

# SW inženýrství informačních systémů v malých a středních podnicích

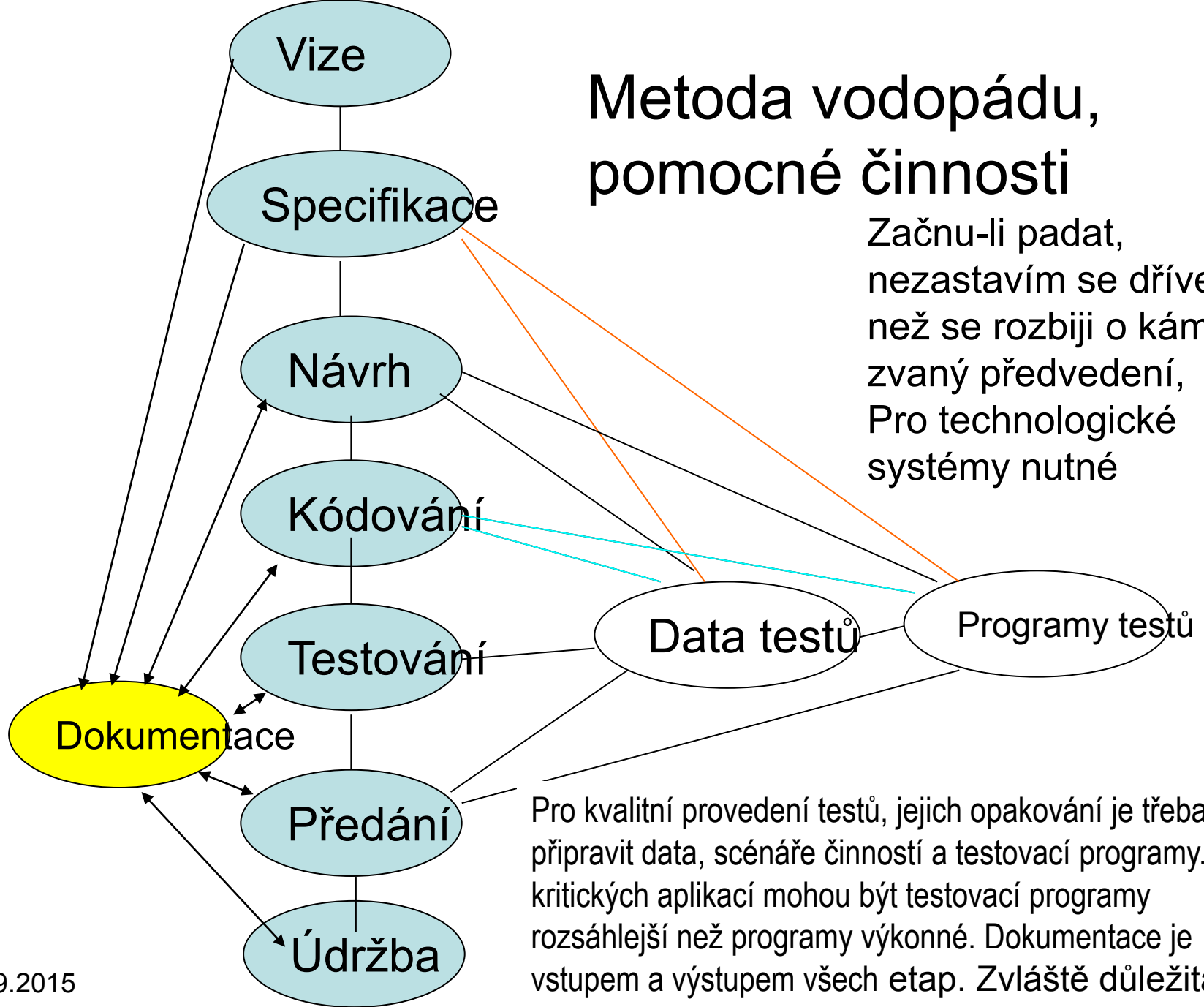
Pragmatický inženýrský přístup pro malé organizace a pro jednotlivce

Pragmatický:

Ad hoc přizpůsobení řešení konkrétní situaci

# Metoda vodopádu, pomocné činnosti

Začnu-li padat, nezastavím se dříve, než se rozbiji o kámen zvaný předvedení, Pro technologické systémy nutné



Pro kvalitní provedení testů, jejich opakování je třeba připravit data, scénáře činností a testovací programy. U kritických aplikací mohou být testovací programy rozsáhlejší než programy výkonné. Dokumentace je vstupem a výstupem všech etap. Zvláště důležitá je pro údržbu

# Pragmatický vodopád

- Rozsah dokumentace a rozsah podpory testování se přizpůsobuje okolnostem (kvalita řešitelů, kritičnost projektu, kvalita vize...)
- Je nutná opatrnost, v IS to obvykle bez pragmatických přístupů nejde, apř. Z důvodů nejasných nebo se měnících požadavků

# Malé a střední podniky small to medium enterprises, SME

- Malé podniky do 20 zaměstnanců
- Střední podniky do cca 200 zaměstnanců
- SW firmy považované za SME mají tendenci být menší než v jiných oborech
- V ČR pracuje v malých a středních firmách 80% ajťáků (průzkum ČSÚ prosinec 2013)
- Jádrem této přednášky bude budování IS pro SME menšími SW firmami

# SW inženýrství informačních systémů v malých a středních podnicích

Mnohé je použitelné i pro velké podniky a státní správu a stává se s růstem složitosti IT stále potřebnější obecně

Platí především pro dokumentově orientované SW architektury

# Důležité jsou odchylky v detailech

- Ďábel je skryt v detailu
- Každý, kdo zažil SME a velkou firmu potvrdí, že jsou rozdíly v mnohém zásadní

# Zaměření přednášky

Zaměříme se hlavně na pragmatická řešení vhodná především pro malé a střední firmy ale také pro státní správu, provoz zdravotních DB, aj.

**Mnohá řešení pro velké firmy mají specifika, které se jen obtížně uplatní v menších firmách. Tato řešení jsou závislá na**

- komplikovaných normách (tisíce stran),
- kultuře velkých výrobců a *velkých uživatelů*,
- dostupnosti velkých souborů dat (např. při optimalizaci byznys procesů, neboť velcí mají více obchodních případů

# IT je technický obor

*Ne zcela si uvědomujeme důsledky toho, že SW má zpravidla charakter nástroje, vývojové prostředky (programovací jazyky) mají daném SW prostředí charakter nástrojů pro výrobu nástrojů, proto mluvíme o **inženýringu***

Důvody, proč se vyvíjejí nové jazyky a jiné zanikají jsou velmi podobné tomu, proč se vyvíjejí nové stroje a nástroje pro ně, viz. spousty kleští u kováře

musí se s ním dobře pracovat při výrobě určitých artefaktů pokud má kovář potřebný talent znalosti a dovednosti, o artefakty musí být zájem

Kovář má objednávky a měl by mít výkresy ....



Neakceptuje je, že tvorba SW  
je inženýrský obor  
SW entity jsou průmyslové  
výrobky

Shody jejich vlastností s  
vlastnostmi klasickými pokročilými  
výrobky

# Inženýr a řemeslník

- Pohrdání řemeslníků (kodérů) s analytiky
- Analytici nemají porozumění pro nuance řemesla (možnosti, cena, ...)
- Obojí spolu se zadavateli nemají zažité přístupy hodnocení plánovaných či existujících SW artefaktů, týká se to i vývojových nástrojů
- Nejsou dostatečně rozvinuté postupy hodnocení a kontroly projektů

# Programovací jazyky jako nástroje

- Vhodné jen pro určitou oblast řešení a znalostní oblast
- Nabízejí připravené artefakty
- Jsou použitelné v daném prostředí
- Jsou dostatečně efektivní a schopné produkovat akceptovatelná (kvalitní) řešení
- Jsou dobře zvládnutelné uživateli („programátory“)
  - dovednosti, znalosti, zvyky

# Podceňované vlastnosti IT

Efekty IT, především IS se obtížně měří, často jsou jinde než se čekalo (uvedeme příklady)

Jsou často dlouhodobé a ty jiné než krátkodobé

Někdy není ochota je uplatnit (př. školství)

Praxe ochrany dat neumožňuje data správně využívat aniž zajišťuje dostatečnou ochranu dat

**Kvalita IT je tedy věcí kvalifikovaného používání, znalostí, zkušeností a někdy i (politické) vůle a boje s předsudky**

# Instrukce se neošoupou?

**Omyl:** Instrukce se neošoupou, software tedy také ne

**Skutečnost:** Softwarové systémy se chovají jako složité systémy:

frekvence selhání sleduje křivku známou z teorie spolehlivosti

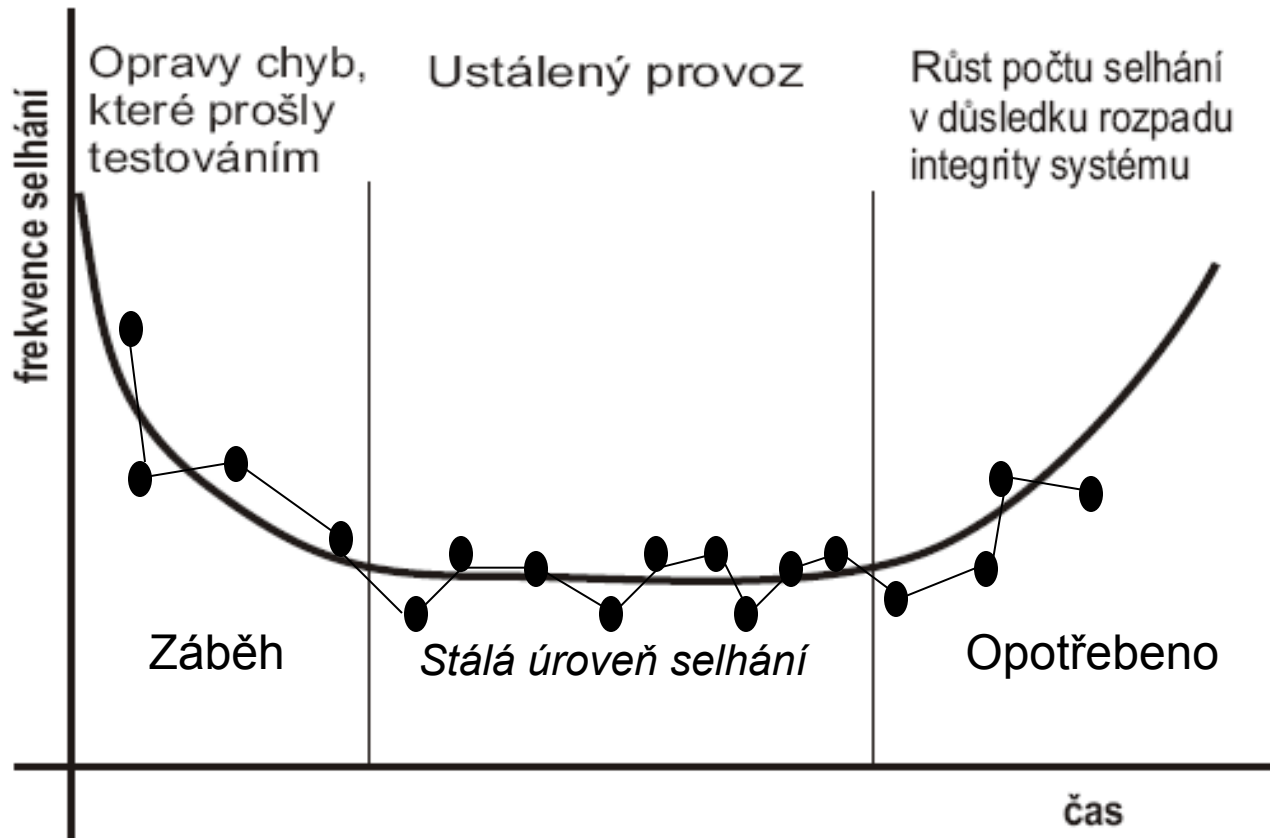
*záběh, provoz, **opotřebení***

*Provoz vyžaduje stále další úpravy (vylepšení, adaptace na nové platformy), to platí pro informační systémy a kupodivu stále více pro vědecké systémy*

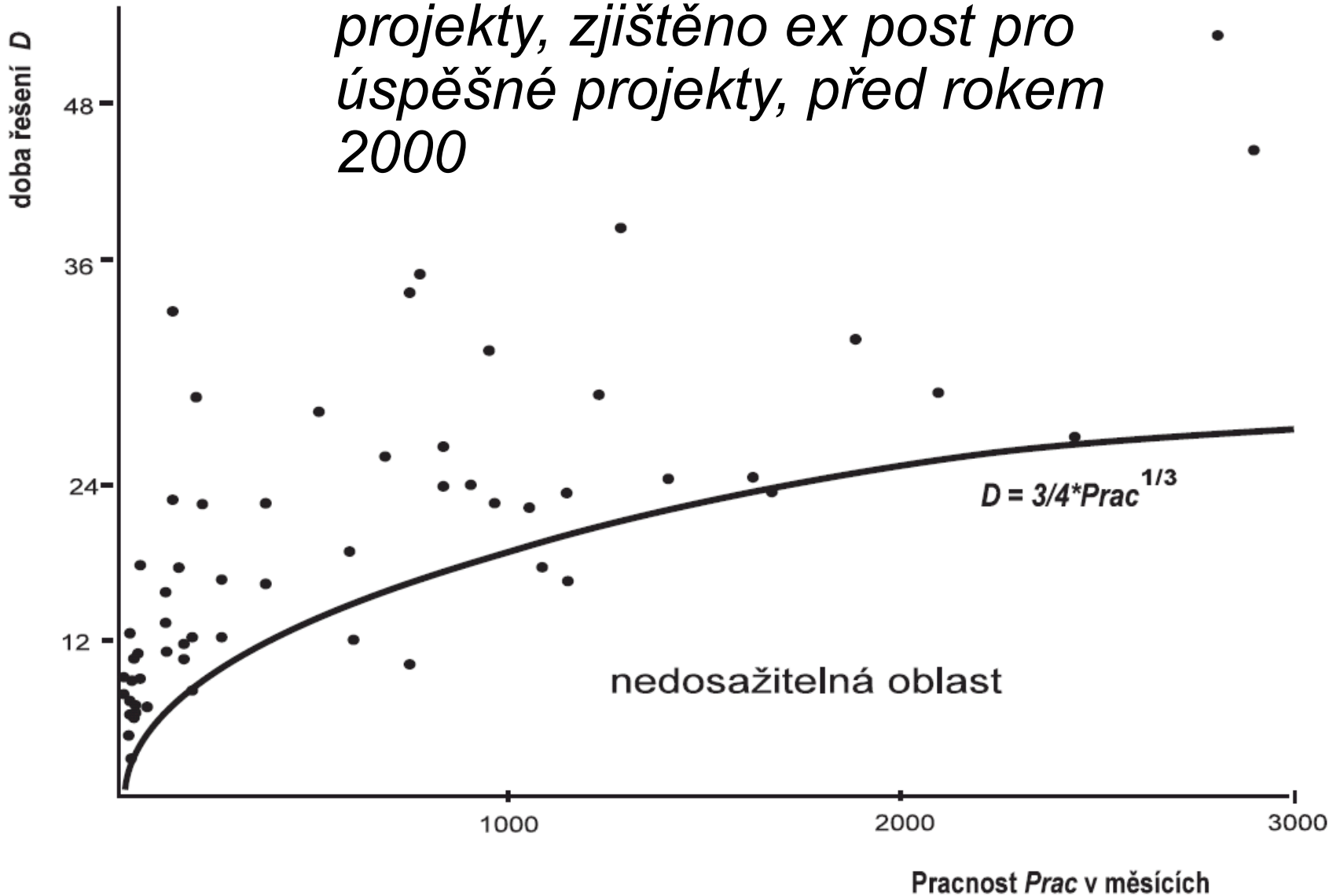
**Výzva.** Po čase se musí systém zahodit nebo přepsat, na to často stačí jen velké podniky

- pokud systém lze někdy přepisovat po částech (inkrementálně), je to dostupné i menším podnikům
- pak ale musí mít SW vhodnou architekturu, dnes obvykle servisní

# Vanová křivka. I SW se opotřebí platí zvláště pro IS a human oriented SW



*Některé starší výsledky pro SW  
projekty, zjištěno ex post pro  
úspěšné projekty, před rokem  
2000*



