

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

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

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 nad 25

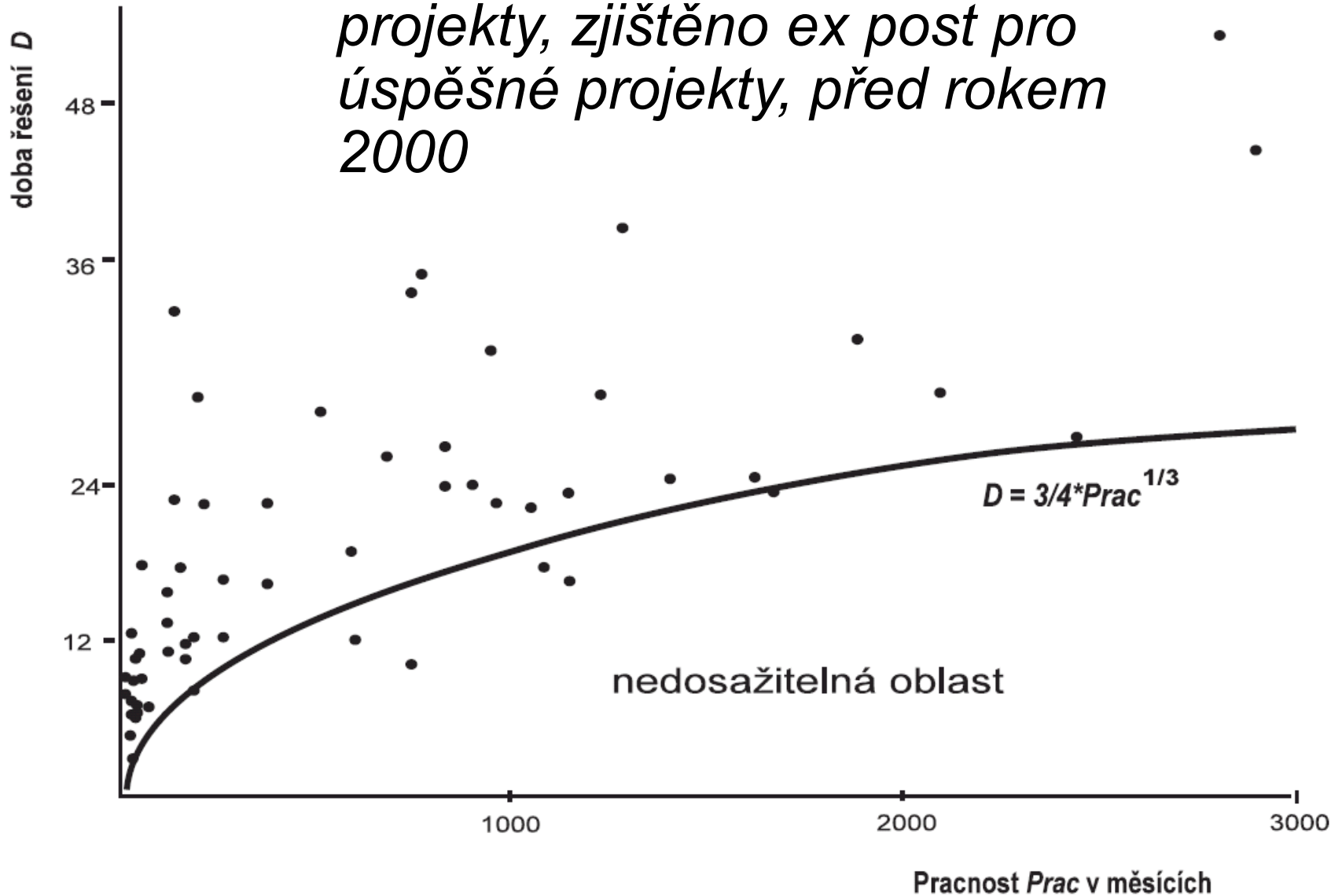
Tempo zastarávání

- 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

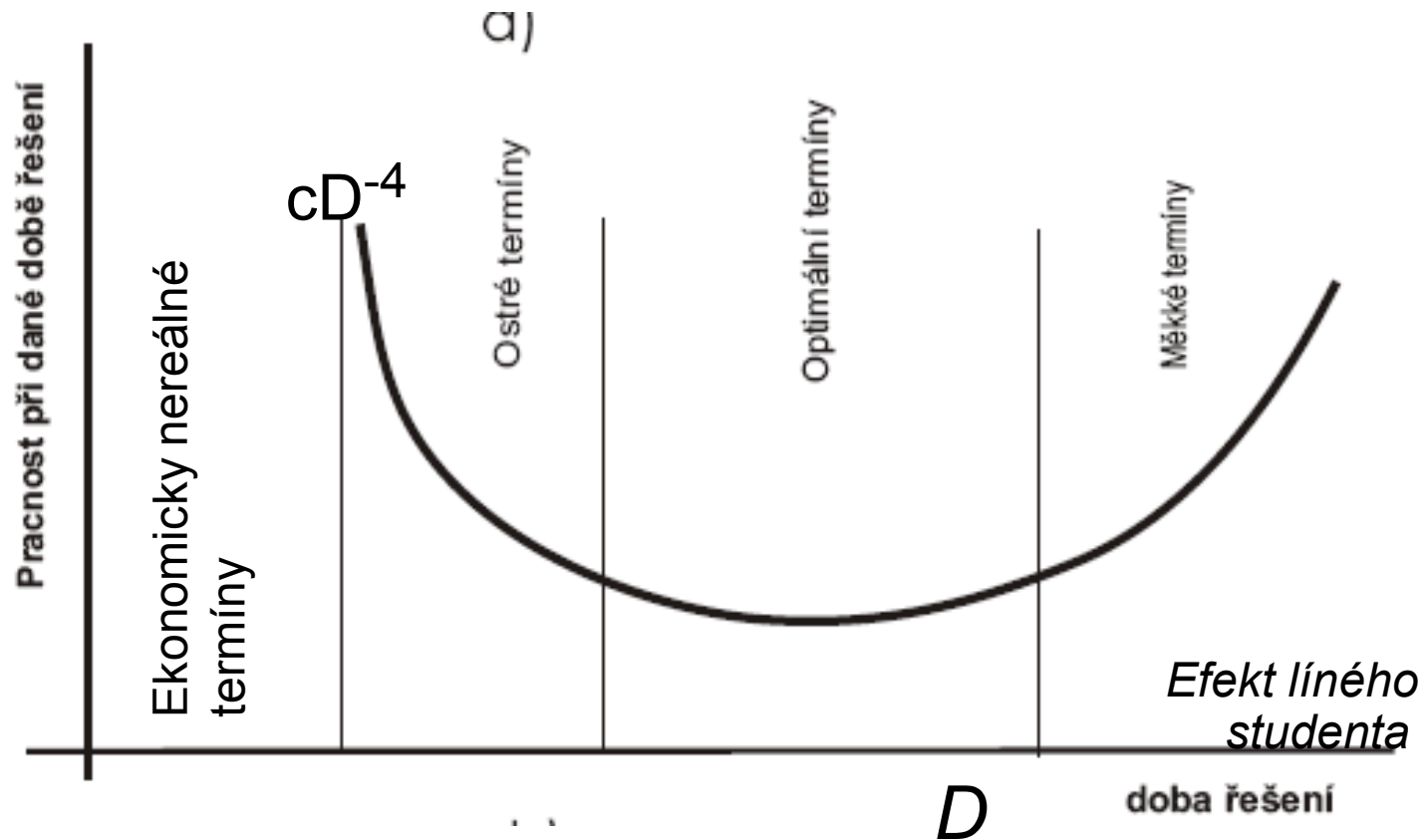
Tempo zastarávání znalostí v IT,

- V technologiích vývoje SW se inovuje 50% znalostí za 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

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



Pracnost projektu při zkracování termínů, nedosažitelné oblasti



Důsledek

- Dobu řešení nelze libovolně zkracovat i při dostatečných zdrojích
 - Zdroje a doba řešení, jsou vždy omezené
- Ťok má asi pravdu. Pokud je nutné přepsat velký systém, bude to prakticky jistě trvat dlouho

**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

Celkový trend

Rostoucí složitost a tempo změn SW systémů
Stále širší integrace do společenských a sociálních
procesů

Integrace autonomních částí

Zapojování uživatelů do vývoje a
údržby a zmenšují se rozdíly mezi vývojem a
údržbou

Dekompozice se stává nutností, především u
SME

Řešení

Hrubozrnná dekompozice

Nejlépe jako SOA, kde služby komunikují pomocí digitálních byznys dokumentů putující přes inteligentní middleware

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

Zaměření přednášky

klíčová témata

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

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

Tyto poznatky 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

