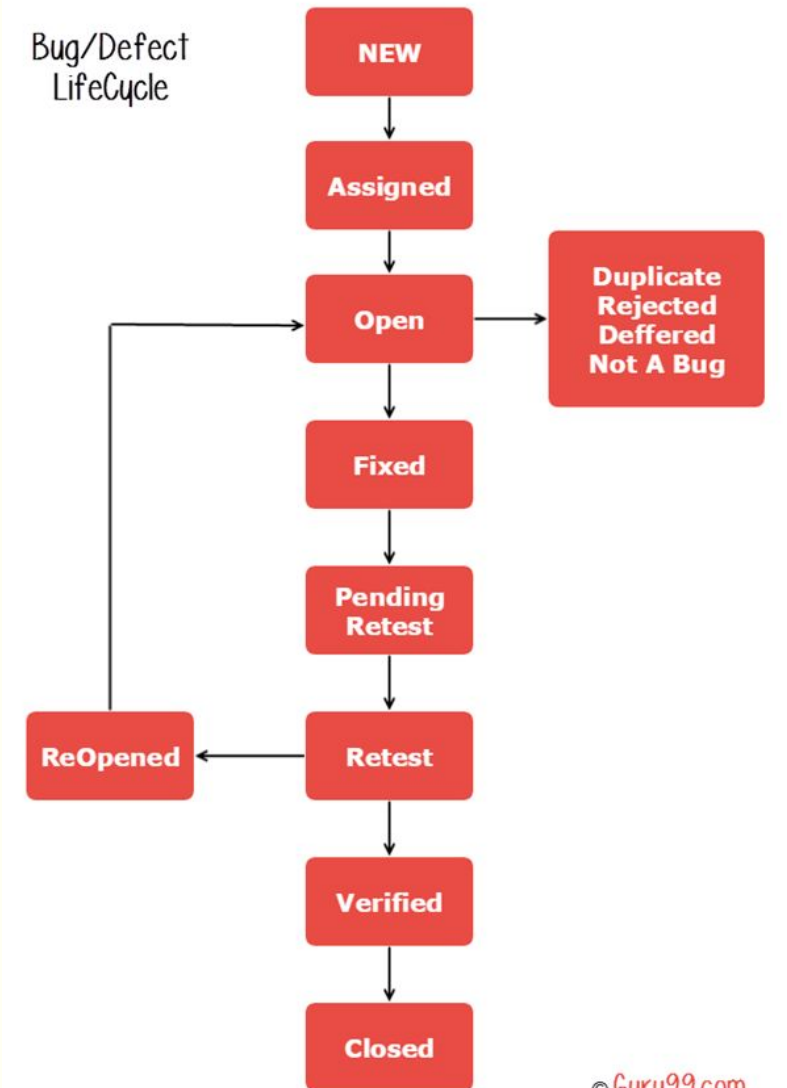
Watchers **2** > Watch issue

Boards > Add to board

III – Errores

Ciclo de Vida de los Errores

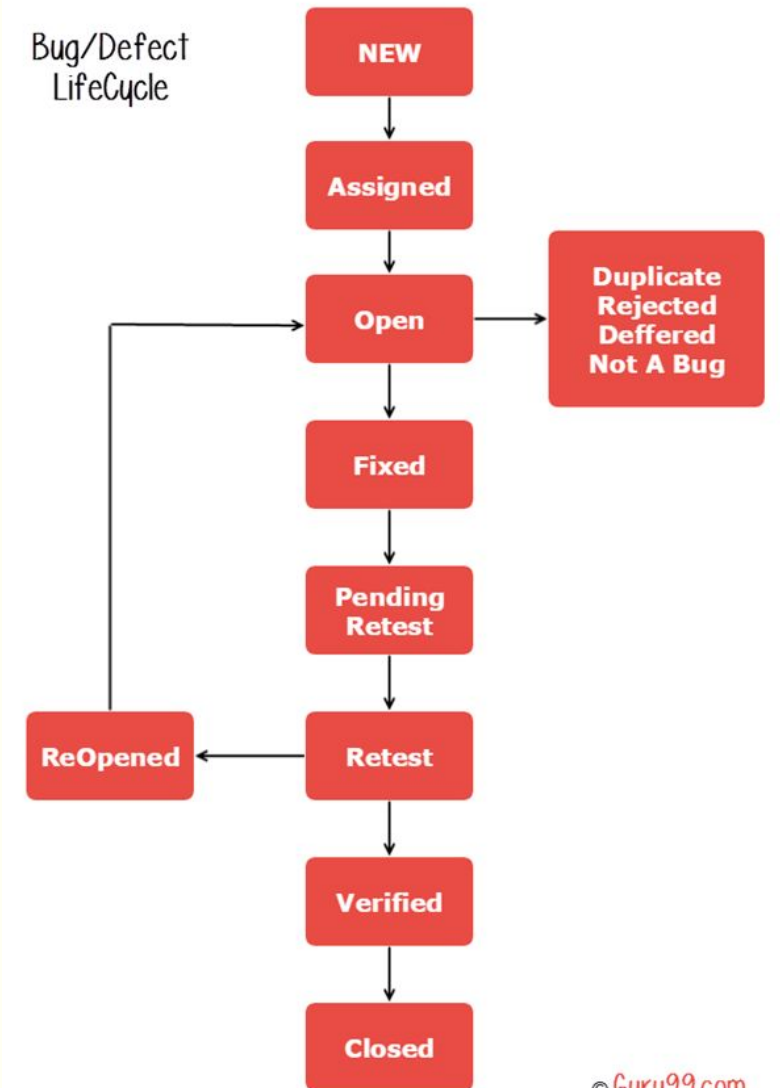
- **New:** Un nuevo defecto es registrado.
- **Assigned:** Se asigna al desarrollador.
- **Open:** El desarrollador analiza el defecto y su causa.
- **Fixed:** Se han hecho los cambios para su arreglo.
- **Pending retest:** Pendiente de confirmación.
- **Retest:** Prueba de confirmación