# Která oblast techniky má s IT nejvíce společných aspektů

Architektura

Kde kdo do toho kecá

Projekty jsou předražené

Vysoký podíl řemesel



# Stavebnictví

Dokonce existují v IT náznaky  
aspektů známých z urbanizace  
Infrastruktury

# Co budeme studovat

- Procesy vývoje SW systémů, celý životní cyklus, především informační systémy (ty zahrnují i lidi) vhodné hlavně pro malé a střední firmy
- Komponentově orientované architektury (hlavně SOA) a na ně vázané postupy (servisně orientované obchodní aplikace, cloud computing, agilní vývoj a agilní byznys, scrum) aplikované hlavně na informační systémy menších podniků
- Mnohé z toho, co je výhodné pro malé podniky se stává nutností i pro velké organizace a velké SW výrobce

# SME mají specifické požadavky

- Jiné a jinak prováděné byznys procesy (podniková kultura)
  - Menší detailnost a formálnost
  - Větší agilita uživatelů
  - Nemohou být příliš podrobné
    - Při provozu na to není dost lidí, kteří by systému rozuměli a měli na něj dost času
    - Nejsou na to prostředky při nákupu i při provozu
    - Jsou použitelné jen o velkých firem, ty ale mají specifickou kulturu, která nebývá vhodná pro malé

# Rychlost změn

- V každém okamžiku se v IT používají znalosti a nástroje mladší pěti let, podstatná část znalostí během této doby zastarává (je to ale méně než 50%)
- Během každých deseti let dojde alespoň jednou ke změně vedoucího paradigmatu
  - V IS nastupuje dokumentově orientované SOA, to bude naše vůdčí téma

# Zkušenosti přednášejícího

- 1959 Statistik se stává programátorem ve assembleru, výpočty pro fyzikální chemii, pitomí politici – nepustíme ani jeden transistor do výroby počítačů, když máme dost relé
- 60-70 Přejít na Fortran, Algol (pokus o kompilátor), formální jazyky, některé mé články z té doby se dodnes citují, mainframy
- 70-80 on line vstupy, minipočítače v přímém řízení systému dílny, první aplikace principu výměny zpráv mezi SW komponentami
- 80 účast na vývoji českých PC, spolupráce se Siemensem na řízení rozsáhlého výrobního systému,
- 90 Soukromý podnik, počítačové sítě a IS , nová varianta analýzy shora, dekompozice IS, počátek www
- 2000 SOA, patterns, účast na projektech malých firem

# Změny v informatice

Roky	Typické úlohy	Technologie
-1960	Vědecko technické úlohy	Sálové počítače, děrné štítky, tiskové sestavy, FORTRAN, Algol
1960-1970	Ekonomické výpočty v dávce, postupný nástup terminálů	Sálové počítače, děrné štítky, tiskové sestavy, COBOL, datové systémy
1970-1980	Ekonomické výpočty v dávce, často interaktivní vstup dat, řízení technologií, <b>krize IT 1980</b>	Sálové počítače s terminály, minipočítače děrné štítky, tiskové sestavy, COBOL, C, Pascal, DB ....
1980-1990	Ekonomické výpočty v dávce, interaktivní vstup výstup, úlohy na PC, <b>krize IT 1990 (meze PC)</b>	Sálové počítače s PC místo terminálů, kancelářské úlohy pro PC, datové báze
1990-2009	Interaktivní výpočty na síti, e-komerce, Internet, sociální SW <b>2002 krize, Internetová bublina</b> <b>2008 krize, IT nepomohlo</b>	Servery, počítačové sítě, Internet, grafika, vývojová prostředí, databáze, globalizace, webové služby, podpora sociálních sítí

# Co se relativně málo změnilo

- Lidé
  - Postoje
  - Cíle
  - Předsudky
- Potřeba a dovednost spolupráce
- Nedostatečné chápání ICT jako technologického oboru
- Manažerské nedostatky

# Hlavní poznatky

- Krize IT se vrací asi po deseti létech, může přijít znovu
- Po krizi se vždy se vrátil zájem o informatiku, ale s podstatně jinými úkoly
- Zdá se, že po krizi v roce cca 2000 a 2008 se zájem o informatiky v mnoha zemích plně neobnovil. O české informatiky je stále zájem, především o kódéry a analytiky
- Současná krize omezuje investice do IT
  - Ne však kriticky
- Nově globalizace vývoje SW, hlavně v kódování
- Není jasné, jak se projeví současná krize v delším výhledu



Budeme využívat zkušenosti z (starších i nedávných) praktických projektů pro malé a střední firmy a nedávné výsledky výzkumu metod SW inženýrství v oblasti SOA metod integrace velkých komponent do velkých systémů s využitím dokumentové orientace

# Celkový trend

Stále širší integrace do  
společenských a sociálních  
procesů

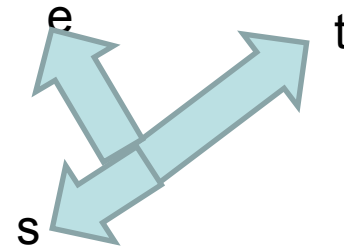
Integrace autonomních částí

Otevřenost

# Obtížnost aspektů IT, hlavně IS

- Klasické IS, např. správa skladu (operativa)

- Ekonomický užitek *i* je dosti zřejmý
- Jak technicky udělat *t* také,
- Sociální aspekt *s* málo ovlivňuje řešení

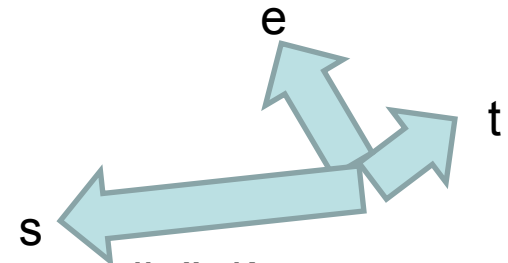


- Informační technologie dnes

- Ekonomický užitek se dostatečně nezvažuje
- Jak technicky udělat: řešení ad hoc, předražené
- Sociální aspekt téměř vždy přítomen, mnohdy zcela zásadní

- Je zdrojem obtížnosti a problémů
  - Nejasnost cílů
  - Postranní zájmy (viz evaluaci škol), korupce
  - Potřeba globálních propojených měnicích se systémů
  - Existence sociálních a zdravotních rizik

- Ekonomické a sociální aspekty zásadním způsobem ovlivňují procesy vývoje a musí se zvažovat při specifikacích požadavků



# Úzká místa informatiky

- Zaměříme se na úzká místa vývoje IS , bez jejich vyřešení nedosáhneme úspěchu
  - specifikace a spolupráce s uživateli, společenské souvislosti
  - Úkoly při řízení projektů, problém sledování úkolů pomocí smart sheetu
  - SW procesy především s aplikacemi na SOA
  - SW normy a SW metriky
- Něco se dá jen obtížně na škole nacvičit, lze jen seznámit (MBA lze studovat až po získání praktických zkušeností)
  - Spolupráce s uživateli
  - Tvorba rozsáhlých systémů
  - Management, zvyky ve firmách
  - Dovednosti a zkušenosti,
  - Pokus o otestování v PV024

# Systemy jsou stále složitější

- Přejít na nový systém resp. vývoj nového systému není možný metodou velkého třesku, to postupně stále více platí i pro velké SW výrobce a velké uživatele
- Řešení: Postupná modernizace částí, využívání legacy systémů, nákup, vývoj částí na zakázku resp. používání otevřeného SW, nově dokumentové SOA

# Halsteadův vztah pro malé programy

Poněvadž je pozorováno, že

$$Soper \cong C \cdot Del^b$$

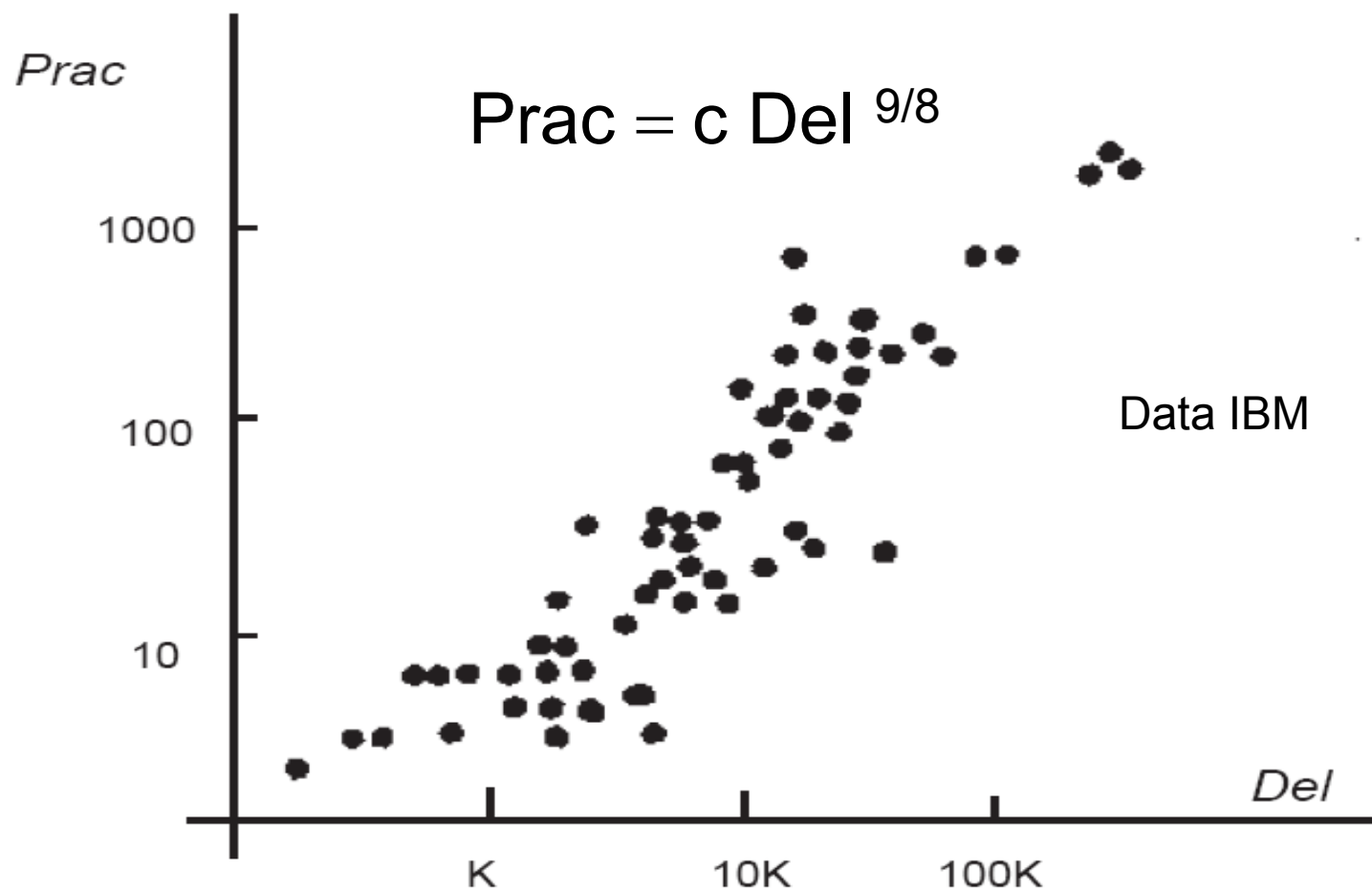
dostaneme

$$Prac = c Del^{1+b} \quad b \cong 0,25$$

Pro velké programy je hodnota  $b$  příliš vysoká. Je to důsledek toho že vztah modifikují takové technologie, jako dekompozice a znovupoužití částí.

Pokud se dekomponuje do malých komponent a systémy se liší jen počtem komponent a transport zpráv se nemusí pracně implementovat, bude  $b$  velmi malé.

# Závislost pracnosti na délce kódu



Hlavní problém velkých kódů:

Udržitelnost



# Jak na věc

- Podmínkou úspěšných řešení je snadná dekompozice do autonomních částí
- integrace částí a postup integrace zdola
  - Pro globální systémy je to i dnes prakticky jediná schůdná možnost pro všechny
  - Vyžaduje to autonomii částí a srozumitelnost jejich rozhraní pro uživatele do značné míry i pro management – dokumentově orientovaná komunikace

# Role informačních systémů

- Kvalitní IS je klíčová podmínka činnosti podniků v globální ekonomice
- Motor zlepšování výroby
- Základ informační společnosti
- Motor změny sociálních aspektů současné společnosti
- Motor přechodu na nová paradigmatata, především v oblasti architektury SW

# Co je na IS klíčový aspekt a co se relativně málo mění

- Lidé, zato musíme zohlednit i malé změny jejich potřeb a znalostí
- Musíme se nějak vyrovnat a řešit jejich předsudky
- **Musíme k nim mít úctu,**
  - Jen oni mnohdy ví nebo alespoň vycítí, co je třeba
    - Vlastně jsou rozhodující pro přijetí systému

# Proč studovat projekty IS pro menší firmy a organizace

IS pro SME musí splňovat požadavky plynoucí ze specifických ekonomických a sociálních podmínek SME

SME tvoří podstatnou část ekonomiky a většinu firem

Start-upy jsou SME, SME jsou nositeli převratných nápadů

Některé SW přístupy vyvinuté v malých SW firmách pro SME jsou významné pro budoucnost oboru

# Proč studovat zkušenosti s projekty IS pro menší firmy a organizace

- Dá to více než mnohá hluboká teorie i mimo SME
- Ukáže to, že nečekané souvislosti mohou být častější a důležitější, než bychom čekali
  - Velký průšvih jsou opomenuté maličkosti a skryté předpoklady
  - Větší průšvih je nevyužití existujících dovedností uživatelů
  - Nevětší průšvih je formulace vizí a požadavků, tedy vlastně spolupráce s uživateli a porozumění jejich potřebám a zvládnutí jejich znalostí, dovedností a legitimních zájmů a nedostatky v managementu

# Podniková kultura v SME

- Pravidla organizace podniku a struktury podnikových procesů jsou relativně jednoduché
- Pravidla spolupráce jsou relativně neformální, jsou častá ad hoc řešení
- Je možná vstřícnost a nepřilíš formalizovaná pravidla spolupráce lidí
- Důsledky omezených kapacit pro aplikaci procesů vývoje

# ZÁKLADNÍ POZOROVÁNÍ

Podniková kultura menší SW firmy se méně liší od podnikové kultury podobně velkých organizací pracujících v jiných oborech (tedy i obvyklých zákazníků) než od podnikové kultury opravdu velkých organizací.

