

Kövesdán Gábor:

Önlab és dolgozatírás tanácsok

Tanszékünkön rengeteg tehetséges hallgatóval találkozunk, azonban azt tapasztaljuk, hogy a szakmaiságon kívül rengeteg egyéb aspektus is befolyásolja az elkészült munkák megítélését. Ebben a rövid dokumentumban összeszedtem azokat a meglátásaimat, amelyekkel tapasztalatom és a kollégák visszajelzései szerint igényes munka készíthető.

[Mitől jó egy dolgozat?](#)

[Témaválasztás](#)

[A dolgozat felépítése](#)

[A feladatkiírás](#)

[Az összefoglalás](#)

[A bevezetés](#)

[A felhasznált technológiák](#)

[A tervezés](#)

[Az implementáció](#)

[A tesztelés](#)

[Némi extra: üzemeltetés, DevOps](#)

[Az összegzés](#)

[Az irodalomjegyzék](#)

[A függelékek](#)

[A munkaszervezés](#)

[A helyesírás](#)

[Toldalékolás idegen szavakhoz](#)

[Külön- és egybeírás](#)

[A mozgószabály](#)

[A vonatkozó névmás helyes használata](#)

[Az ami kötőszó](#)

[Az és kötőszó](#)

[A dolgozat beadása](#)

[A prezentáció és a védés](#)

[Az adminisztráció](#)

[Néhány további tanács](#)

Mitől jó egy dolgozat?

Milyen egyáltalán egy jó dolgozat. Összeszedtem néhány jelzőt, hogy szerintem milyen, illetve milyen nem kellene, hogy legyen:

- **Objektív:** mivel egy mérnöki munkáról van szó, a dolgozatodnak alapvetően objektívnek, tényyszerűnek kell lennie.
- **Szubjektív:** ugyanakkor tartalmaz szubjektív meglátásokat is, hiszen a saját munkád bemutatásáról van szó. Elmondhatod, mit tartottál nehéznek, érdekesnek, hasznosnak stb. a feladat során, hiszen ezek a benyomások is igen érdekesek. A lényeg, hogy derüljön ki, hogy valamit véleményként közölsz, és nem tényként állítasz.
- **Referenciaszerű:** vannak olyan tantárgyak és feladatok, ahol formális jelöléseket kell precízen és minden részletre kiterjedően használni, a szakdolgozat és a diplomatervezés azonban nem ilyen.
- **Olvasmányos:** Sokkal jobbnak tartom, ha egy érdekes, olvasmányos írás születik, mintha referenciaszerűen felsorolva írnál le mindent.
- **Érdekes:** ez persze szubjektív, de mindenképp jó, ha tudod érdekesen tálni a témát. Ebben segít, ha olyan témát választasz, amiért egyébként is lelkesedsz.
- **Alapos:** a témát körültekintően meg kell vizsgálnod, és megfelelő módszereket, technológiákat választanod. Érvelned kell tudni a döntéseid mellett, az nem elég érv, hogy ezeket a technológiákat ismerted.
- **Kerek egész:** a jó dolgozat egy logikus vezérfonalon halad végig a probléma bemutatásától az elkészült megoldás kiértékeléséig, és így egy kerek történet áll össze a dolgozatban.
- **Logikus, érthető:** Ebben a kerek történetben a munka egyes fázisait, a döntéseidet logikusan felépítve, érthető módon kell visszaadnod.
- **(Tudományos:)** nem kell egyáltalán tudományosnak lennie, hiszen mérnökként a meglévő módszerek és technológiák alkalmazása a célod. Ha azonban olyan témát választasz, előfordulhat, hogy részben tudományos aspektusa is lesz a dolgozatodnak, pl. megalkotsz egy új algoritmust, és megméred a teljesítményét.

Témaválasztás

A témalabor és az önálló labor témaválasztásában sokkal nagyobb a szabadság, ugyanis nem szükséges hogy a félév végére egy kerek történet álljon össze, mint a szakdolgozat és a diplomatervezés esetében. Bár a félév végi prezentáció is számít, az értékelés nagyrészt a konzulens feladata, és figyelembe tudja venni azt is, ha sok időt töltöttél új technológiák tanulásával vagy olyan próbálkozásokkal, amelyek végül nem vezettek érdemi eredményre. Ezért ezen félévek alatt van inkább lehetőséged jelentősen új területeket megismerni. Viszont szakdolgozat és diplomatervezés esetén ne válassz olyan témát, amelyben:

- Sok számodra új technológia van.
- A feladat egésze idegen számodra.
- Nem látod előre, hogyan, milyen lépéseken át jutsz majd el a megoldásig.
- A feladat kísérletező jellegű, nem biztos, hogy szép vagy használható végeredmény fog kijönni.

Ha bizonytalan vagy, kérdezd a konzulenszt, és segít eldönteni, hogy mekkora lélegzetű a feladat, és belevághatsz-e.

