



# Szoftvertechnológia

## 1. Bevezetés

BSc kurzus

Dr. Balla Katalin



# Bevezetés / Kurzus információk

## ■ Oktatók:

- Dr. Simon Balázs

- Dr. Balla Katalin

- Irányítástechnika és Informatika Tanszék

- Szoba: IL 415

- Tel: 463 1848

- E-mail: [balla@iit.bme.hu](mailto:balla@iit.bme.hu)

- <https://www.iit.bme.hu/targyak/BMEVIIIAB01?language=hu>



# A kurzus tartalma

- UML 1-4
- 1: Bevezetés. A szoftverről és a szoftverfejlesztésről. Aktuális trendek és problémák. Folyamatok és modellezésük.
- 2: Népszerű életciklus modellek. Hagyományos és agilis szoftverfejlesztés.
- 3: Folyamatfejlesztési modellek. Elemek, célok és gyakorlatok. Folyamatcsoportok.
- 4: Követelmények. Egyeztetés, elemzés, meghatározás, dokumentálás. A követelmények változásának menedzselése.
- 5: Tervezés és implementálás.
- 6: Tesztelés. Alapfogalmak, a tesztelési folyamat, tesztelési dokumentumok, teszt típusok.
- 7: Tesztelési technikák. Hagyományos és agilis tesztelés.
- 8: A szoftver projektek irányítási vonatkozásai. Becslés, tervezés, követés, vezérlés.
- 9: Kockázatkezelés. Támogató folyamatok: konfigurációkezelés.
- 10: Támogató folyamatok: minőségbiztosítás , mérések.
- 11: RUP és Scrum: egy hagyományos és egy agilis megközelítés összehasonlítása
- 12: Napjaink néhány aktuális problémája, „divatos” megközelítések.

# Források, hivatkozások

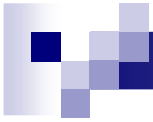
- Ez a diasor.
- Essentials of Software Engineering. Fourth Edition. Jones & Bartlett Learning, authors: [Frank Tsui, PhD](#), [Orlando Karam](#), [Barbara Bernal](#)  
[www.jblearning.com](http://www.jblearning.com).<http://www.jblearning.com/catalog/9781284106008/>
- **[Software Engineering 10th Edition](#)**. 2015. Ian Sommerville's book website.  
<http://iansommerville.com/software-engineering-book/about/>
- Balla K: Minőségmenedzsment a szoftverfejlesztésben. Budapest: PANEM, 2007
- Agile Alliance documents - <https://www.agilealliance.org>
- ISTQB Glossary of Terms used in Software Testing <http://glossary.istqb.org>
- ISTQB Foundation Level Syllabus et.al. <http://www.istqb.org/downloads/category/2-foundation-level-documents.html>
- A Glossary of Requirements Engineering Terminology Version 1.6 May 2014
- IREB Foundation Level Syllabus, Version 2.2.1 , July 24th 2017
- Requirements Engineering Fundamentals, Klaus Pohl, Chris Rupp, 2nd edition, Rocky Nook Inc. (April 2015)
- [Many sources from Internet – for example \(all references shown in the lectures\):](#)
  - <https://theagileadmin.com/what-is-devops/>
  - Kanban boards step by step.htm
  - <https://www.processmodel.com/blog/what-is-a-process/>
  - <https://www.computer.org/web/swebok>
  - <http://iso25000.com/index.php/en/iso-25000-standards>



# Források, hivatkozások

## ■ Szabványok, modellek:

- CMMI-DEV v1.3, [www.cmmiinstitute.com](http://www.cmmiinstitute.com)
- ISO/IEC 12207:2008 / 2015: Systems and software engineering -- Software life cycle processes
- ISO 15504: 2012. Information technology -- Process assessment -- Part 5: An exemplar software life cycle process assessment model
- ISO 9001:2015 Quality management systems -- Requirements
- ISO/IEC 25000:2014 . Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE
- Automotive SPICE PAM, V3.0.  
[http://www.automotivespice.com/fileadmin/software-download/Automotive\\_SPICE\\_PAM\\_30.pdf](http://www.automotivespice.com/fileadmin/software-download/Automotive_SPICE_PAM_30.pdf)
- ISO/IEC/IEEE 29119-1, 29119-2, 29119-3 and 29119-4 (2013) IEEE Standard for Software Test Documentation
- ISO/IEC 25010, System and software engineering – Systems and software Quality Requirements and 6 Evaluation (SQueRE) System and software quality models. 2011
- ... és sok más



# Kérdések?



# 1. előadás: Bevezetés

- A szoftverről és a szoftvermérnökségről
  - IT és szoftver, definíciók
  - Miért olyan fontos a szoftver? Különbözik-e a szoftver más “javaktól”? Miért, vagy miért nem?
  - A szoftverfejlesztés alapelvei
  - Példák szoftverhibákra és ezek következményeire
  - A fegyelem szükségessége a szoftverfejlesztésben
  - Próbálkozások a szoftverfejlesztés modellezésére az elmúlt 50 évben
  - Folyamatok, meghatározásuk és elemeik
  - Folyamatmodellek a szoftverfejlesztés szolgálatában



# Definíciók

## ■ IT: Information Technology / Információ technológia

- Bármire vonatkozhat, ami **számítógépes technológiával kapcsolatos**, például hálózati kommunikációra, hardverre, szoftverre, az Internetre vagy azokra az emberekre, akik ilyen technológiával dolgoznak. Sok cégnek van „IT részlege”, melyek feladata a cég üzleti tevékenységével kapcsolatosan a hálózatokat, számítógépeket stb. üzemeltetni és felügyelni.
- Az „IT job” kategóriában találunk programozót, hálózati adminisztrátort, szoftvermérnököt, webes fejlesztőt, tesztelőt, műszaki támogatást nyújtót (rendszergazdát) és még sok más foglalkozást.

## ■ ICT: Information and Communication Technology

- Az ICT az Information and Communication Technology angol kifejezés rövidítése. Az **Információ és Kommunikáció Technológia**, az **infokommunikáció értelmezését jelenti**, de az **adattfeldolgozás mellett az adattovábbítással kapcsolatos** műveleteket is magába foglalja..
- Az ICT területhez tartoznak többek között az alábbi szolgáltatások:
  - Felhő alapú adattárolás, IT infrastruktúra kialakítása, Rendszer integráció, IT kiszervezés, IT és alkalmazás üzemeltetés, SSL tanúsítvány beállítása, Szerver szolgáltatások (bérlés, hosztig, virtualizáció stb)
- <https://techterms.com/definition>, <http://www.szervervalaszto.hu/ict-jelentese/>





# Mennyire fontos az IT / szoftver?

- Képzeljük el egy napunkat nélküle....
- Mi hiányoznánk legjobban?





# A szoftver fontos...

## ■ Szoftver...

- „vezeti” a repülőgépeket
- gyorsítja, fékezi a gépkocsikat
- „mozgatja” a pénzünket
- ... életünk minden területébe „beleszól”

- Almost 90% of Dutch employees uses a computer at work.
- And they can depend heavily on it.
- Only one in 10 can function normally without.
- Others have problems:
  - either work stops (69%)
  - or it is seriously hampered (21%).

Source: research Trendbox (2007)

Kusters& Trienekens: IT Governance, Course.  
Technische Universiteit Eindhoven

# Az IT / szoftver része a mindennapjainknak

- This is a list of the leading social networks based on number of active user accounts as of April 2017.
- Facebook: 1,968,000,000 users
- WhatsApp: 1,200,000,000 users
- YouTube: 1,000,000,000 users
- Facebook Messenger: 1,000,000,000 users
- WeChat: 889,000,000 users
- QQ: 868,000,000 users
- Instagram: 600,000,000 users
- QZone: 595,000,000 users
- Tumblr: 550,000,000 users
- Twitter: 319,000,000 users
- Sina Weibo: 313,000,000 users
- Baidu Tieba: 300,000,000 users
- Snapchat: 300,000,000 users
- Skype: 300,000,000 users
- Viber: 260,000,000 users
- Line: 220,000,000 users
- Pinterest 150,000,000 users



From Wikipedia. "Most famous social network sites worldwide as of April 2017, ranked by number of active users (in millions) | Statista". Statista. Retrieved 14 July 2017. Picture: <https://www.aokmarketing.com/social-media-time-waster-waste-time/>

# Az IT drága

	2017 Spending	2017 Growth (%)	2018 Spending	2018 Growth (%)	2019 Spending	2019 Growth (%)
Data Center Systems	181	6.3	188	3.7	190	1.1
Enterprise Software	352	8.8	391	11.1	424	8.4
Devices	663	5.1	706	6.6	715	1.3
IT Services	933	4.4	1,003	7.4	1,048	4.6
Communications Services	1,392	1.3	1,452	4.3	1,468	1.1
Overall IT	3,521	3.8	3,740	6.2	3,846	2.8

Source: Gartner (April 2018)

Worldwide IT Spending Forecast (Billions of U.S. Dollars)

<https://www.gartner.com/newsroom/id/3871063>



# Definíciók

## ■ Szoftver:

- Számítógépes programok, folyamatok és esetlegesen a számítógépes rendszer üzemelésére vonatkozó dokumentációk és adatok. [IEEE 610].
  - A szoftver megjelenhet koncepciók, ügyletek vagy eljárások alakjában. Egy példa a szoftverre a számítógépprogram

## ■ Szoftvertermék:

- Számítógépi programok, eljárások, adatok és a hozzájuk tartozó dokumentáció komplett készlete [ISO/IEC 12207:1998]

## ■ Egyéb meghatározások

- Szoftver = olyan szellemi alkotás, amely magában foglalja valamely adatfeldolgozó rendszer működését biztosító programok, eljárások, szabályok és a hozzájuk tartozó dokumentáció összességét. (ISO 9000-3)
- Szoftvertermék = számítógépi programok, eljárások, adatok és a hozzájuk tartozó dokumentáció olyan komplett készlete, amelyet valamely felhasználónak való szállításra terveztek. (ISO 9000-3)
- Szoftver: Parancsok sorozata, amely a hardver működését ellenőrzi.



# A szoftver mint karrierlehetőség és üzlet

- Önök itt mind a tudomány, technológia, mérnökség vagy matematika területén szeretnének elhelyezkedni („a student interested in a career in science, technology, engineering, or mathematics (STEM)”. Milyen lesz a munkaerőpiac, amikor végeznek?

- [https://cs.calvin.edu/documents/computing\\_careers](https://cs.calvin.edu/documents/computing_careers)

- Az U.S. Bureau of Labor Statistics (US-BLS) kétévente két lehetséges forgatókönyvet készít a következő 10 évre:
  - **Új állások** – olyan állások, amelyek korábban nem léteztek
  - **Összes állás** – olyan új munkahelyek, amelyet bővítés miatt hirdetnek meg, vagy a nyugdíjba vonulók pótlására.
- **Mindkét esetben – az US-BLS előrejelzése szerint- , a számítógépes terület lesz a legbiztosabb karrierlehetőség a belátható jövőben.**