# Co je typické pro malé (SW) firmy

- Rychlé změny
  - Diktuje je trh, malá tržní síla firmy – nemůže si dupnout
  - Úkoly i metodiky se rychle mění v důsledku požadavků uživatelů a změn vývojových ekosystémů
  - Menší firmy jsou pružnější obecně
- Nejsou zdroje:
  - Nelze nasadit mnoho specialistů, nejsou k dispozici
- V malých a středních SW firmách se snáze uplatní superprogramátoři a nové nápady



# Co je typické pro malé SW firmy

- Často nelze použít hotová komplexní řešení
  - Hotová řešení často dělají velcí pro velké, to není nic pro svět malých firem, mnohdy platí i pro normy, mnohé se musí použít znovu
  - Speciální požadavky: ne velké náklady ani dlouhá doba řešení
  - Kupodivu leccos vhodné pro malé firmy se uplatní i v e-governmentu a tam, kde IS silně ovlivňují přímo uživatele (dokumentově orientovaná rozhraní),

# Omezení pro malé výrobce SW

- SME mají omezené zdroje (investice, čas, zkušenosti, ...) a tedy nemohou zpravidla vyvíjet či modernizovat velké a dokonce ani středně velké systémy metodou velkého třesku *(uvidíme, že to pro velmi velké systémy platí i pro firmy s prakticky neomezenými zdroji)*
- Musí se vyrovnat s tím, že jejich zákazníci potřebují ne zcela malé systémy a vyžadují, aby bylo možné některé části svého a systému outsourcovat
- Nové systémy musí být integrovatelné se s existujícími systémy
- Musí být snadné je udržovat a při instalaci přizpůsobovat specifickým podmínkám svého zákazníka

# Důsledky omezených zdrojů v malých organizacích obecně

- Méně lidí: role musí mít širší škálu úkolů, člověk ale zvládne jen omezený počet pravidel  $\Rightarrow$  podnikové procesy nemohou být detailně specifikovány
- Malá uživatelská firma nemá prostředky na nákup komplikovaného systému, a ani nevytvoří podmínky pro jeho používání (zaučení, změny podnikové kultury), totéž platí pro vývojová prostředí SW firem
- Velké IS vytváří velcí pro velké, nejsou tedy vhodné z kulturních důvodů pro malé, malí je nejsou schopni vzhledem k výše uvedeným omezením vyvinout naráz

# Řešení omezených zdrojů v SME

- Vyrábět co nejjednodušší systémy,
  - princip maximální možné lenosti
  - K čemu a proč je to třeba
  - Stálé podceňování pracnosti úprav
- Přebírat
- Používat otevřený SW a adaptovat ho (ERP5, AGdampiere,..) nebo používat hotová řešení (systémy třetích stran nebo legacy)

# Řešení omezených zdrojů v SME

- Nepoužívat Big Bang, nedělat monolitická řešení, používat architektury s autonomními komponentami
  - Moderní architektury, integrace systémů, SOA, EDA, cloud...
  - Inkrementální vývoj a údržba
  - Integrace s existujícími systémy (doplňky),
  - Využití komunitních systémů jako mikrokomponent (Excel, Alfresco, některé varianty využití cloudu)

# Nadměrná složitost řešení škodí i velkým

- Složitá řešení se ne vždy vyplatí,
- I když velcí výrobci SW na to mají, koncoví uživatelé leckdy ne
- Normy resp. dokumentace o tisících stránek, která se často mění
- Mění se přístupy, inovace velmi často už během pěti let

# Filosofie řešení malých se stává důležitá i pro velké firmy

- Systémy se stávají natolik komplikované, že je ani velcí nezvládají pokud nepoužijí triky malých SW firem
  - Příliš drahé
  - Permanentně zastaralé
  - Obtížně udržovatelné
  - Nespolehlivé

# Zaměření přednášky

## klíčová témata

Počáteční etapy vývoje IS (jsou rozhodující, závisí na analýze a spolupráci s uživateli) a management projektů je úzkým místem ICT

Poznátky diskutované v přednášce tvoří jádro SW inženýrství pro IS a nejen pro IS

Tyto poznátky jsou často klíčem k profesnímu uplatnění i mimo oblast kódování, tj. ve vedoucích a jiných velmi žádaných pozicích i mimo IT, které lze vykonávat do vysokého věku, jsou prestižní i dobře placené a rozhodují do značné míry o úspěchu či neúspěchu softwarových projektů



# System, definice

- Strukturovaná entita
  - Zdroje (materiál, energie, data)
  - Prostředky (stroje, nástroje, lidé, znalosti a dovednosti)
  - Vazby mezi částmi (komponentami)
  - Procesy umožňující za daných podmínek dosahovat určité cíle, u IS poskytovat informace, doporučovat opatření případně přímo řídit technologie a organizace a měnit svět

# Informační systémy

-Informační systém (IS) je systém umožňující sběr, ukládání a správu dat s cílem získávání a presentaci informací.

-IS je systém, tj. strukturovaný komplex technik, nástrojů, a zdrojů umožňující získávání, ukládání a poskytování informací uživatelům a jiným systémům. V širším smyslu mohou být výstupem IS přímo rozkazy osobám a signály procesům reálného světa (avionika letadla, reaktor, ...). **IS tedy může být i řídicím systémem a obvykle do určité míry jím je (to je velmi významný fakt).**

-IS nemusí využívat SW, my se ale budeme zabývat případem, kdy IS využívá softwarovou podporu.

-IS jsou základním nástrojem globalizace světové ekonomiky, informatizace společnosti a změn ve výrobních procesech a změn ekonomických procesů.

-Větší efekty jsou ve výrobě

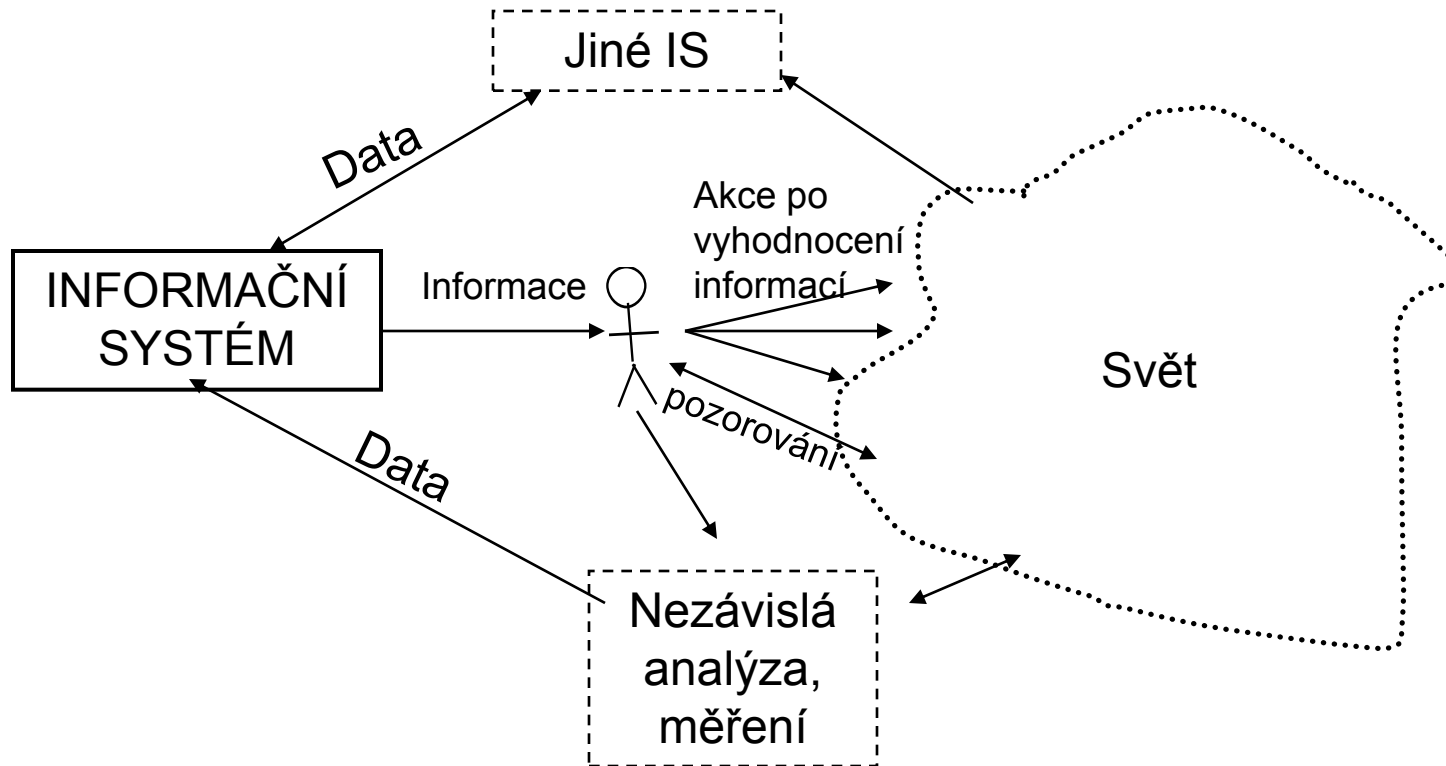
-Pozor na BPR – měním i to co dobře funguje

# Zvláštnosti informačních systémů

- IS jsou klíčovou částí IT
  - Většina aplikací IT jsou informační systémy, IS mají největší dopad na ekonomiku i globalizovaný svět a náš každodenní život
    - Podpora byznys procesů a managementu
    - Motor někdy spíše brzda výkonu státní a místní správy
  - Největší dopady: Sociálně ekonomická oblast
    - Podmínka globalizace, podpora celosvětové spolupráce
    - Zasahují do soukromí, legislativně je to špatně ošetřeno (chrání se, co by se nemuselo, blokáda důležitých informací)
  - Technické problémy
    - Největší SW systémy,
    - Moderní SW architektury, nová paradigmatata (architektury)
    - Velké soubory dat

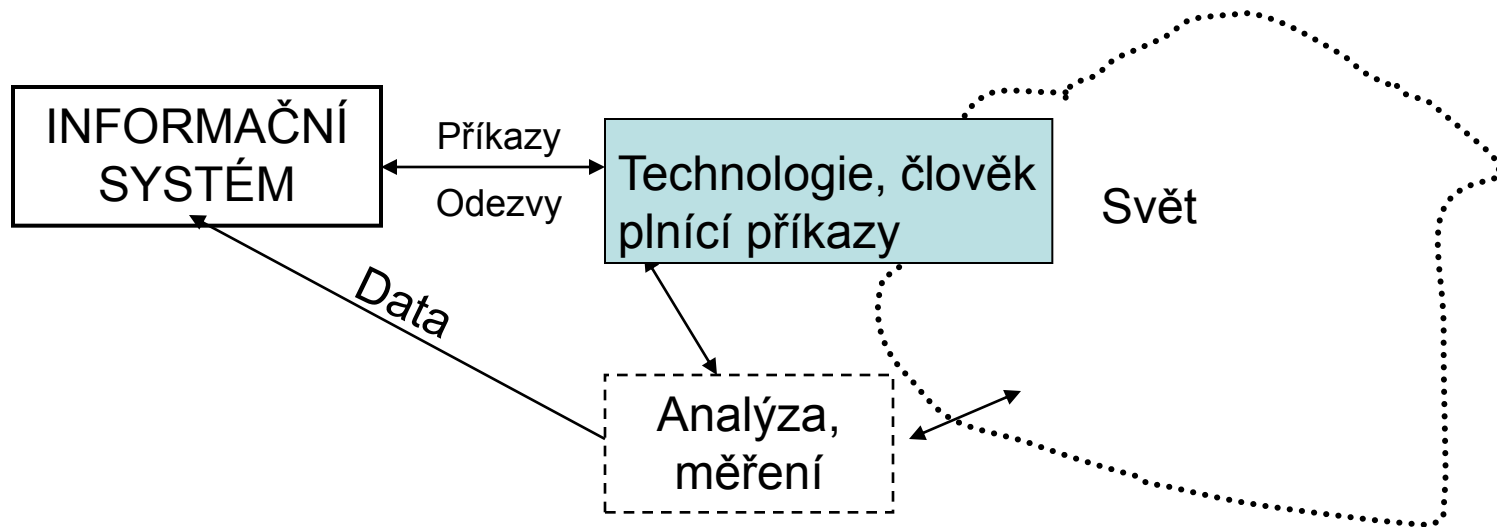
# Informování versus řízení

- Informování



# Informování versus řízení

- Řízení, z hlediska technologie programů zdánlivě téměř totéž, existují skryté hluboké rozdíly (kritičnost automatických akcí, za které musí být někdo odpovědný).



I u technologických systémů je výhodné, aby byly komunikace srozumitelná uživatelům, uvidíme možnosti použití

# Hlavní varianty IS, typ úkolů

- Technologicky orientované systémy (protokoly, avionika, software auta)
  - Problém není co, ale spíše jak
  - Záhy vím, co je třeba, pak se to (moc) nemění
- Human oriented, klíčová varianta SW
  - Cíle a potřeby nejsou úplně jasné, u různých zainteresovaných osob mohou být různé, vliv lidského faktoru postupně roste
  - V důsledku měnících se společenských podmínek se mění
  - Může existovat snaha o optimalizaci jen pro některé a ke škodě jiných
    - Příklad: Ochrana osobních dat versus ochrana života
  - Mohou být zdravotní rizika
    - fyzická (klouby, poruchy životních funkcí) a psychická poškození (Gamblerství, sociální defekty typu izolace od lidí, závislost na sociálních sítích)

# Sociální a společenské efekty informatiky jsou stále významnější

- Globalizace
  - Ekonomické efekty významné
  - Zranitelnost světové ekonomiky roste (Fukušima a celosvětová výroba aut ohrožena výpadkem výroby jedné součástky)
  - Rostoucí vliv softwarových dinosaurů (velkých dodavatelů SW)
- Sociální sítě a odcizování lidí
- Vzdělávání nedežní krok s potřebami
- Nedomyšlená legislativa, nejen u nás

# Sociální dimenze informačních systémů

- Zvláště silná v řadě oblastí
  - Především menší a střední podniky, e-health, e-government
- Silná závislost na znalostním oboru cílech uživatele
- Uživatel silně zapojován do vývoje, požaduje on-line změny byznys procesů a byznys partnerů integraci artefaktů podporujících základní byznys činnosti (finance)
- Nutnost dynamické spolupráce s IS partnerů uživatele s možností on-line ovlivňování spolupráce uživateli a umožňující řešení nečekaných byznys problémů (výpadky) a obchodních sporů
- Potřeba dokumentově a uživatelsky orientované dekmpozice



# Technická dimenze

- Kvalita služeb (service level agreement), podrobnosti jsou věcí techniků
  - Dostupnost služeb včetně doby odezvy
  - Spolehlivost a reakce na nečekané události
  - Zabezpečení
  - Bezpečnost
  - Kapacity a metody outsourcingu, cloudy,
    - Je to věcí ajťáků
  - Modifikovatelnost
  - Otevřenost
  - Nové technologie, potřeba globálního přístupu, SW architektury
- Každých deset let nové paradigma pro IS
  - Objekty, Komponenty, SOA, otevřené systémy, webové služby, cloudy
  - Dnes Cloud, EDA, SOA a web, dokumentové orientovaná komunikace

# Zvláštnosti IT, hlavně IS

- Prudký vývoj
  - Za pět let pokrývají současné znalosti jen 50 procent potřebných znalostí
    - Naučit se získávat nové znalosti
  - Moorův zákon (vzhledem k paralelizaci vlastně stále platí)
    - Zdesateronásobení výkonu (běžné sítě) procesorů za pět let
    - Za 50 let  $10^{10}$  zvýšení kapacit, místo 320 let jedna sekunda
  - Nové možnosti a nové požadavky
    - Zábava, sociální sítě
    - Správa, vedení podniků, přístup k informcím
    - Ochrana i hrozby
  - Každých deset let nové paradigma
    - , DFD, Objekty, Komponenty, SOA, otevřené systémy, REA
- Manažerské aspekty jsou stabilní, manažeři hlavní hrozba
  - Procento krachujících projektů se dlouhodobě drží kolem 25%

# Průběh v čase

## Standish Findings By Year

	1994	1996	1998	2000	2002	2004	2009
Succeeded	16%	27%	26%	28%	34%	29%	32%
Failed	31%	40%	28%	23%	15%	18%	24%
Challenged	53%	33%	46%	49%	51%	53%	44%

Úspěšných projektů	39%
S nedostatky	43%
Krachující	18%

## **Jak jsme na tom s úspěšností projektů**

Stav v roce 2012, Standish  
Group Chaos Report

# Co způsobuje největší problémy

Zdánlivé hlouposti (blbosti), dokonce i skutečné triviality

Skryté předpoklady, absence správných otázek

Předsudky

# Nejhorší je srážka s blbcem Nejfatálnější důsledky má opomenutí blbosti (samozřejmosti)

- Etapy vývoje: vize, specifikace požadavků, návrh, kódování, testování, předání, údržba.
  - Přes 85% nákladů na opravy chyb jde na pochybení v etapách vizí a specifikací požadavků, ne na profesní pochybení programátorů
  - Analýza příčin chyb ukazuje, že kritické jsou různé apriorní představy, neuvěřitelné logické chyby, opomíjení souvislostí atd. na straně uživatelů a problém blokovaných znalostí, tunely
    - Je to si jeden z důvodů používání krabicových a customizovatelných (např. od SAP) řešení, pořizujeme si ale konfekci, odstraňujeme konkurenční nevýhodu, nezískáváme konkurenční výhodu
- Zkušenosti tohoto druhu se těžko předávají, pro plné pochopení a hlavně akceptování se musí zažít

# Příklady z historie

- Turing a prolomení šifrování Enigma
  - Na straně Hilera slepá víra v kvalitu kódování a neexistenci výpočtové kapacity
- Bombardování měst a oslabení boje s ponorkami – hrozba prohry ve válce
  - Zřejmá chyba, zjevná ale jen při položení správné otázky a při vyhodnocení zkušeností s nálety na Londýn, ale nikdo si to nepřipustil
- Hloupé chyby dělají podniky i v době míru
- Víra v neprůchodnost Arden v roce 1940

# Nejhorší je srážka s blbcem

## Zvláště fatální je varianta pilný blbec - tunelář

- Mnohé efekty IT se těžko měří a není k dispozici dostatečně kvalitní a uznávaná metodika hodnocení kvality
  - To především mimo podniky, ale i v podnicích umožňuje zakázky zdražovat
    - Přímo, vyhnat cenu za málo
    - Nepřímo. Zakázku nafukovat
- Informatika se de facto nechápe jako inženýrský obor, viz výše



