Statikus tesztelés

### Micsoda

A szoftver statikus tesztelése során az ellenőrök a forráskód és a dokumentáció elemzésével próbálják azonosítani a potenciális hibákat és problémákat.

A dokumentációs hibák kiküszöbölése és az értékes dokumentáció létrehozása rendkívül fontos a szoftvertervezés és fejlesztés folyamatában, mivel segít csökkenteni a hibák számát, növeli a fejlesztői produktivitást, és javítja a szoftver használhatóságát és megbízhatóságát.

### Miket talál meg

**Ami project feladat szempontjából érdekes:**

Dokumentáció hiánya vagy hiányos volta: A leggyakoribb hiba az, ha a szoftvernek nincs vagy csak részleges dokumentációja. A hiányzó információk megnehezíthetik a fejlesztők és tesztelők munkáját.

Elavult dokumentáció: Amikor a dokumentáció nem követi a szoftver aktuális verzióját, az zavarhatja a fejlesztőket és vezethet olyan hibákhoz, amelyeket az elavult információk miatt követtek el.

Ellentmondásos információk: Ha a dokumentáció különböző részei ellentmondanak egymásnak, a fejlesztők könnyen összezavarodhatnak, és ez hibákhoz vezethet.

Rosszul formázott vagy nehezen olvasható dokumentáció: Ha a dokumentáció nehezen érthető vagy rosszul formázott, az megnehezítheti az olvasók számára az információk megtalálását és értelmezését.

Nem megfelelő felhasználói dokumentáció: Ha a felhasználók számára szánt dokumentáció nem világos vagy hiányos, az zavarhatja őket, és emiatt nem tudják megfelelően használni a szoftvert.

**Amikre még használható:**

Hiányzó vagy hibás példák: A jó dokumentáció példákat is tartalmaz, amelyek segíthetnek a fejlesztőknek a megfelelő módon használni a szoftvert. Hiányzó vagy helytelen példák hibákhoz vezethetnek.

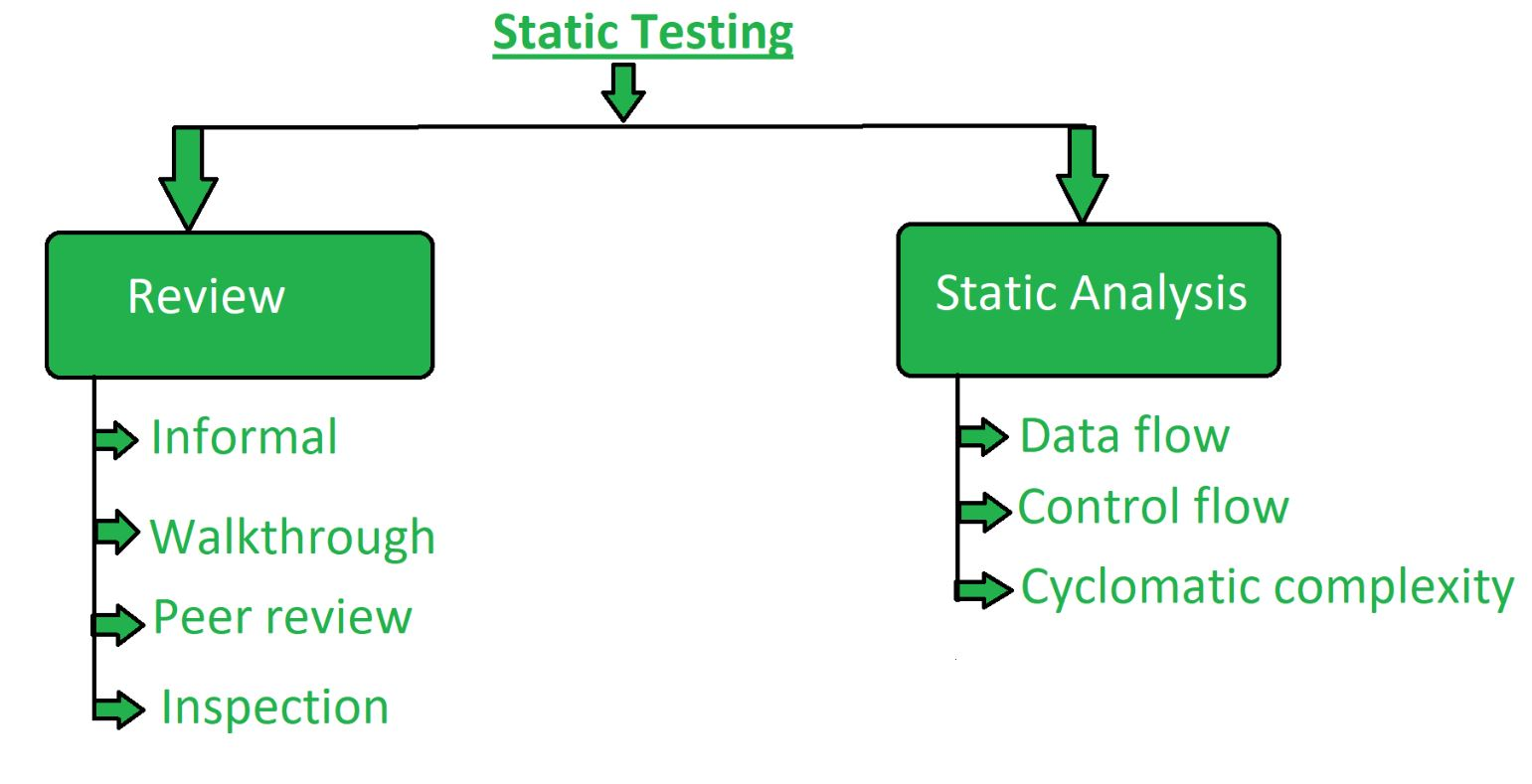
Nem frissült hiba- vagy probléma-jegyzék: Ha a dokumentációban nem szerepelnek a szoftverrel kapcsolatos ismert hibák vagy problémák, a fejlesztőknek nehéz lehet ezeket elkerülni vagy megoldani.