A jó szakdolgozat és diplomaterv feladat egy kerek egész történet, amelynek van eleje és vége, a részfeladatok egymásra épülnek benne. Lehetséges társulni is más hallgatókkal, és egy még nagyobb feladatot felosztani, pl. backend, frontend és mobil részekre. Néha előfordul, tipikusan külső, céges témánál, ahol a hallgató a cég érdekeihez is próbál igazodni, hogy több, lazábban kapcsolódó feladatrészből tevődik össze a dolgozat. Ebben az esetben is legalább olyan szempontból képezzem egy egységet a feladat, hogy ugyanazon alkalmazás részeiről vagy ugyanazon üzleti folyamat elemeiről legyen szó. Megértjük, hogy a céges munkát szeretnétek elszámolni egyetemi munkaként is, de az egyetem részéről támasztott követelményeknek és elvárásoknak is meg kell felelni.

A szakdolgozat és a diplomaterv tárgyak elején mindig először egy feladatkiírást kell készítened a konzulenssel, ebben fogalmazzatok meg a feladatot, és később ez majd az értékelés alapjául is szolgál. Ami ide bekerült, azt mindenképp meg kell majd csinálnod. Céges téma esetén akkor is meg kell csinálnod a feladatot, ha közben esetleg a cégnek már nem lenne rá szüksége.

Bár életszerű helyzet az is, hogy egy adott technológiát (pl. Spring Boot, Angular, Android fejlesztés) szeretnél megtanulni, azt javaslom, hogy mindenképp egy érdekes témát válassz, ne egy mindennapi középszerű alkalmazást, mint pl. webshop vagy mobilos bevásárlólista. Sokkal motiválóbb és érdekes egy innovatívabb, kreatívabb témán dolgozni. Az egyetemi emlékeimből tudom, hogy bőven akadnak olyan tantárgyak is, amelyek nem érdekelnek, használd ki, hogy itt részben van választásod, és keress egy érdekes témát, amelyben van kihívás, és amelyre majd büszkén gondolsz vissza, vagy mutatsz be mondjuk egy állásinterjún is. Én konzulensként csak olyan témákat írok ki, vagy vállalom el, amelyek vagy a témájukat vagy a technológiai megvalósításukat tekintve újszerűek, kreatívak. Pl. egy virtuális állatmenhely a témáját tekintve érdekes, nem megszokott, egy gráfadatbázissal készült közösségi háló pedig a megvalósítását tekintve újszerű. Esetleg az is érdekes lehet, ha a téma maga nem túl izgalmas és bonyolult, de több technológiával is elkészítéd, és összehasonlítod az elkészült megoldásokat.

A dolgozat felépítése

A színvonalas dolgozat egyik kulcsa a jól megtervezett logikus felépítés. Így lesz a dolgozat olvasható és érthető. Véleményem szerint a jó dolgozat végigvezet a mérnöki munkafolyamaton:

- Adott egy probléma
- Elemzem, megvizsgálom pontosabban a problémát
- Megvizsgálom, számba veszem az elérhető technológiákat, megoldásokat, és kiválasztok egy ideálisat

- Megtervezem a megoldást
- Kivitelezem a megoldást
- Ellenőrzöm a megoldásom működőképességét
- Értékelem az elkészült megoldást

A legtöbb esetben azt javaslom, hogy ezeken a lépéseken haladj végig a dolgozatodban, pl. Így:

Probléma felvetése	Feladatkiírás, Összefoglalás
Probléma elemzése	1. Bevezetés
Eszközök vizsgálata	2. Felhasznált technológiák
Megoldás megtervezése	3. Tervezés
Kivitelezés	4. Implementáció
Működőképesség ellenőrzése	5. Tesztelés
Értékelés	6. Összegzés
	Irodalomjegyzék
	(Ábrajegyzék)
	(Függelékek)

Egy alternatív felépítés lehet, hogy ha a feladat több, lazábban kapcsolódó részből áll, akkor a 3-4-5. fejezet helyett 1-1 fejezet dedikálsz az egyes részfeladatoknak, és azokon belül haladsz végig a tervezés, implementáció és tesztelés fázisain. Most az első javasolt felépítést vesszük végig részletesebben, de természetesen ezt az alternatív javaslatra is könnyen tudod alkalmazni.

A feladatkiírás

Szakdolgozat és Diplomaterv 1 esetén szükséges lesz egy feladatkiírás. A feladatkiírás a feladat nagyon magas szintről vett összefoglalását tartalmazza. Ez egy hivatalos dokumentum, és a kiadott sablon szerint kell elkészíteni. Vagy a konzulens készíti, vagy megkér, hogy készítsd el, és utána átnézi. Ez egy egyoldalas dokumentum, de a lényegi rész benne kb. fél oldal. Először a feladat folyószöveges ismertetését tartalmazza, utána pedig pontokba szedve tartalmaz néhány előírást. Az értékelésnél a konzulens, a bíráló és a bizottság vizsgálja majd, hogy minden teljesül-e, és neked is illik a dolgozat végén reflaktálni rá.

