

سوال: تفاوت بین Failure و Fault را در حوزه آزمون نرم افزار بیان کنید.

پاسخ: در حوزه آزمون نرم افزار، Fault و Failure دو مفهوم کلیدی هستند که با وجود ارتباط نزدیکی که با یکدیگر دارند، تفاوت های مهمی دارند.

• تعریف Fault (خطا یا نقص)

Fault که به آن Bug یا Defect نیز گفته می شود، به هرگونه اشتباه یا خطایی گفته می شود که در کد منبع نرم افزار وجود دارد. این اشتباه می تواند در نتیجه اشتباهات برنامه نویسی، طراحی، یا حتی تحلیل نیازمندی ها به وجود آمده باشد. به بیان ساده، Fault به آن بخشی از نرم افزار اشاره دارد که از حالت صحیح انحراف داشته و ممکن است باعث بروز مشکلات در عملکرد نرم افزار شود. مثال: فرض کنید که در یک برنامه بانکی، یک خطا در کدنویسی باعث شود که محاسبات مربوط به سود بانکی به درستی انجام نشود. این خطا (Fault) ممکن است بلافاصله به عنوان یک Failure قابل مشاهده نباشد اما وجود دارد و ممکن است در آینده مشکل ایجاد کند.

• تعریف Failure (شکست)

Failure به شرایطی گفته می شود که نرم افزار نمی تواند عملکرد صحیح خود را انجام دهد و خروجی نادرست یا رفتاری غیرمنتظره از خود نشان می دهد. زمانی که یک Fault در زمان اجرای نرم افزار باعث شود که نرم افزار عملکرد خود را به درستی انجام ندهد، این حالت را Failure می نامیم. بنابراین، Failure نتیجه نهایی وجود یک یا چند Fault در نرم افزار است که به صورت مشکل در کارکرد سیستم ظاهر می شود.

مثال: همان خطای بانکی که در بخش قبلی مطرح شد، اگر هنگام محاسبه سود به کاربر نتیجه ی نادرست نمایش دهد، این رفتار نادرست نرم افزار را Failure می نامیم. بنابراین، وقتی کاربر متوجه خروجی اشتباه می شود، در واقع Failure رخ داده است.

• ارتباط بین Fault و Failure

ارتباط بین این دو مفهوم به این شکل است که هر Fault لزوماً منجر به Failure نمی شود. برخی از Fault ها ممکن است هرگز به کاربر نشان داده نشوند یا در شرایط خاصی رخ دهند که تشخیص داده نشوند. همچنین، ممکن است Fault هایی وجود داشته باشند که تنها تحت شرایط خاصی (مانند یک ورودی غیرمنتظره یا یک تعامل نادر با سیستم) به یک Failure تبدیل شوند. این تفاوت می تواند به دلیل پیچیدگی سیستم های نرم افزاری و حالت های مختلفی که سیستم ممکن است در آن اجرا شود باشد.

• مفهوم Fault Latent (نقص نهفته)

گاهی اوقات، Fault ها در نرم افزار پنهان می مانند و بلافاصله باعث Failure نمی شوند. این نوع از Fault ها را Fault Latent می نامند. این نقص ها ممکن است تنها تحت شرایط خاصی آشکار شوند که ممکن است در مراحل ابتدایی تست یا حتی تا مدت ها پس از استفاده از نرم افزار نیز قابل شناسایی نباشند.

• مفهوم Error (خطا)

برای درک بهتر این موضوع، باید به مفهوم دیگری به نام Error نیز اشاره کنیم. Error معمولاً به معنای اشتباهی است که توسط توسعه دهنده یا تحلیل گر در طول فرآیند توسعه ایجاد شده و منجر به ایجاد یک Fault می شود. بنابراین، Error علت است و Fault نتیجه.

• مثالی برای درک تفاوت Failure و Fault

فرض کنید شما در حال استفاده از یک نرم افزار حسابداری هستید. اگر کد برنامه به دلیل یک اشتباه برنامه نویسی طوری نوشته شده باشد که به جای جمع دو عدد، آن ها را ضرب کند، این یک Fault است. اما اگر این قسمت از کد هرگز توسط کاربر استفاده نشود یا ورودی هایی که منجر به بروز خطا می شوند هرگز ارائه نشوند، هیچ Failure ای رخ نمی دهد. در عوض، اگر کاربر ورودی های مربوطه را وارد کند و نرم افزار خروجی نادرست تولید کند، یک Failure اتفاق افتاده است.

• پیشگیری و شناسایی

تیم های توسعه ی نرم افزار باید به طور مداوم به دنبال یافتن و اصلاح Fault ها باشند تا از وقوع Failure ها جلوگیری کنند. آزمون های نرم افزاری به این منظور طراحی می شوند که Fault ها را شناسایی کنند تا پیش از آنکه به یک Failure منجر شوند، برطرف شوند. با این حال، در پروژه های بزرگ و پیچیده، امکان شناسایی تمامی Fault ها بسیار سخت است.