# Verifikáció vs. validáció

Mindkét tevékenység fontos eleme a minőségirányítási rendszernek. Egymástól független eljárások, melyeket együtt kell alkalmazni annak érdekében, hogy a rendszer követelményeknek való megfelelését igazolni tudjuk és növeljük a termék irányában a bizalmat. A szoftver fejlesztése során végzett ún. V&V tevékenységek segítenek az élesítés előtt előkerülő hibák kiküszöbölésében.

|  |  |  |
| --- | --- | --- |
|  | ISTQB | ÁLTALÁNOSSÁGBAN |
| VERIFIKÁCIÓ | Az adott követelmények teljesülésének vizsgálata és konfirmálása. | **Jól gyártunk**, fejlesztünk egy terméket? Jó minőségű a termék? ***🡪 A gyártó véleménye a minőségről.*** |
| VALIDÁCIÓ | Annak vizsgálata és konfirmálása, hogy a szoftver tervezett felhasználási céljának megfelelő követelmények teljesülnek-e. | **Jó terméket** gyártunk, fejlesztünk? Megfelelő a termék? ***🡪 A felhasználó véleménye a minőségről.*** |

## **Verifikáció:**

* Alapját a specifikáció jelenti, objektív szempontok alapján ellenőrizzük, hogy a termék megfelel-e a specifikációnak.
* **Statikus tesztelés**i feladatok elvégzéséből áll, többek között dokumentumok, tervek és a kód áttekintéséből.
* Több módszert is alkalmazhatunk erre, ide tartoznak a walkthrough-k, a review-k (formális és informális) és az inspekciók (inspection).
* Már ebben a korai fázisban kiküszöbölhetünk hibákat, vagy javíthatjuk pl. a specifikációt, ha nem egyértelmű, vagy a kódot, ha szintaktikai vagy logikai hibát találunk.

## **Validáció:**

* Alapját a követelmények jelentik, itt már lehetnek szubjektív szempontok is.
* Azt igazoljuk, hogy a termék a felhasználó elképzeléseinek megfelelően működik, alkalmas a kívánt feladat végrehajtására. A kimenetre összpontosítunk, pl. a használhatóságra, a felhasználói felület kialakítására, a teljesítményre, kompatibilitásra.
* **Dinamikus tesztelés**i feladatok elvégzését jelenti, manuális vagy automatizált funkcionális (pl. unit tesztek, integrációs tesztek, rendszertesztek, UAT) és nemfunkcionális tesztek formájában.
* A verifikáció mindig megelőzi a validációt, de azokat a hibákat, amiket a verifikáció során nem találtunk meg, a validáció során még azonosíthatjuk és kijavíthatjuk, ezzel is csökkentve a javítási költségeket és erőforrásigényt.

## Hétköznapi példa:

Étel rendelése egy étteremben. Amikor kihozzák a rendelést, mielőtt megkóstolnánk, verifikáljuk, hogy úgy néz-e ki, ahogy kell. Amikor megkóstoljuk, akkor tudjuk validálni, hogy valóban ezt rendeltük, olyan az íze, az állaga, amilyet szerettünk volna.

## Szoftveres példa:

Egyszerű belépési felület. A verifikációs fázisban ellenőrizzük, hogy megfelel-e a kinézete a terveknek, vannak-e helyesírási hibák, minden gomb ott van-e, ahol a specifikáció szerint lennie kell. A validáció során azt vizsgáljuk, hogy a gombok a követelményeknek megfelelően működnek és a kívánt feladat végrehajtására alkalmasak (a belépés gombra kattintva belépünk).

Források:

<https://dcs.uni-pannon.hu/files/docs/users/csertangyorgy/swtech/22_Verifikacio_validacio.pdf>

<https://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/doktorandusz-t%C3%A1rgyak/szoftver-verifik%C3%A1ci%C3%B3-%C3%A9s-valid%C3%A1ci%C3%B3/11/SZVV_EA01_intro.pdf>

<https://vik.wiki/images/5/55/Swsv_zvTetelKidolgozas_2020jan.pdf>

<https://www.lambdatest.com/blog/verification-vs-validation/>

<https://www.softwaretestinghelp.com/what-is-verification-and-validation/>

+1 olvasnivaló – Szoftvertesztelés összefoglaló jegyzet:

<https://aries.ektf.hu/~gkusper/SzoftverTeszteles.pdf>