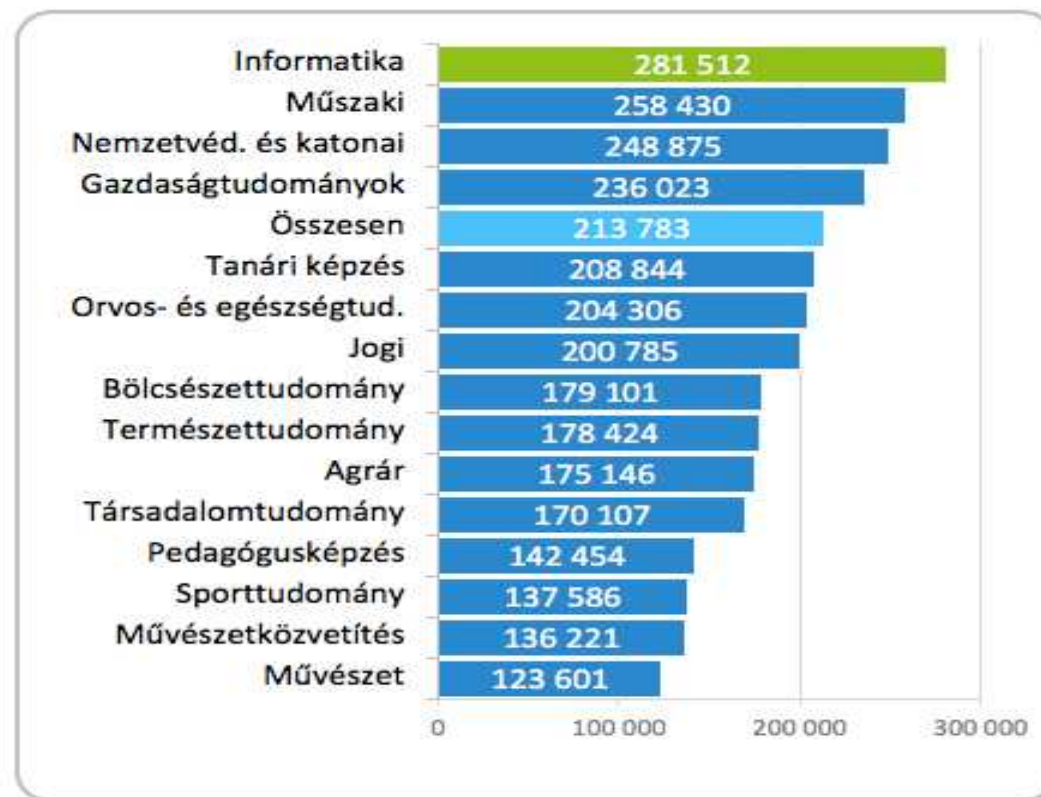


# A szoftver mint karrierlehetőség és üzlet Magyarországon

- <http://ivsz.hu/oktatas/kutatas-az-informatikus-munkaerohianyrol/>
- „A hazai informatikai munkaerő-piacon **22 000 új állás** tölthető be. Ez a bővülés a közvetett multiplikátor-hatásokkal összesen 72 000 embernek adhat munkát a nemzetgazdaságban. Ennek fő feltétele a szakember-kibocsátás mennyiségi növelése és minőségi fejlesztése. Az infokommunikációban közvetlenül több mint 160 000 fő dolgozik. Az ICT szektor 122 ezer főt alkalmaz, további több mint 40 ezren dolgoznak az ágazaton kívül informatikai munkakörben.
- A szektor a nemzetgazdasági foglalkoztatás 4,1%-át, a versenyszféra 6,2%-át adja közvetlenül. A havi bruttó átlagkereset az ICT Szolgáltatás szektorban a második legmagasabb az ágazatok között. A nemzetközi és hazai informatikai munkaerő-piacon nagymértékű a munkaerő-hiány, amely exponenciálisan nő. A hiány gátolja a növekedést, veszélyezteti a versenyképességet.”



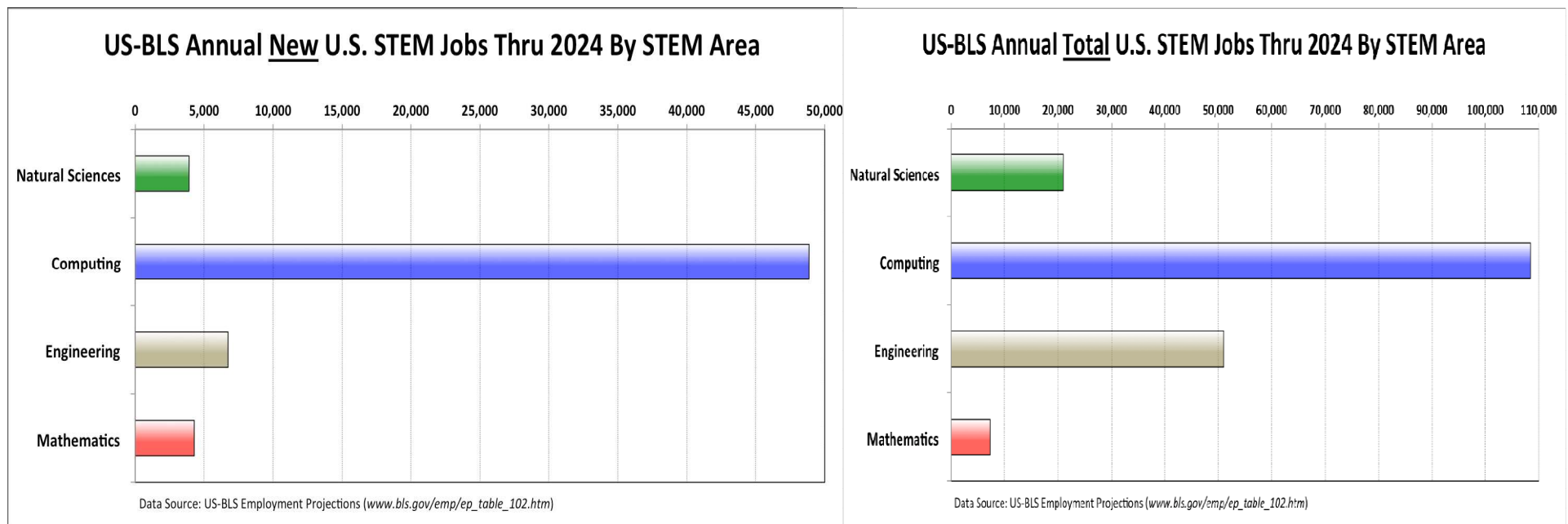
# A szoftver mint karrierlehetőség és üzlet Magyarországon



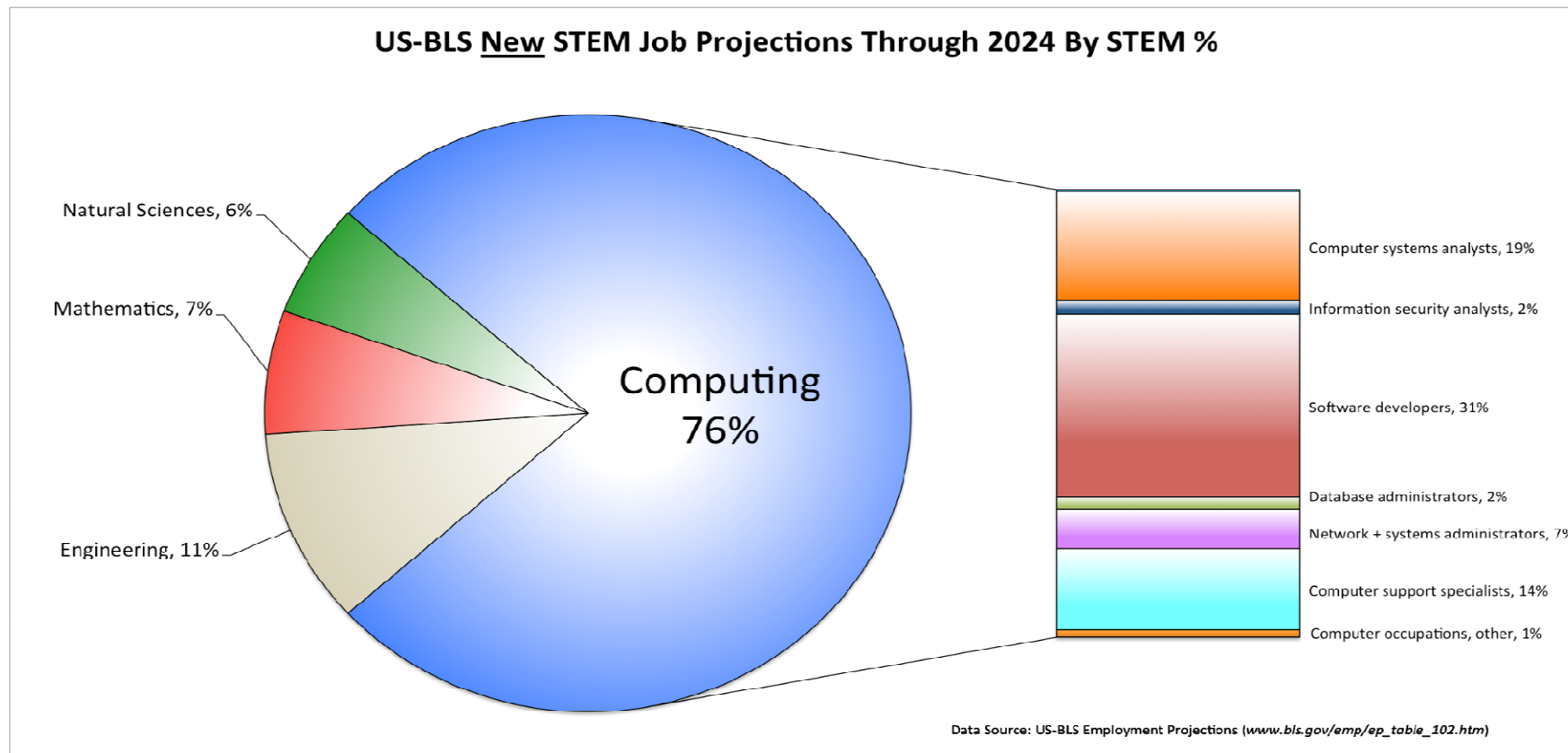
**A frissdiplomások bruttó havi összjövedelmének átlaga a vizsgált képzés képzési területe szerint (Ft/hó)**

Adminisztratív adatbázisok integrációja 2013, Educatio

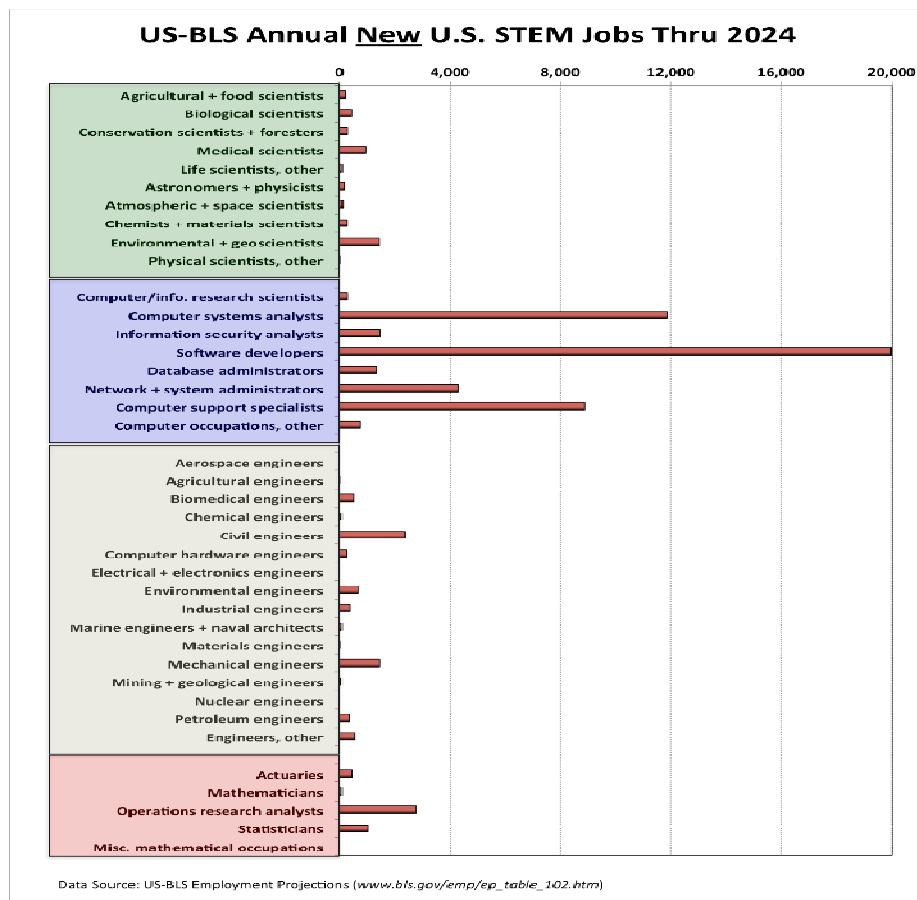
# Software as a career and as a business



# Software as a career and as a business



# Software as a career and as a business



Why will there be so many software development/engineering jobs?

- One reason is the mobile computing market.
- Another reason is that manufacturers are increasingly embedding computers into appliances like refrigerators, ovens, water heaters, and so on; creating the so-called Internet of Things.
  - [https://cs.calvin.edu/documents/computing\\_careers](https://cs.calvin.edu/documents/computing_careers)

# Problémák a szoftverrel kapcsolatban

- Hibás szoftver:

- Emberi életet veszélyeztethet
- pénzügyi csődöt eredményezhet
- világhátróhoz vezető folyamatokat indíthat el
- ...

