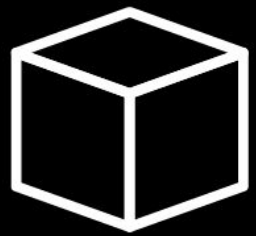
attrecto
NEXT TECH DIGITAL SOLUTIONS

Horváth Veronika & Csizmadia János

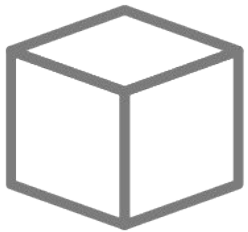
Black-Box / White-Box Tesztelés

BLACK BOX



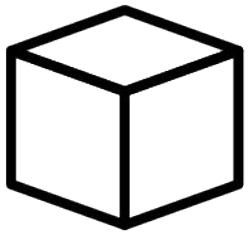
ZERO KNOWLEDGE

GRAY BOX



SOME KNOWLEDGE

WHITE BOX



FULL KNOWLEDGE

Black-Box Tesztelés
(Specifikáció alapú tesztelés)

Alapja

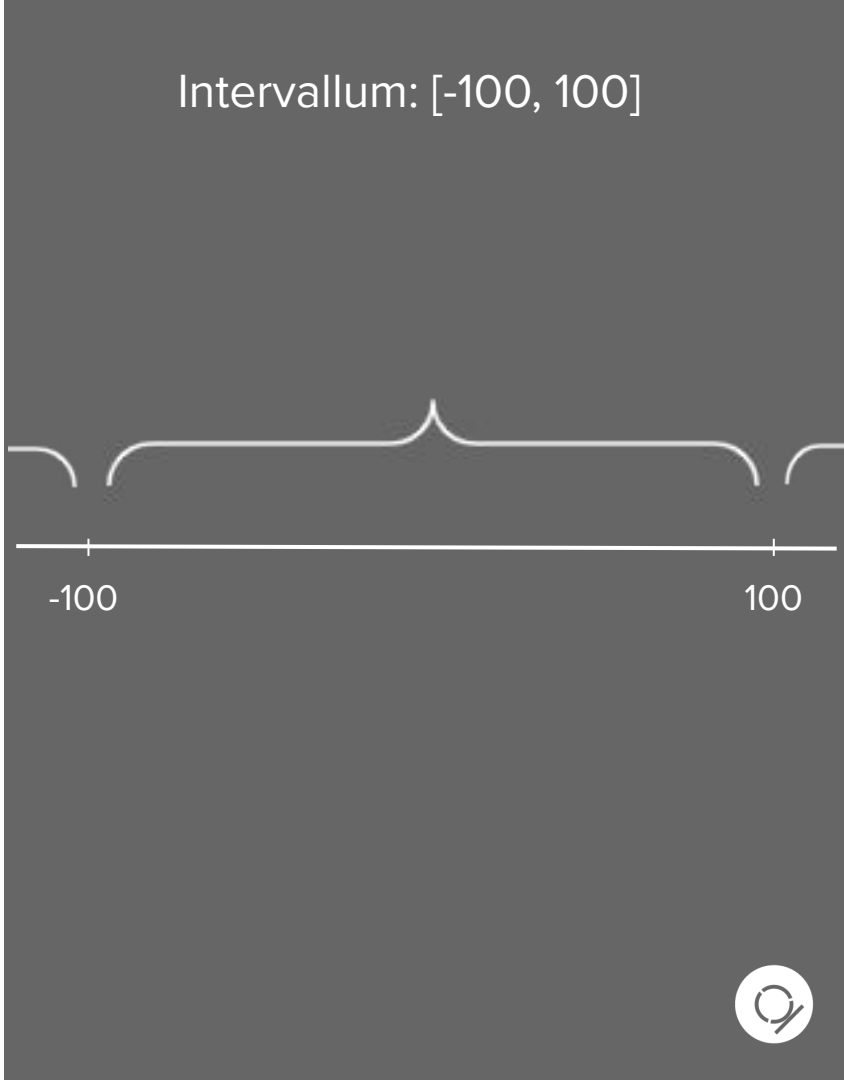
- Specifikáció alapján készülnek el a tesztesetek, nem látjuk a mögötte lévő kódot
- Adott bemenetre előre meghatározott kimeneteket határozunk meg
- A program futásakor a kimenetet össze hasonlítjuk az elvárt kimenettel

Black-Box tesztelési típusok

1. Ekvivalencia Particionálás
2. Határérték Analízis
3. Döntési Tábla
4. Állapot átmenet
5. Use Case

Ekvivalencia Particionálás

- Lecsökkenti a szükséges Tesztesetek számát
- Nagyobb csoportokba szervezzük az azonos inputtal vagy outputtal rendelkező teszteseteket Input Partíciók / Output partíciók
- Meghatározunk Valid Partíciókat és Nem Valid Partíciókat
  - Valid Equivalence Partitions
  - Non-valid Equivalence Partitions
- Minden partícióból csak 1-et választunk