# Čím se budeme zabývat

- Budeme studovat celý proces vývoje IS v malých firmách, především ale ty etapy a podmínky, které jsou v praxi klíčové, ale na škole jen obtížně získatelné a při tom jsou hlavní příčinou problémů
- To jsou počáteční etapy vývoje a celkové podmínky, za nichž se SW vyvíjí, především
  - Vize, cíle systému (proč)
  - Specifikace požadavků (co a zčásti jak)
  - Podmínky (legislativa, předsudky veřejnosti, zdravotní ohrožení, získávání znalostí, bezpečnost, ...)
  - Nevyřčené nealgoritmické podmínky – service level agreements (SLA), např. ochrana dat
  - Spolupráce s uživateli

# Vize (cíle) požadavky

Nedomyšlené vize, cíle a požadavky jsou příčinou, že

- jen kolem 30% projektů plně uspěje
- zkrachuje při tom cca 1/5 projektů, toto číslo je po řadu let pozoruhodně stálé, příčina je většinou v managementu
- 80% vícenákladů na nápravu chyb(Standish Group, Gartner Group) jde na vrub nedostatkům ve vizích a požadavcích, jen procento jde na účet chyb v kódování
- Platí to především pro systémy jejichž činnost zahrnuje aktivity lidí, tj. většinu IS

# Statistiky z USA o egovernmentu devadesátá léta

- Šestina IT projektů včas a v plánovaných nákladech Cca 25 procent se nedokončí
- Skoro polovina se dokončí s omezeními (v horší kvalitě) a nebo podstatně draž
- Dnes jen o trochu lepší

# Zkušenosti z ČR

- Jsme na tom podobně, existují ale i „specifika“:
  - Předražené státní maturity
  - Skandál se sociálními dávkami
    - Důsledek toho, že se doba vývoje nedá libovolně zkracovat, ale nikdo na to nebere ohled, z této ignorance může mít i prospěch
  - Aféra s registry vozidel a jinými registry
    - Důsledek toho, že sítě aplikací je nutné vyvíjet a udržovat jinak, než jednotlivé aplikace
- V SW dnes dochází k principiální změně, změně paradigmatu (cloud, SOA, webové služby, dokumenty)
  - Starého psa novým kouskům nenaučíš, obtíže se zvládáním

# Registry podrobněji

- Nepracují dobře po mnoha měsících ač se konceptuálně a rozsahem požadavků na výkon jedná o poměrně jednoduché úlohy
  - Chyběla rozumná vize a zčásti i specifikace, příliš interaktivnosti, řešit i v dávce nebo pro celý dokument
  - Chybí rozumná dekompozice
  - Nesprávné využívání sítě, asi díky protokolům (programováno jako pro jeden program, jemnozrnné rozhraní bojující s mnoha tagy)
  - Nesprávná orchestrace, omezené kapacity sítě - úzká místa

# Registry podrobněji

- Dodavatel nedodal v dostatečné kvalitě, ztráty občanů
- Dodavatel si díky tomu namastil kapsu, stálo to majlant
- Nebyl „stavební dozor“
- Svinstva přes anonymní firmy
- **Typické pro mnoho státních zakázek, nejen ale hlavně v IS**

# Případ GAČR

- I vědci podceňují inženýrský aspekt IS.
- GAČR navrhl v r. 2012 nový informační systém hodnocení výsledků podle RIV bodů, neprošlo to řádnou oponenturou a dodnes se zjišťují slabá místa koncepce,
  - především přecenění možností IS a nedostatky RIV bodů
- Byla provedena rychlá změna podpůrného IS a jako ve státní správě systém IS dlouho nefungoval
- Vstup dat je pracný, výzkumné instituce si musí budovat vlastní IS pro vstup dat, sebraná data nejsou dostatečně dostupná a jsou i jiné problémy (doba „platnosti“, hodnocení při změně místa se jaksi ztratí, nezachycuje nové obory a správně nehodnotí zvláště významné výsledky (Einstein a Turing))
- 2015 ČR je poslední v EU ve spolupráci vědců s podniky

# Reakce jednoho z uživatelů IS GAČR

- *„Pracnost vstupů pro RIV mi nevadí, mám sekretářku“*
  - To je indikátor bídné kvality!
  - Důsledek zanedbání faktu, že IS je složitý výrobek, projevilo se to výše uvedených problémech, nejen na pracnosti vstupů
  - tak začíná předražování IT, mnoho úředníků a jejich horší práce, skleněné pustiny
  - Nejsou kontroloři provádějící dohled nad vývojem, např. jako na stavbě



# Naše zkušenosti

- Příklad registrů ale i s-karet a opencard ukazuje , že je nutno zvládnout i aspekt SW architektury
  - Rozdělit na rozumně velké a rozumně autonomní části
  - Orchestrace zatížení sítě
    - Rozumně velké uživatelsky (dokumentově ) orientované zprávy: srozumitelné uživatelům, málo metadat, nedělat nutně vše interaktivní, dávky se někdy hodí
    - Vhodná kapacita sítě a pomocných programů
    - Nevyhýbat se dávkovým řešením
  - Dělat jen užitečné věci (s měřitelným efektem), užitečné pro uživatele, nejvíc zatěžují nedomyšlené požadavky, a zbytečnosti

# Stokrát nic umoří nejen osla ale i vývojáře a uživatele

- Je de facto rozšířeno skryté přesvědčení, že v IS s různé funkce snadno doplní a nic to nestojí, což je hluboký omyl
  - **Syndrom** *Ještě by se hodilo tohle .....*
  - Vede to ke skluzům, změnám a nejvíce na to často doplatí to, co je nejužitečnější, protože to slouží často pouze těm, co dělají černou práci a nemohou do toho kecat
- Dodavatelé SW na tom rýžují peníze a odkládají termíny dodávek.

# Stokrát nic umoří nejen osla ale i vývojáře a uživatele

- Vše je důsledkem toho, že se nedoceňuje, že IS jsou složitá technická díla, jejichž účel a vlastnosti je nutné pečlivě a použitím specifických znalostí a dovedností specifikovat, budovat a provozovat. Švindle!!!
- Kdyby se tak jednalo v jiných technických oborech tak by bylo např. běžné,
  - že by se a technické a uživatelské vlastnosti mostu neustále měnily a řádně nespecifikovaly a považovalo by se za normální, že bude most stát včas, nebude dražší a nespadne i když do toho budou kecat laici a stavět neumětelové a budou strukturu mostu stále měnit

## Skrytý postoj

Vím jak postavit most, protože vím,  
jak se má přes něj jezdit

Výsledkem obvykle je, že se  
nespecifikuje, **proč, za kolik a jak**  
se bude po IT „mostě“ jezdit a  
nezajistí se, aby „most“ nespadol  
Je to dosti rozšířený předsudek  
uživatelů ICT

# O které profese je především zájem

- 6 profesí, studie Gartner Group 2010.
  - Byznys architekt (netriviální použití SW pro byznys)
  - Datový vědec (vyždímat data)
  - Architekt sociálních medií
  - Expert na mobilní technologie
  - Vývojář podnikové mobilní platformy
  - Architekt cloudů, virtualizace - Saas, IaaS, Paas, ....
- Obecně roste potřeba multidisciplinárních znalostí a dovedností (stálá spolupráce s lidmi různých oborů, porozumění jejich oborům, sociální dovednosti)

# O které profese je především zájem, střední úroveň managementu 2013

- Byznys architekt
- Datový architekt
- Project architect
- Analytik sociálních sítí
- .....

# Manažeři

- Výkonný ředitel, měl by ví leccos vědět o IT technologii
- Marketingový ředitel, byznys znalosti, základní IT znalosti o projektu
- Projektový manažer, technolog, chápe i ekonomické a sociální souvislosti a potřeby druhých managerů

# Kteří informatici nejméně chybí (Forrester Research 2006)

Klasické programování, statické HTTP,  
Cobol,...

*Podobné výsledky má i Manpower,  
celosvětová personální agentura*

*Kódování je dobře zvládnuto metodicky, o dobré kodéry je  
zájem, není výjimkou, že jeden udělá sám, co dvacet  
jiných.*

*Kódování není ale problém, ví se jak na to,  
Kodéři se v průměru proto hůře platí než analytici,  
Kódování se dá outsourcovat do Indie*

*Ve stáří se hůře kóduje, kódér je job pro mladé,*



# O které profese je největší zájem (Forrester Research 2014)

Největší poptávka je po SW architektch a lidech, kteří dokáží s uživateli spolupracovat a použít to, co existuje, často lokalizují existující systémy (SAP atp.), lze to jen obtížně outsourcovat do Indie

– Často se kódují jen doplňky, např. sběr dat

Pozor: SAP se nehodí pro malé firmy. Je drahý a je koncipován pro velké firmy a vytvořila ho velká firma, to platí i pro mnohé normy

Možnost přílepků, čtu s databáze

Na znalosti o specifikacích, architekturách, týmech, zavádění a zákonitosti se zaměříme v dalších přednáškách

Abychom mohli takové profese vykonávat

Nebo alespoň se s takovými profesemi dovedli dohodnout a spolupracovat

# Problémy softwaru jsou zpravidla důsledkem problémů s lidmi

- Lidé jsou součástí IS, mohou se bránit nebo prostě nespolupracovat, je nutné je získat a vyškolit, prosazovat i jejich prospěch
- Musí se z nich vytáhnout, co je třeba udělat
  - Mohou mít potíže si na to vzpomenout (blokováná, skrytá znalost), vzpomenou si až při navození situace či při správně formulované otázce,
  - není pravda, že nevědí, co chtějí, jen si nevzpomenou
  - I když si vzpomenou, nedovedou to zformulovat!!

# Problémy softwaru jsou zpravidla důsledkem problémů s lidmi

- Uživatelé nemohou vyvíjet IS sami, nelze ale IS vyvíjet bez nich, nutno rozumět jejich potřebám
- Nemohou zpravidla sami přesně požadavky, neboť v tom nemají praxi, neodhadnou obtížnost, závislost doby odhadu na rozsahu požadavků, na důležité si nevpomenou
  - Tento aspekt zesiluje a bývá důvodem přechodu na nákup velkých a drahých systémů

# Software není problém, problémy jsou s lidmi

- Nadměrné či nekonsistentní požadavky
- Neschopnost efektivní specifikace požadavků a adaptace na nové podmínky
- Snaha tunelovat
- Pocit ohrožení a proto neochota spolupracovat
- Práce navíc pro uživatele bez ocenění

# Zvláštnosti IS

Velké peníze na vývoj, na práci IS závisí velké peníze

IS jsou často kritické systémy

Mohou způsobit škody,  
někdy rozhodují o životech

Mohou mít nečekané dlouhodobé efekty

Jsou to velké systémy, největší, co známe

Jsou otevřené a rychle se mění

Je nutné je stále častěji bdovat jako  
společenství autonomních komponent

# Zvláštnosti IS, shrnutí

Stálá změna funkcí

Otevřenost a prvky reálného času (včasnost odpovědi nezbytná, některé akce nelze vrátit zpět), je nutno uplatňovat moderní SW architektury (componenty, SOA, cloud computing, gridy, atd)

Service level agreement, nefunkcionální vlastnosti (přístup, doba reakce, bezpečnost, .....

Spolupráce IT odborníků s uživateli při vývoji i provozu je nutností, často musí být velmi úzká, např. při agilních variantách vývoje, otevřené systémy

# Úkoly a profese, mimo celkového managementu

Obchodníci znají technologie  
a potřeb uživatelů

SW inženýři a obchodníci  
schopní spolupráce s  
uživateli, uživatelé,  
architekti

---

Designéři

Kodéři

Testéři

Provozní programátoři

- Sehnat kšeft a nástřel „**co**?“
- Dohodnout podrobněji **co** s vědomím, že to lze udělat
- architektura
- Zpřesnit **co**, zásady **JAK**

– Struktura i detaily **JAK**

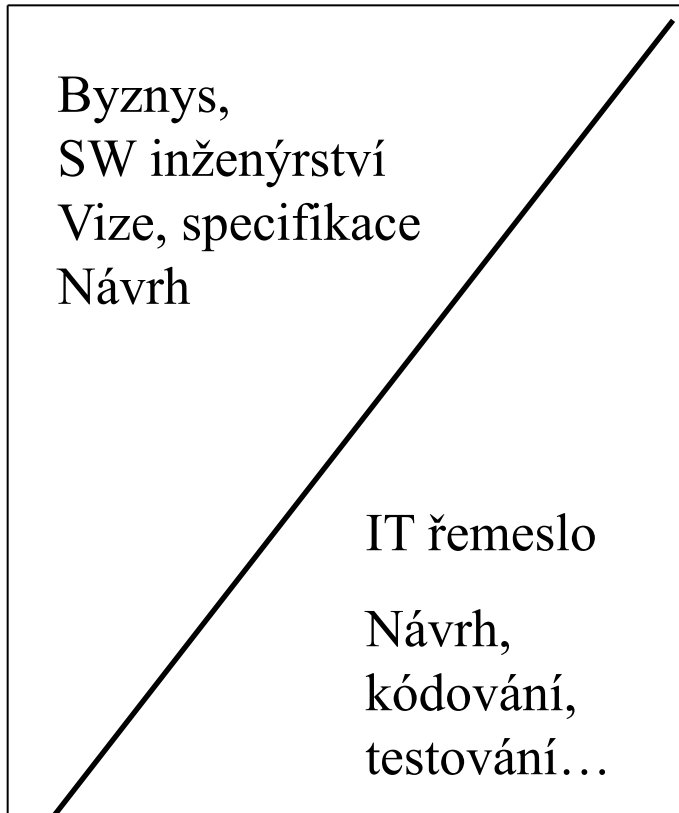
– Kódování

– Testování

– Údržba a provoz

**co** = **co** je třeba

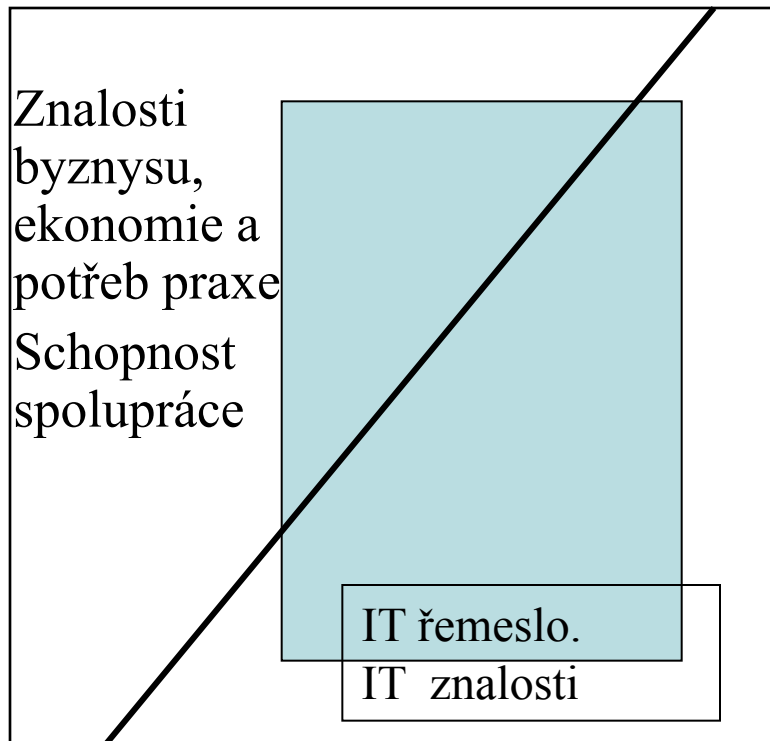
# Úkoly a profese



- Sehnat kšeft a nástřel CO
- Dohodnout podrobněji CO s vědomím, že je to rálné, tj. lze to udělat
- Zpřesnit CO, zásady JAK, architektura úkolu, SW
- Podrobně JAK
- Kódování
- Testování
- Údržba a provoz



# Úkoly a profese a zaměření přednášky



- Sehnat kšeft a nástřel CO
- Dohodnout podrobněji CO s vědomím, že lze udělat
- Zpřesnit CO, zásady JAK
- Podrobně JAK
- Kódování
- Testování
- Údržba a provoz