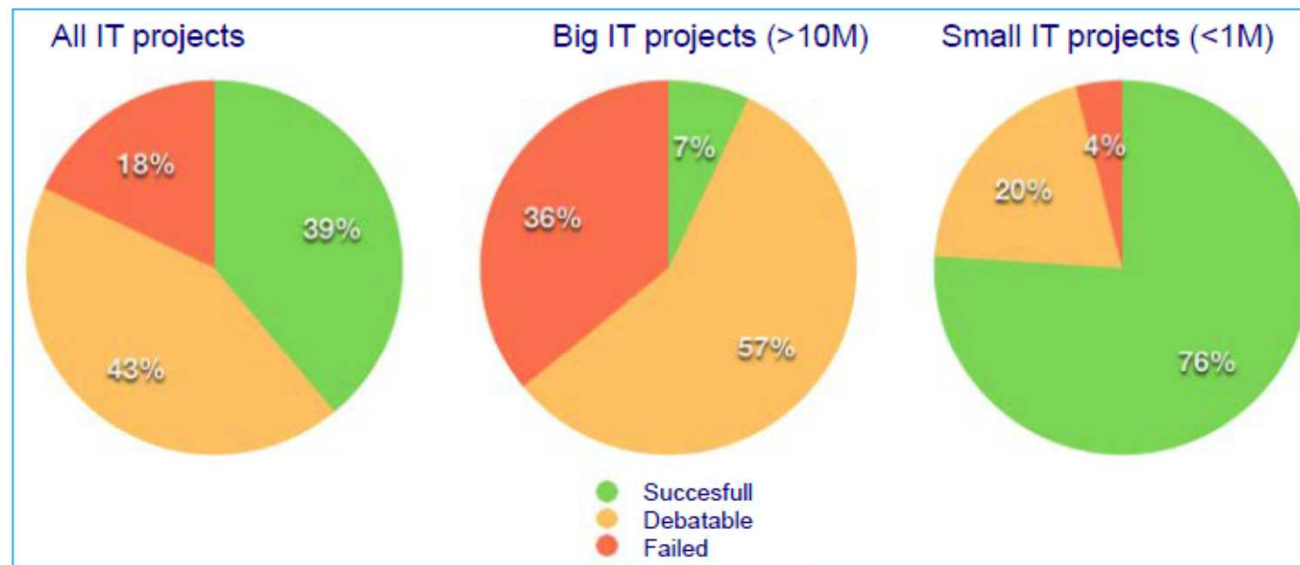
- És mégis...

- A tapasztalat azt mutatja, hogy

- a szoftverben hibák vannak
- a szoftverprojektek nagy része problémákkal küzd

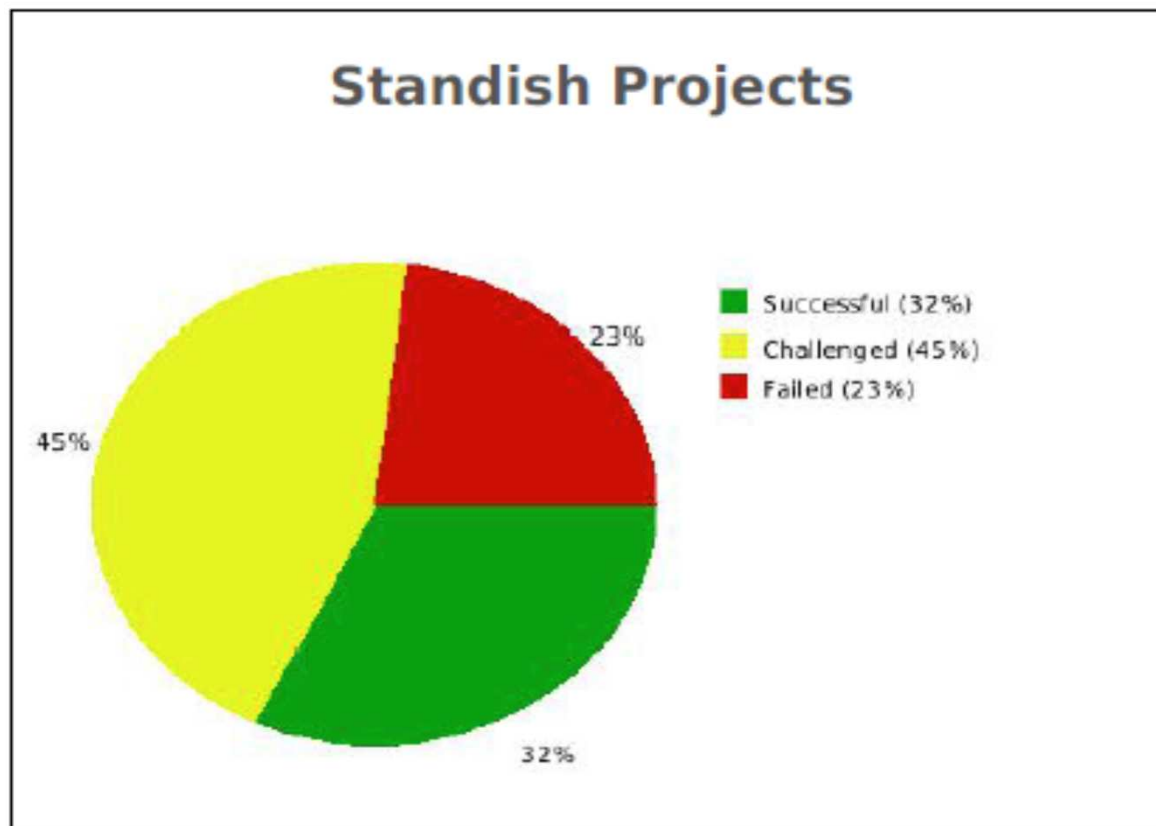


# Problémák az IT projektekkel kapcsolatban



The Standish Group International Inc., 2013

# Néhány számadat / Standish Group, Chaos Study (2015-ből)

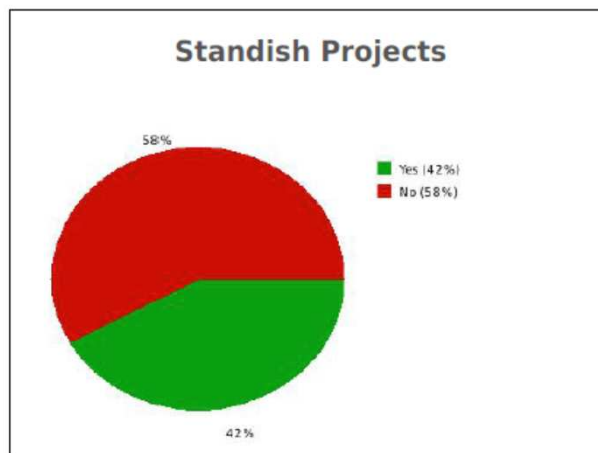


Measures projects against the CHAOS database for **on time, on budget, with satisfactory results**.

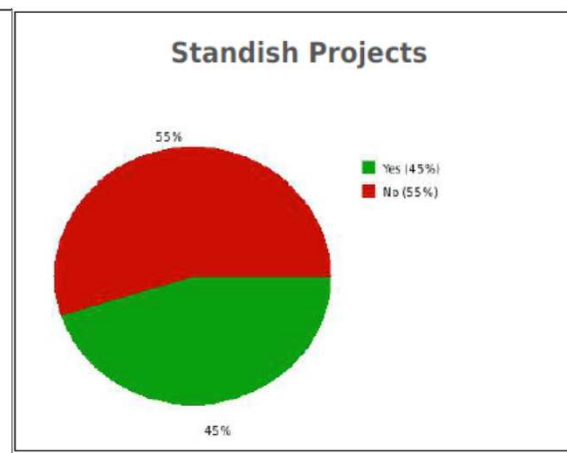
Challenged projects are projects that were completed but late, over budget, with unsatisfactory results. Failed projects were cancelled or not used.

# Néhány számadat / Standish Group, Chaos Study (2015-ből)

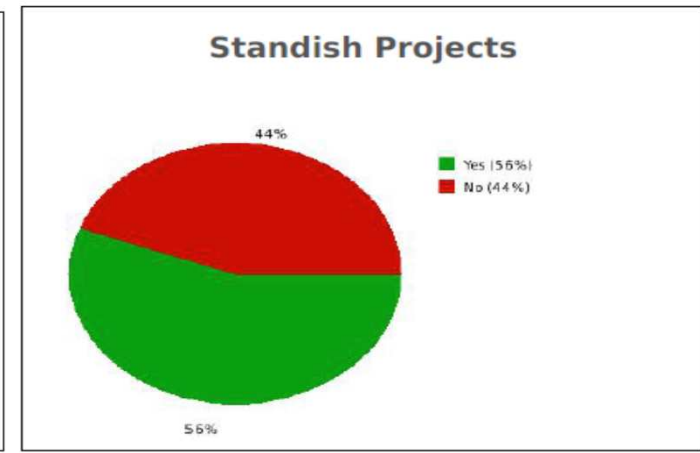
On Time



On Budget



On Target







# A szoftvernek „jónak” kell lennie

- „Normális elvárás”, hogy a szoftvertermék a legmagasabb szinten ...
  - ☐ Elégítse ki felhasználójának követelményeit
  - ☐ Rendeltetésének megfelelően működjék
- Korábban: 1 fejlesztő, 1 felhasználó
  - ☐ „a program fusson le, a várthoz hasonló eredményekkel”
- Később: több „alkotó”, több felhasználó
  - ☐ kevesebb kapacitás igénye
  - ☐ érthetőség, „olvashatóság”, hordozhatóság
  - ☐ felhasználó-barátság, tanulhatóság
- Manapság:
  - ☐ Hatékonyság, megbízhatóság, hibamentesség, adatok biztonsága, adatvesztés nélkül legyen képes újraindulni



# De: a szoftverben vannak hibák...

- Mert:

- ☐ A szoftvert emberek írják, és “tévedni emberi”
- ☐ Komplex feladatok elvégzésénél az emberek követnek el hibákat, ez elkerülhetetlen

- Tapasztalt programozók átlagban minden 7-10 forrássorban vétenek 1 hibát
- Ezen hibák felét a gépnyelvre történő fordításkor kijavítják
- A tesztelés során további hibák is kijavulnak, de a hibák 15%-a bent marad az ügyfélnek való átadáskor

(Watts Humphrey: „What if your life depended on software?” Előadás a 2000-s EuroSPI konferencián, Koppenhága, 2000. április)



# Hibák a szoftverben

## ■ Okozóik:

- ☐ 1. Az emberi tényező
- ☐ 2. Kommunikációs hiba ←
- ☐ 3. Irreális(an rövid) fejlesztési idő ←
- ☐ 4. Rossz tervezés
- ☐ 5. Rossz kódolási gyakorlat
- ☐ 6. Verziókezelés hiánya
- ☐ 7. Hibás, harmadik féltől származó eszközök
- ☐ 8. Képzett tesztelők hiánya
- ☐ 9. Változtatások az utolsó percben
  - <http://www.softwaretestingtricks.com/2008/12/why-are-bugsdefects-in-software.html>



# Hibák a szoftverben

- Kommunikációs hibák (kommunikáció okozta hibák)
  - A követelményekkel kapcsolatban
    - A követelmények összegyűjtésére, egyeztetésére, dokumentálására nem szánnak elég időt
    - Egy fejlesztő megpróbál másik fejlesztő által fejlesztett kódot módosítani
- Irreálisan rövid fejlesztési idő
  - Nagyon gyakran a szoftvert irreálisan rövid idő alatt vállalják kifejleszteni, kevés erőforrással. Nagy valószínűséggel ilyen esetekben a követelmények betartását nem veszik eléggé szigorúan (mert lényegesebb, hogy határidőre átadjanak „valamit”).



# Hibák a szoftverben

- A közelmúltban (2017 augusztus) hibák voltak a BKK rendszerében, melyet a T-Systems Hungary fejlesztett. Egy fiatalembert, aki a hibát reprodukálta, majd jelentette a BKK-nak, rendőri intézkedés alá vontak.
- Hol hibázott a T-Systems:
  - „Havi bérlet 50 Ft-ért”
  - Plain text jelszavak
  - E-mail-ben visszaküldött jelszó
  - Elérhető váltak felszanalói profilok
  - Ellenőrzés nélkül kitörölt felhasználói regisztráció
  - SSL DROWN sebezhetőség
- Hogyan reagált a T-Systems és a BKK, amikor kiderültek a hibák?
  - Az esetről lásd a holnapra feltett leírást



# 1. példa

## ■ Feladat:

- ☐ Írjon egy programot, mely kiszámolja  $n$  valós szám átlagát, és a szórást.



# 1. példa:

## ■ Egyszerűnek tűnik, ám mégis:

□ Több döntést is meg kell hozni; több elemet jól meg kell érteni, mielőtt nekifogunk programozni!  
Mit kérdezne meg?

□ Pl:

■ Milyen programozási nyelvben írom a programot?

□ Abban, amelyik a legjobb az adott esetre.

■ Honnan „jönnek” a számok?

□ „A program az  $n$  valós számot be tudja olvasni a képernyőről, egy file-ból vagy egyéb forrásból.”

■ Milyen megoldást használjak?