III – Errores

Ciclo de Vida de los Errores

- **Verified:** Arreglo confirmado y verificado.
- **Reopen:** El fallo persiste.
- **Closed:** El fallo ya no existe.
- **Duplicate:** El fallo ya había sido registrado.
- **Rejected:** El fallo no es aceptado o no se puede reproducir.
- **Deferred:** No prioritario, pasaría al backlog.
- **Not a bug:** No es un fallo, puede ser una mejora u otra cosa.



III – Errores

3.2 Categorización de Defectos

- Defectos bloqueantes:** Son aquellos defectos, detectados durante la ejecución de alguno de los servicios, que impiden continuar con su ejecución de forma total o parcial.
- Defectos no bloqueantes:** Son aquellos defectos detectados durante la ejecución de los servicios, pero que no impiden su ejecución. Dichos defectos serán asignados inicialmente al Gestor de Proyecto correspondiente, para que evalúe si realmente se trata de defectos como tal y por tanto requieren ser resueltos por el Equipo de Proyecto, o si por el contrario, no deben considerarse como defectos, y deben ser desestimados.

III – Errores

3.2 Categorización de Defectos

- Errores funcionales:** Son aquellos defectos donde el software no se comporta como se espera.
- Errores de comunicación/contenido:** Mensajes de ayuda/error no claros o no aparecen.
- Typos:** Errores de escritura o gramaticales.
- Errores de gestión de errores:** Los casos negativos no se gestionan correctamente.
- Errores visuales:** La aplicación no cumple con los diseños.
- Errores de usabilidad:** La aplicación no es usable.
- Errores de entorno:** La aplicación está fallando debido al entorno en el que está instalada.

III – Errores

3.3 Priorización de Defectos

Software defects by severity

Dependiendo del impacto técnico que tiene un defecto en el sistema, podemos clasificar:

- **Critical defects:** el sistema al completo está bloqueado o una funcionalidad de él. El testing no puede continuar. Un ejemplo es un error del servidor al intentar acceder a una aplicación.
- **High-severity defects:** afecta a funcionalidades principales y se comportan de forma muy diferente a lo que está establecido en los requisitos, por ejemplo un mail que no permite añadir más de una dirección de correo electrónico.
- **Medium-severity defects:** funciones menores no se comportan como se espera. Por ejemplo un link roto a la página de términos y condiciones.
- **Low-severity defects:** defectos visuales por ejemplo el color o el tamaño de un botón diferentes a lo establecido.

III – Errores

3.3 Priorización de Defectos

Software defects by priority

Dependiendo del impacto en negocio. La prioridad la decidirá el PO o algún miembro del equipo de Producto.

- **Urgent defects:** se tienen que arreglar en menos de 24 horas. Un defecto de cualquier severidad puede tener prioridad urgente.
- **High-priority defects** son los errores que deben arreglarse en la iteración actual, pueden llegar a ser bloqueantes para la entrega del equipo.
- **Medium-priority defects** son los errores que se pueden arreglar en futuras iteraciones.
- **Low-priority defects** suelen ser errores cosméticos que no impiden entregar el producto.

Fuentes

- <https://www.getzephyr.com/insights/agile-strategies-managing-bug-fixes>
- <http://www.juntadeandalucia.es/servicios/madeja/contenido/procedimiento/16>
- <https://www.scnsoft.com/blog/types-of-bugs>
- <https://www.guru99.com/defect-life-cycle.html>
- <https://academy.test.io/en/articles/2541938-types-of-bugs>
- <https://www.softwaretestinghelp.com/types-of-software-errors/>