Az összefoglalás

Ez egy kb. 1-1,5 oldalas számozatlan rész a dolgozat elején, amelyet majd külön is fel kell töltened a diplomaterv portálra. Tulajdonképpen a feladatkiírás tartalmát fejt ki némileg bővebben, olvasmányosabban. A könyvek végén található kedvcsináló fülszöveghez hasonlítható, ugyanaz a célja, vagyis, hogy az olvasó megismerje, hogy miről lesz szó, és kedvet kapjon az olvasásához.

A bevezetés

Itt már részletesebben ki kell fejtened a problémát. Milyen részekből áll, milyen kihívások vannak benne? Milyen fő módszerek léteznek a probléma megoldására? Vannak-e esetleg kész megoldások, amelyekre alternatívát fogsz kínálni? Miben lesz más a megoldásod? Természetesen nem elvárás, hogy üzletileg is működőképes alkalmazást alkoss, de érdemes azért röviden kötnöd az elérhető alternatívákhoz. Ennek a résznek a hossza kb. 3-5 oldal. Mindig feltételezd, hogy alapvetően szakmabeliek olvassák a dolgozatot, ezért nem kell a bibliai teremtéstörténettel, Neumann Jánossal, lyukkártyákkal, de még csak az internet történetével sem kezdenek az írást. A saját feladatod érdekes felvezetésére koncentrálj.

A felhasznált technológiák

Itt bemutatathatod, hogy milyen technológiákat használsz a megoldáshoz. Fontos, hogy nem szakkönyvet és nem tutorialt készítesz, nem kell az olvasót megtanítanod a technológiák használatára, illetve alapvető szoftveres előismereteket feltételezhetsz is tőle. A mérnöki szemlélet alkalmazása itt is fontos, a technológiák fő feature-eire koncentrálj, emeld ki az előnyöket, hátrányokat, de ne menj szakkönyvekre jellemző részletekbe. A technológiákat pedig valamilyen logikus sorrendben mutasd be, és azt javaslom, hogy alulról felfelé haladj. Pl. egy full-stack webalkalmazásban mutasd be az adatbázisban, majd a backendben, végül a frontendben használt technológiákat, végül esetleg a további eszközöket, mint a különböző IDE-k, git, github és egyebek.

A tervezés

Ide jöhetnek a terveid olvasmányos magyarázatok és diagramok formájában. "Egy kép többet ér ezer szónál" – tartja a mondás, nem véletlenül, ezért javaslom, hogy bőven használj ábrákat, diagramokat. A referenciaszerű felsorolásokat (pl. use case-ekét) viszont kerülném, inkább folyószövegben foglald össze a lényegét, és a részletek pedig az ábrán legyenek láthatóak. Az UML-diagramok jól mutatnak itt, különösen az osztálydiagramok az entitásaidból és ha van bonyolultabb interakció, pl. social login oauth2-vel, akkor 1-1 szekvenciadiagram is jól jöhet. Viszont pl. architektúra szemléltetéséhez a komponensdiagramnál jobbnak tartom a nemszabványos saját ábrát ikonokkal. A komponensdiagramban minden egy ugyanolyan téglalap, saját ikonokkal viszont az adatbázis lehet egy diszk ikon, a mobilkliens mobiltelefon stb. Adok pár ötletet, hogy mire térhetsz ki ebben a fejezetben, persze elég az alkalmazásodban releváns elemekre koncentrálnod a listából:

- Felhasználói szerepkörök és jogosultságok
- Entitások és kapcsolataik, adatbázisséma ha abból indultál ki, és nem kódból generáltad
- Főbb use case-ek
- Összetettebb interakciók szekvenciadiagramja
- Architektúra, modulokra bontás
- REST API endpointjainak specifikációja

- Felülettervek

Az implementáció

Ahogy korábban említettem, itt is rétegenként alulról-felfele javaslom a haladást. Itt is javaslom kerülni a referenciaszerű felsorolásokat, pl. ne sorold fel az összes entitást vagy az összes felhasználói felületet. Ezek kevés értéket adnak a dolgozatodhoz, és unalmasak is. A célod annak a megmutatása, hogy rendelkezzel a megfelelő mérnöki gondolkodásmóddal, és ezt legjobban szerintem úgy éred el, ha a rétegenként felfelé haladva kiemelsz problémákat, kihívásokat, és ezekre mutatsz megoldást illetve példakódot. Pl. ha szükség volt adatbázisba mentett entitásokra, akkor elmondhatod röviden, hogy a JPA-val oldottad meg, ez annotációkat használ, ezek közül melyek a legfontosabbak, és végül mutatsz egy példakódot, ahol ez megfigyelhető. De ne sorold fel az összes entitást, az szükségtelen, csak 1-2-őt emelj ki, amelyeken jól bemutatom a JPA lehetőségeit.