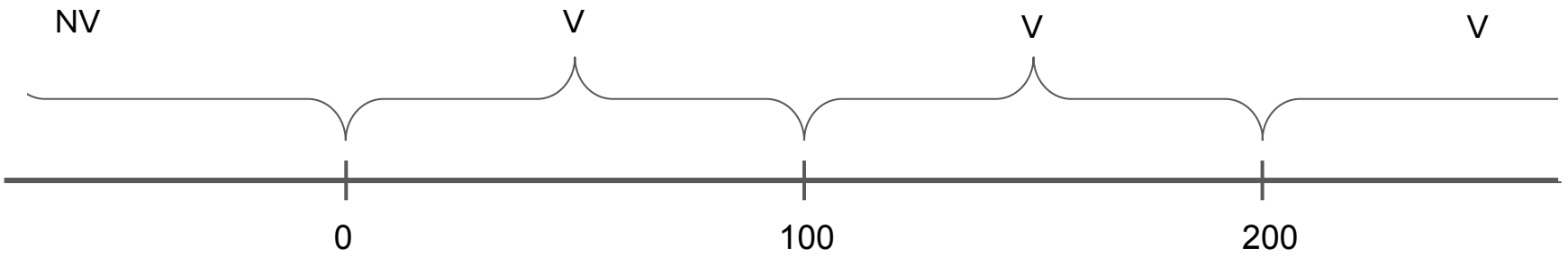
Ekvivalencia Particionálás - Feladat

Van egy banki program ami különböző kamatozásokat nyújt nekünk a számlán lévő összeg után a következőképpen:

- 0.5% az első 100.000 Ft-ra.
- 1% a következő 100.000 Ft-ra.
- 1.5% az e feletti összegekre

1. A számla legkisebb összege 0.
2. Csak egész számok halmazán gondolkodjunk

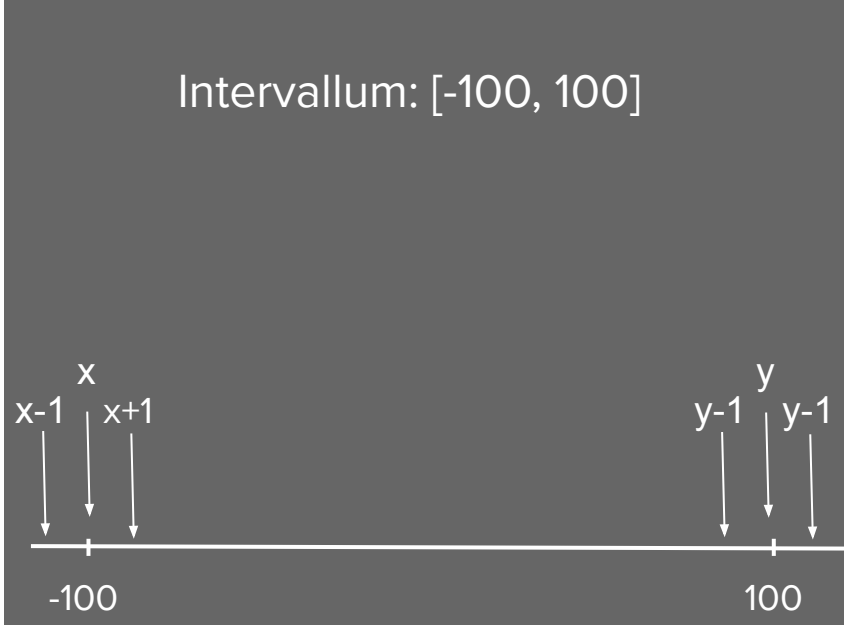
Ekvivalencia Particionálás - Feladat



1.]-∞, 0 [
2. [0, 100]
3.]100, 200]
4. [201, ∞ [

Határérték Analízis

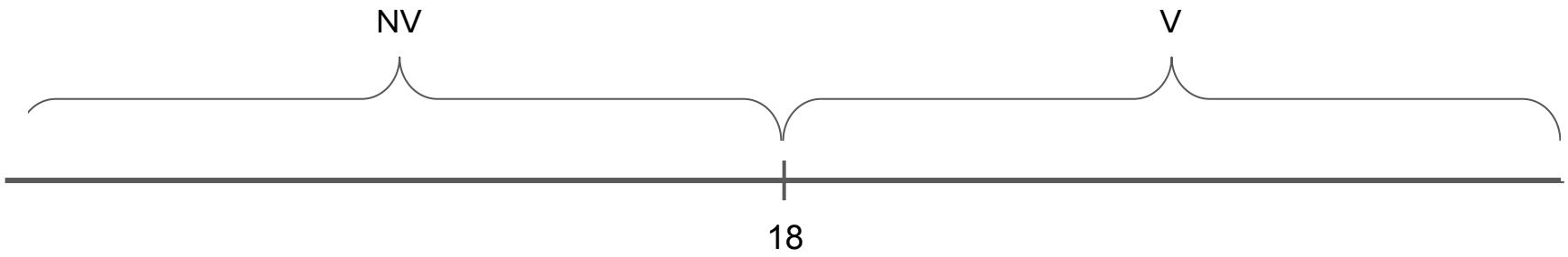
- A meghatározott partíciók határait vizsgálja
- A maximum és a minimum értékei egy partíciónak a határai.
- A határértéke egy valid partíciónak a valid határérték
- A határértéke egy nem valid partíciónak a nem valid határérték
- Gyakran mondják az Ekvivalencia particionálás kibővítésének



