**Tagek és metaannotációk használata**

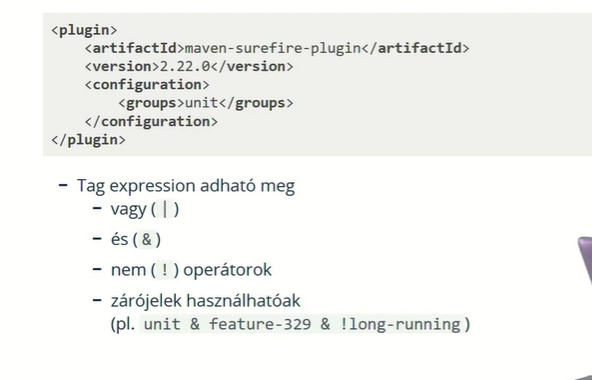
**Tagek**

a jUnit unit teszteket el lehet látni tagekkel is; egy tesztesetre akár több taget is rá lehet tenni; ezeket a tag annotációkat metódusszinten és osztályszinten is használhatjuk



**Filterelés Mavennel**

a tagek alapján tudunk filterelni, hogy milyen taggel ellátott esetek fussanak csak le -> Maven esetén a surefire plugin beállításánál ezt konfigurálni kell: a group tagen belül megadjuk a kifejezést, hogy mire szűrjön a Maven; itt nem csak egy nevet lehet megadni, hanem bonyolultabb kifejezéseket is, operátorokkal



**Metaannotációk**

mi magunk létrehozhatunk annotációkat és ezekre további annotációkat tehetünk

létrehozunk egy UnitTest annotációt -> public @interface UnitTest; rátesszük a @Test és @Tag annotációkat erre

az általunk létrehozott annotációt mintegy rövidítésként használhatjuk a tesztmetódusokon

Itt meg vannak adva a következőek:

* az új annotáció neve (public @interface ServiceTest)
* mire tehető rá a létrehozott új annotáció (@Target(ElementType.TYPE, ElementType.METHOD), vagyis osztályra és metódusra),
* mikor kerüljön feldolgozásra ez az új annotáció (@Retention(RetentionPolicy.RUNTIME), vagyis futásidőben),
* az, hogy tesztesetnél akarjuk használni ezt az annotációt (@Test)
* és kerül rá egy tag is (@Tag("service")).



**meta arra is jó, ha elgépelném a tageket; csak egyser kell jól írni aztán meg előhívni**

**új Annotáció létrehozása**

**mint az osztályt(jobb klikk new), csak Annotaitont kiválasztva!**

****

**Tesztesetek csoportosítása a gyakorlatban**

1.ráraktunk különböző tageket a teszt osztályokra (@Tag("service") és @Tag("entity")

2.fent empapp in empappra klikk és edit configuraitons -> all in package helyett tags

3.mavenbe is kofiguráltuk -> surefire config groups