□ Használjon láncolt listát az  $n$  valós szám tárolására. Ha szükséges, akkor az adatok tárolására használható egy statikus vagy dinamikus tömb, adatbázis vagy egyéb adatszerkezet.

# 1 példa:

## ■ Mi az „átlag” és a „szórás”?

- Átlag :
- Egy számsor átlaga: a számok összege elosztva azzal, ahány számunk van.
- Szórás: A szórás megmutatja, milyen messze esnek egy lista számai az átlaguktól. A számok többsége nagyjából egy szórásnyi távolságon belül van az átlagtól. Csak nagyon kevés esik két vagy három szórásnyi távolságnál messzebb.

## ■ A szórás kiszámításának lépései:

1. Kiszámítjuk az adatok számtani közepét.
2. Kiszámítjuk az adatok eltérését a számtani közepétől (adat - számtani közép)
3. Vesszük ezeknek az eltéréseknek a négyzetét.
4. Kiszámítjuk ezeknek az "eltérés négyzeteknek" a számtani közepét.
5. Végül ebből négyzetgyököt vonunk.

The formula for calculating the standard deviation is

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - x_{avg})^2}{n-1}}$$



# 1. példa:

## ■ Hogyan kell kinéznie a feladatmegoldás folyamatának? (Legalább...):

- Olvassuk el a feladatkírást
- Értsük meg a feladatot
- Tervezzük meg a megoldást
  - Használjunk fel korábbi tapasztalatokat, meglévő megoldásokat, rendelkezésre álló tervezési és programozási elemeket
- Implementáljuk a tervet (kódoljunk)
- Teszteljük a megoldást
  - Javítsuk a hibákat
  - Újratesztelés
  - (amíg nem találunk már több hibát)
- Adjuk át a megoldást

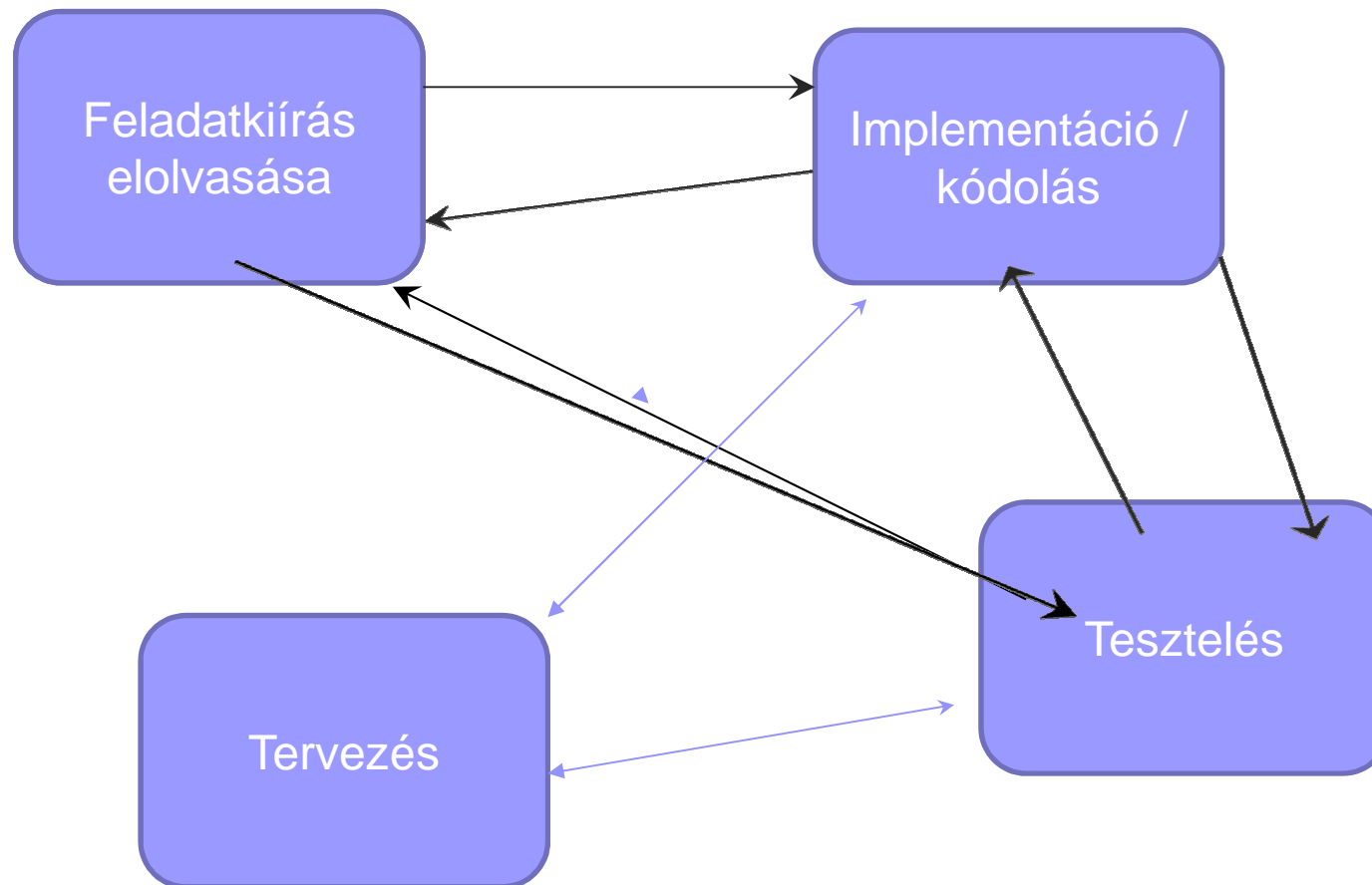


## 1. példa:

- Milyen lesz a feladatmegoldás folyamata ? Mire nem szeretne majd időt vesztegetni? (legalább):
  - Olvassuk el a feladatkiírást
  - Értsük meg a feladatot
  - Tervezzük meg a megoldást
    - Használjunk fel korábbi tapasztalatokat, meglévő megoldásokat, rendelkezésre álló tervezési és programozási elemeket
  - Implementáljuk a tervet (kódoljunk)
  - Teszteljük a megoldást
    - Javítsuk a hibákat
    - Újratesztelés
    - (amíg nem találunk más több hibát)
  - Adjuk át a megoldást

# 1. példa:

Mi fog „történni”?



# 1. példa:

- Tudja, mennyi idő alatt végzi el a feladatot?
  - Van erről sejtése, legalább?
  - Jobb lenne, ha lennének korábbi adatok hasonló feladatról!
    - Hogy legyen ilyen listája, **saját magának** kell létrehoznia!
  - Mennyire jó kódot írt?
- Van erről fogalma?
  - Miért van vagy miért nincs?



# 1. példa:

- Hogyan kell kinéznie a feladatmegoldás folyamatának?
  - Sokféleképpen nézhet ki, jó, ha van útmutató!

PSP1 Development Script

<b>Purpose</b>	To guide the development of small programs
<b>Entry Criteria</b>	<ul style="list-style-type: none"><li>- Requirements statement</li><li>- Project Plan Summary form with estimated program size and development time</li><li>- Time and Defect Recording logs</li><li>- Defect Type standard and Coding standard</li></ul>

Step	Activities	Description
1	Design	<ul style="list-style-type: none"><li>- Review the requirements and produce a design to meet them.</li><li>- Record in the Defect Recording log any requirements defects found.</li><li>- Record time in the Time Recording log.</li></ul>
2	Code	<ul style="list-style-type: none"><li>- Implement the design following the Coding standard.</li><li>- Record in the Defect Recording log any requirements or design defects found.</li><li>- Record time in the Time Recording log.</li></ul>
3	Compile	<ul style="list-style-type: none"><li>- Compile the program until there are no compile errors.</li><li>- Fix all defects found.</li><li>- Record defects in the Defect Recording log.</li><li>- Record time in the Time Recording log.</li></ul>
4	Test	<ul style="list-style-type: none"><li>- Test until all tests run without error.</li><li>- Fix all defects found.</li><li>- Record defects in the Defect Recording log.</li><li>- Record time in the Time Recording log.</li><li>- <i>Complete a Test Report template on the tests conducted and the results obtained.</i></li></ul>

<b>Exit Criteria</b>	<ul style="list-style-type: none"><li>- A thoroughly tested program that conforms to the Coding standard</li><li>- <i>Completed Test Report template</i></li><li>- Completed Time and Defect Recording logs</li></ul>
----------------------	---

## 2. példa:

- Egy kisvállalkozást szeretne indítani, melynek keretében Interneten kirándulójegyeket értékesítene azok számára, akik Roatan szigete mellett szeretnék a bújárákórást kipróbálni, a közép amerikai nagy korallzátonynál.
- Megteheti, hogy „csak kódol”?
  - Miért igen, vagy miért nem?





## 2. példa:

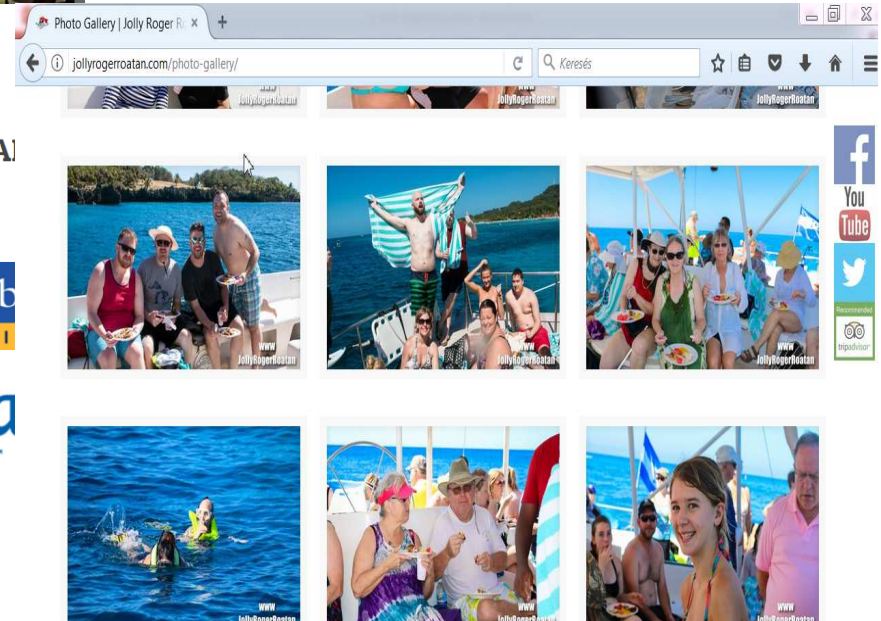


- Szükséges / elengedhetetlen, például (nem fontossági sorrendben):
  - ☐ Lehetővé tenni, hogy az érdeklődők kérdéseket tegyenek fel
  - ☐ Lehetővé tenni, hogy bankkártyával fizessenek
  - ☐ Interneten keresztüli fizetést lehetővé tenni
  - ☐ Vonzó weblapot készíteni
  - ☐ A weblapnak „folyamatosan” elérhetőnek kell lennie
  - ☐ Megállapodni olyan cégekkel, társaságokkal, amelyek az ügyfeleket az Ön honlapjára irányítják (hajótársaságok, utazási irodák, búvároktatók...)
  - ☐ Elégedett ügyfeleket „összegyűjteni” (és közzétenni) – ők segítik majd az üzlet továbbvitelét ...





**GET READY FOR A ROATAN CRUISE EXPERIENCE OF A LIFE TIME...  
EXCURSIONS DEPARTING FROM MAHOGANY BAY OR PORT OF ROATAN!  
BOOK WITH OUR SHORE EXCURSION PARTNERS**





# 3. példa:

- Ön a Lufthansa Systems Hungary-nál dolgozik. Csapata azt a feladatot kapta, hogy a légitársaság meglévő „On-line check-in” rendszerét egészítse ki olyan modullal, amelyik lehetővé teszi, hogy az Air China légitársasággal közösen szervezett járatokra on-line módon be lehessen jelentkezni.

Lufthansa online check-in available for:

- Lufthansa worldwide
- Austrian between Germany and Austria
- LOT for departures in Germany (with the exception of Hamburg and Stuttgart)
- Adria Airways and Croatia Airlines for departures from Germany
- SAS for departures from Germany (except for Dusseldorf, Hamburg, Berlin Tegel)

No online check-in is available for departures from:

Country	Airport
France	Bastia/Corsica (BIA)
Italy	Lamezia Terme (SUF)

For flights by all other airlines that are not specified here, as well as for so-called codeshare flights, i.e. flights which are operated by another airline but booked on a Lufthansa flight number, you cannot check in using Lufthansa online or mobile check-in. You can recognise codeshare flights with a Lufthansa flight number by the identifier 'operated by' in your booking confirmation.