Határérték Analízis - Feladat

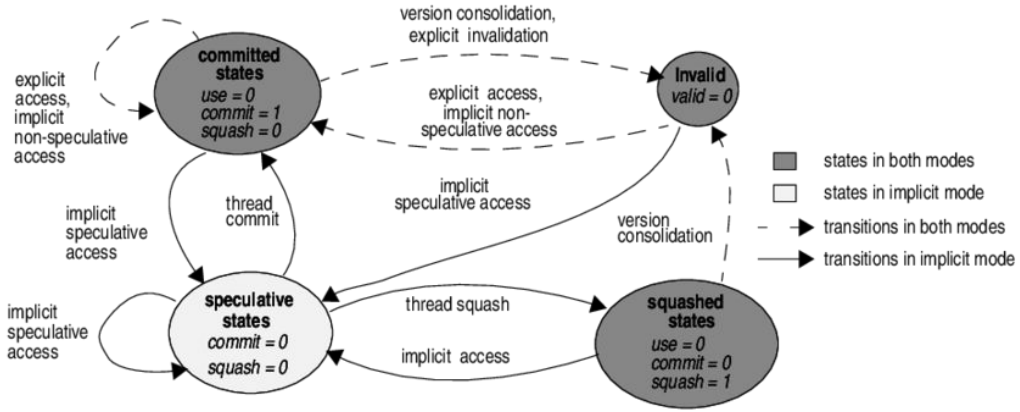
Egy alkalmazást csak 18 év felettiek használhatnak. A regisztrációkor meg kell adni a kort, ahol azonnal le is vizsgáljuk, hogy regisztrálhat-e vagy sem.