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

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.

SME vývojář má větší šanci rozumět zákazníkovi který je SME

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), Viz používání papírových dokumentů

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 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 náraz

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 servisně orientované architektury (hrubozrnné SOA), dokumentová orientace
 - integrace systémů, přílepky cloud...
 - Inkrementální vývoj a údržba
 - Integrace s existujícími systémy (doplňky),
 - Změny skryté v komponentách
 - Údržba splývá s vývojem

Nadměrná složitost řešení škodí i velkým

- Složitá řešení se ne vždy vyplatí,
- I když velcí výrobci SW na to někdy mají, koncoví uživatelé zpravidla 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 SW výrobci nezvládají pokud nepoužijí triky malých SW firem. Jinak jsou jejich systémy
 - Příliš drahé
 - Permanentně zastaralé
 - Obtížně udržovatelné
 - Nespolehlivé

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 jsou významné

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

Informační systémy

- Jsou klíčovým prvkem ITC a motorem vzniku většiny nových SW paradigmat. Stále více podporují i propojují procesy reálného světa.
- Roste potřeba snadného a flexibilního propojování existujících i nově vytvářených informačních systémů a SW komponent.
- Výsledné systémy jsou **obrovské** a stále se mění. Proto **musí** být permanentně **zastaralé** jsou-li budovány jako **katedrály** .