Zaměření přednášky

# Úzká místa

- Sehnat kšeft, vize
  - Přesně a správně: proč a rámcově co je třeba udělat
- Stihnout včas za dané peníze
- Řemeslo je důležité, ale ne kritické
  - Na českých univerzitách, především an FI a MFF, velmi dobrá výchova v kódování návrhu i ve světovém srovnání
  - V praxi je kódování menší problém než analýza (vize specifikace, návrh), tam se uplatňují inženýři z technik
  - Roste důležitost inženýringu, hlavní úkoly
    - Specifikace požadavků
    - Procesy vývoje SW, byznys procesy a byznys intelligence
    - Architektury velkých SW systémů
    - Schopnost se domluvit

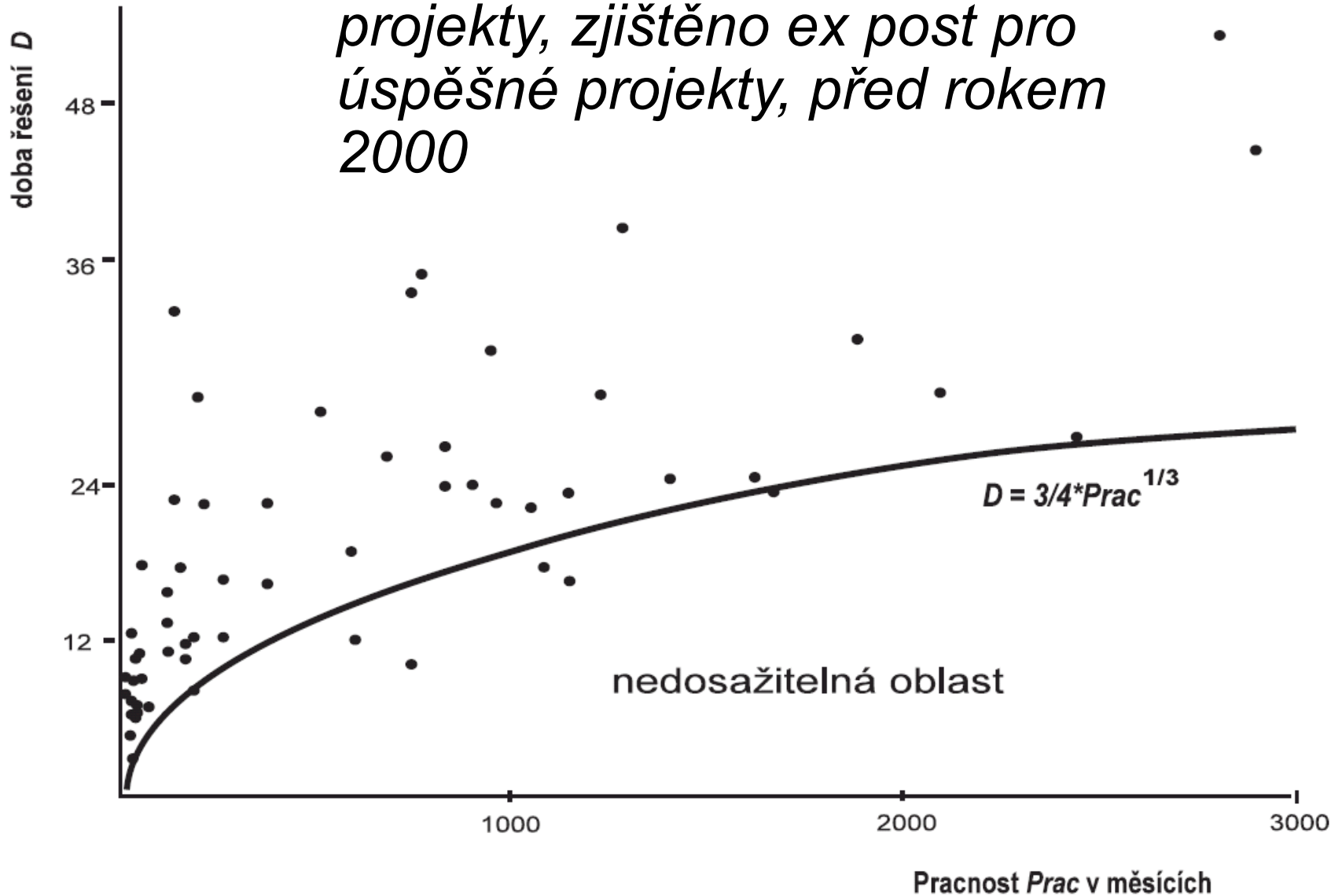
# Tempo zastarávání

- V technologii vývoje SW je 50% používaných znalostí mladší pěti let
  - neplatí to pro principy vývoje
- V analýze je tempo změn mírnější
  - Lidé se mění pomalu
    - Kamkoliv pohlédneš, všude jsou lidé stejní pitomci
      - Nápis na sumerské tabulce z Uru (hlavní město sumerské říše ve druhém tisíciletí před Kristem, podle Bible rodiště Abraháma)
    - Nemá smysl se učit všechny systémy, klíčové ale ano, učit se zvládat nové, umět to s lidmi

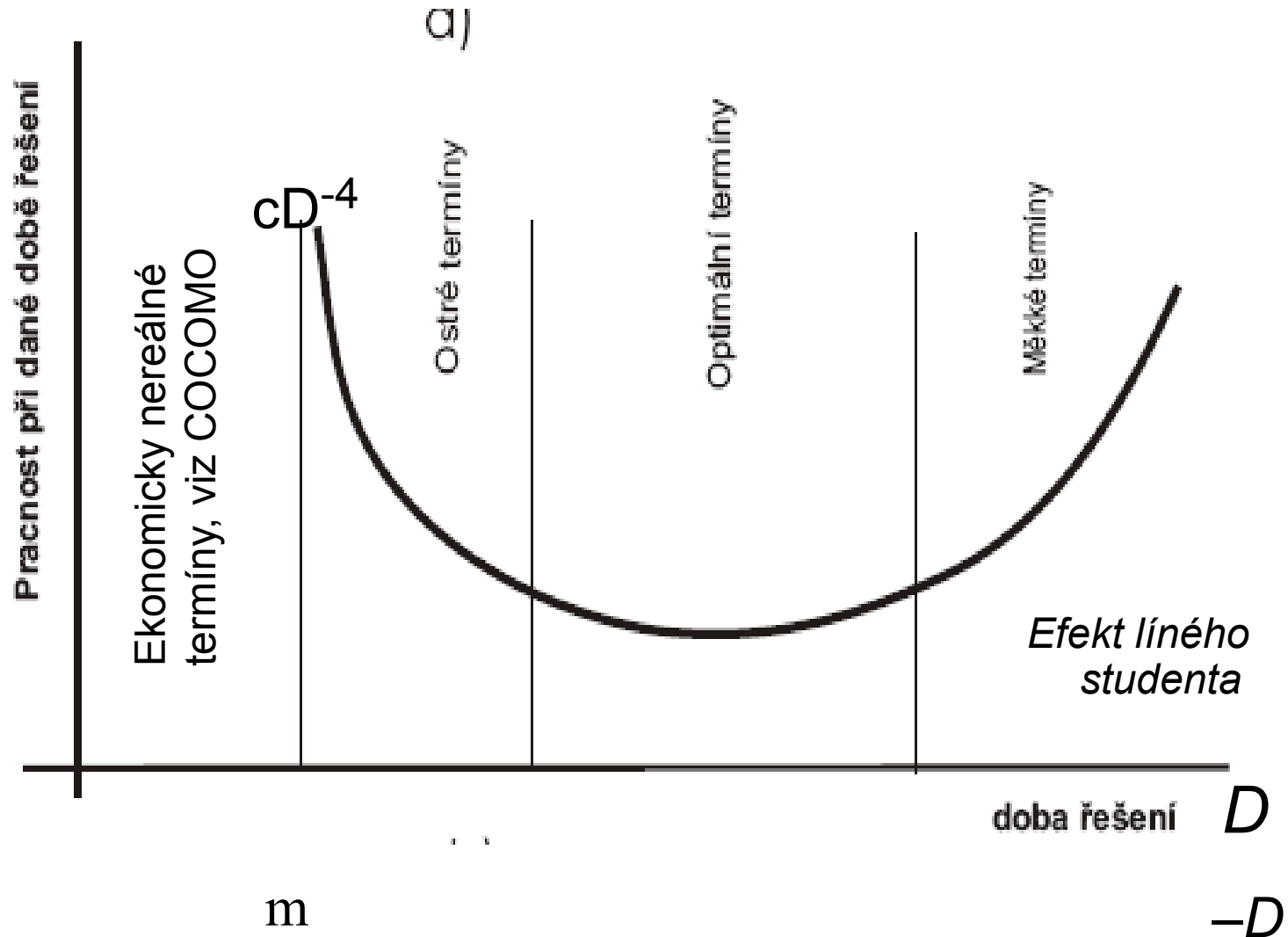
# Kódování

- Řemeslo v IT
- Rychlý vývoj
- Nejméně kritická etapa
- Dobrý kodér je výhra, nahradí i desítky průměrných
  - manažeři ale nechtějí být závislí na špičkových kodérech
- Scrum – místo řemeslníka pomocný dělník

*Některé starší výsledky pro SW  
projekty, zjištěno ex post pro  
úspěšné projekty, před rokem  
2000*



# Pracnost projektu při zkracování termínů, nedosažitelné oblasti



# Důsledek

**ANI VE VELKÝCH FIRMÁCH  
NEMOHU DOBU ŘEŠENÍ  
LIBOVOLNĚ ZKRACOVAT  
VELKÉ SYSTÉMY A VELKÉ TÝMY JSOU  
OBTÍŽNĚ ZVLÁDÁNY.**

# Boehm

Late team increase makes  
projects late

?? Velký tým hned od začátku??  
To taky nejde



# Příklad: Aféra se sociálními dávkami

Nemohlo být hotovo včas  
Je to příklad problému setrvalé  
zastaralosti:

Než udělám opravy je už třeba systém  
zase přepsat, u velkých monolitních  
systémů je to nevyhnutelné

# Tempo zastarávání znalostí v IT,

- V technologiích vývoje SW je 50% znalostí mladších než 3-5 let
  - Systém může zastarat dříve, než se dokončí jeho vývoj
  - Doba vývoje závisí na velikosti systému a nedá se libovolně zkracovat ⇒ dilema stálého zastarávání
  - Budeme hledat řešení v servisních systémech složených z AUTONOMNÍCH ČÁSTÍ, služeb
- Každých asi deset let dochází ke změně paradigmatu (struktury, objekty, cloudy, služby...)
  - Zaměříme se na věci, které mají šanci přežít
    - Inkrementálnost vývoje, zapojení uživatelů, otevřenost

# Důsledek

Pokud je systém velký a aktualizuje se celý naráz pak

**musí být stále zastaralý,**

doba změny je tak velká, že vše mezitím zastarává.

Často se to obchází tím, že se předává nespolehlivý systém

Platí to i pro velké výrobce

# Etapa opotřebení

- Zachovává se celková filosofie architektury řešení
- Nakonec může být zastaralá filosofie hlavním důvodem přepsání systému
- I to je vlastnost technických děl
  - Srv. staré auto

# Celkový trend technik dekompozice a propojování

- Program
- Program a podprogramy, knihovny
- Skupiny podprogramů (objekty) a jejich propojování a modifikace
- Propojování objektů do komponent
- Autonomní komponenty a služby
- SOA s dokumenty, cloudy, sítě komponent, nejnověji dokumenty

# Celkový trend

Stále širší integrace do  
společenských a sociálních  
procesů

Integrace autonomních částí

Otevřenost

# Servisní systémy

- Nezvládnutí paradigmatu SOA byla hlavní příčinou aféry s registry vozidel
  - Viníci jsou za nekvalitu de facto odměněni (za peníze přepsali)
  - Dalo se dělat i pomocí cloudu
  - Přehnaná interaktivnost
    - Každý ťuk se ověřuje v centru

# Úzká místa informatiky

- Zaměříme se na úzká místa vývoje IS , bez jejich vyřešení nedosáhneme úspěchu
  - specifikace a spolupráce s uživateli, společenské souvislosti
  - úkoly řízení projektů,
  - SW procesy především s aplikacemi na SOA
  - SW normy a SW metriky
- Něco se dá jen obtížně na škole nacvičit, lze jen seznámit (MBA lze studovat až po získání praktických zkušeností)
  - Spolupráce s uživateli
  - Tvorba rozsáhlých systémů
  - Management, zvyky ve firmách
  - Dovednosti a zkušenosti



# Co je třeba se naučit

1. Poznatky, které v daném okamžiku tvoří jádro znalostí oboru - jako vklad do začátku
2. Zvládnout při tom dovednost zvládat nové poznatky
3. K tomu je třeba také pochopit a nacvičit používání „filosofii oboru“
4. Každých deset let se ale filosofie mění (změna základních přístupů: strukturovaný, objektový, servisní, ...)
5. Lidé se mění relativně málo, především v přístupu k věci

# Čím se ajťáci liší od ekonomů

- Podobná témata, my je ale probíráme z hlediska IT (např. jak a proč implementovat SOA, triky v SOA, návrhové vzory) a jako součást technických děl
- Máme zkušenosti s realizacemi IS a SW obecně hlavně z technického hlediska
- Musíme být schopni se s ekonomy (manažery) domluvit, budou nám často velet, zvláště pokud budeme kóděři či testěři
- Musíme být schopni spolupráce s uživateli a také s jinými technickými profesemi, se kterými musíme spolupracovat
- Top profese: Project (product) manager

# Potřebné znalostní profily

1. Ekonomické-manažerské znalosti a dovednosti s menšími avšak důležitými znalostmi IT, výkonný ředitel
  1. adaptace SW balíčků
2. Vedoucí vývoje IT s dobrými znalostmi IT i oboru aplikace – pro významnější inovace resp. vývoj a adaptace SW balíčků, projektový manažer
3. Vedoucí prodeje, ne zcela malá znalost technologie IS
4. Technologové – řešení systémových a technických otázek
5. Vlastní programátoři – kódéři ale se schopnostmi návrhu a testování projektů
6. Testéři- ve větších projektech

Uplatníme se v bodech 2-6. Ti s manažerským talentem i v 1. Ti pak jsou velmi žádáni a velmi užiteční

# Co se relativně málo mění

- Lidé
  - Postoje
  - Cíle
  - Předsudky
- Potřeba a dovednost spolupráce
- Nedostatečné chápání ICT jako technologického oboru
- Manažerské nedostatky

# Nové směry v SW, od 1995

- Masové používání objektových technik, UML v modelování
- XML jazyky, web, Java
- ***Servisní orientace!!!***, po roce 2000, komponentové architektury
- Sociální sítě, web 2.0, globální IS
- Aspektové programování, agilní vývoj
- Cloud computing, gridy, (využití volných kapacit, balancování zátěže,)
- Normy ITIL, CMM, COBIT, SOA,

# Hlavní poznatky

- Krize IT se vrací asi po deseti létech, může přijít znovu
- Po krizi se vždy se vrátil zájem o informatiku, ale s podstatně jinými úkoly
- O české informatiky je stále zájem, především o kódéry
- Nově globalizace vývoje SW, hlavně v kódování, správa dokumentů
- Není jasné, jak se projeví dkrize v delším výhledu

# Různé technologie dospějí za různou dobu

- Fast Track 2-3 roky, př. SMS, instant messaging.
  - Podpora od velkých výrobců
  - užitečnost, jednoduchost použití,
  - relativní snadnost implementace a nasazení (využití existující infrastruktury, poměrně malé potíže se smysluplným zadáním).
- Standardní: 5-8 let
- Long fuse: až 20 let, i více vrcholů, nutnost nové infrastruktury, výzkum, zákony, paradigmatata. Internet, nanotechno, data integration, SOA, ?Cloud

# Průběh popularity technologií a systémů. Hype křivka

1. Velká stále rostoucí popularita, **líbánky**
2. Stále bolestnější zjišťování mezí možností dané technologie a proto **desiluze** čili vystřízlivění a cosi jako kocovina.
3. **Krize**, objevení či neobjevení hlavního užitku a optimálního využití
4. Rozvod, nebo soužití - plató těch, co to dokázali správně využít, **fáze dospělosti**



