**Tesztelési Követelmények**

**1. A tesztek céljai**

A szoftver tesztelésére azért van szükség, hogy az esetleges hibákat, nem kívánatos eseményeket,

szintaktikai, és logikai problémák felfedésre kerüljenek. Így beazonosíthatóvá válik, hogy egyes problémák hol találhatóak a kódban, ezért egyszerűbb azok elemzése és helyes javítása.

Mivel a szoftvernek a Szoftver Követelmény Specifikáció (SRS), és a Követelmény Analízis (RA)

dokumentumokban foglaltaknak eleget kell tennie, ezért ezen dokumentumok alapján a kódbázis egyszerűbben aktualizálhatóvá tehető. Ez azt jelenti, hogy tudni fogjuk, hogy egyes komponenseknek, illetve rendszerfunkcióknak milyen viselkedést kell produkálniuk.

A hangsúly természetesen a helyes kimeneteken van, ehhez pedig ellenőrzött keretrendszerben kell végrehajtanunk a teszteket. A következőkben áttekintésre kerülnek a tesztelési módszerek.

**2. Statikus tesztek**

* Szemantikai teszt: áttekintjük, hogy a program kódjainak részei logikailag a követelményekben meghatározott definíciókat követi-e. Ellentmondás esetén változtatunk  
  a programrész struktúráján, vagy az adott algoritmuson.
* Szintaktikai teszt: ez automatizált, lévén, hogy az IDE automatikusan elvégzi helyettünk.

**3. Dinamikus tesztek**

* Egységteszt: bemenetek, kimenetek, és hibakezelés megfeleltetése az elvárt eredményekhez. Ehhez tesztkeretrendszert használunk.
* Formai teszt: megfelelő elnevezések, konvenciónak megfelelő használata.   
  Technikailag a projektben előforduló mappák, fájlok, komponensek, osztályok, változók stb.  
  tartalmi vonatkozását teszteli. Ezt manuálisan kell megoldani.

**4. Fekete doboz tesztelési módszerek**

A dinamikus tesztekhez kötődik, tehát a program egységeit fekete dobozoknak vesszük, az elvárt

I/O, valamint hibakezelési kimeneteket összevetjük az elvárásokkal. Ehhez tesztadatokkal kell dolgoznunk.

**5. Hibakeresési módszerek**

Az alábbiak segítségével tudjuk megvalósítani a hatékony hibakeresést:

* IDE – Fejlesztőeszköz
* NUnit – Tesztelési keretrendszer
* Logging – Események naplózása a programon belül
* StackTrace – Hibák visszavezetése

**6. Lehetséges hibák**

I/O & hibakezelés szempontból:

* Helyes bemenet esetén nem az elvárt működés áll elő
* Nem keletkezik kivétel akkor, amikor kellene
* Kivétel keletkezik akkor, amikor nem kellene
* Helytelen bemenet esetén, nem keletkezik kivétel, a program  
  az adott algoritmus szemszögéből megszegi a szabályokat