Website Feedback



## 3. példa:

- Megteheti, hogy „csak kódol”, mint az 1. példában?
- Miért igen , vagy miért nem?
- Figyelembe kell vennie (minimum):
  - A meglévő rendszert / rendszereket
    - Korlátozásokat
      - Architektúrát
      - Interfészeket
      - Adatbázisokat
      - Felhasználói kapcsolódási lehetőségeket
      - ...




# Példák – Következtetés (szerűség)

- Minél nagyobb a feladat / projekt, annál több elemet kell észben tartani
- A sok elem között könnyű elvesztődni
- Módszerek, számarvezetők, útmutatók, minták ... **szükségesek!**
- Egyszerűen: nem engedheti meg magának senki, hogy „rossz” szoftvert fejlesszen!



# Probléma...

- Lehet „nulla hibával rendelkező” szoftvert fejleszteni, csak iszonyúan drága!
  - „It is possible to achieve zero defects but it is also costly. NASA was able to achieve zero defects for the Space Shuttle Software, but **at a cost of thousands of dollars per line of code**. If people will die because there are bugs in the software then that kind of cost makes sense. Most projects simply cannot afford the same level of testing as NASA.”
  - Source: <http://labs.sogeti.com/how-many-defects-are-too-many/>



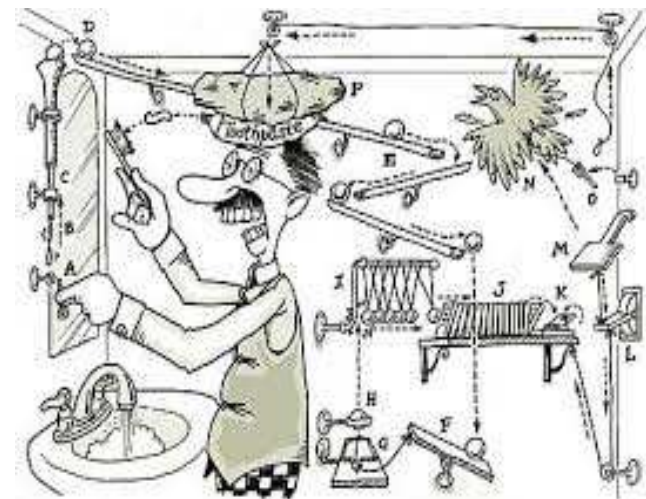
# Mitől „más” a szoftver?

## Más-e...?

- Nincs fizikai „léte”
- Előállításának sajátos életciklusa van
- Nagyon gyorsan változik
- Alkalmazkodnia kell a mindig újabb hardver-szoftver környezethez
- A felhasználónak nehézségei vannak elvárásai megfogalmazásában (ha korábban nem használt szoftvert)
- Nagyon magasak a felhasználónak a szoftverrel kapcsolatos elvárásai
- Nehéz a szoftver minőségi jellemzőit meghatározni
- Nehéz a minőségi jellemzők értékét mérni
- Pontos másolat készítése, csomagolás nem jelent gondot

# A szoftverfejlesztés...

- Komplex feladat
- Projektben végezzük (amely átnyúlhat kontinenseken, országokon ...)
- Szükségessé tesz sokféle szakterülethez kapcsolódó tudást, szakértelmet
  - Üzleti terület (adatok, folyamatok, emberek, törvények stb.)
  - Kommunikáció – különböző érdekelt felekkel
  - Csapaton belüli kommunikáció
  - Projekt tervezés, követés, vezérlés
  - Szoftverfejlesztési módszerek és eszközök
  - ...





# Mit mond a józan ész?



A COMPLEX TASK IS  
LIKE AN UNPEELED ORANGE...



... SO BREAK IT DOWN  
INTO SUBTASKS

# Mit tehetünk mi, emberek, akik szoftverfejlesztéssel foglalkozunk?

- **Legyünk nyitottak!** Mindig tanuljunk új dolgokat!
- Tanuljunk másoktól! Dolgozzunk együtt másokkal!
- Kövessünk létező jó gyakorlatokat!
  - Használjunk fel módszertanokat, modelleket, szabványokat – de úgy, ahogyan azok a saját esetünkben hasznosak!
- Osszuk meg a tapasztalatainkat!
- Tanuljunk mások tapasztalatából!
- Fegyelmezetten fejlesszünk szoftvert!





# Szoftvermérnökök legyünk, ne csak kódoljunk!





# Mi a szoftvermérnökség / szoftverfejlesztés / Software Engineering?

- David Parnas – több személy által, több verziójú szoftver fejlesztése
- Sommerville – “egy mérnöki diszciplína, mely magas minőségű szoftver-rendszerek költség-hatékony fejlesztésére törekszik”
- Pfleeger – számítógépes eszközök alkalmazása problémák megoldására.
- CMU/SEI-90-TR-003 – “egyfajta mérnökség, amely a számítástechnikai tudomány elveit és matematikát alkalmaz szoftveres problémák költség-hatékony megoldására.

From: Essentials of Software Engineering, Fourth Edition  
Frank Tsui, Orlando Karam, and Barbara Bernal  
Copyright © 2018 by Jones & Bartlett Learning, LLC, an Ascend  
Learning Company



# Mi a szoftvermérnökség / szoftverfejlesztés / Software Engineering?

- Módszeres, fegyelmezett, számszerűsíthető megközelítés, mely szoftver fejlesztésére és karbantartására irányul; azaz: a mérnöki tudományok szoftverre történő alkalmazása.
  - [IEEE Std 610.12-1990]



# Mi a szoftvermérnökség / szoftverfejlesztés / Software Engineering?

- A szoftver fejlesztéséhez és karbantartásához kapcsolódó széles terület, mely kapcsolódik a következőkhöz / érinti a következőket:
  - Folyamatok
    - Fejlesztési / műszaki, menedzsment, támogató, szervezeti ...
    - Módszerek / módszertanok (folyamatok csoportja, melyek bizonyos szempontok szerint összetartoznak)
  - Termékek
    - Jellemzők, mérőszámok...
  - Emberek
    - Képességek , csapatmunka...
  - Projektek



# IEEE-CS/ACM Version 5.2 Report

- 8 principles\*\* for ethics and professional practices in software engineering
  - Software engineers shall act consistently with the public interest.
  - Software engineers shall act in a manner that is in the best interest of their client and employer, consistent with the public interest.
  - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
  - Software engineers shall maintain integrity and independence in their professional judgment.
  - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
  - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
  - Software engineers shall be fair to and supportive of their colleagues.
  - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

From: Essentials of Software Engineering, Fourth Edition  
Frank Tsui, Orlando Karam, and Barbara Bernal

\*\* Under each of these major principles, there are further sub-principles.



# Néhány egyszerű viselkedési szabály

- Tiszteljük a többieket!
- Legyünk korrektek, becsületesek!
- Legjobb képességeink szerint végezzük el a feladatot!
- Tartsuk tiszteletben a törvényt!

From: Essentials of Software Engineering, Fourth Edition  
Frank Tsui, Orlando Karam, and Barbara Bernal



# A "Szoftvermérnöki szabály"

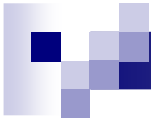
- **Nincs** „Abszol-út”, nincs mindenki által elfogadott szoftverfejlesztési / szoftvermérnöki megközelítés, szabály vagy törvény!
- Ami van: sok megközelítés, melyek mindegyike a szoftverfejlesztés különböző aspektusaival foglalkozik
- Mindegyik megközelítés jó lehet bizonyos körülmények között
  - Sokkal valószínűbb, hogy több megközelítésből tudjuk összerakni a számunkra legkedvezőbb, saját megoldást!
- **Vannak**
  - Általánosan érvényes „tudás-elemek”, amelyeket meg kell jegyezni és alkalmazni kell!



# Ne feledjük!

- Nem vagyunk a szoftverfejlesztés pionírai;  
rendelkezünk 50 év tapasztalatával; használjuk fel!
  - ☐ Nem kell mindig feltalálni a spanyolviaszt!
  - ☐ **De** meg kell ismernünk a meglévő megközelítéseket, és  
meg kell tanulnunk megfelelően alkalmazni őket!
- A szoftvermérnökség **sokkal több** a kódolásnál!
- A szoftver minősége **sokkal több** a kód minőségénél!!!
- Jó minőségű termékek elkészítéséhez szükséges a  
fegyelmezett munkavégzés !
  - ☐ Határozzuk meg folyamatainkat!
  - ☐ Alkalmazzuk a meghatározott folyamatokat!
  - ☐ ... **akkor is alkalmazzuk folyamatainkat, ha éppen pánikba estünk, mert  
szorít az idő!**





# Lehetőségek...

- Egy lehetséges kockázatmérséklési tevékenység ahhoz, hogy jó minőségű szoftverterméket állítsunk elő az, ha **modellezzük** a szoftverfejlesztést.
- Meglepőnek tűnhet...

de...

- működik!



# A kezdetek és a múlt

- „Szoftverfejlesztés” : megszületett Garmisch-ban, 1968-ban.

Report on a conference sponsored by the NATO SCIENCE COMMITTEE  
Garmisch, Germany, 7th to 11th October 1968

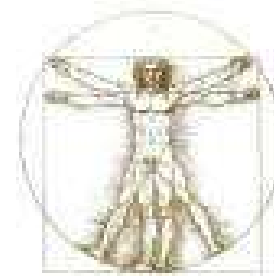
□ Kb. 50 éves.

- Az elmúlt 40-50 évben a „szoftver” életünk igen fontos eleme lett
- Nagyon gyors fejlődésen ment át!
- A szoftver(fejlesztés) fejlődése erőteljesen befolyásolta az informatikától különböző tudományágakat, a művészetet ... az egész életet!



# A szoftverfejlesztés modellezése

- Már korán elkezdődött
- Többféleképpen is próbálkoztunk...
  - ...aktuális tudásunkat tükrözve
  - ...az aktuális eszközkészletet használva
  - ...testre szabva
  - ...teljességében
  - ... részleteiben
  - ...az aktuális pillanatban fontosnak vélt elemekre figyelve
  - ...bizonyos elemeket részletezve
  - ...és még tovább részletezve...
  - ...



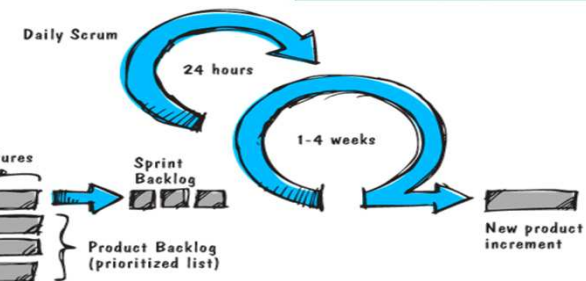
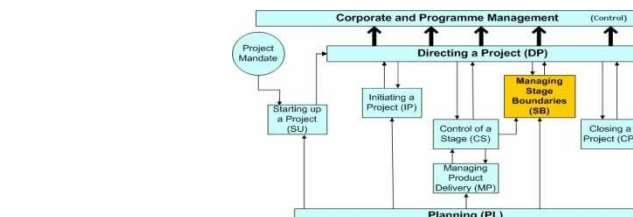
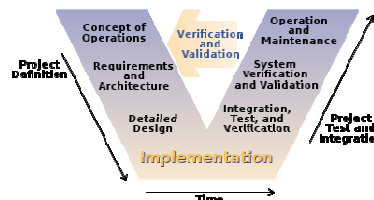
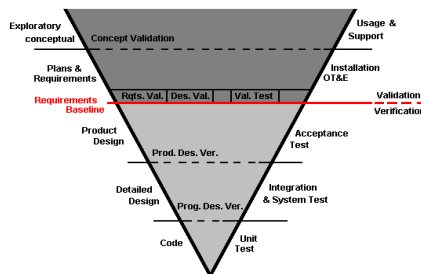
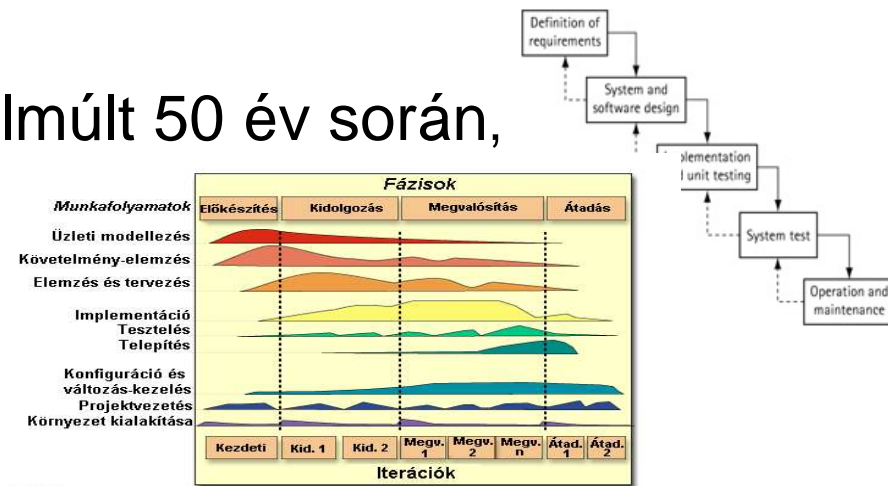


# A szoftverfejlesztés modellezése

- A szoftver jó minősége mindig is fontos volt, ám nem lehet „általánosan” meghatározni
- Talán emiatt is...
  - „Mindent” modelleztek az elmúlt ~ 50 és során, ami kapcsolódik a szoftverfejlesztéshez :
    - Terméket
    - Folyamatokat
    - Erőforrásokat
    - Definíciókat és metrikákat határoztak meg
    - ... és még sok mást...

