

تفاوت میان Fault و Failure

استاد: خانم دکتر فرشته جدیدی

علیرضا سلطانی نشان

۲۶ شهریور ۱۴۰۳

۱ Defect یا نقص

زمانی رخ می‌دهد که بین خروجی واقعی و خروجی مورد انتظارمان تفاوت وجود داشته باشد [۲]. این نوع نقص‌ها را مهندسان آزمون نرم‌افزار مطلع می‌شوند و آن‌ها را به توسعه‌دهنده کد آن برنامه ارجاع می‌دهند که بایستی اصلاحات را انجام دهد. نقص‌ها انواع مهمی دارند:

- نقص‌هایی که بایستی به ترتیب اولویت بر طرف شوند مانند (High, Medium, Low).
- نقص‌هایی که باید از نظر اهمیت برطرف شوند (Critical, Major, Miner, Trivial).

۲ Error

Error زمانی رخ می‌دهد که یک اشتباهی در کد وجود داشته باشد [۲] و مانع انجام شدن آن در کامپایلر یا مفسر شود به گونه‌ای که برنامه نوشته شده با وجود آن خطا اجرا نخواهد شد. تفاوت اصلی Error با باگ‌ها این است که در نرم‌افزاری که حاوی باگ است می‌توان فرایندی را انجام داد و تنها آن قسمتی که باگ دارد نمی‌تواند وظیفه‌اش را کامل انجام دهد اما در نرم‌افزاری که با Error رو به رو شده است برنامه اصلاً در مرحله اجرا و خروجی وارد نمی‌شود [۱]. معمولاً می‌تواند توسط مهندسان نرم‌افزار و مهندسان واحد آزمون نرم‌افزار گزارش شود. می‌تواند انواع مختلفی داشته باشد:

- خطای نوشتاری (گرامری) در برنامه نوشته شده
- خطای منطقی
- خطاهای محاسباتی
- خطای هنگام کامپایل مانند استفاده از ماکروهایی که قبل از کامپایل انجام می‌شوند.

۳ Fault یا عیب

یک وضعیتی است که باعث می‌شود نرم‌افزار نتواند یک عملکرد ضروری را انجام دهند که انسان با انجام اشتباه باعث رخ دادن آن می‌شود [۲]. به بیانی بهتر، Fault یک رفتار ناخواسته یا نادرستی است که توسط برنامه‌های کاربردی آن را مشاهده خواهیم کرد [۱]. عیب‌ها باعث ایجاد اختلال در برنامه می‌شوند. اگر برطرف نشوند، ممکن است منجر به شکست در کار کد مستقر شده شوند. اگر اجزای مختلف کد نوشته شده به یکدیگر متصل باشند (به عنوان ماژولی از یکدیگر باشند) یک عیب ممکن است رفتار کل سیستم را تحت تاثیر خود قرار دهد [۱]. دلایل مختلفی می‌تواند وجود داشته باشد که باعث ایجاد عیب در سیستم شوند. یک عیب در یک برنامه می‌تواند زمانی رخ دهد که یک عمل نادرست در فرایندها یا تعریف داده‌ها صورت گرفته باشد یا می‌تواند به صورت ناسازگاری در برنامه‌ها شناخته شود. برای مثال در یک برنامه پایتون از کتابخانه‌ای جهت خواندن فایل‌های مخصوصی استفاده شده است. این برنامه شامل هیچ خطا و باگی نیست و به درستی عمل می‌کند. این برنامه برای فرمت‌های مشخص شده می‌تواند به درستی عمل کند ولی به محض آن که فایلی با فرمت مشخص نشده به آن وارد می‌شود ممکن است آن را اصلاً نپذیرد یا خروجی مورد نظر را ایجاد نکند و باعث خروجی از برنامه شود. یا اینکه برنامه‌ای نوشته شده است که به یک پروتکلی با نسخه‌ای مشخص متصل می‌شود اما وقتی آن برنامه را به همان پروتکل ولی نسخه‌ای قدیمی‌تر متصل می‌کنیم تغییراتی که بین نسخه‌ها وجود دارد باعث می‌شود برنامه به درستی وظیفه خود را انجام ندهد. عیب‌های مختلفی می‌تواند در یک محصول نرم‌افزاری در بخش‌های مختلف آن ظاهر شوند [۲]:

- عیب‌های مربوط به Business Logic

- عیب‌های عملکردی و منطقی
- عیب‌های نمایش (GUI)
- عیب‌های اجرایی
- عیب‌های امنیتی
- عیب‌های سخت‌افزاری/نرم‌افزاری

۴ Failure یا شکست

گاهی وقت‌ها، زمانی که برنامه در حال اجرا است سیستم ممکن است نتیجه غیرمنتظره‌ای را به عنوان خروجی تولید کند که می‌تواند به برنامه اجازه دهد که به شکست منجر شود. در موقعیت و محیط‌های مشخصی نقص‌ها می‌تواند دلیلی برای ایجاد شکست در برنامه باشد [۱]. یا به عبارتی دیگر، اگر برنامه‌ای تعداد زیادی از نقص‌ها را داشته باشد منجر به شکست می‌شود یا باعث شکست می‌شود. این شکست‌ها در تست‌های دستی معمولی توسط مهندسان در فرایند چرخه توسعه نرم‌افزار کشف می‌شوند [۲].

شکست در برنامه می‌تواند توسط خطاهای انسانی توسط کاربرانی که با نرم‌افزار تعامل دارند رخ دهد. برای مثال زمانی که کاربر ورودی اشتباهی به برنامه کاربردی وارد کند باعث شود که نرم‌افزار با شکست رو به رو شود. با این وجود می‌توان گفت که یک شکست می‌تواند عمداً یا سهواً توسط یک فرد در سیستم ایجاد شود [۱].

مراجع

- [1] GeekFlare. Difference between bug, defect, error, failure, and fault in software testing. <https://geekflare.com/difference-bug-defect-error-failure-fault-testing/>, 29 Nov 2022.
- [2] Javapoint.com. Difference between bug, defect, error, fault & failure. <https://www.javatpoint.com/bug-vs-defect-vs-error-vs-fault-vs-failure>.