Határérték Analízis - Feladat



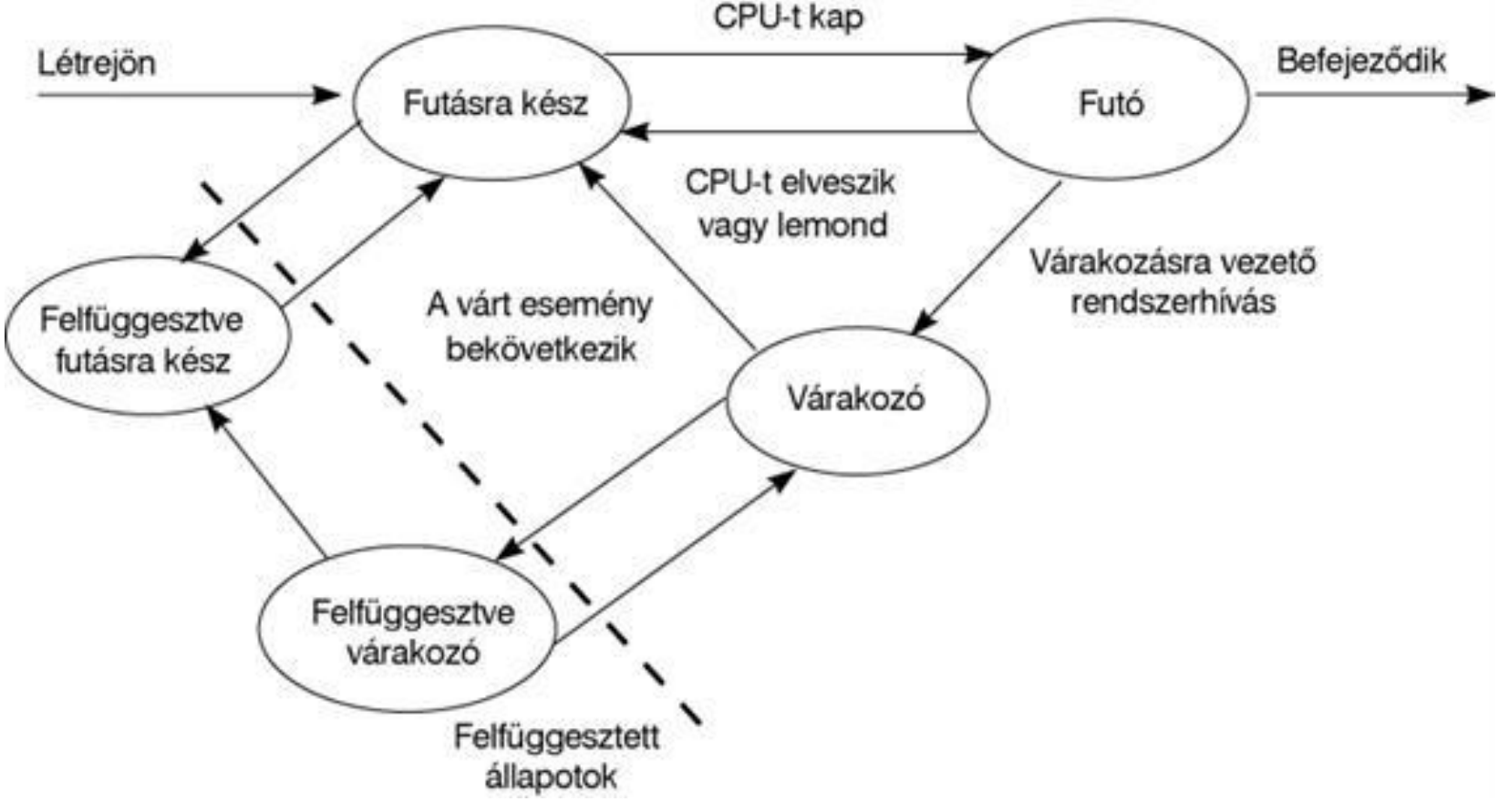
1. $]-\infty, 17]$
2. $[18, \infty[$

Állapot Átmenet

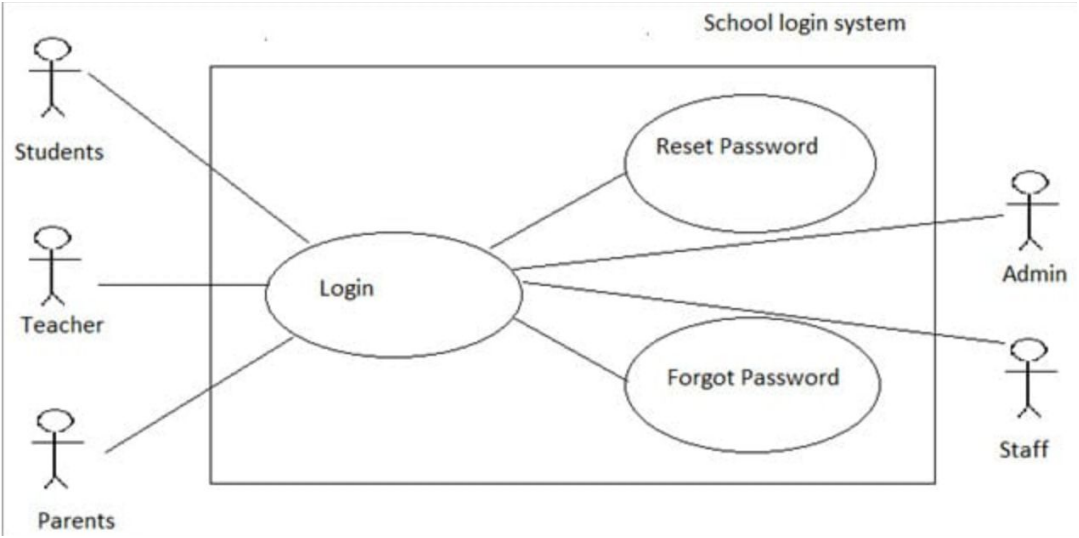


- Hasznos, ha bizonyos események az input változásával vannak triggerelve
- Ennek a tesztnek az alapja, az Állapot Átmenet Diagram
- Az egyes körök az állapotok, a köztük lévő vonalak az egyes átmeneteket az állapotok közt

Állapot Átmenet Diagram



Use Case



- A felhasználók és a rendszer high-level szintű közti kapcsolatot mutatja
- Célja a hibák felderítése a folyamatokban
- Minden Use Case-nek van:
  - Preconditions: Amelyeknek meg kell felelnie a Use Case elkezdéséhez
  - Post-Conditions: Use Case forgatókönyv sikeres lefutását követően az eredménye a rendszernek vagy a végső állapota.

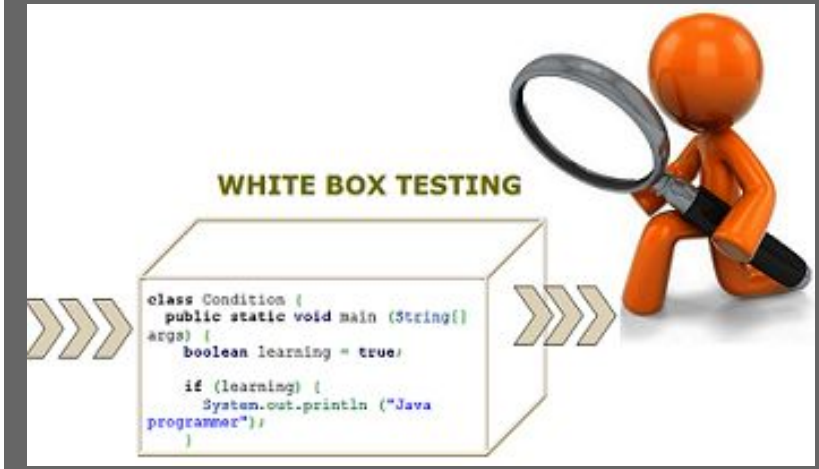
White-Box Tesztelés

(Struktúra alapú tesztelés)

Háttere, alapja

- Alapja egy meghatározott struktúrája a tesztelt szoftvernek vagy rendszernek

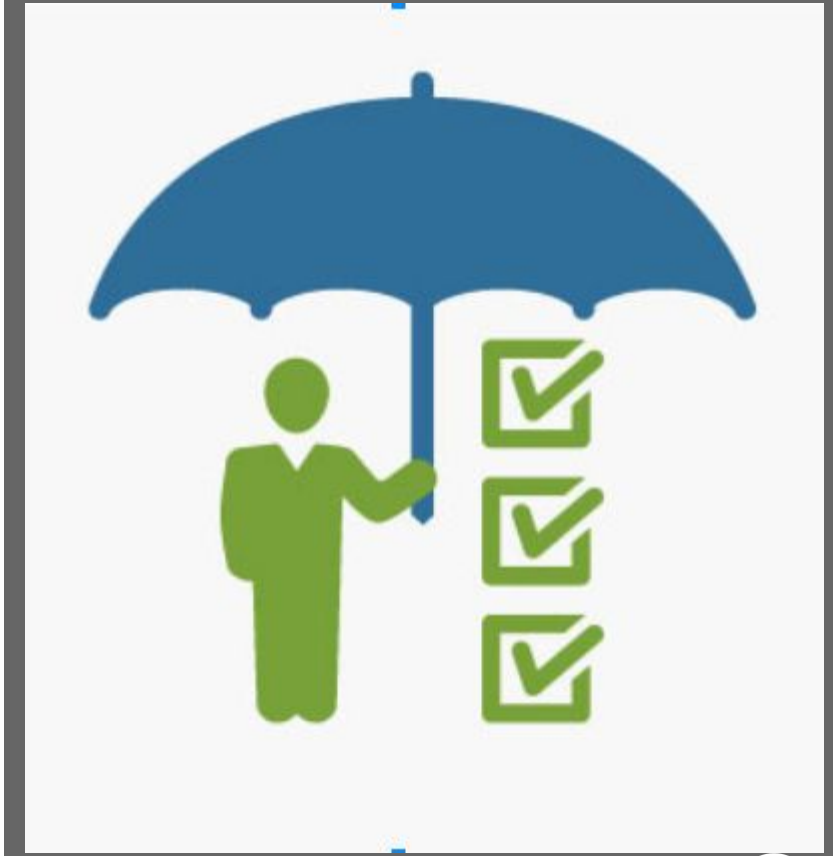
- **Komponens szinten:** A szoftver komponens struktúráján alapul (utasítások, döntési pontok, ágak, utak)
- **Integrációs szinten:** A struktúra lehet például egy hívási fa, diagram, ahol a modulok más modulokat hívnak
- **Rendszer szinten:** A struktúra egy teljes menürendszer, üzleti folyamat, vagy weboldal