# A szoftverfejlesztés modellezése

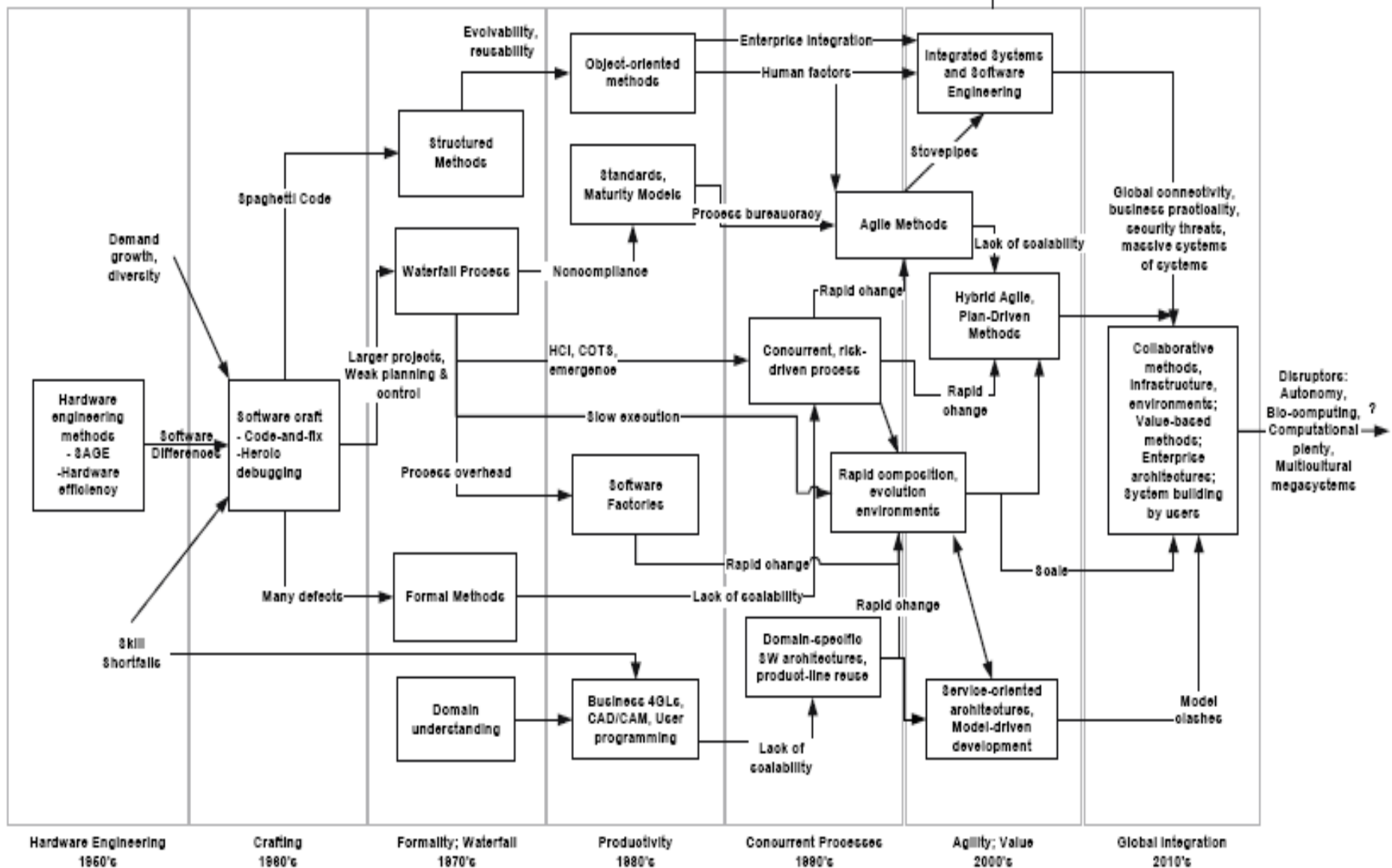
- „Mindent” modelleztek az elmúlt 50 év során,
- **folyamatokat** nagy előszere
  - PM folyamatok
  - Támogató folyamatok
  - Szervezeti folyamatok
  - Fejlesztési folyamatok
- ✦ Életciklus modellek





# Folyamatok a szoftverfejlesztésben

- Következő előadásainkban bemutatunk néhány folyamatot és folyamatmodellt, amelyek fontosak a szoftvermérnökök számára
- A folyamatokat a következő csoportokba szokás sorolni:
  - Fejlesztési / műszaki / mérnöki folyamatok
    - Követelményfejlesztés, tervezés, kódolás, tesztelés, karbantartás
  - Projektmenedzsment folyamatok
    - Projekttervezés, Projektkövetés és -vezérlés, kockázatkezelés
  - Támogató folyamatok
    - Konfigurációmenedzsment, minőségbiztosítás, mérés
  - Szervezeti folyamatok
    - Stratégiai tervezés, szervezeti képzés, folytonos folyamatfejlesztés
- Azt a gondolatot, hogy a szoftvercégek által alkalmazott folyamatokat csoportokba sorolják, az ú.n. „folyamatfejlesztési modellek” erőteljesen támogatják („Software Process Improvement” (SPI) models)



Boehm, B. (2006). A view of 20th and 21st century software engineering International conference on Software engineering (ICSE). <http://isr.uci.edu/icse-06/program/keynotes/boehm.html>



# Mi a folyamat?

- Egy folyamat egy lépéssorozat és döntések sorozata, melyek a munka elvégzéséhez vezetnek, és megadják az elvégzés módját .

- <https://www.processmodel.com/blog/what-is-a-process/>

- Akciók és műveletek sorozata, melyek eredményre vezetnek. Különösen a gyártásban használt fogalom.

- <https://www.merriam-webster.com/dictionary>

- Egymással kölcsönhatásban álló tevékenységek sorozata, mely bemeneteket kimenetekké alakít, egy adott cél elérésére.

- Ez a meghatározás konzisztens a következő szabványokban megjelenő meghatározásokkal: ISO 9000, ISO 12207, ISO 15504, and EIA 731.

- Forrás:

- [http://cmmi.institute.com/system/files/models/CMMI\\_for\\_Development\\_v1.3.pdf](http://cmmi.institute.com/system/files/models/CMMI_for_Development_v1.3.pdf)



# Folyamatok elemei

(Curtis et al., 1992)	(Kellner et al., 1999)	(Bhuta et al., 2006)	(C. J. Pardo et al., 2009)	CMMI v1.3 (2010)	Derived Process Element
process step	key activity and task	project activity	activity, special activity, task	activity	activity
agent	resources	–	resource	–	resource
role	–	–	role	role	role, responsibility
artefact	primary objects	input and output (indirectly)	product	input, output	artifact
–	activity dependency, flow of object, sequence, loop, structural interdependency	logical dependency	–	–	element relation
–	–	input and output interface, precondition, postcondition	objective, process, process group, process category, indicator, measurement, tool	purpose, measures, verification step, entry and exit criteria	further process elements

Zádor Dániel Kelemen: Process based unification for multi-model software process improvement. PhD thesis. 2013. <https://pure.tue.nl/ws/files/3583930/741509.pdf>



# A folyamatok modellezése egy „legjobb gyakorlat”

- **“Alapvetően minden modell hibás; némelyikük hasznos.”**

George E.P. Box

- A modelleknek három fontos tulajdonságuk van:
  - ☐ Reprezentáció: a modellek a valóságot tükrözik
  - ☐ Redukció: a modellek redukálják a valóságot / csak bizonyos aspektusaival foglalkoznak
  - ☐ Pragmatikus tulajdonság: a modelleket bizonyos céllal hozzák létre



# Mi az a folyamatmodell?

- Egy olyan keretrendszer, amelyben hasonló folyamatok egy csoportját írják le, pl. egy teszt fejlesztési modell.

- <http://www.astqb.org/glossary/>

- Annak leírása, hogy milyen tevékenységeket és milyen sorrendben kell végrehajtani ahhoz, hogy a kitűzött célt elérjük.

- Essentials of Software Engineering. Fourth Edition. Jones & Bartlett Learning, [www.jblearning.com](http://www.jblearning.com)



# Egy folyamatmodell leírása

- A „leírás” többféle formában történhet.
  - Szövegesen, diagramokkal, sajátos jelölések használatával, mindezek kombinációjával
- Sokféle folyamatmodellezési nyelv, eszköz, szabvány ...van
  - Ezek segítik a különböző típusú folyamatok bizonyos célból történő modellezését (üzleti, követelmények, tervezés, tesztelés...)
  - Különböző részletezettségűek (rendszer szintű... komponens szintű...funkció szintű...)
  - Különböző elemekre koncentrálnak, melyeket adott esetekben fontosnak tekintenek (emberek, szerepkörök, feladatok, kommunikáció stb. )

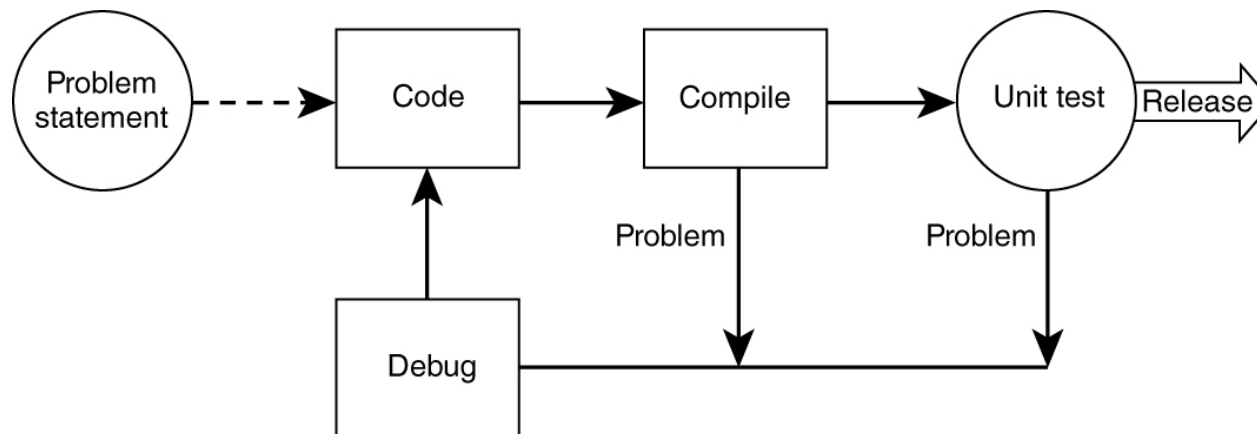
# Szöveges folyamatleírás

Process element (Table 17)
Process name
Parent process
Process description
Activity
Role, responsibility
Data object
Element relations
Entry/exit criteria

## 5.4.4.4. Decision: Are there further processes to be improved?

Activity ID	4.4
Parent ID	4 Validation
Activity name	Decision: Are there further processes to be improved?
Activity description	Y: in case if there is one or more further process to be improved then select the next process (activity 1.2) N: end of the PBU process.
Inputs	List of processes to be improved
Outputs	Y/N
Roles/ responsibilities	SEPG team

# Egy „egyszerű és ismerős” folyamat



Essentials of Software Engineering. Fourth Edition.