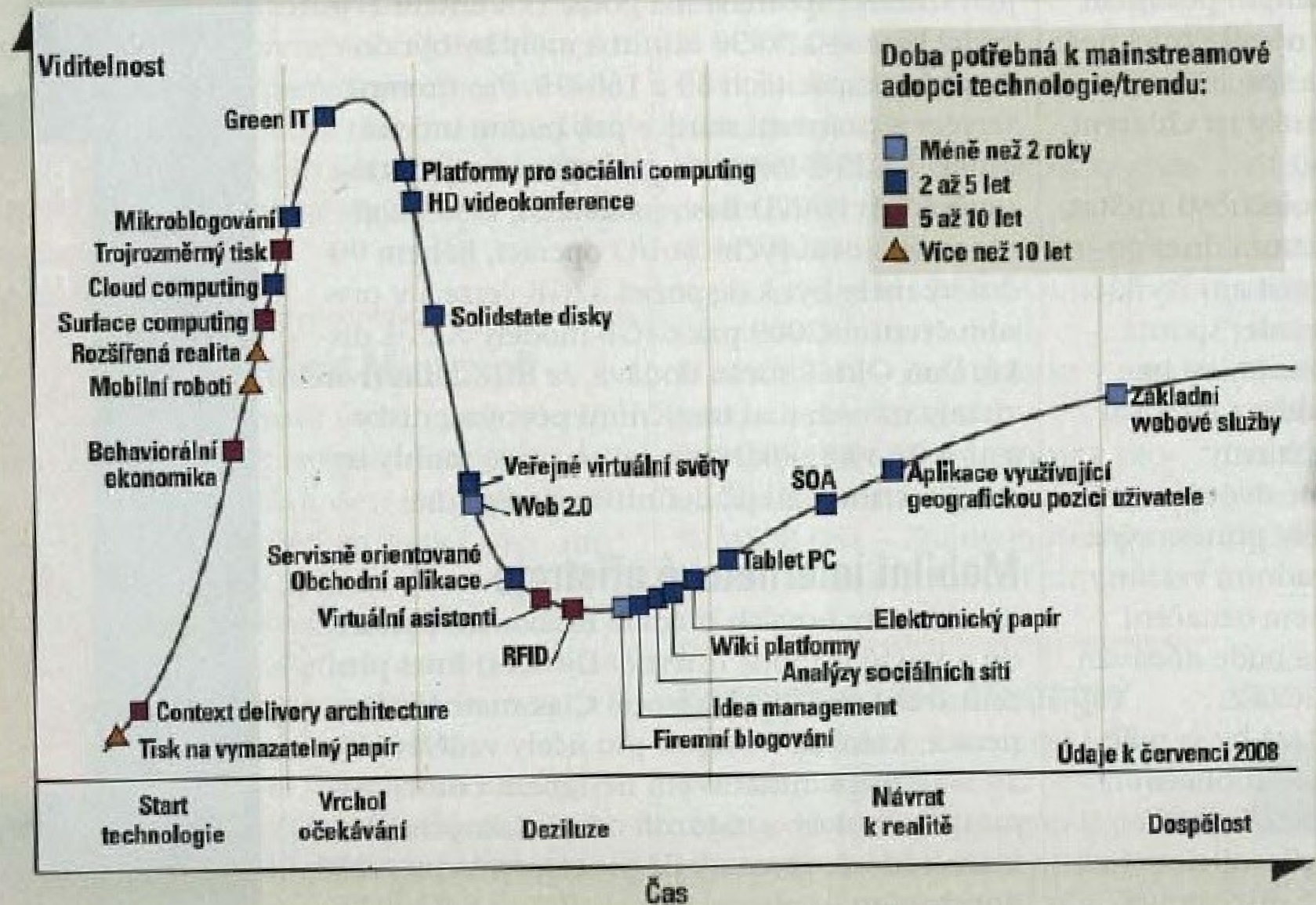
# Příklady na hype křivce

- Etapy přijímání nové technologie
  - **Objev** nového řešení
  - **Líbánky**, nadšení, řeší zajímavé problémy
  - **Vystřízlivění**, má to mouchy, na leccos se nehodí
  - **Plató přijetí**, rutinní používání tam, kde se to hodí nebo postupná ztráta zájmu (rozvod).

# Průběh popularity technologií a systémů. Hype křivka

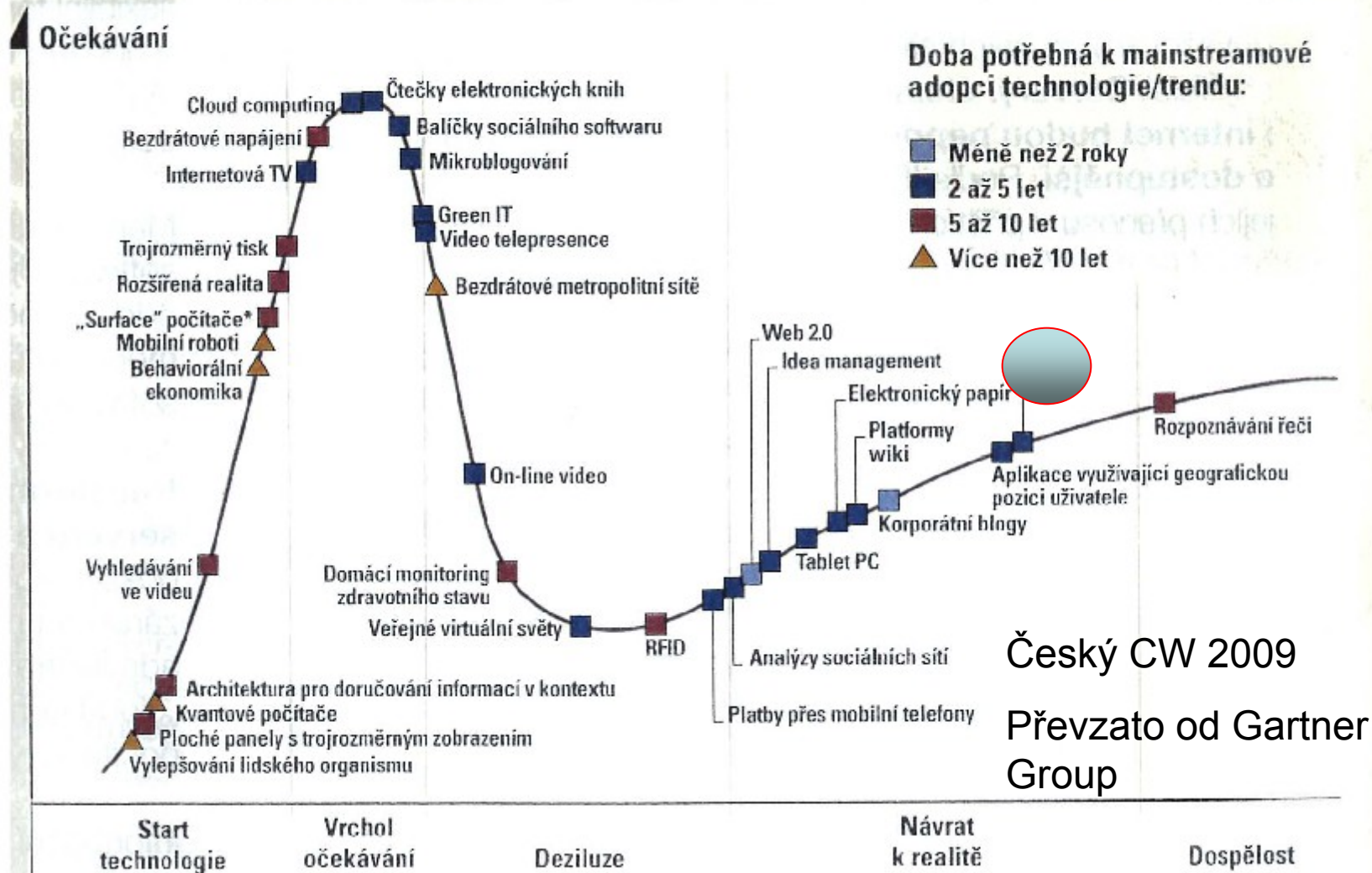
1. Velká stále rostoucí popularita, **líbánky**
2. Stále bolestnější zjišťování mezí možností dané technologie a proto **desiluze** čili vystřízlivění a cosi jako kocovina.
3. **Krize**, objevení či neobjevení hlavního užitku a optimálního využití
4. Rozvod, nebo soužití - plató těch, co to dokázali správně využít, **fáze dospělosti**

# Hype křivka rozvíjejících se trendů a technologií



# SOA – servisně orientovaná architektura

## Hype křivka rozvíjejících se trendů a technologií



# Hodnocení informatiků včetně techniků zaměřených na informatické problémy

- Rychle programují
- Nelze je pustit, alespoň zprvu, k uživatelům, jsou neochotní spolupracovat s odborníky uživatelských domén a s uživateli vůbec
- Často nadměrně nafoukaní a neochotní
- Neochotní používat hotové a podřizovat svůj rozlet přejímání existujících programů
  - Zdá se, že je to méně silné u absolventů technik

# Vyhlídky kódérů

- Vysoce kvalifikované řemeslo, z VŠ velmi kvalitní příprava, nebývá výhodné pro vůdčí pozice
  - Existence superprogramátorů (kódují 20krát rychleji než průměr) Moudrý, Galamboš, Demner, ....
- Je jich kupodivu stálý nedostatek, není ale to, co bývalo (rozhraní, více znovupoužívání, tlak na kodéry dodržovat standardy, nové metody vývoje, sázky manažerů na jistotu)
  - Rychle poměrně vysoké platy, vhodné pro mladé do 35 let, jsou výjimky, pak už menší růst platů a i ztráta schopností, staří pomaleji píšící, lépe si ale věci rozmýšlejí,
  - Poločas rozpadu znalostí do 5 let každých 5-7 let nový programovací jazyk nebo vývojové prostředí
- Každých 10 let nové paradigma (gotoless, strukturovanost, OO, SOA, cloud, sociální SW) a nové procesy vývoje (scrum jako práce o pásu)

# Král – profesní vývoj jako příklad

- 1959 Absolvent MFF UK, matematická statistika, první program na Ural 1, programování v absolutních adresách, není ani assembler, paměť cca 12KB, 100 op/sec, numerika (velmi přesná aritmetika), **zaostávání komoušů**
- 1959 – 1975 Numerická matematika, (generátor náhodných čísel), grafové úlohy pro programy (segmentace), hash metody (chování pro konečné tabulky), servis pro akademii věd, 4 publikace, svépomocný binárně kódovaný assembler (není podpora textů), práce na pč. s pamětí 32kB

# Král – profesní vývoj jako příklad

- 1967 –2005. Makroprocesory, kompilátory, formální jazyky, sítě procesorů. Hlavní výstup kvalitní generátor pseudonáhodných čísel založený na sečítání v 16tubitové aritmetice, nikoliv na násobení, analýza téměř shora, generátor LR analyzátorů založených na rekurzivním sestupu, kompilační techniky, vlastnosti formálních jazyků (nested iterated substitution)



# Král – profesní vývoj

- 1975 – dosud. Řízení výrob a technologií, cca 8 projektů, pět úspěšných. Jedno z prvních uplatnění architektury SOA. Několik desítek publikací
- 1985 - dosud. Architektura SW (SOA), architekturní služby
  - vlákna v COBOLu,
- 1990 - dosud výuka informatiky a její problémy.
- 2005 Vzdělanost, zdravotnictví a kvalita dat

# Od řemesla k analýze a k nekvalifikovaným činnostem

- Nejprve NC stroje, přibyli NC programátoři, ubyli řemeslníci přibyli odborníci na řízení NC strojů, mezioperační skladování a doprava
- Dnes obráběcí centra, úbytek nekvalifikovaných činností, problém programování obráběcích center a analýzy a prodeje

# IS a legislativa

Ochrana dat kontra přínos  
ochrany a ztráty ochrany

# IT a legislativa

## Př. problém ochrany osobních dat

Jak se za cenu velkých ztrát a nežádoucích efektů dosáhne opaku deklarovaného cíle

# Takzvaná ochrana osobních dat jako příklad ztráty kvality dat a chybného řešení

1. Pravidlo: Osobní data se smí shromažďovat, udržovat a používat jen k tomu účelu, ke kterému byla pořízena
2. „Řešení“: Vymazat data, pokud není 1 splněno.
  - Příklad zrušení dat o vydaných receptech
  - Ztrta kvality dat – dostupnosti dat

Výzva: Chrání to skutečně životní zájmy jednotlivých občanů?

Zamlčený předpoklad: Tím se podstaně lepší ochrana osobních dat z hlediska občana

# Main failure of public information systems

Should provide crucial services, e.g. access to any important publishable information, and assure personal data security and **also** other basic human rights

The systems, however, provide  
neither personal data security , neither access to  
publishable info

nor the quality of crucial services

Moreover, it threatens some human rights

# Brute data security rules implied by laws

1. Personal data can be collected and used for the purposes only they were created
2. Any information computed from sensitive (personal) data must be treated as sensitive.
  - Such information is not open irrespective of its content that could/should be open
3. Any collection of personal data not satisfying previous points must be erased

# Limits of brute rules

- Reasons for the rules
  - (Mis)interpretation of Universal Declaration of Human Rights
  - Prejudices of public especially the Big Brother hysteria
    - State has too much information on me
  - Lobby interests
- Undesirable effects of the rules
  - The rules to some degree protect some rights but threaten others
  - They increase the threats of Big Godfathers
  - They in principle cannot assure personal data security



# Brute rules in action 1

System for the prevention of the production of the narcotic  
Pervitin

- Principle:
- On line control of the cases, when anyone has purchased too many drugs containing pseudoefedrin (needed for the production of the narcotic Pervitin) lately all over the Czech Republic
  - Personal ID's had to be used in a database of all drug purchases. The data disclose health information.

# Brute rules in action 2

System for the prevention of the production of the narcotic  
Pervitin

- **Results:**

- A very effective use of SOA, effective implementation
- The production of Pervitin was substantially reduced after the system had started
- Potential opportunities
  - On-line prevention of improper medication, it could save a lot of lives (hundreds, maybe thousands in Czech Rep.) and prevent hundreds of thousands health damages (estimations for USA 50000 and 1.2 milions a year respectively)
  - Evaluation of health institution (hospitals)
  - Optimization of health expenses
  - Epidemic prevention and control
  - Support of medical research and the on line control of the drug use effectiveness, Etc

# Brute data security rules in action 3

- Pity end of the system: The system was forbidden as its database did not comply with the brute rules
- Consequences
  - *The production of Pervitin was resumed*
    - Tragedies of drug consumers and their families
    - Growing power of criminal structures
  - *The opportunities were lost*
    - Lost lives due to a wrong cure (not too small number of cases, comparable with the number of deaths in traffic incidents)

# Brute data security rules in action 4

- Hidden consequences:
  - The analysis of the huge amount of data collected by health assurance institution and health institutions is blocked,
    - The above opportunities are missed
    - The physicians and health institutions are not properly informed on patients health and medications and their effects
    - People are induced to apply improper security disciplines (not to use info critical for patients)
    - Many cure procedures are needlessly repeated

# Summary of brute rules effects

- The effects of the rules are the negations of declared goals as they threaten some rights
  - The effects jeopardized lives or health of patients
    - It threatens the right for live
  - Some open information is not available although it should be used to evaluate/access
    - Aspects of the health care quality
    - Epidemic information
    - Etc.
  - People must invest into health more than necessary
  - Personal data leakage prevention is not substantially enhanced by the current data security regulations as there are many data leakage channels. People and majorities are not aware of it.

# Multiple data leakage channels

- Open data state institutions (land or enterprise registers, ....), data leakages from state institutions
- Financial institutions, e-commerce, etc
- Social software and generally Internet if somebody is not careful enough,
  - It is often very difficult to be careful enough
  - Dangerous habits
- Mobile phones
  - Can be tapped or monitored, even from satellites (positions, the communication can be decoded, ...)
  - Log files at the operators of mobiles
- Servers are not fully immune against hacker attacks,
- etc.