Lefedettség

- A lefedettség értelmezése a struktúra szintjével változik

- **Integrációs szinten:** a modulok és interfészek hány százaléka lett letesztelve
- **Funkcionális szinten:** hány százaléka lett a menü struktúrának letesztelve
- **Rendszer szintű:** Hány százaléka lett letesztelve az összes lehetséges útnak, egy üzleti folyamatra nézve



Kódlefedettség

A struktúrális tesztelés magába foglalja hogy a teszteseteinket magából a kódból (pszeudó kód) hozzuk létre.

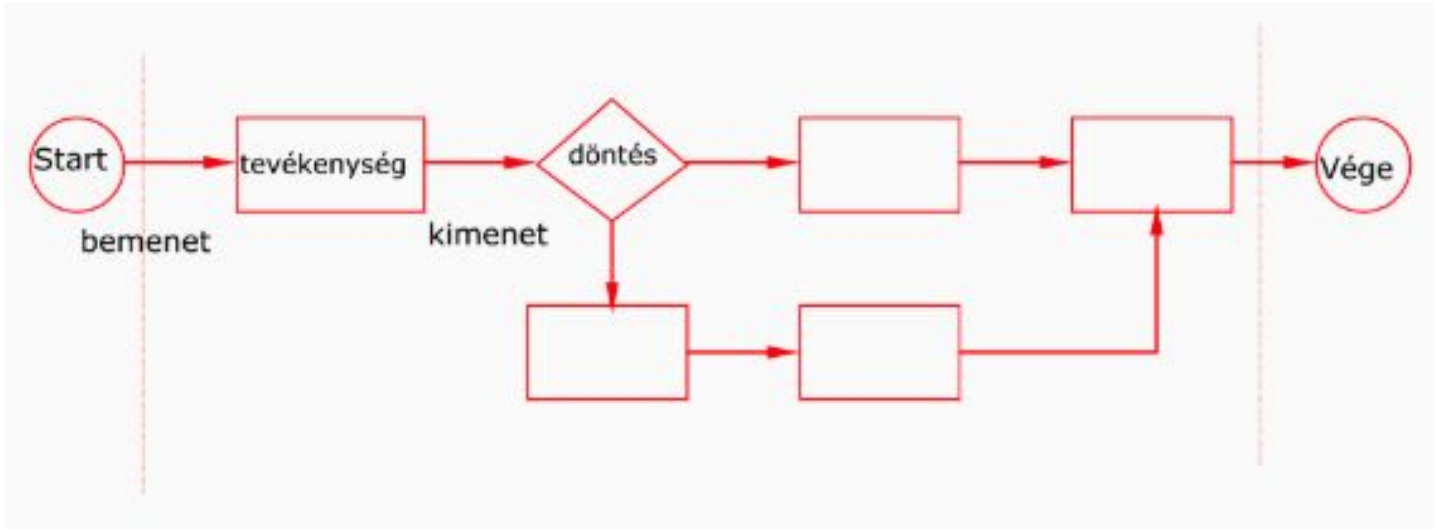
A kódlefedettséghez 2 főbb struktúrális (whitebox) technikát használhatunk.

- **Utasítások alapján:** Utasítás tesztelés (és lefedettség)
- **Döntések alapján:** Döntés tesztelés (és lefedettség)