IS často selhávají

- Příklad 1: **Bakalář**, Aktualizovaná instalace malém podniku - gymplu s následujícími důsledky pro uživatele
 - Nutnost naučit se jiné rozhraní
 - Po zvládnutí rozhraní se zjistí, že je systém líný
 - Když začnou dělat všichni, spadne.
- Příklad 2: **Registry**: Výpadky, rozsah služby by neměl představovat skoro žádnou zátěž, ale každý ťuk se zapouzdří a cestuje na konec světa a zpátky.
- Hypotéza: **Chybná architektura**
- Řešení: Tlustý klient jako služba s **dokumentově orientovaným rozhraním** na zbytek systému

Ekonomické a sociální aspekty

- **Nedodržování** termínů a nákladů
 - Úspěšnost kolem 1/6, krach kolem 1/3 projektů
- **Závislost** na dodavateli (antipattern vendor lock in)
- **Předražená** údržba a obtížný provoz
 - Mýtný systém
 - Projekt Opencard (daly se snížit náklady na přepsání téměř stokrát)
- Permanentní **zastaralost**
- **Lidé** jsou de facto součástí systému, význam této skutečnosti roste (UI), musí se více zohledňovat než dříve
- Potřeba agilní spolupráce a **integrace** informačními systémy obchodních partnerů, často formou sourcingu
- Potřeba po čase systém celý **přepsat**, ten ale může být tak rozsáhlý, že je to prakticky nemožné pokud nezmodernizujeme architekturu systému.

Permanentní zastaralost velkých monolitických SW systémů (katedrál)

- Termíny implementace monolitických SW systémů (katedrál) nelze libovolně zkracovat.
- Důsledkem jsou dlouhé termíny realizace a permanentní zastaralost – mezitím dojde ke změně požadavků a mění se i technologické prostředí i požadavky
 - Do 5 let je 1/2 potřebných IT znalostí mladší než pět let
 - Za méně než deset let změna paradigmatu
 - Systém nutně i jinak nemá časem dobrou kvalitu
- Řešení: **Nestačí jen katedrály ani jen bazar**
 - Budovat město – je nutná **infrastruktura**

Klíčové paradigma

- SOA s rozhraním používajícím byznys dokumenty, systémy správy dokumentů a případně s využíváním cloudů a inteligentní dokumentové rozhraní.
- Je obtížné přejít na nové paradigma. Např. vzory řešení pro objektový přístup mohou být špatná řešení pro SOA a naopak.
- V praxi SW firem je pocit, že SOA je prázdné heslo (viz. antipattern whats new) a že se bezdůvodně vzdáváme toho, co dobře umíme, zvláště když máme cloudy). Je nutné stále více spolupracovat s uživateli.
 - Klesá význam/rozsah kódování, samotné kódování se mění.
 - Do vývoje i do běhu systému agilně vstupují uživatelé.

Architekturní (infrastrukturní) služby

- Inteligenci middlewaru lze zvýšit pomocí infrastrukturních služeb, jako jsou směrovače, transformátory zpráv. Osvědčuje se, aby činnost infrastrukturních služeb byla ovladatelná on-line.
- Uživatelské rozhraní systému se pro ostatní služby jeví jako standardní služba. Tuto službu lze s malými modifikacemi použít ke spolupráci s jinými systémy, prototypování, řešení výpadků ...
- Výsledný systém má rysy města (a není to povrchní podobnost, viz infrastrukturní služby a prostředky není to ani katedrála ani bazar.

Virtualizace architektury

- Přenos dokumentů může být v cloudech virtuální, nastavením příznaků v metadatech dokumentů uchovávaných v cloudu.
- Virtualizace nabízí další možnosti především díky možnostem řízení daty.
- Je důležité, aby **se zachovala možnost vidět a používat i pak datové operace jako komunikaci dokumentů mezi službami.**
- Virtualizace má meze při celosvětové spolupráci byznys partnerů, zatím není celosvětový cloud a jsou i jiné zádrhely.

Navrhované principy umožňují splnit požadavky na to, aby se zachovala logická struktura existujících uživatelských procesů a umožnila vysokou agilitu uživatelů

Gartner Group 2015

Massimo Pezzini,...

Shake Up Your Integration Strategy to Enable Digital Transformation

Massimo Pezzini,...

Unleash DiY Citizen Integration to Enable Digital Business Transformation

[John Brandon](#) Why paper still rules the enterprise

<http://www.cio.com/article/3025928/printers/why-paper-still-rules-the-enterprise.html>

Některé publikace (některé nejsou zadarmo)

[Elizabeth Golluscio](#) | [Massimo Pezzini](#)

Unleash DIY Citizen Integration to Enable Digital Business Transformation

<https://www.gartner.com/doc/3172117/cio-action-shake-integration-strategy>

<https://www.gartner.com/doc/3184624?ref=AnalystProfile&srcId=1-4554397745>

[John Brandon](#) CIO Magazine: Why paper still rules the enterprise

<http://www.cio.com/article/3025928/printers/why-paper-still-rules-the-enterprise.html>

The Application Integration Middleware Market Embraces Digital Transformation

Pezziniho analýzy (a také druhý mod bimodálního řešení diskutovaný níže) lze nejspíše implementovat pomocí hrubozrnných komponent s dokumentovým rozhraním ve formě XML textů Brandonova zjištění navíc implikují, že je dobré počítat s tím, aby se digitalizovaly byznys dokumenty, jako objednávky, tak, aby se snadno daly převést do papírové formy. Podle zkušeností z praxe to má mnoho zásadních výhod pro kvalitu řešení i procesů vývoje a údržby.