A tesztelés

Bár látok néha olyan dolgozatokat is, amelyekben egy szó sem esik a tesztelésről, a mérnöki munkának mindenképp része a tesztelés, ezért illik foglalkoznod vele, ha igényes, színvonalas munkát szeretnél kiadni a kezeid közül. Tesztelhetsz manuálisan is, akár különböző eszközöket alkalmazva, pl. a REST API-k teszteléséhez a Postmant, de az igényesség minimum szintjét inkább ott mérjük, hogy a kritikusabb funkciókhoz automatizált unittesztjeid is legyenek. Ez manapság alapvető kompetenciának számít, nemcsak tesztelésre szakosodott szakemberektől, hanem hétköznapi fejlesztőktől is elvárható. Ezért azt javaslom, hogy legalább ennek az alapjait, illetve ehhez kapcsolódóan a függőségek mock objektumokkal történő helyettesítését tanuld meg. Ambíciózusabbaknak szélesebb paletta is rendelkezésre áll a témakörben: linting, coverage report, integrációs tesztek, UI tesztek, elfogadási tesztek, teljesítménytesztek stb.

Némi extra: üzemeltetés, DevOps

Nem elhanyagolható kérdés az sem, hogy az alkalmazást hogyan lehet majd üzemeltetni, és ez a kérdéskör egyre inkább összefonódik a fejlesztéssel, hiszen már fejlesztés során gondolni kell rá. Ebből született meg a DevOps mozgalom. Nem elvárás egy átlagos fejlesztési témánál, de ha érdekel, beleveheted a feladatba ennek a vizsgálatát is. Megvizsgálhatod pl. a konténerizáció lehetőségét Dockerrel, a Continuous Integration és Continuous Delivery technikáit, a konfigurációmenedzsmentet, a felhőben való telepítést.

Az összegzés

Itt összegezheted az elért eredményeidet. Mindenképp utalj a feladatkiírásra, hogy mennyire sikerült a feladatot megvalósítani. Érdemes kitérned a hiányosságokra és a továbbfejlesztési lehetőségekre is. Nyugodtan beszélhetsz itt némi szubjektív benyomásról is: mit éreztél kihívásnak, mi volt benne az érdekes stb.

Az irodalomjegyzék

A feladatod fejlesztése során mindenképpen kutakodni fogsz megoldások után, és a forrásaid hivatkoznod is kell. Kb. 10 hivatkozás tekinthető minimum elvárt szintnek. Ezek hitelessége, színvonalas sem mindegy, csökkenő sorrendben kb. így festenek:

- Tudományos források: nem elvárás, de ha olyan a témád, akkor lehet ilyeneket is hivatkozni
- Szabványok, amelyeket használsz
- Szakkönyvek, technológiák hivatalos dokumentációi
- Technológiai blog, fórumtéma: mértékkel, de hivatkozatsz ilyeneket, lehetséges, hogy 1-1 problémára pont itt találod a megoldást
- Wikipedia: csak ha nincs jobb

A függelék

Egy képernyőképes végigvezetés itt szemléletes lehet, esetleg tehetsz ide telepítési útmutatót, ha nem triviális a futtatás. Egyes dolgozatokban látok szótárat, de én ezt nem gondolom indokoltnak, hiszen az alap rövidítéseket, kifejezéseket az olvasó ismeri, amit pedig nem ismer, azt a felhasznált technológiák fejezetben egyébként is megismeri.

A munkaszervezés

Szintén fontos szerepet játszik a sikerben a munkaszervezés. Ezeknek a tárgyaknak nem pusztán az a célja, hogy a végére legyél készen a feladattal, hanem folyamatos munkát várunk. Ez azért is van így, hogy a konzulensed folyamatosan visszajelzéseket tudjon adni az eddig elért munkádról, és ezeket alapján javítva, finomítva az eddigi megoldásod, szakmailag többet tudj fejlődni. Érdemes is ezt a lehetőséget kihasználnod, elvégre ideálisan olyan témát választottál, amely érdekel, és kaptál egy hozzáértő konzulens is, aki személyre szabott tanácsokkal, javaslatokkal tud ebben segíteni. Viszont a konzulenseknek akár 15-30 hallgatójuk is lehet félénként, és ha minden hallgató az utolsó pillanatban küldi a kész dolgozatot, akkor a konzulens sem tud gyors visszajelzést adni, és ha kapsz is visszajelzést, neked sem lesz már idő a beadás előtt érdemben kijavítani, átdolgozni a visszajelzései alapján. Ezért hasznos, ha félév közben is küldesz elkészült fejezeteket a konzulensednek. A probléma bemutatását és a felhasznált technológiákat is már korán, a félév elején meg tudod írni. Érdemes ezt gyorsan elkészítened, és elküldened a konzulensnek. Azért is jó, ha így dolgozol, mert a fejlesztéssel szemben a dolgozatírás munkaigényét sokan alábecsülik. Ha időben elkezdted a dolgozatot, nyugodt lesz a féléved vége, nem fogsz majd stresszelni azon, hogy sikerül-e befejezni, vagy nem. Sokan a pótbeadásra számítanak, de a pótbeadás MSc-re felvételiző BSc hallgatók esetén eleve nem megengedett, és a többi esetben sem arra szolgál, hogy egy hét alatt írd meg a dolgozatot. A konzulensed csak akkor fogja ezt engedélyezni, ha egyébként is elfogadható

már a dolgozat, és csak kisebb finomítások maradtak, pl. kicsit még át kell szervezni a dolgozatot, vagy kell még írni néhány tesztet.