Jones & Bartlett Learning,  
[www.jblearning.com](http://www.jblearning.com)

# Folyamatok modellezésének lehetőségei

## UML aktivitás diagramok elemei

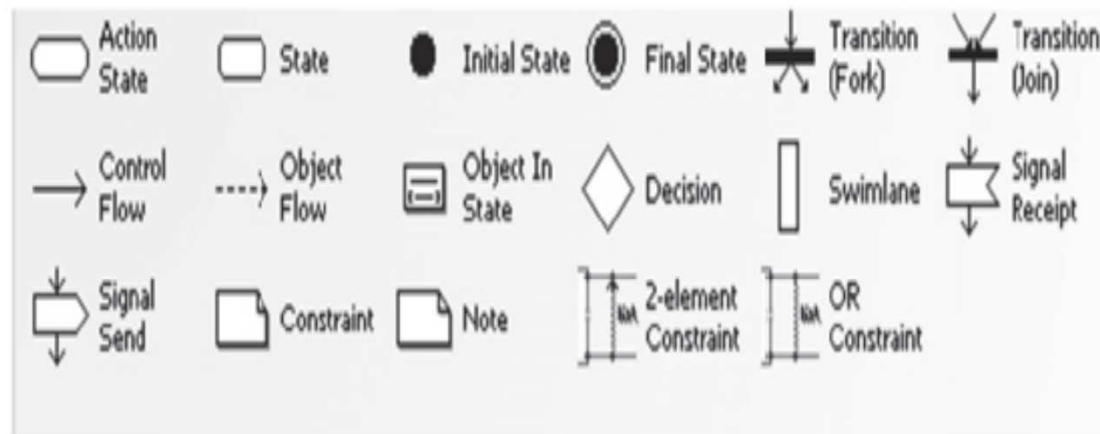


Figure 17 – Elements of UML activity diagrams, source: MS Visio

# Folyamatok modellezésének lehetőségei

## EPC diagramok

(Event Driven Process Chain)

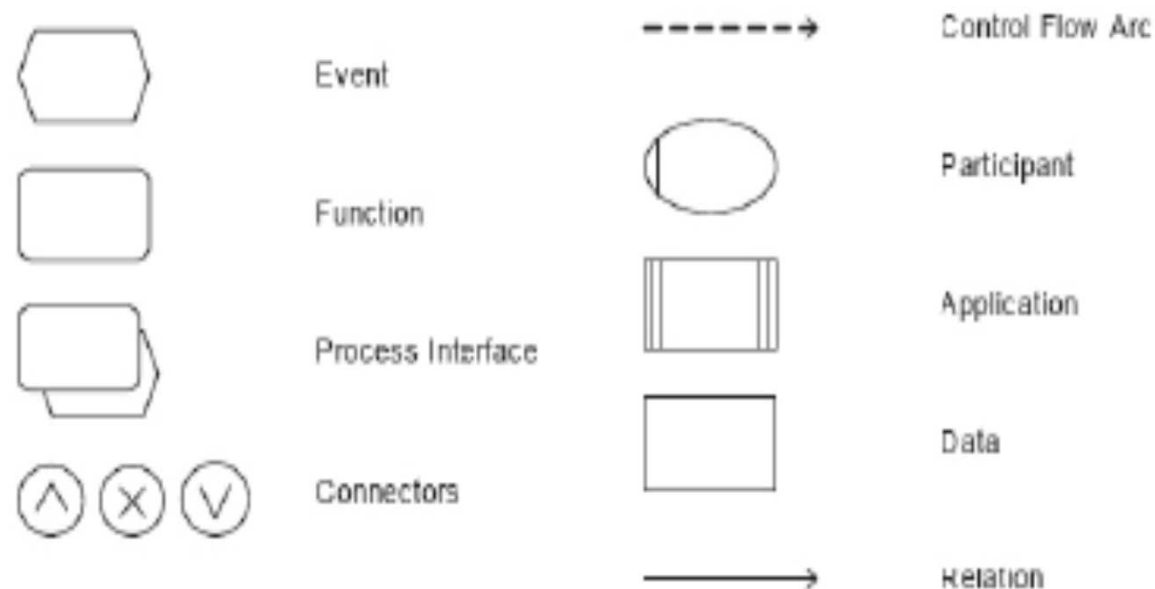
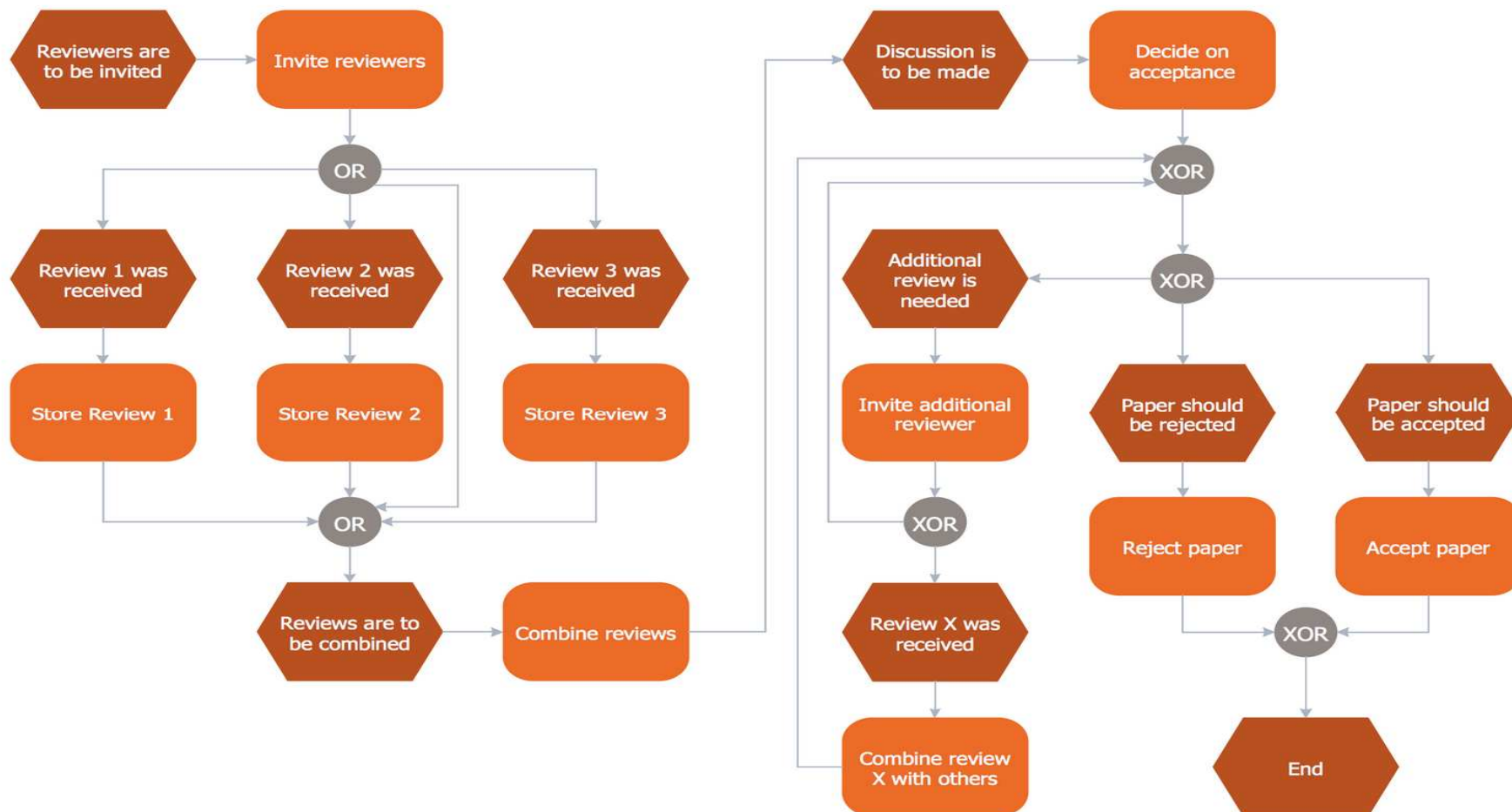


Figure 16 – Symbols of EPC diagram, source: (Mendling & Nüttgens, 2005)



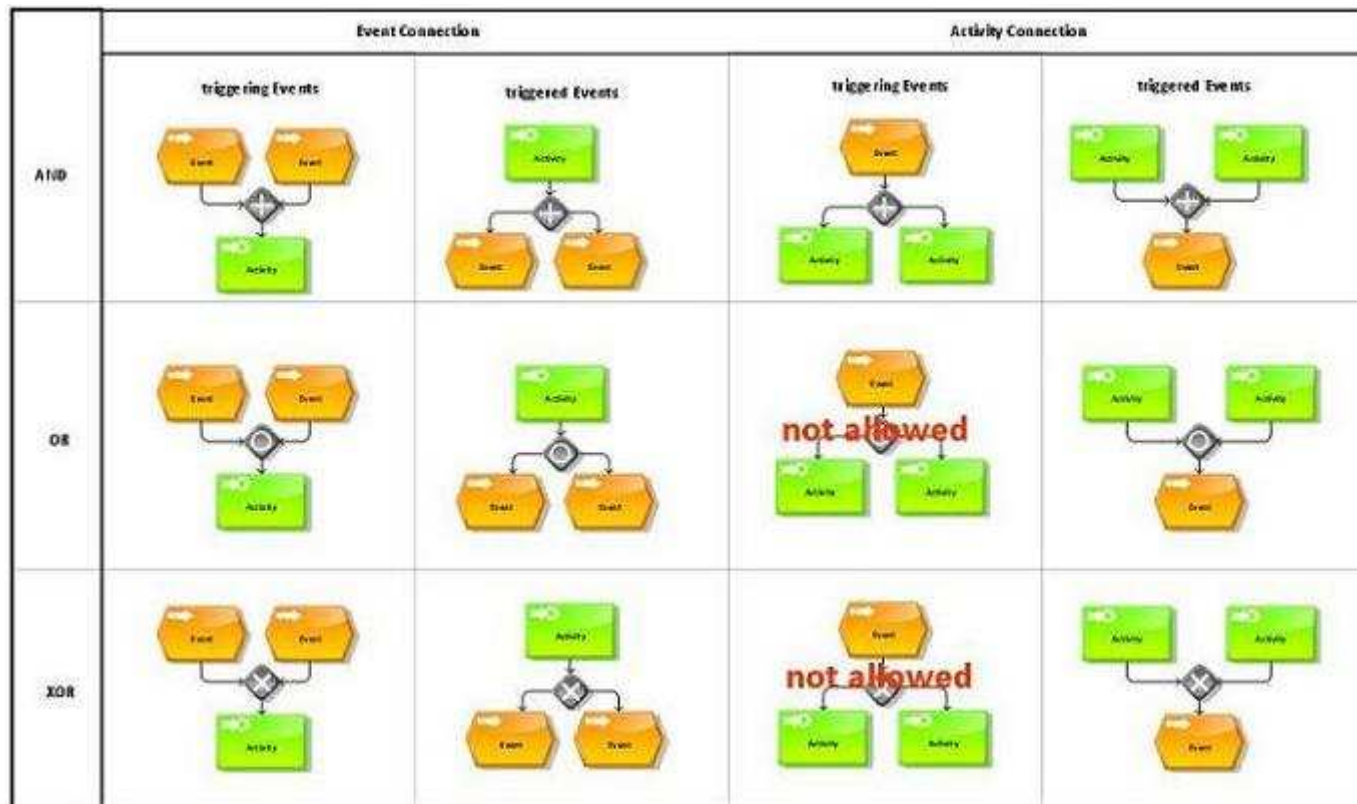
# Folyamatmodellek (példák)

## EPC modell



# Folyamatmodellek (példák)

## EPC (Event Driven Process Chain) modellezés szabályai



<http://www.ariscommunity.com/users/rbaureis/2010-03-22-basic-rules-epc-modelling>