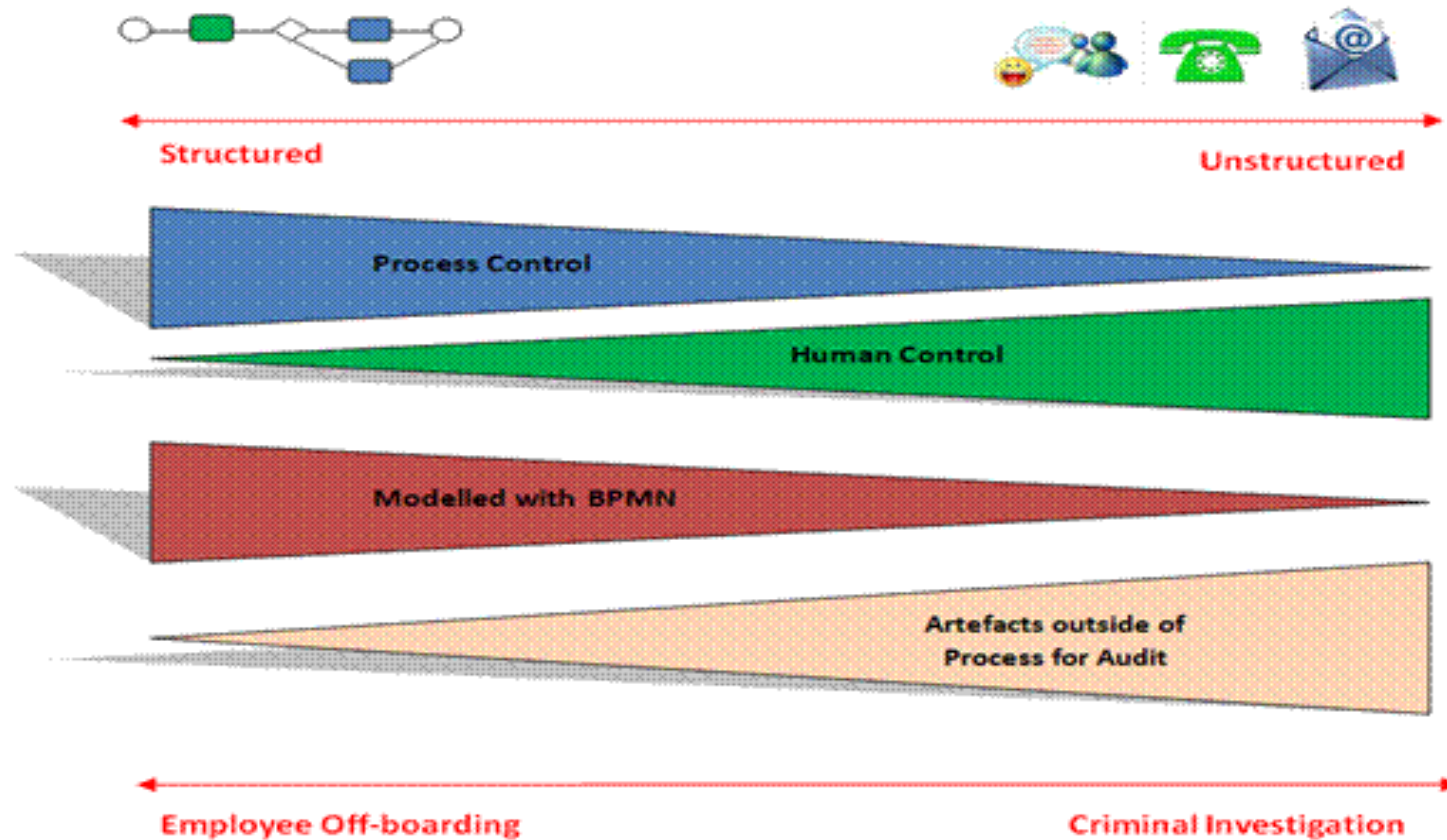
Důležité závěry

- Pezziniho analýzy (a také druhý mod bimodálního řešení diskutovaný níže) lze nejnáze implementovat pomocí hrubozrnných komponent s dokumentovým rozhraním ve formě XML textů
- Brandonova zjištění navíc implikují, že je dobré počítat s tím, aby se digitalizovaly byznys dokumenty, jako objednávky, tak, aby se snadno daly převést do papírové formy.
- Podle zkušeností z praxe to má mnoho zásadních výhod

Bimodal IT (Oracle)

- Mode 1 is traditional and sequential, emphasizing safety and accuracy.
- Mode 2 is exploratory and nonlinear, emphasizing agility and speed.
 - Druhý mód vyžaduje digitalizaci uživatelských procesů, **jádrem druhého modu je u větších systémů digitalizace dokumentů**
 - Zdá se, že druhý mód má tendenci převládat

<http://www.ateam-oracle.com/case-management-part-1-an-introduction/> Oracle case management



Dříve převažovala tendence posunu vlevo, dnes se to zčásti obrací.

Gartner glossary

<http://www.gartner.com/it-glossary/bimodal/>

Bimodal is the practice of managing two separate but coherent styles of work: one focused on predictability; the other on exploration. *Mode 1* is optimized for areas that are more predictable and well-understood. It focuses on exploiting what is known, while renovating the legacy environment into a state that is fit for a digital world. *Mode 2* is exploratory, experimenting to solve new problems and optimized for areas of uncertainty. These initiatives often begin with a hypothesis that is tested and adapted during a process involving short iterations, potentially adopting a minimum viable product (MVP) approach. Both modes are essential to create substantial value and drive significant organizational change, and neither is static. Marrying a more predictable evolution of products and technologies (Mode 1) with the new and innovative (Mode 2) is the essence of an enterprise bimodal capability. Both play an essential role in the digital transformation.