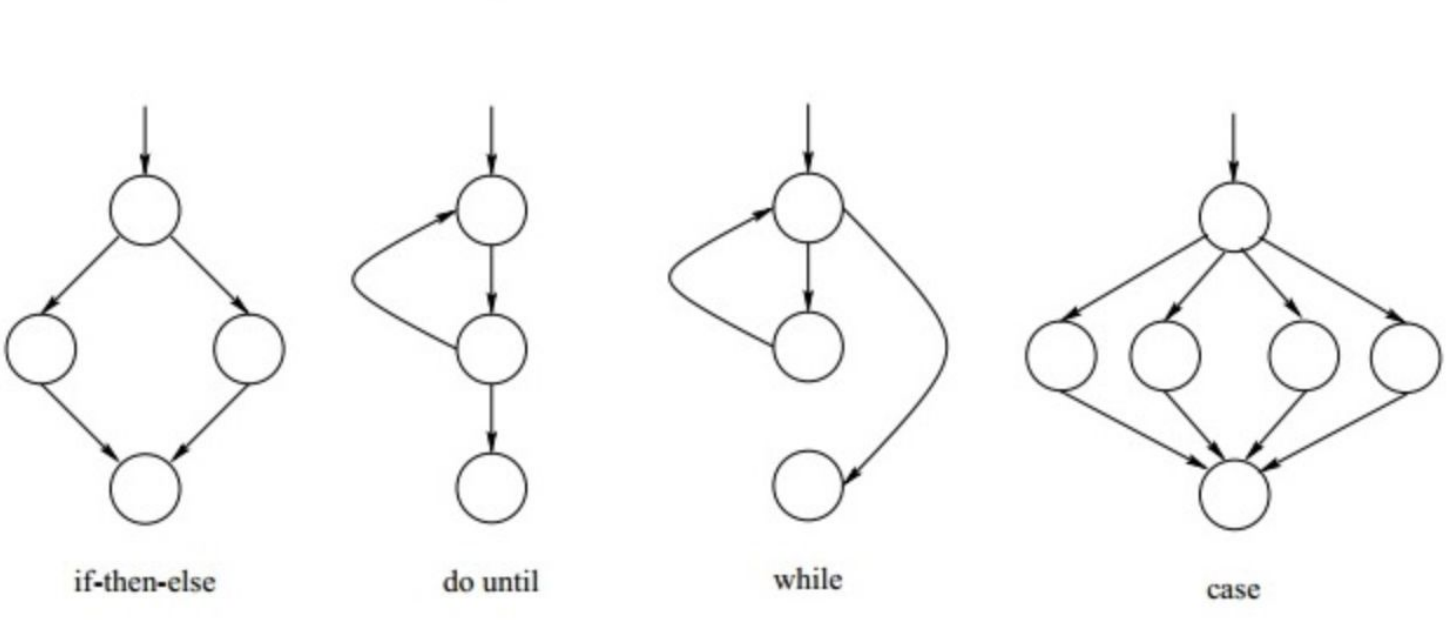
Pszeudo kód

```
let a = 5
let b = 10
let c = 7
var MAX = 0

if a > b {
    if a > c {
        MAX = a
    } else {
        MAX = c
    }
}
else if b > c {
    MAX = b
} else {
    MAX = c
}

print(MAX)

```

Folyamat gráf (flow chart)

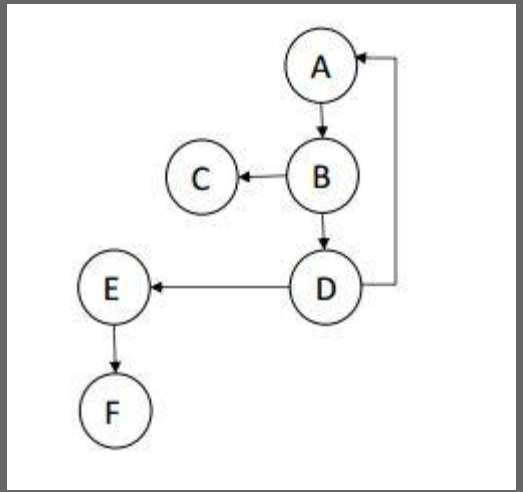


Vezérlési folyamat ábra (control flow graph)



Utasítás tesztelés és lefedettség

- Célja: hogy minden utasítás legalább egyszer végre legyen hajtva
- Csak a futtatható utasítások számítanak
- Lefedettség = tesztelt utasítás / összes utasítás * 100%
- 100% utasítás lefedettség nem jelenti hogy mindent leteszteltünk (hiányzó else utasítás), de egy jó viszonyítási alapot ad.
- Alapvetően “gyenge” metrika ahhoz hogy megfelelő eredményeket produkáljon és csak ezt az egy módszert alkalmazzuk

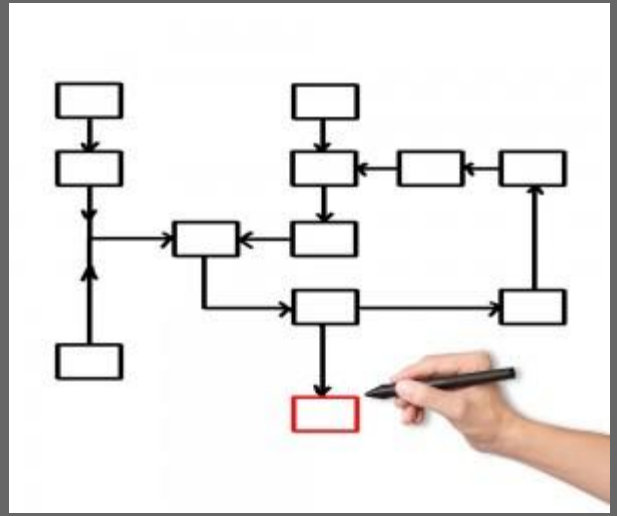


Folyamatábra

- Leírja milyen sorrendben kell egy adott tevékenységet (feladatot elvégezni)
- Segíti az üzleti folyamatok megértését és elemzését

Folyamatábra készítésének lépései

- Bemenetek / kimenetek meghatározása
- Lépések meghatározása
- Lépések összekapcsolása
- Diagram helyességének ellenőrzése