A konzultáció gyakoriságát illetően a konzulenséd útmutatásait is kell követned, de jelezheted neki a preferenciádat. Én abban hiszek, ha nem rendszeresítünk időpontot, hanem igény szerint egyeztetünk, amikor egy nagyobb részt jönnél megmutatni, vagy kérdéseid lennének a haladásoddal kapcsolatban. Így hatékonyan töltjük el az időt, és nem kell azért bejőnnöd, hogy szabadkozz, amiért az utóbbi egy hétben nem sikerült haladnod, hiszen annak nincs értelme. Minimum elvárásként viszont kérek kéthetente státuszjelentést a munkádról, hogy tudjam, hogyan haladsz, és ezeket az értékelésnél figyelembe is veszem. Nálam ez a rendszer bevált, és a hallgatóim nagy része időben elkészül, és ezért tudok is adni nekik érdemi visszajelzést a munkájukról.

A visszajelzések technikai kivitelezésével kapcsolatban hatékonynak találtam, ha Word vagy Google Docs formátumban kapom a félkész dolgozatokat, és ezekben kommentek formájában jelzem az észrevételeimet. Szintén hasznos a programkódot githubon tárolni, mert akkor meg tudom nézni, és erről is tudok visszajelzést adni.

A helyesírás

Nem vagyunk nyelvészek, és valószínűleg egy szakdolgozat vagy diplomaterv sem lesz teljesen mentes a helyesírási vagy nyelvhelyességi hibáktól, de azért érdemes az igényességre törekedni. A Word, az OpenOffice és a LibreOffice is tudnak magyarul helyesírást ellenőrizni, ha pedig Latex-ben készíted a dolgozatot, akkor is át tudod copy-paste-elni az ellenőrzés erejéig. A helyesírás befolyásolja az összképet, és a bírálók is meg szokták jegyezni, ha sok a hiba a dolgozatban. A továbbiakban néhány gyakori helyesírási és nyelvhelyességi kérdést tekintünk át.

Toldalékolás idegen szavakhoz

Ha az utolsó betűt úgy ejted ki, mint ahogyan magyarul is ejtenéd, akkor nincs kötőjel, egybeírva jön a toldalék, pl. Microsofttal, Amazonon, Javában.

Ha az utolsó betűt másképp ejted ki, kell a kötőjel: Skype-on, Windows-zal, pendrive-ot

Külön- és egybeírás

Ha az első szó a másodikat jellemzi, és melléknév, akkor külön kell írni: reguláris kifejezés, virtuális gép, logikai diszk.

Ha az első szó a másodikat jellemzi, de nem melléknév, akkor egybe kell írni (kivéve ha legalább három tagból és toldalékok nélkül számolva hat szótagból áll, mert akkor kötőjellel). Az ilyen összetételeket jelentéstömörítő összetételeknek nevezzük, mert a jelentésük csak bonyolultan fejthető ki. Pl. a vámvizsgálat egy vámmal kapcsolatos vizsgálat, vagy a csigalépcső egy csiga formájú lépcső. Példák még erre az esetre: backendalkalmazás, adatbázis-kezelő (három tagú és több mint hat szótag), alkalmazásintegráció, felhőalkalmazás.

Ha nem vagyunk biztosak valamiben, érdemes megnézni az MTA eszközében is: <https://helyesiras.mta.hu/helyesiras/default/kulegy>

A mozgószabály

Többféle mozgószabály van, most a leggyakoribbat tekintjük át. Pl. a meleg víz kifejezést külön írjuk, mivel a meleg melléknév. Mi történik azonban ha az összetételhez hozzáadjuk a csap szót is? Hogyan írjuk helyesen? Ha azt írnánk, hogy meleg vízcsap, az azt jelentené, hogy a vízcsap meleg, nem pedig a belőle folyó víz. Lehet ennek az írásmódnak is értelme, pl.: Egy forró kádfürdőt vettem, és a meleg vízcsap megégette a kezemet. Tehát a vízcsap felmelegedett, és a vízcsap volt meleg. De általában nem ezt akarjuk mondani, hanem egy olyan csapról beszélünk, amelyből meleg víz jön. Ilyenkor az egyébként különírandó meleg víz egybeolvad, és a csap kötőjellel követi: melegvíz-csap. Talán elsőre furcsa ez az alak, de jól tükrözi, hogy a meleg jelző a vízhez tartozik, és nem a csaphoz. A műszaki nyelvben is sokszor találkozhatunk ilyen összetételekkel, pl. reguláriskifejezés-implementáció, hidegindítás-kezdeménnyezés, többszintűmodell-generálás.