# Great financial data leakage, German government must act against crime

- German government has bought and will again buy disks with financial data of Swiss banks to prevent tax evasions of German citizens

# Data security and education

Issues and effects similar to those discussed above,

- especially important in post-communistic countries,
- Obama and others complain on U.S. education quality
- What schools and study profiles should be preferred
  - Measure: What schools have the most successful graduates/alumni according to the criteria of individual evaluators.
  - What education profiles are the most successful ones
- Problem of STEM (science, technology, engineering, mathematics) education
  - falling popularity,
  - what are the STEM job perspectives



# Data security rules threaten basic human rights in education

- It is difficult for people to decide properly, it is dangerous to the prosperity of whole society
  - Lack of needed professions, decay of basic education services, fall of the quality of education
  - International competition issues
- The investments into education are not optimal, attempts to introduce school-fees
- Lost talents (i.e. it is in fact equal to the vasting of a limited natural resource)
- Danger of social instability

# Other consequences of improper data security procedures

- Macroeconomic data
  - Monopoly of analytical firms, the firms, however, have failed to forecast coming crisis
  - State organizations have failed as well
- Copyright practices
  - Monopoly of large editors
  - Copyright transfers from people to companies

# The brute rules threaten whole IT

- The brute data security rules reduce in fact substantially the data quality,
- It seems to be the bottleneck of many crucial IT projects
  - The effects of the projects cannot be bettered unless the rules are changed
  - Many aims of IT cannot be then achieved irrespective massive investments
- It is a danger for the whole IT (informatics)

# Takzvaná ochrana osobních dat 2

## Skartace zdravotních dat

V daném případě:

- Mohou chybět data o právě prováděných medikacích a pak může být fatální průšvih
  - Stává se omylem lékaře nebo tím, že je nějaký incident jako dopravní nehoda
  - Pacient na léky, které užívá, neupozorní
  - Ohrožuje to životní zájmy lidí, stovky, spíše tisíce úmrtí, kterým nemuselo dojít, statisíce poškození zdraví (v USA 1.2 milionu ročně)
- Blokuje se optimalizace výdeje léků
  - Mnohamiliardové ztráty
- Uvolňuje se prostor pro produkci drog, příběh pervitinu
- Zhoršují se podmínky zdravotnického výzkumu
- Klíčová osobní data se stejně neochrání
- Naprostá většina informací, které jsou pro občany zajímavá, jako je kvalita škol, by měla být zpřístupnitelná ze zákona, jsou často jsou blokována

# Takzvaná ochrana osobních dat 3

## Analýza efektivity vzdělávání

### Hrozba

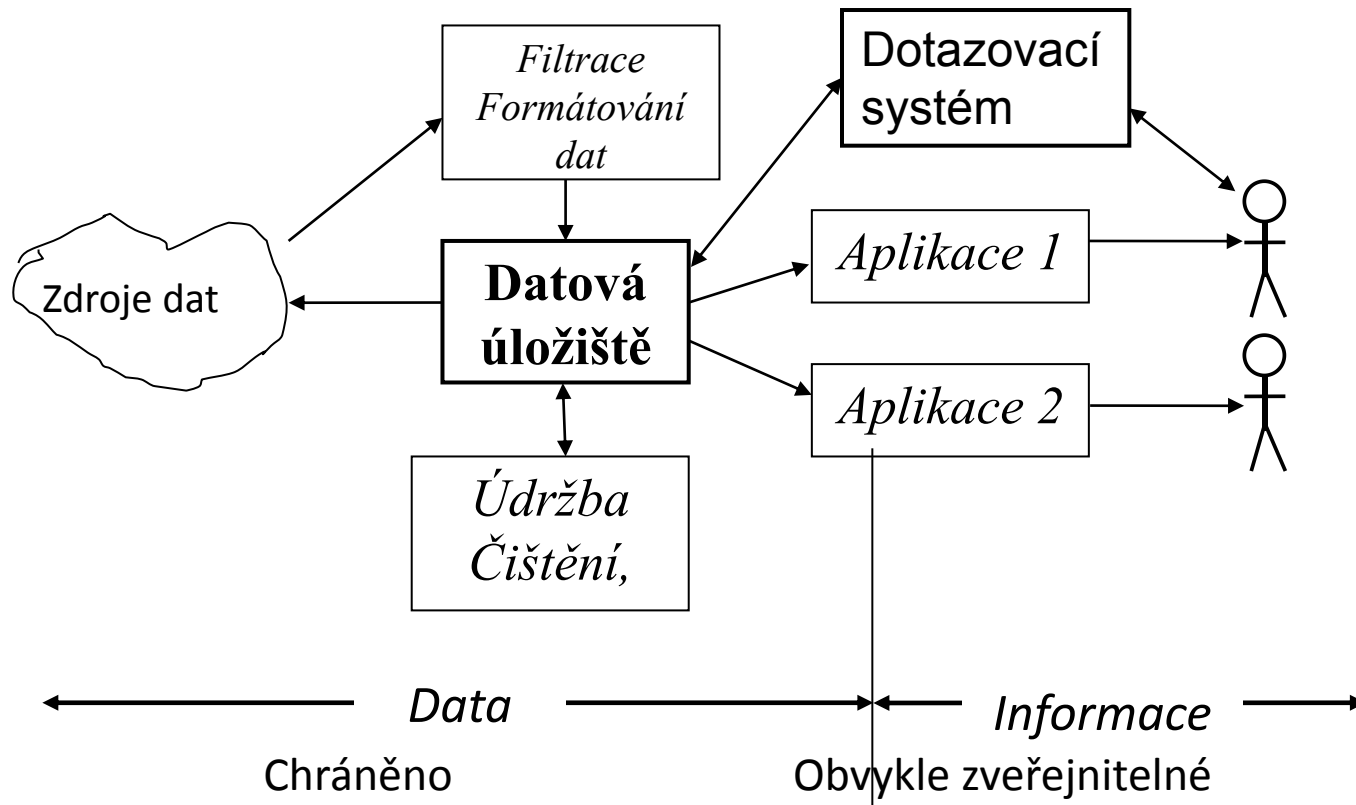
- Dosti rozšířený a asi správný pocit, že se naše školství vyvíjí špatným směrem
  - Opomíjení STEM, To pocítujeme i u nás, je problém v podnicích (průzkum Manpower)
  - Opomíjení tréninku dovedností. Včetně trivia a cizích jazyků, pologramotnost
  - Nerovnoprávné postavení učitelů
    - Za stížnosti žáků je trestán učitel, za nedodržení osnov fakticky nikoliv, příklad Zborovská, Evropská, náznaky i na MFF
  - Administrativní náročnost chodu škol
    - Ekonomická samostatnost – inspekce sledují účty a ne výuku
    - IT by mohlo sledovat úspěšnost všech zařízení podle úspěšnosti absolventů
- Jistou pomocí by mohlo být sledování úspěšnosti absolventů
  - Není o to zájem, nedalo by se tunelovat, rodiče by se museli o ratolesti více starat
  - Naráží to na problém ochrany dat
  - Je to lepší než nic
  - Ohrožuje to i informatiku, nebudou odborníci

# Takzvaná ochrana osobních dat 4

## Informatický výzkum, kvalita IS

- Nelze plně využít výsledky informatického výzkumu, poněvadž přístup k datům není „hladký“
  - Sémantický web
  - Použití metod umělé inteligence
  - Možnosti jednotného pohledu a hladkého přístupu ke všem dostupným datům

# Chráněná data a otevřené informace



Informace je zde to, co je výstupem příslušné aplikace (pro uživatele)

# Takzvaná ochrana osobních dat 4

## Analýza efektivnosti datové skartace

Virtuální skartace je v principu neúčinná, neboť nemůže zásadně omezit únik dat (vlastně těch nejdůležitějších)

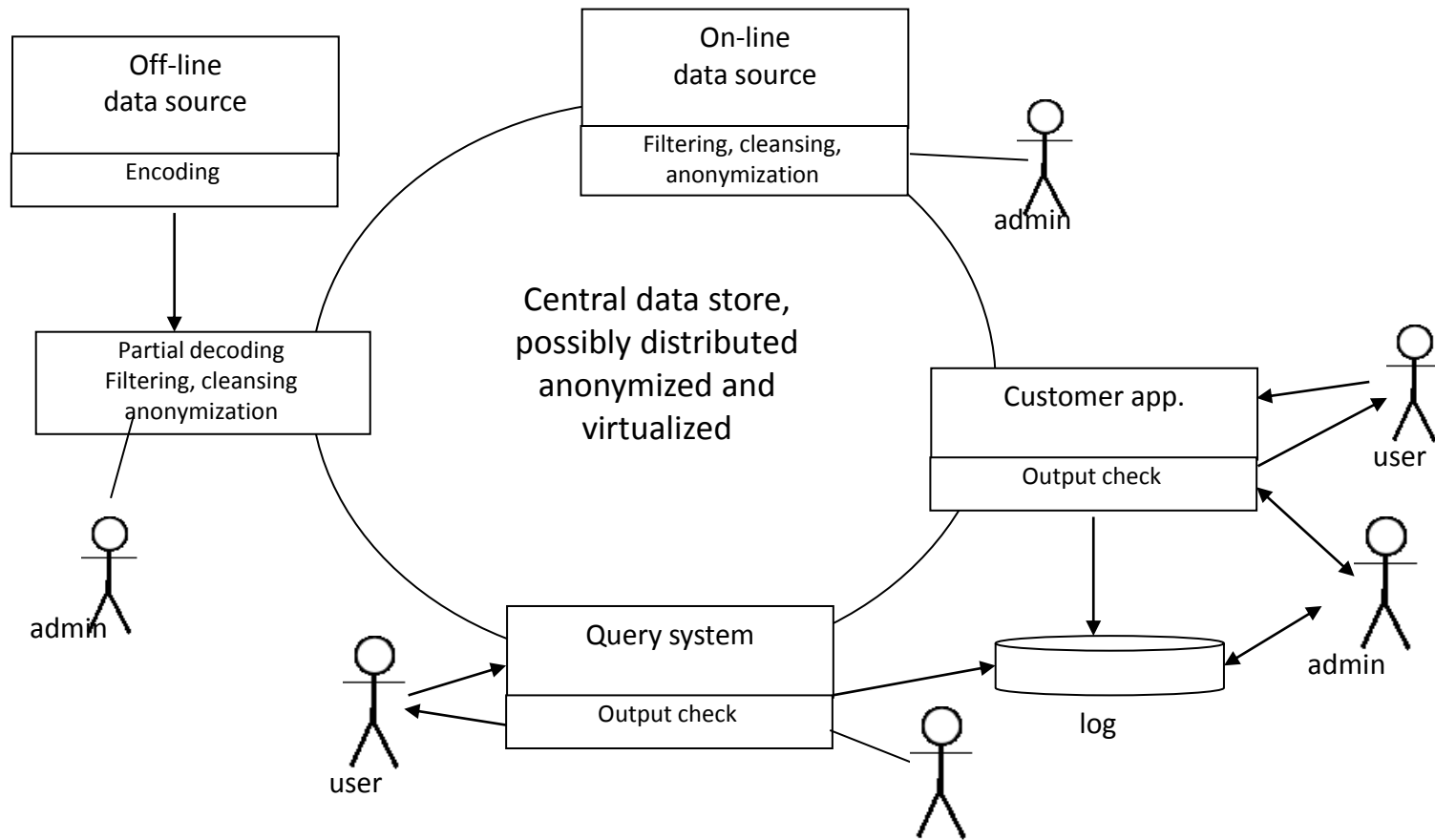
- Pro mnohé je nesmyslná, proto ji nepodporují, a tudíž není úplná a včasná
- Osobní data se shromažďují za různými účely a mohou tedy unikat - a také unikají - mnoha dalšími cestami, legálními i nelegálními
  - Sociální sítě
    - To je výzva i pro nás, jak to zlepšit
  - Různé rejstříky (obchodní, katastry, ...)
  - Finanční instituce
  - Mobily, ty může sledovat i družice
  - Webové služby
    - a-maily, e-komerce
    - Různé procedury zpřístupňující otevřené dokumenty na webu
  - Imigrační a cestovní procedury, např. USA



# Výpočet otevřených informací s použitím osobních dat

- Data se shromažďují u důvěryhodné instituce
  - instituce data a soukromí chrání
    - Organizačně s použitím známých technik
    - Částečným zakódováním (identifikátorů subjektů)
    - Kontrolou výstupů výpočtů, zda se z informací nedá odvodit, ke komu se vztahují
  - Instituce poskytuje základní dotazovací systém a umožňuje po prověření nezávadností aplikací vyvinutých uživateli integraci a používání těchto aplikací

# System chránící data a zpřístupňující otevřené informace



# Co chybí při ochraně osobních dat

- ÚOOÚ by měl prokazovat, že je skartace skutečně nezbytná,
  - měl by doporučovat opatření (důvěryhodnost instituce, anonymizace dat, ...), opatření by měl navrhnout sám, nebo si je dát navrhnout od expertů s cílem vyhnout se skartaci
  - Měl by mít povinnost vyhodnocovat ztráty vyvolané skartací a to i na podnět občanů a institucí
- Ale to je legislativní změna!

# Od dostupnosti k nedosažitelnosti

## Zavalí nás informační odpadky?

- Stále větší množství smetí, přes které se musím probojovávat k tomu, co potřebuji, leccos nakonec nenajdu
- I užitečných/relevantních informací je stále více a pro menší skupiny je příliš nákladné všechny využívat
  - Některé informace jsou jen v knihách a reportech, jiné v dokumentaci SW artefaktů
  - Je obtížné se k některým dokumentům prokousat
- Informační prostor stále více vyplňují velcí vydavatelé a vnucují ostatním svoje pravidla hry

# Od dostupnosti k nedosažitelnosti

## Konec copyrightu?

- Copyright:
  - Založeno na principech z 19. století
  - Vede dnes i k tomu, že se musíme vzdávat svého autorského práva ve prospěch vydavatelů
    - Nemohu jako autor říci, že daný vzorec lze bez omezení používat,
      - Einstein by měl dnes možná potíže
    - Problémy s návrhem a využíváním digitálních knihoven
      - Není to velký problém technický, ale legislativní

# Základní pojmy

Výchozí stanoviska a postupy

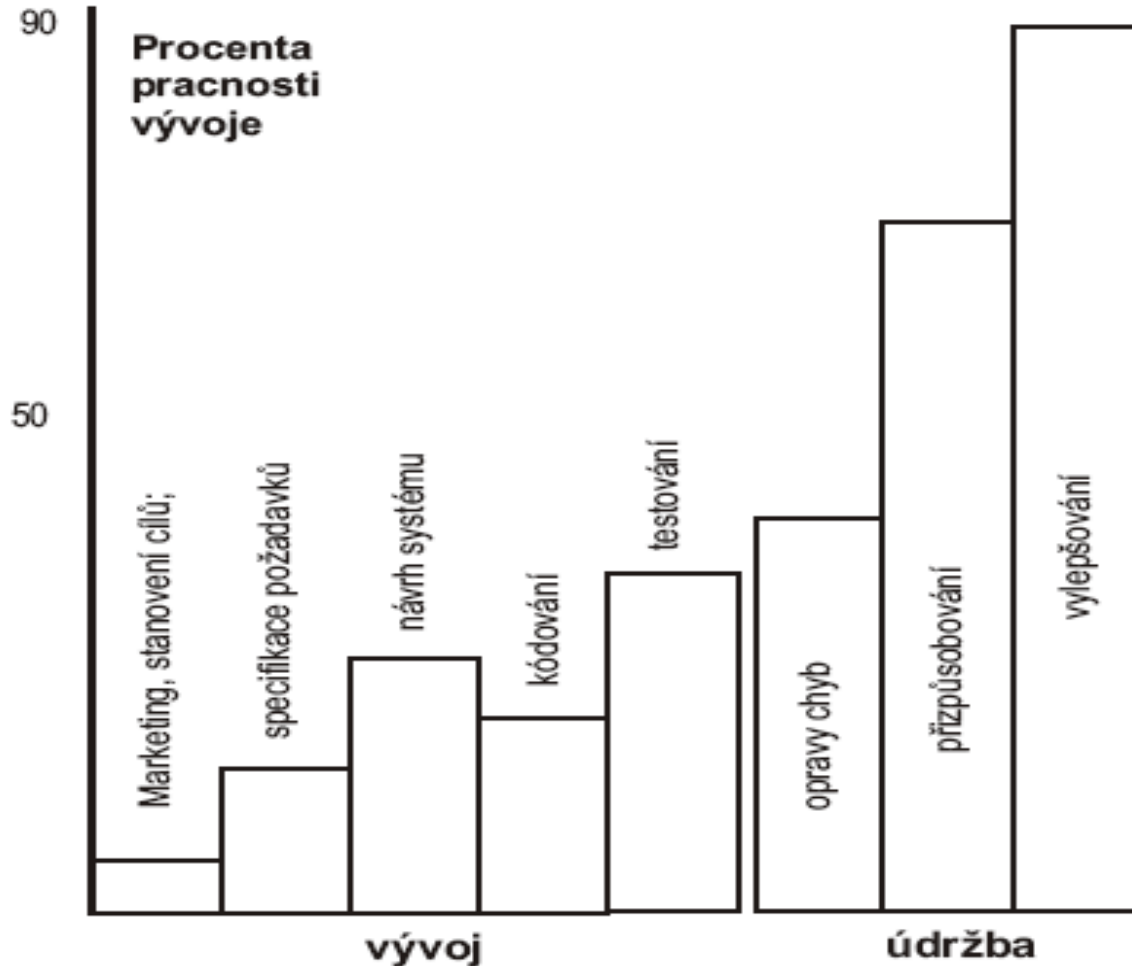
Shrnutí známých skutečností

# Životní cyklus projektů a „vodopád“

- Vize (**proč** je co třeba)
- Specifikace požadavků (**co** je třeba)
- Návrh (**jak** to asi udělat)
- Kódování (psaní programů, u nás často spojováno s návrhem)
- Testování (kodér: testování částí; testér: testování integrační, funkcí, systému)
- Předání
- Údržba
- Zrušení

Pokud se takto člení i projekt a nemáme možnost se vracet (měnit např. požadavky při kódování), mluvíme o metodě vodopádu. Ta může být vhodná, dokonce podle příslušných norem nutná, pro technologicky orientované systémy. Vodopád v původní verzi se neosvědčuje pro human oriented systems, především pro IS, moderní procesy vývoje, většinou využívají modifikace vodopádu (iterace, inkrementy, prototypy)

# Podíly pracnosti ve vodopádu



Údržba stojí podstatně více než vývoj (obvykle dvakrát více).