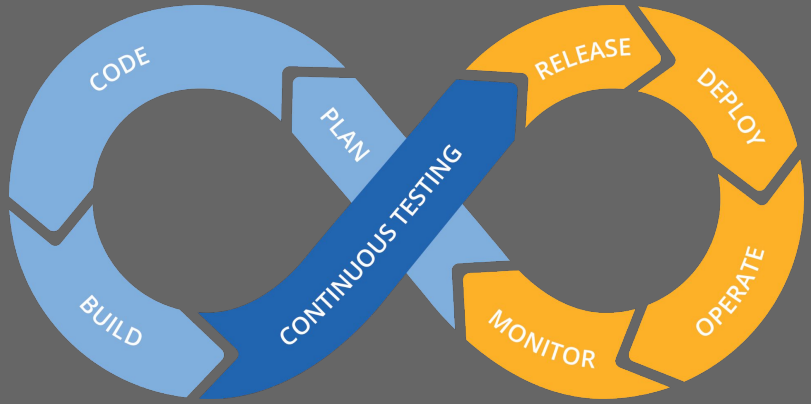


Folyamatábra a tesztelésben

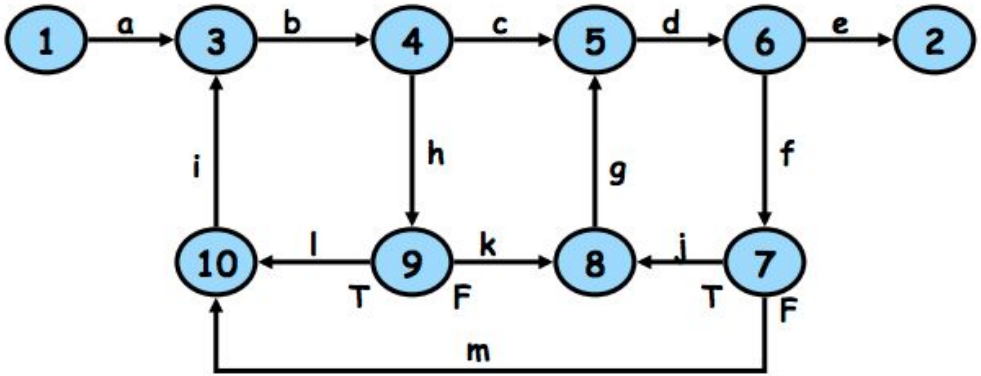
- Gráf előállítása a forráskódból (manuálisan vagy cél sw segítségével)
- Lefedettség (cél) meghatározása
- Tesztesetek meghatározása a gráf alapján
- Tesztesetek végrehajtása
- Eredmények kiértékelése és konklúziók levonása (lezárás, javítás..)

Mikor “kész” a tesztelés?

- Minden út a bemenettől a kimenetig legalább egyszer tesztelve lett
- Minden “statement” legalább egyszer tesztelve lett
- Minden döntési ág legalább egyszer tesztelve lett



Control flow graph



PATHS	DECISIONS				PROCESS LINKS											
	4	6	7	9	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*	*l*	*n*
abcde	*T*	*T*			*	*	*	*	*								
abhkgde	*F*	*T*		*F*	*	*				*	*	*					
abhlibcde	*TF*	*T*		*T*	*	*	*	*	*				*	*		*	
abcdfjgde	*T*	*TF*	*T*		*	*	*	*	*	*	*	*			*		
abcdfmibcde	*T*	*TF*	*F*		*	*	*	*	*	*				*			*

Ciklomatikus komplexitás

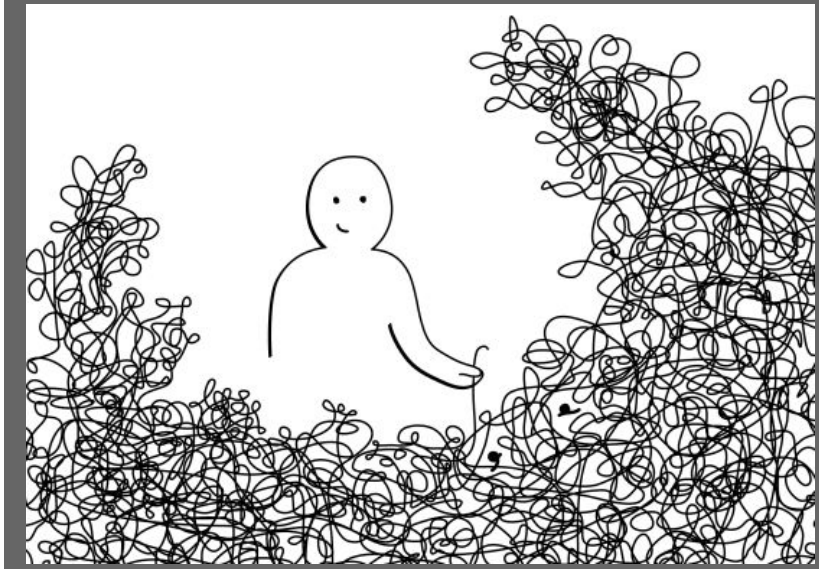
M = E - N + 2P

E = edges/link (1 élet jelöl)
N = nodes (statement - utasítás)
P = components

VAGY

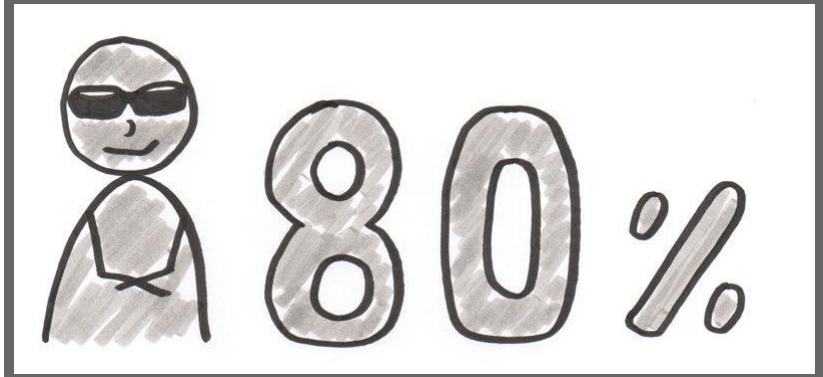
M = D + 1

D = Decision node (döntési pont, melyből 1-nél több nyíl indul ki)