Hiányzó vagy helytelen API leírások: Ha a szoftverhez tartozó API-k nem megfelelően vannak dokumentálva, a fejlesztőknek nehéz lehet helyesen használni őket, ami hibákhoz és kompatibilitási problémákhoz vezethet.

Nem dokumentált függőségek: Ha a szoftver függ más komponensektől vagy könyvtáraktól, de ezek nincsenek dokumentálva, akkor a fejlesztők nem tudják pontosan megérteni a szoftver működését és annak követelményeit.

Hiányzó biztonsági irányelvek: Amennyiben a dokumentáció nem tartalmazza a szoftver biztonsági szempontjaira vonatkozó irányelveket és tanácsokat, az növelheti a biztonsági hibák kockázatát.

### Hogy néz ki a statikus tesztelés?



A statikus tesztelés egy olyan szoftverteresztési módszer, amelyet a szoftver kódjának végrehajtása nélkül végeznek. Ezzel ellentétben a dinamikus tesztelés során a kódot végrehajtják a hibák felderítése érdekében. A statikus tesztelést a fejlesztés korai szakaszában végzik, hogy elkerüljék a hibákat, mivel könnyebb megtalálni a hibákat és hibák forrásait, és könnyen javíthatók. Azok a hibák, amelyeket dinamikus teszteléssel nem lehet felfedezni, gyakran könnyen megtalálhatók a statikus teszteléssel. Statikus tesztelési technikák:

**Review**: A statikus tesztelés során a felülvizsgálat egy olyan folyamat vagy technika, amelyet a szoftver tervezetének potenciális hibáinak felderítésére végeznek. Ez egy olyan folyamat, amely segít hibákat felfedezni és eltávolítani a különböző támogató dokumentumokból, mint például a szoftver követelmények specifikációja. Az emberek átnézik a dokumentumokat, és kijavítják a hibákat, redundanciákat és homályos pontokat. A felülvizsgálatot négy típusba soroljuk:

* **Informális review**: Az informális felülvizsgálat során a dokumentumok készítője az információkat az emberek elé tárja, és mindenki kifejti a véleményét, így a hibákat a korai szakaszban azonosítják.
* **Walkthrough**: Általában tapasztalt vagy szakértő személy végzi el a hibák ellenőrzésére, hogy később ne legyenek problémák a fejlesztés vagy tesztelés során.
* **Peer review**: A peer review azt jelenti, hogy egymás dokumentumait ellenőrzik a hibák felderítésére és kijavítására. Alapvetően csapaton belül végzik.
* **Inspection**: Az ellenőrzés alapvetően a dokumentumok ellenőrzése a magasabb hatóság által, például a szoftverkövetelmények specifikációjának (SRS) ellenőrzése.

**Statikus elemzés**: A statikus elemzés magában foglalja a fejlesztők által írt kód minőségének értékelését. Különböző eszközöket használnak a kód elemzéséhez és annak összehasonlításához a szabvánnyal. Segít az alábbi hibák azonosításában:

1. Használatlan változók
2. Halott kód
3. Változó meghatározatlan értéke
4. Hibás szintaxis

A statikus elemzésnek három típusa van:

* Adatáram (data flow)
* Vezérlés (control flow): Alapvetően azt mutatja be, hogyan hajtódnak végre a statementek vagy utasítások.
* Ciklometrikus komplexitás: A ciklometrikus komplexitás meghatározza a független utak számát a kód vagy annak vezérlését leíró gráfjából, hogy az egyes független utakhoz minimális számú teszteset legyen tervezhető.