U customizovaných systémů je podíl údržby pro jednotlivé instalace podstatně menší (platí se tím, že se zákazník musí přizpůsobovat)



# Co budeme dělat

Celý životní cyklus IS, důraz na počáteční etapy vývoje, architekturu SW a management SW prací se zaměřením na IS.

- Životní cyklus softwaru, prvé poznatky o SW architekturách
- Co je IS a proč je jeho vývoj složitý
- Servisně orientovaná architektura a jiné architektury
- Vize, před uzavřením smlouvy, smlouva, správa rizik
- Základní techniky při specifikaci požadavků
- Společenské a zdravotní souvislosti IS
- Kvalita dat
- Varianty životního cyklu, agilní formy vývoje
- Perspektiva profese informatik
- Práce v týmu (základní poznatky)

# Problém vejce a slepice

- Témata vzájemně souvisí – je nutný výklad po etapách a vracet se, k tématům dnešní přednášky se vrátíme a probereme podrobněji
- Mnohé problémy zasahují mimo kyberprostor
  - Důležité pro analýzu a vaše uplatnění mimo informatiku a pro dobré joby v informatice
  - Důležité pro možnost získat lukrativní místa
  - Obtížné, hackerský syndrom (já jako programátor se starám jen o programy respektive o SW obecně, vše ostatní je blbost, v této oblasti skutečně převyšuji ostatní)

# Co způsobuje největší problémy

Zdánlivé hlouposti (blbosti), dokonce i skutečné triviality

# Nejhorší je srážka s blbcem Zvláště fatální je varianta blbec – tunelář a ignorant

- Mnohé efekty IT se těžko měří a není k dispozici dostatečně kvalitní a uznávaná metodika hodnocení kvality
  - To především mimo podniky, ale i v podnicích umožňuje zakázky zdražovat
    - Přímo, vyhnat cenu za málo
    - Nepřímo. Zakázku nafukovat
  - Dlouhodobě to škodí všem

# Pozor na hackerský syndrom

- Přesvědčení, že svět mimo kyberprostor je nezajímavý a že hlavní je jak rychle píše programy a detailní znalost systémů, pohrdání uživateli IS
  - Neschopnost a neochota spolupracovat s uživateli
  - Neschopnost domluvy s jinými hackery
  - Neochota pracovat v týmu a dokumentovat své výtvoary, antivzor Corncob
  - Neschopnost zvládat cizí znalostní obory
  - Neochota přebírat to, co je.
- *Hackeri jsou použitelní jako programátoři úloh s jasným zadáním vyvíjenými obvykle od začátku*
  - *Takových úkolů je stále méně*
- *Jak je to s open source?*

# Využitelnost přednášky podruhé

- Přednáška je zaměřena především na znalosti potřebné pro
  - SW inženýry a softwarové architekty
- Přednáška obsahuje poznatky využitelné
  - Kodéry, testéry, údržbáři a obchodníci
- Zčásti se týká i témat, které jsou dosud převážně ve stadiu nekomerčního výzkumu a vývoje.
- Problém“nezakusíš – nepochopíš“
  - Některé zkušenosti jsou obtížně sdělitelné, mohou se zdát i samozřejmé dokud jejich těžké dopady nepocítíte na vlastní kůži

# SW architektura

- Organizace a struktura systému ve velkém
  - Dekompozice na nejvyšší úrovni do kooperujících částí (pokud možno autonomních), skládání komponent do sítí-sestav-vrstev
  - Principy spolupráce s uživateli
  - Základní vlastnosti částí a jejich rozhraní
    - P2P – typ komunikace
    - Klient-server, tři vrstvy (i prostřednictvím stanovení funkcí uzlů sítě – globální členění)
- Struktura tvořená SW komponentami, jejich vztahy, principy vývoje a integrace
- Jiné formy, především API

# SW architektura - účel

- Specifikace a návrh ve velkém
- Dekompozice
  - Dá se pak mentálně a organizačně zvládnout i velký a komplikovaný systém
    - Nezávislý vývoj komponent
    - Znovupoužitelnost komponent
    - Nabízí i technické výhody (prototypování, údržba, ...) a možnost specifických funkcí), inkrementální vývoj
- Na architekturu vázané procesy a funkce (decentralizace).
  - Inkrementální či iterativní vývoj
- Evoluce, škálovatelnost a modifikovatelnost systému
- Distribuovanost



# Komponentové architektury

- Větší celky (komponenty), základní funkce
- Konektory – entity umožňující komunikaci a spolupráci komponent
  - Umožňují obvykle výměnu zpráv
- Principy podobné jako b v objektové orientaci

# Servisně orientovaný systém

- Vazby mezi komponentami jsou volné, komponenty spolu komunikují podobně jako služby reálného světa nebo webovské služby na internetu – vyřizují požadavky z fronty požadavků - jinými slovy systém se chová jako virtuální p2p síť s asynchronní komunikací.
  - Sekvenční komunikace je možná, je doplňkovou možností
- Je to vedoucí paradigma současného SW inženýrství, zvláště v případě velkých informačních systémů a na Webu
- Nutné používat standardy (UML, w3c, OASIS, Open Group)

# Potíž s human oriented systems

často nevíme ani my sami ani uživatelé si často s nevybaví, co by měly přinést a jak to, co přinese, měřit, požadavky se mnohdy mění podle toho, jak se mění byznys

- Efekty IT jsou často jinde, než se čeká
- Uživatelé si obtížně uvědomují své potřeby. Vybaví si je až když nastane příslušná situace
- Obtížně se měří
- Projeví se až po jisté (někdy dosti dlouhé) době
- Dlouhodobé přínosy jsou jinde než krátkodobé

Otevřený problém: jak vlastně  
IT ovlivňuje svět  
Vliv IT na makroekonomické  
ukazatele (1992)

Podle T.K. Landauer, *The Trouble  
with Computers*. MIT Press, 1993

Pouze pro studijní účely

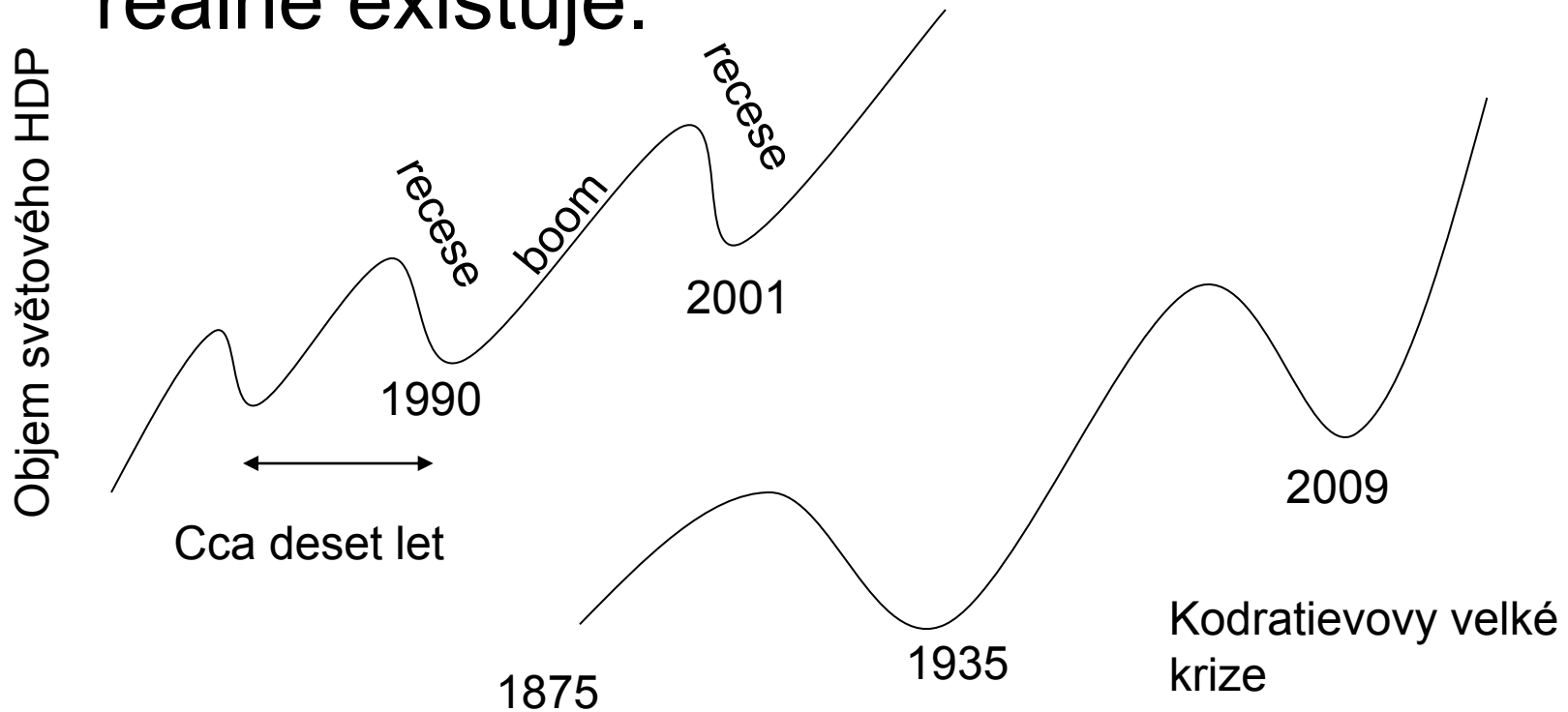
# Produktivita po zavedení počítačů rostla pomaleji

	1950-1973	1973-1990
Francie	5%	3%
Japonsko	7,5%	3,5%
Německo	6,0%	2,5%
Britanie	3,5%	2,0%
USA	2,7%	1,0%

McKinsey Global Institute 1992

# Námitka

Hranice zvolených období padla do recese (důsledek první ropné krize). Problém ale reálně existuje.



# Trendy růstu produktivity v jednotlivých odvětvích

Porovnání období 1948 – 1973 proti 1973 – 1987

- Růst:
  - Zemědělství
  - Některé výrobní obory (elektro, textil, stroje)
- Zmírnění růstu
  - Průmysl celkem,
  - Služby, obchod
- Pokles
  - Banky, publikování, obchod z nemovitostmi

# Důvody

- Banky přecházely na nové typy činností (karty), dnes jsou v balíku
- Nízké počty ks na jedno vydání **knihy**, rychlost vydávání, autoři si sázejí sami, nedomyšlenost



# Možné vysvětlení

- Editace – nová kvalita: rychlost publikování, autoři si sázejí knihu sami
- Banky
  - nové služby (platební karty), které ještě nebyly plně zvládnuty
  - Lidé museli po určitou dobu podporovat staré procesy,
  - Práci s novými technologiemi a museli naučit pracovníci i zákazníci

Nevýrazná až záporná korelace  
mezi investicemi do IT a výší  
dividend a také produktivitou

Jedna z příčin:

Do IT investují ti, jimž teče do bot

Ale také ti, co jsou předvídaví a investují do  
budoucnosti a zisk “neprojí“, takže není v  
dividendách, banky byly v balíku a nic je k  
investicím nenutilo,  
mohlo to být tehdejší fází cyklu (boom)

# Klíčové problémy jsou spojeny se systémy poskytující informace tj. informačních systémů

- Do takových systémů jde více než 90% investic a zaměstnávají přímo či nepřímo 90% aťáků.
- S informačními systémy jsou spojeny hlavní výzvy a hrozby a také teoretické problémy informatiky
- V dalším budeme analyzovat hlavně problémy spojené s informačními systémy, mnoho těchto problémů je spojena s tvorbou a užitím softwaru obecně
  - Softwarové systémy se navíc vyvíjejí směrem k otevřeným informačním systémům

# Přínosy informatiky

- Nelze popřít, že informatika podstatně přispěla k tomu, že se máme docela dobře a možná i k tomu, že se neválčí
  - Efektivnost výroby a logistiky (to hlavně)
  - Výkonnost zemědělství
  - Pokrok vědy a technologie.
  - Zdraví, tam jsou velké rezervy!!!
  - Globální ekonomika (ale zranitelnost, Fukušima a výpadek výroby jediné továrny na výrobu klíčových komponent aut)

# Hrozby a bariéry

- Copyright proti potřebě
  - zdravotní data a léčba
  - Informace o kvalitě vzdělání
- Sociální sítě a gamblerství a mezilidská spolupráce
- Nejasné efekty

# Zvláštnosti informatiky a IS

Silně ovlivňuje chod společnosti a umožňuje plynutí a usnadňuje úspory.

Přínos IT je ve společnosti často záporný nebo jiný, než se čekalo,

*IT* přináší zbytečnou složitost do společenských procesů (e-government)

Může to ohrozit celou společnost, viz Tainter:  
Kolaps složitých společností, ty zahynou na rostoucí složitost procesů státní správy i velkých korporací

# Zvláštnosti informatiky a IS

Složitost ve zdánlivě jednoduchých věcech.

Často je odpověď na určitou otázku (technicky) jednoduchá, problém je s formulací správné otázky (odpověď může být doprovázena poznámkou, „to je jasný“ a při tom se důsledky odpovědi neakceptují), příliš se zapomíná a to, že součástí systému jsou vlastně lidé a jejich zájmy zvyka a znalosti

Př. Lůžka intenzivní péče a rozhraní pronižší zdravotnický personál, použití tabulkových kalkulátorů jako klientské vrstvy

Př. COBOL kontra PL/1 a Algol, skutečné potřeby účtařů

# Zvláštnosti výzkumu v informatice

- **Výzva:** Moorův zákon: jak v IT publikovat a jak se školit, když za rok je mnoho znalostí zastaralých (do pěti let je polovina nových),
- Další problémy
  - Jak hodnotit SW artefakty
  - Časopisy nejsou dost rychlé, takže se v nich dá rozumně publikovat jen něco, navíc není výjimkou, že se nové objevy tématicky nehodí do žádného časopisu
    - Problém hodnocení informatického výzkumu
    - Meyer et al, CACM April 2009
  - Důsledek: Mnoho poznatků je publikováno jen v knihách, sbornících, výzkumných zprávách a dokonce v dokumentaci SW systémů



# Zvláštnosti informatiky

- Výzva: Celoživotní vzdělávání je v informatice nutností
  - Má to dělat i univerzita?
  - Je to problém i mimo informatiku!!
  - Staří mají jiné dispozice než mladí, má cenu do nich investovat?
    - Některé dispozice starších se dají v IT dobře využít, někdy se lze bez nich dost těžko obejít.
  - Jak na to při vzdělávání?

# Zvláštnosti informatiky

- Výzva: Celoživotní vzdělávání v informatice je nutností
  - Má to dělat i univerzita?
  - Je to problém i mimo informatiku!! Knihy o IT jsou zčásti zastaralé už když vyjdou
  - Staří mají jiné dispozice než mladí, má cenu do nich investovat!
    - Některé dispozice starších se dají v IT dobře využít, někdy se lze bez nich dost těžko obejít.
  - Jak na to při vzdělávání?

# Zvláštnosti výzkumu informatiky

- Výzva: jak publikovat, když za rok je mnoho znalostí zastaralých (do pěti let je polovina nových)
  - Jak hodnotit artefakty
  - Časopisy nejsou dost rychlé, takže se v nich dá rozumně publikovat jen něco, navíc není výjimkou, že se nové objevy tématicky nehodí do žádného časopisu
    - Problém hodnocení infromatického výzkumu
    - Meyer et al, CACM April 2009
  - Důsledek: Mnoho poznatků je publikováno jen v knihách, sbornících, výzkumných zprávách a dokonce v dokumentaci SW systémů