Döntési tesztelés és lefedettség

- Célja hogy a program döntési ágait tesztelje (futassa)
- Lefedettség = (letesztelt döntési kimenetelek száma/ összes lehetséges döntési kimenetel) x 100%
- 100% döntési lefedettség = 100% utasítás lefedettség (fordítva nem igaz - else)



Döntési tábla

Feltételek		**Szabályok**								
F1	1. zh nincs	**Y**	**Y**	**Y**	N	N	N	N	N	N
F2	1. zh = 1	N	N	N	**Y**	**Y**	**Y**	N	N	N
F3	1. zh >= 2	N	N	N	N	N	N	**Y**	**Y**	**Y**
F4	2. zh nincs	**Y**	N	N	**Y**	N	N	**Y**	N	N
F5	2. zh = 1	N	**Y**	N	N	**Y**	N	N	**Y**	N
F6	2. zh >= 2	N	N	**Y**	N	N	**Y**	N	N	**Y**
Tevékenységek										
T1	Helyből UV	X	X		X	X				
T2	Pót 2. zh			X						
T3	Átlagjegy felkerek.						X	X	X	X

Miért hasznos?

- Kompakt és struktúrált módon képes bemutatni az üzleti folyamatot
- Könnyebben kimutatja a hibákat (inkonzisztencia, vagy hiányzó kritérium)

Döntési tábla

- Általában 1 oszlop = 1 teszteset
- Oszlopok száma általában a feltételek négyzetével egyenlő
- Egyes esetek összevonhatóak, ha a feltétel “don’t care”
- Egyforma oszlopok összevonhatóak
- Invalid (értelmetlen) kombinációk is előfordulhatnak, ezeket szintén törölni kell (pl vki senior és junior egyszerre ...stb)
- Minden oszlophoz tevékenység meghatározása (a követelményekből derül ki a tevékenység)
- Oszlopok elnevezése (R1, R2.. vagy kifejező elnevezés is lehet)

Példa

Egy vevő az ATM ből szeretne pénzt felvenni

Szabály az ATM ben, hogy csak akkor fizeti ki az összeget, ha a vevőnek van elegendő pénz a számláján (R1), **VAGY** a tranzakció jóvá lett hagyva (R2)

Feltételek	**R1**	**R2**	**R3**
Felvevendő összeg <= egyenleg	T	F	F
Tranzakció jóváhagyva	-	T	F
Tevékenységek			
Pénzfelvét jóváhagyva	T	T	F

Tesztesetek elkészítése

Feltételek	**R1**	**R2**	**R3**
Felvevendő összeg <= egyenleg	T	F	F
Tranzakció jóváhagyva	-	T	F
Tevékenységek			
Pénzfelvét jóváhagyva	T	T	F

Test case R1:
Egyenleg = 200
Kért összeg = 200
Elvárt eredmény: Pénzfelvét jóváhagyása

Test case R3:
Egyenleg = 100
Kért összeg = 200

“Hitel elutasítva”
Elvárt eredmény: Pénzfelvét megtagadva

Test case R2:
Egyenleg = 100
Kért összeg = 200

“Hitel jóváhagyva”
Elvárt eredmény: Pénzfelvét jóváhagyása

Döntési Tábla - Feladat

Don’t Care kategória

	Női köszöntő	Férfi Köszöntő	Női köszöntő évforduló	Férfi köszöntő évforduló	Női köszöntő szilveszter	Férfi köszöntő szilveszter
Hölgy a címzett?	T	F	T	F	T	F
Évforduló?	F	F	T	T	F	F
Szilveszter?	F	F	F	F	T	T
Napi SMS Férfi	N	Y	N	Y	N	N
Napi SMS Hölgy	Y	N	Y	N	N	N
Plusz 500 MB	N	N	Y	Y	N	N
BUÉK	N	N	N	N	Y	Y

Döntési Tábla - Feladat

Minden nap SMS küldés Hölgy vagy Férfiak számára külön megszólítással.

Ha évfordulója van, akkor + 500 MB net üzenetet is kap.

Ha Szilveszter van, akkor BUÉK üzenet küldése mindegy, hogy férfi vagy nő, viszont ebben az esetben napi köszöntőt nem küldünk!

Egyéb technikák

- **Entry/exit coverage** Minden lehetséges hívás és visszatérés meg lett hívva a függvényben?
- **Function coverage** Minden függvény meg lett hívva ami a kódbázisban van?
- **Path coverage** Minden lehetséges út be lett e járva a kód egy adott részén / moduljában. Az út lefedettség magába foglalja a döntési és utasítás lefedettséget és az belépési/kilépési lefedettséget.
- **Loop coverage** ...

Köszönjük a figyelmet!

Attrecto Zrt.
Attrecto Next Tech Digital Solutions

H-9024 Győr, Wesselényi str. 6.
info@attrecto.com