**Think
marathon
runner**

	Mode 1	Mode 2
Goal	Reliability	Agility
Value	Price for performance	Revenue, brand, customer experience
Approach	Waterfall, V-model, "high-ceremony IID"	Agile, Kanban, "low-ceremony IID"
Governance	Plan-driven, approval-based	Empirical, continuous, process-based
Sourcing	Enterprise suppliers, long-term deals	Small, new vendors; short-term deals
Talent	Good for conventional processes and projects	Good for new and uncertain projects
Culture	IT-centric, removed from customer	Business-centric, close to customer
Cycle times	Long (months)	Short (days, weeks)

**Think
sprinter**

Kritika bimodálnosti Gartnera

- V praxi malých firem jsou běžné systémy, kde se občas nebo standardně vyžaduje, aby některé akce prováděli lidé a zasahovali do procesů. Je to důležité i pro inkrementální vývoj. To se dosti liší od modu 2 podle Gartnera.
- Celkově se pro SME otevírají nové možnosti, vlastně praxe SME značně ovlivňuje praxi velkých SW firem a SME má větší možnosti úprav a vývoje velkých systémů (e-government)

Další trendy

- **CIO Call to Action: Shake Up Your Integration Strategy to Enable Digital Transformation**
- <https://www.linkedin.com/pulse/cio-call-action-shake-up-your-integration-strategy-enable-pezzini>

Důležité jsou odchylky v detailech

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

Celkový trend

Množí se případy
ve kterých se dobře uplatní
malé a střední SW firmy

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 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
- Neodborníkovi těžko, pokud vůbec, proč je třeba tolik jazyků

Podceňované vlastnosti IT

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

Jsou často dlouhodobé a ty jsou 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

Která oblast techniky má s IT nejvíce společných aspektů

Architektura

Kde kdo do toho kecá

Projekty jsou předražené

Vysoký podíl řemesel

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, krize IT 1980	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, krize IT 1990 (meze PC)	Sálové počítače s PC místo terminálů, kancelářské úlohy pro PC, datové báze
1990-2009	Interaktivní výpočty na síti, e-komerce, Internet, sociální SW 2002 krize, Internetová bublina 2008 krize, IT nepomohlo	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

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 Autonomní komponenty a SOA
- 2000 SOA, patterns, účast na projektech malých firem, konfederativní SOA, práce v týmu , dokumenty řízené systémy, role osob v SW systémech

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-paradigmaty
- O české informatiky je stále zájem, především o kódéry a návrháře
- Nově globalizace vývoje SW, hlavně v kódování
- Mnoho lidí funguje jako servis běhu SW systémů
- Vývoj od začátku není častý, hlavně v menších firmách
- Účast na vývoji velkých firem v jejich pobočkách
- Mnoho lidí programuje podporu e-governmentu

Podceňované vlastnosti IT

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

Jsou často dlouhodobé a ty jsou jiné než krátkodobé

Někdy není ochota je uplatnit (př. školství)

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

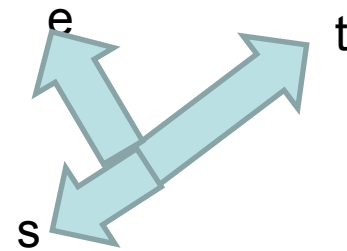
Kvalita IT je tedy věcí kvalifikovaného používání, znalostí, zkušeností a někdy i (politické) vůle a boje s předsudky

Budeme využívat zkušenosti z (starších i současných) 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

Obtížnost aspektů IT, hlavně IS

- Klasické IS, např. správa skladu (operativa)

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

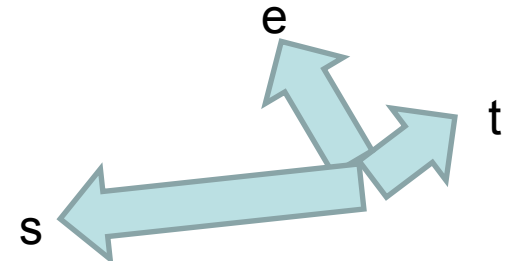


- Informační technologie dnes

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

- Je zdrojem obtížnosti a problémů
 - Nejasnost cílů
 - Postranní zájmy (viz evaluaci škol), korupce
 - Potřeba globálních propojených měnících 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