A vonatkozó névmás helyes használata

A vonatkozó névmást arra használjuk, hogy kijelöljön, határozottá tegyen egy dolgot. Sokszor hibásan akkor is alkalmazzuk, amikor eleve konkrét dologról van szó, és egyszerűen hozzá akarunk fűzni valamit, pl.:

A CI szerver lefordítja a backendalkalmazást, amelyet a felhőbe telepítünk.

Itt a backendalkalmazás már önmagában is konkrét, határozott névelős szó. Helyesebb, ha ezt írjuk:

A CI szerver lefordítja a backendalkalmazást, és ezt a felhőbe telepítjük.

Mutatok egy helyes példát is a vonatkozó névmás használatára:

A backendalkalmazást, amelyet a felhőbe telepítettünk, a CI szerver fordította le.

Jelen esetben a vonatkozó mellékmondat kijelöli, hogy melyik backendalkalmazásról van szó.

Az ami kötőszó

Az "ami" kötőszó nem ekvivalens az "amellyel." Előbbit csak teljes tagmondatokra történő visszautaláshoz használhatjuk. Helytelen ezért a következő mondat, mert itt az "ami" a backendalkalmazásra, vagyis egy konkrét mondatrészre utal vissza:

A backendalkalmazást, amit a felhőbe telepítettünk, a CI szerver fordította le.

Helyes viszont a következő mondat:

A backendalkalmazást a felhőbe telepítettük, ami nagyban csökkentette az üzemeltetési költségeinket.

Itt ugyanis az "ami" arra az egész tényre utal vissza, hogy a backendalkalmazást a felhőbe telepítettük.

Az és kötőszó

Nagyon sokan tévesen azt gondolják, hogy az “és” előtt sosem szükséges vessző. Ez azonban csak akkor igaz, ha az “és” mondatrészeket (szavakat) és nem tagmondatokat választ el. Pl. jelen esetben nincs vessző, hiszen két tárgyat kapcsol össze a kötőszó:

A backendalkalmazást és a frontendalkalmazást is kifejlesztettem.

A következő esetben viszont két tagmondatot kapcsol össze, ezért vesszőt kell elé tenni:

A backendalkalmazást megterveztem, és ki is fejlesztettem.

Keresd mindig az állítmányokat (ezek gyakran igék, de nem mindig), hogy el tudd dönteni, szükséges-e vesszőt tenned.

A dolgozat beadása

Azt javaslom, hogy a dolgozat beadása előtt többször nézd át, és legyen hozzá egy szisztematikus átnézési stratégiád is. Mindig egy adott szempont szerint nézd át, pl.:

1. Teljes átnézés, figyelve az érthetőségre és a helyesírási, nyelvhelyességi hibákra.
2. Képek: jó helyen vannak-e, olvashatóak-e a feliratok, érdemes-e valahova további képet tenned? Érdemes a képeket számozott képaláírással ellátnod, így a szövegben tudsz rájuk egyértelműen hivatkozni, pl.: “A 2.3. ábrán látható a kifejlesztett alkalmazás architektúrája.”
3. Kódrészletek: jól szemléltetik-e, amit meg akarsz mutatni, szépen vannak-e formázva, behúzva, szükséges-e további kódrészlet?
4. Irodalomjegyzék: minden fontos technológiához létezik hivatkozás, és egységes formátumban vannak?
5. Tördelés

A tördeléssel kapcsolatban összefoglalom a legfontosabb tudnivalókat. Ha mód van rá, alkalmazz elválasztást, úgy lesz igényes a tördelés. Alapvetően hagytatkozz a saját ítéltőképessegedre, és nézd végig, hogy minden szépen helyezkedik-e el a szövegben. Van néhány alapszabály a tördelésben:

- Mondatkezdő “a” névelő után ne legyen sortörés, inkább vidd át ezt is új sorba. (Előtte nyomj Shift-Entert.)
- A fejezetcímek maradjanak együtt a fejezet első bekezdésével, ne legyen köztük sortörés.
- A képaláírás maradjon a képpel együtt. A táblázatnál viszont ha alkalmazol címet, azt a táblázat elé kell tenni, mert a táblázat lehet hosszú, és tödelődhet. Táblázat tördelése esetén az oszlopok nevét meg kell ismételni az új oldalon.
- Kerüld az árva- és fattyúsorokat! Minden bekezdésből min. 3 sor maradjon a tördelés előtt és után.

A végén pörgesd át még egyszer, nézd meg, nem maradt-e benne semmi furcsa, pl. komment vagy korrektúraszöveg. Ezután készítsd elő a mellékletet. Ha a dolgozatod nem titkosított, és nem sérülnek üzleti érdekek, akkor töltsd fel a forráskódot mellékletként. Viszont előtte távolítsd

el belőle a főleges részeket, pl. a lebuildelt kódot. A függőségek kezeléséhez az a jó, ha valamilyen build eszközt használsz (pl. Java esetén Maven, JavaScript esetén npm), ezek ma már elvárt alapvető tudásnak számítanak. Ha a kódot githubra vagy egyéb publikus helyre is feltöltötted, és erre a dolgozatban hivatkozol, akkor beadás előtt nézd meg, hogy a kódok ténylegesen elérhetőek-e ott.