# Folyamatok modellezésének lehetőségei

## BPMN

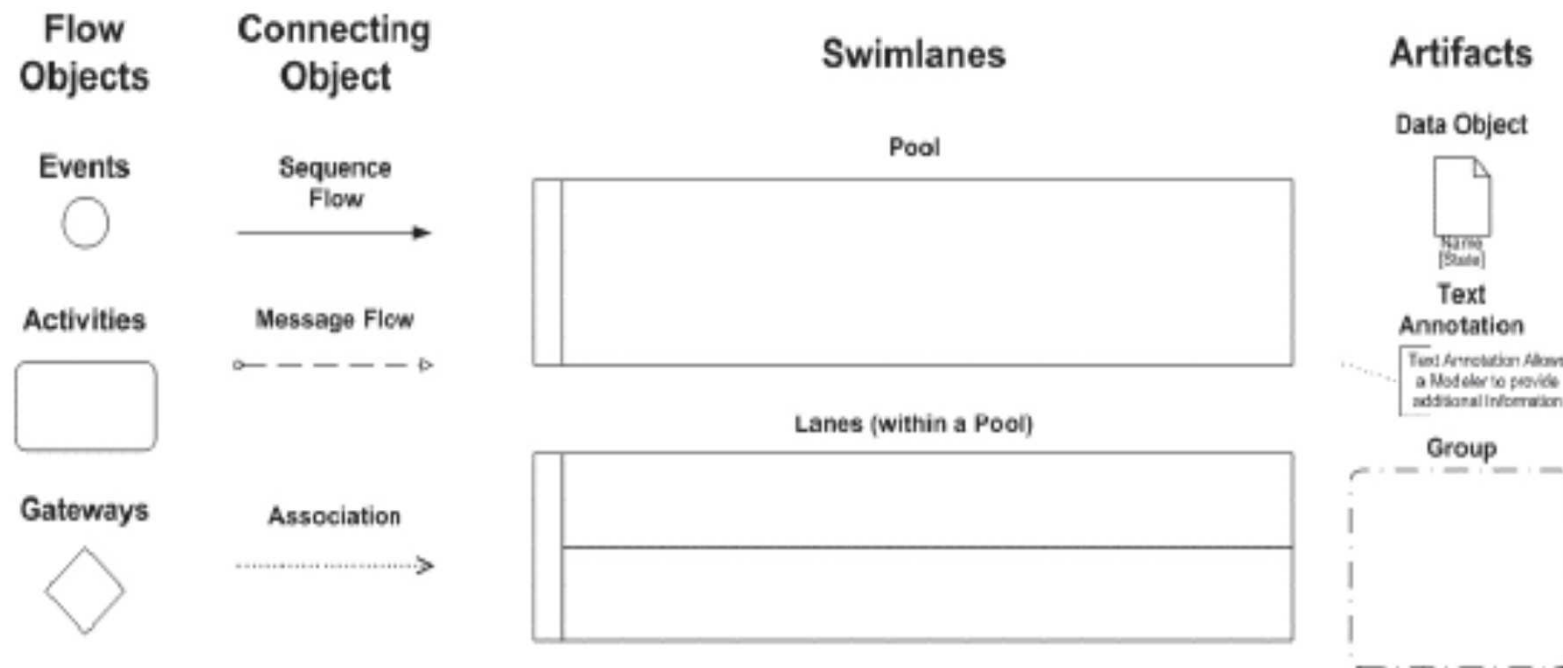
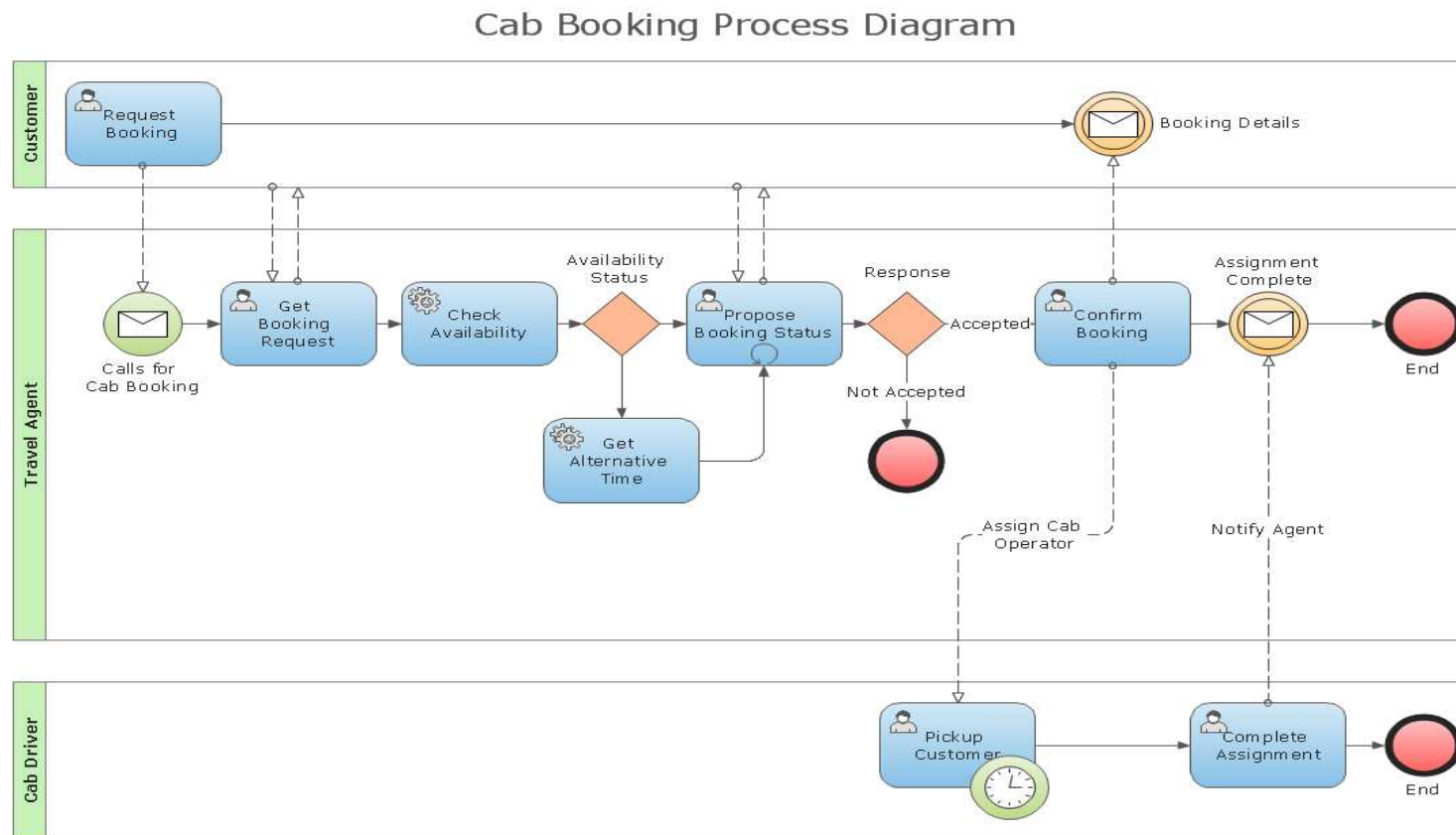


Figure 15 – Basic symbols of BPMN, source: (OMG, 2012)

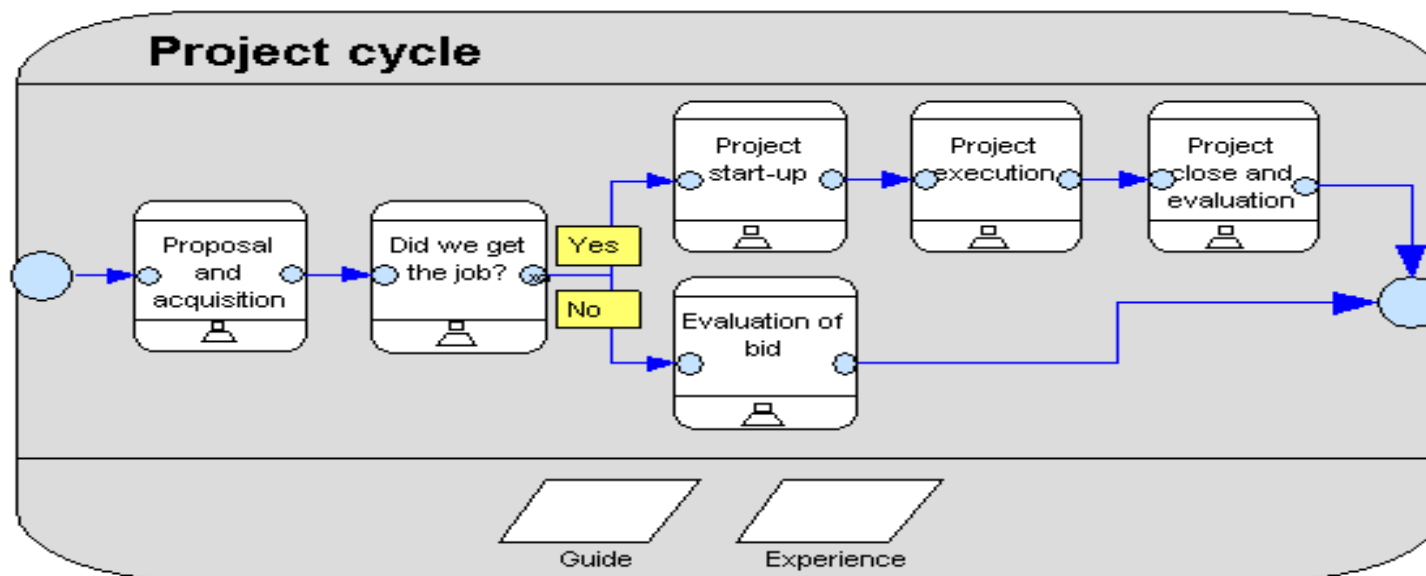
# Folyamatmodellek (példák)

## BPMN, BPML



<http://www.conceptdraw.com/How-To-Guide/process-flowcharts>

# Folyamatmodellek (példák)

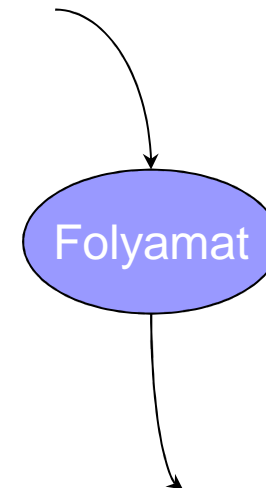


A simple transformational process model, with organizational roles and information resources.

<https://activeknowledge modeling.com/2009/03/31/12-different-ways-to-model-business-processes/>

# Folyamatok elemei

- Az egyéni folyamatokra koncentrál
  - Input és output
  - A folyamat célja
  - A folyamat tevékenységei
  - Felelősségek és hatáskörök
  - Ellenőrzési pontok
  - Folyamat mértékek
  - Teljesítmény - elvárások
  - Szükséges kompetenciák



Based on CMMI-DEV V1.3. Continuous representation



# További folyamat-jellemzők

- Az egyéni folyamatokhoz tartozó eljárásokat végrehajtják-e (legalább informálisan)?
- A folyamatot megfelelően tervezték-e?
- Követik-e a tervet?
- Ellenőrzik-e a végrehajtást?
- Követik-e a végrehajtást és hoznak-e korrekciós intézkedéseket?
- Szabványosított-e a folyamat?
- Értik-e a folyamatot mennyiségileg is?
- Folyamatosan javítják-e a folyamatot?

Based on CMMI-DEV V1.3. Continuous representation

# Folyamatmodellezési lehetőségek

Table 21 – Coverage of process elements

Process element	BPMN	EPC	UML Activity diagram
Process name	Process name	Function name	Activity name
Process parent	Parent-child relation (unlimited levels)	–	Activity-Action relation
Process purpose	In textual descriptions	In textual descriptions	–
Activity	Activity	Function	Activity
Role, responsibility	Lanes	Organisation unit	Partition or Swimlane
Artifact	Artifact	Document, Information object	Object
Element relation	Element relations	Element relations	Control flow, Object flow
Entry/exit criteria	–	–	Action constraints (pre and postconditions)

Zádor Dániel Kelemen: Process based unification for multi-model software process improvement. PhD thesis. 2013. <https://pure.tue.nl/ws/files/3583930/741509.pdf>





# Folyamatleírási lehetőségek

Table 22 – Comparison of modeling languages

Guideline	BPMN	EPC	UML activity diagram
Intelligibility	+		
Coverage of process elements	+	+	+
Ability of expressing workflow patterns	+		
Software support			+
Portable format			+
Widespread			+

Zádor Dániel Kelemen: Process based unification for multi-model software process improvement. PhD thesis. 2013. <https://pure.tue.nl/ws/files/3583930/741509.pdf>



# Érett / magas képességi szintű folyamatok

- Egy érett szoftverfolyamat:

**Meghatározott, dokumentált, tervezett, követett, mért, ellenőrzött, hatékony, alkalmazásban van, alkalmazását oktatják, szabványos és javulásra képes.**

- Ezek közül minél több elem van a helyén egy folyamatra vonatkozóan, annál érettebb, annál magasabb képességi szintű az adott folyamat.



# Miről volt szó...

- A szoftverről és a szoftvermérnökségről
  - IT és szoftver, definíciók
  - Miért olyan fontos a szoftver? Különbözik-e a szoftver más “javaktól”? Miért, vagy miért nem?
  - A szoftverfejlesztés alapelvei
  - Példák szoftverhibákra és ezek következményeire
  - A fegyelem szükségessége a szoftverfejlesztésben
  - Próbálkozások a szoftverfejlesztés modellezésére az elmúlt 50 évben
  - Folyamatok, meghatározásuk és elemeik
  - Folyamatmodellek a szoftverfejlesztés szolgálatában