$$Soper \cong C Del^b$$

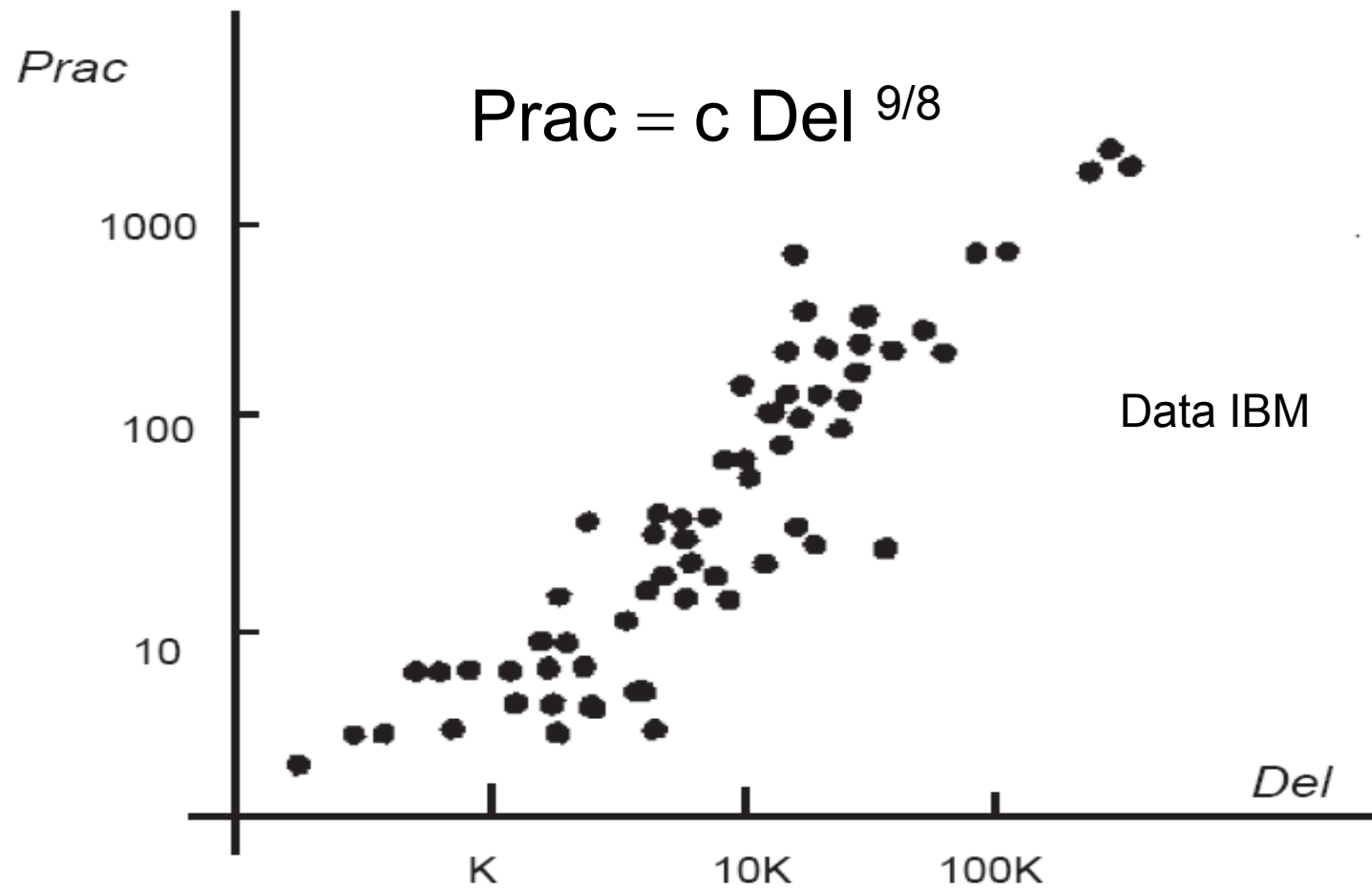
dostaneme

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

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



Pozoruje se

$$Duration \cong c Effort^a \quad a \cong 0,4$$

c lze ovlivnit technologií návrhu a dekompozicí
 a i *Effort* architekturou

Architektura umožňuje zlepšit celkovou kvalitu řešení
a kvalitu vývojových procesů a má řadu dalších
výhod

Hlavní problém velkých
systémů:

Udržitelnost

Jak na věc

- Podmínkou úspěšných IT 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

Mnohé 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 nepřiliš formalizovaná pravidla spolupráce lidí
- Důsledky omezených kapacit pro aplikaci procesů vývoje