Fontos, hogy megnyomd a beadás gombot is a diplomaterv portálon. A pusztán feltöltés nem elegendő, az olyan, mintha semmit sem csináltál volna. Akkor véglegesítetted a beadást, ha már nem tudsz többé szerkeszteni.

A prezentáció és a védés

Témalabor esetén csak a konzulensednek kell bemutatnod az elkészült munkát, viszont önálló labor esetén félév végi szóbeli beszámoló van, szakdolgozat és diplomaterv esetén pedig védés. Az önálló labor beszámolóján két tanszéki kolléga és hallgatótársaid, a védésen pedig a záróvizsga-bizottság tagjai és a vizsgáztatók lesznek jelen. Az önálló labor beszámolón kb. 10 perc, a védésen 12-15 perc áll rendelkezésedre a munkád bemutatására. Mostantól a védésre fókuszálok, de a legtöbb dolog az önálló labor beszámolójára is alkalmazható.

A védés menete a következő lesz:

- Bemutatod a munkádat kb. 12 percben.
- Előzetesen a bírálótól fogsz kapni 1-3 kérdést. Ezekre otthon készülhetsz. Készíts ezekhez is 1-1 diát, amelyen rajta van a kérdés. Ha szükséges további információt vagy diákat is rátehetsz, de az is elég, ha szóban válaszolsz. A munkád bemutatása után magad is mondhatod, hogy most rátérsz a bíráló kérdéseire, mert egyébként a bizottság úgyis ezt fogja kérni.
- Ezután rögtönzött kérdéseket is kaphatsz. Ezekre igyekezz a legjobb tudomásod szerint válaszolni, de mivel nincs felkészülési idő, ha olyat kérdeznek, amire nem tudod a választ, azt is mondhatod, hogy ennek még utána kellene nézned.
- Ezután következik a szóbeli vizsga, BSc-n egy, MSc-n két tárgyból.

„Csak azért lett ez a levelem hosszabb, mert nem volt időm rövidebbet írni.” – írta Blaise Pascal. Talán elsőre úgy érzed majd, hogy az egy vagy akár többfélèves muinka túl nagy ahhoz, hogy ilyen rövid idő alatt bemutasd. Azonban ez semmiképpen sem lehetetlen, sőt tanulható. Az igazán érdekes előadások mindig azok, amelyek tényleg a lényegre tudnak fókuszálni. A munkádat szakmabeliek előtt fogod bemutatni, és nem szükséges mindent az alapoktól elmondanod. Ami igazán érdekli őket, az a saját munkád. Milyen kihívások voltak a munkád során, és mennyit dolgoztál, mit készítettél el? Tulajdonképpen hasonló menetet javaslok a bemutatónak, mint a dolgozatnak, pl.:

- Probléma (1 dia)
- Technológiák (1-2 dia)
- Tervezés (2-3 dia)
- Implementáció (3-4 dia)
- Tesztelés (1-2 dia)
- Összegzés (1 dia)

Az a kényelmes, jól követhető tempó, ha kb. percenként 1 diát készítesz. A diák legyenek vizuálisak, esetleg kulcsszavakat tartalmazzanak, a többit majd úgyis szóban elmondod. Ha túl sok szöveg van a diákon, akkor a hallgatóság azt fogja olvasni, és nem rád figyel, ráadásul hamarabb elolvassák, mint elmondod, és unatkozni fognak. Azt is fontosnak tartom, hogy legyen minél több konkrétum a munkádról. Tegyéél bele néhány screenshotot és érdekesebb kódrészletet is, így jobban képet kaphatunk a munkádról.

A kérdéseket illetően tényleg véd meg a munkád! Adj tényszerű érveket a döntéseidhez, persze azt is elmondhatod, ha utólag, a mostani tudásoddal már másképp döntenél. A lényeg, hogy mutasd meg, hogy mérnöki szemmel gondolkodsz, látod 1-1 döntés előnyét, hátrányát. Ne mondd például azt, hogy valamiért azért használtál, mert már ismerted, vagy mert manapság divatos, még ha egyébként ezért is választottad. Legalább sorolj fel akkor pár előnyt. Ha céges témán dolgozol, akkor előfordulhat, hogy olyan döntésre kérdeznek, amelyet nem neked kellett meghoznod, hanem a feladat részeként készen kaptad. Természetesen ilyenkor hivatkozhatasz erre.

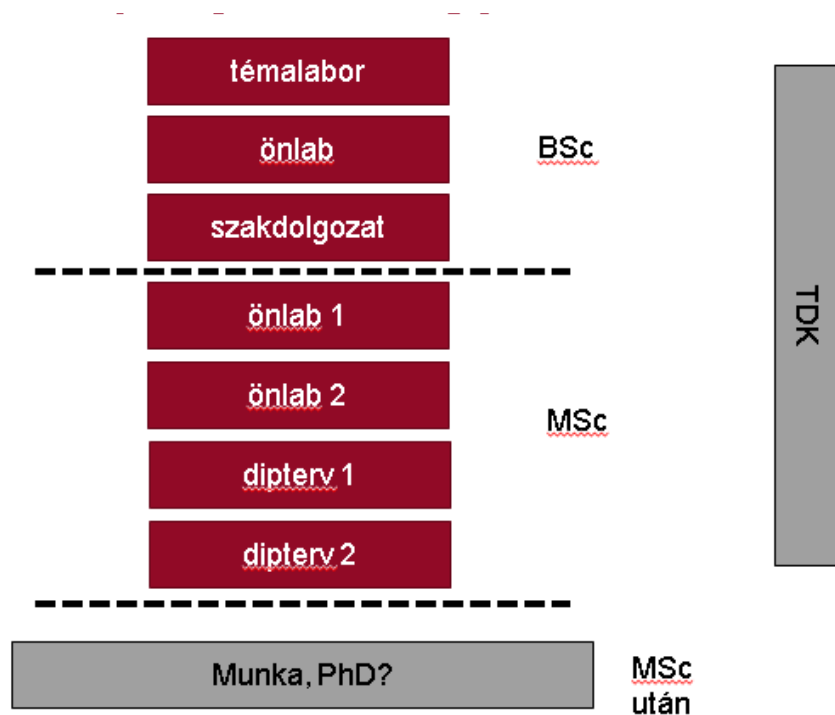
Az adminisztráció

Ebben a részben összefoglalom, hogy milyen adminisztratív teendők vannak a tárgyakkal.

- Jelentkezés témára a tanszéki portálon. Először mindenképp itt kell jelentkezned. Három fő "tématípus" létezik:
 1. A konzulens saját ötleteként kiírta.
 2. Neked van saját ötleted, és egy konzulens támogatja, kiírja neked.
 3. Céges témán dolgozol, és ehhez találsz egy belső konzulenset.
- Érdekes már jelentkezéskor írnod egy emailt is a konzulensnek, akkor is, ha előre kiírt témájára jelentkezel. Leírhatod milyen korábbi tapasztalatod van, miért érdekel a téma, és egyeztethetsz vele egy személyes megbeszélést az induláshoz.
- Ha szakdolgozatot vagy diplomatervet készítesz, akkor a konzulensnek a diplomaterv portálon is meg kell nyitnia a témát számodra.
- Ha szakdolgozatot vagy diplomatervet készítesz, el kell készítened a feladatkiírást. Lehetséges, hogy a konzulens megírja, de az is lehet, hogy megkér rá, hogy készítsd el, és ő csak átnézi, majd feltölti a diplomaterv portálra.
- Ha szakdolgozatot vagy diplomatervet készítesz, a beadás évében (tehát Diplomatervezés 1-nél nem) ki kell majd töltened egy adatlapot, ebben a záróvizsga szóbeli tárgyait is meg kell jelölnöd. BSc esetén kötelező a tanszéki tárgyat választani, kivéve ha a téma nagyon indokolna egy másik tárgyat. MSc esetén pedig a kettő közül az egyiknek tanszéki tárgynak kell lennie.
- A szorgalmi időszak végén be kell adnod az elkészült feladatot.

Néhány további tanács

A projekt tárgyakkal a BSc-MSc képzés során hét félévet is eltöltesz. Érdekes jól kihasználni ezeket a féléveket, és olyan témákat választani, amelyek érdekelnek.



Megtervezheted tudatosan ezeket a féléveket úgy, hogy fokozatosan összeszedj magadnak egy olyan technológiai portfóliót, amellyel hatékonyan el tudsz majd helyezkedni a munkaerőpiacon, és olyan dolgokkal foglalkozhatsz, amelyeket szívesen csinálsz. Például elindulhatsz a Java és Spring backendtechnológiákkal, és a későbbi félévekben hozzátanulhatsz egy kliensoldali keretrendszerrel, pl. A Reactet vagy az Angulart, és ezzel fullstack webfejlesztővé képezheted magad. Vagy elkezdheted az Android fejlesztést, és később megtanulhatod az iPhone-t is, így egyszemélyben képes leszel mindkét platformra megoldást szállítani.

Egy másik tudatos stratégia is lehet, hogy funkcionálisan bővíted tovább ugyanazt a témát, és így több félév alatt egy komolyabb alkalmazást tudsz felépíteni. Ennyi idő alatt, ha motivált vagy, készíthetsz akár életszerű, ténylegesen értéket képviselő alkalmazásokat is, és akár startup ötletekbe is belevághatsz. Akár választhatsz kutatósabb témákat is, amelyekből az egyetemen TDK dolgozatot írhat, illetve a konzulensed segítségével tudományos konferenciákon, folyóiratokban publikálhat. Ha érdekel ez a lehetőség, ezzel felkészülhetsz a PhD felvételre, hiszen a pontszámítás jelentős részét a korábbi munkák, publikációk adják.