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

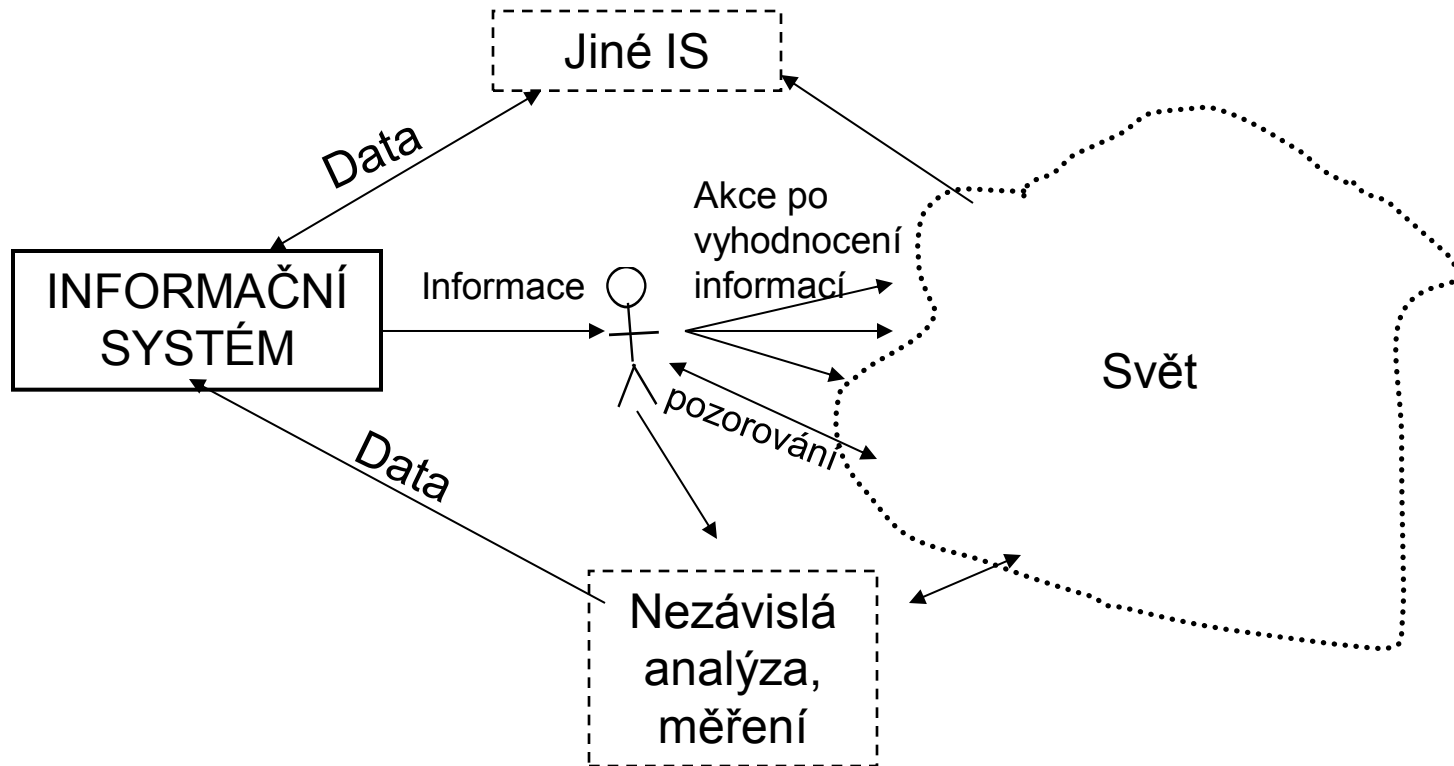
-Pozor na BPR – měním i to co dobře funguje

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

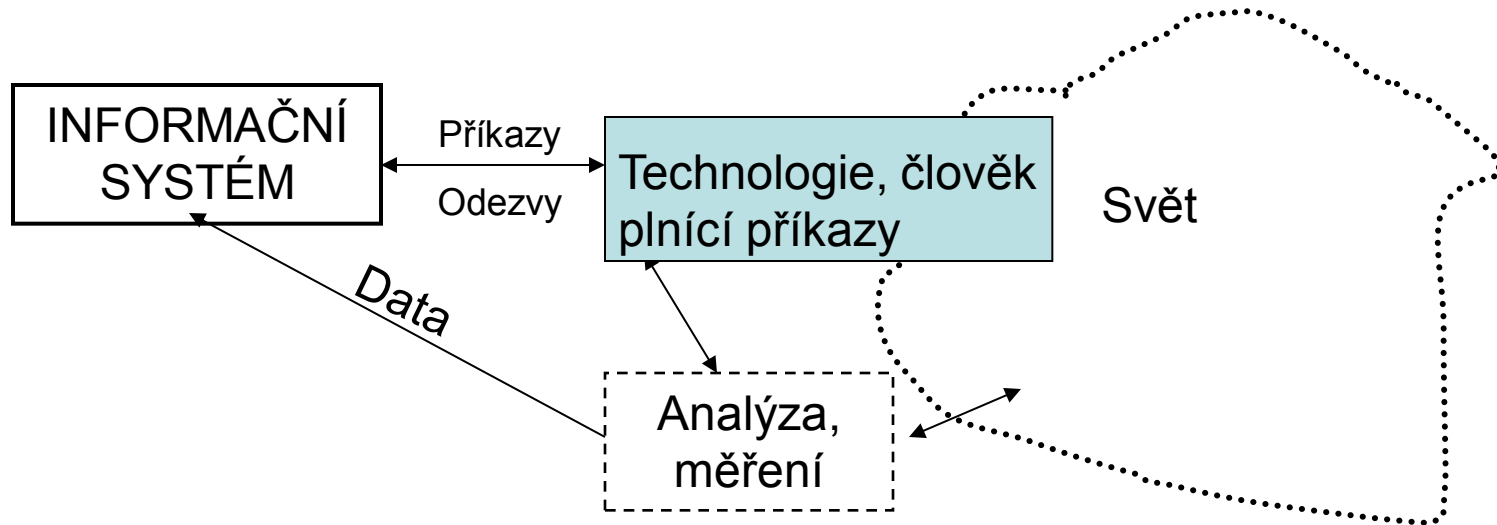
Informování versus řízení

- 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%

V roce 2012 se zvýšilo procento úspěšných projektů, Je problém, v posunu definice projektu v souvislosti se změnami paradigmat IT.

Úspěšných projektů	39%
S nedostatky	43%
Krachující	18%

Stav v roce 2012, Standish
Group Chaos Report

Co způsobuje největší problémy

Antivzory jako podceňování realizovatelnosti
(vše hned, vše so mne napadne
realizovat, cena, ...)

Zdánlivé hlouposti (blbosti), dokonce i
skutečné triviality

Skryté předpoklady, absence správných
otázek

Problémy s IS

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

Nejfatálnější důsledky mívá opomenutí 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 blokových 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 opomenutí z historie

- Turing a prolomení šifrování Enigma
 - Na straně Hitlera 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
- Víra v neprůchodnost Arden v roce 1940
- Hloupé chyby dělají podniky i v době míru

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 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
- Také je důvodem nákladnosti údržby (dvakrát až třikrát více než náklady na vývoj). Platí to především pro systémy zahrnující aktivity lidí, tj. většinu IS. Málo se snažíme zachovat a využít existující znalosti a osvědčenou praxi

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, dokumentově orientovaná komunikace)
 - Nové technologie znamenají obvykle změnu zvyků a dovedností, u dokumentové orientace lze naopak podpořit znalosti

Registry - výzvy

- Nepracují zpravidla dobře po mnoha měsících ač se konceptuálně a rozsahem požadavků na výkon jedná o poměrně jednoduché úlohy
 - Chyběla rozumná specifikace: příliš interaktivnosti, je třeba použít operace 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, každý ťuk putuje přes celý svět)
 - Nesprávná orchestrace, omezené kapacity sítě - úzká místa

Registry, časté scénáře

- Nesprávné specifikace a termíny a nedokonalá kontrola uskutečnitelnosti obecně (poručíme větru, dešti) i kvality dodavatele.
- Nebyl „stavební dozor“
- Dodavatel nedodal včas a v dostatečné kvalitě, ztráty občanů i státu.
- Dodavatel si díky tomu namastil kapsu,
- 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 IS zprvu dosti 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
- 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 lítaček 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), systémy zprávy digitálních dokumentů
 - 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
- Nepohrdat znalostmi, dovednostmi uživatelů

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, není to ale bez rizik
 - Přehnaná interaktivnost
 - Každý ťuk se ověřuje v centru

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

Skrytý postoj podruhé
Vím jak postavit most, protože vím,
jak se má přes něj jezdit
Na druhé straně je důležité, aby se
systém uživatelům jevil jednoduchý
a bylo jasné co vlastně z hlediska
uživatele dělá
Jak se vlastně jezdí po mostě

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
- Obecně roste potřeba multidisciplinárních znalostí a dovedností (stálá spolupráce s lidmi různých oborů, porozumění jejich oborům, sociální dovednosti)

Jak je to s kodéry

- Je to základní řemeslo, dobří kodéři jsou pro SME výhodou
 - Existují analýzy, podle kterých se v každé stovce kodérů vyskytuje několik kodérů s produktivitou 25 krát větší než je medián produktivity
- Pro malou firmu bývá superprogramátor terno,

Manažeri

- Výkonný ředitel, měl by 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 manažerů
- V SME mohou a často musí být tyto role zastávány i jediným člověkem a nemusí být jejich obsah podrobně vymezen

Kteří informatici nejméně chybí (Forrester Research 2006)

Před deseti léty 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 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, dobře lokalizují existující systémy (SAP atp.), lze to jen obtížně outsourcovat do Indie

- Často se kódují jen doplňky, např. sběr dat, UI, řízení procesů

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 SW normy

Doplňky jsou dobrou příležitostí pro menší SW firmy

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!!
 - To neplatí pro případy systémů s dokumentovým rozhraním když používané dokumenty jsou transparentní digitalizací stávající praxe (byznysu)

Problémy softwaru jsou často důsledkem problémů s lidmi

- Uživatelé nemohou vyvíjet IS sami, nelze ale IS vyvíjet bez nich, nutno rozumět jejich potřebám
- Nemohou zpravidla sami přesně formulovat požadavky, neboť v tom nemají praxi, neodhadnou obtížnost, závislost doby odhadu na rozsahu požadavků, na důležité si nevzpomenou
 - Tento aspekt bývá důvodem přechodu na systémy, které jsou daný účel zbytečně velké a drahé a bývají nevhodné pro podnikovou kulturu v uživatelů, zvláště malých, je to jeden z důvodů bimodálního řešení

Software není problém, problémy jsou s lidmi díky SW

- Netransparentní potřeby
- 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 budovat 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

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

Kódování

- Nejméně kritická etapa, rychlý vývoj nástrojů
- Dobrý kodér je výhra, nahradí i desítky průměrných
 - manažeři ale nechtějí být závislí na špičkových kodérech
- Scrum – místo řemeslníka pomocný dělník
- Některé SW SME se specializují na detailní návrh, kódování a ožívování, často se stávají

Tempo zastarávání znalostí v IT,

- V technologiích vývoje SW se inovuje 50% znalostí za 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 na implementaci změny je tak velká, že vše mezitím zastarává. Často se to obchází tím, že se předá nespolehlivý systém

Platí to i pro velké výrobce SW, je to kritické pro malé SW firmy

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

Co s tím, zvláště v SME

- Dekompozice do téměř nezávislých entit s jasnou sémantikou (sklad, účetnictví) spolupracujících pomocí výměny digitalizovaných dokumentů
- Sémantika je zřejmá všem zúčastněným
- Komunikace SW entit pomocí chytré sítě

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

Celkový trend

Stále širší integrace do společenských a sociálních procesů

Integrace autonomních částí pomocí dokumentových rozhraní, SOA

Spolupráce vývojářů s uživateli při vývoji údržbě.

Údržba a vývoj splývají

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)

Výsledky IT nejsou jednoznačné

- IT propojuje svět, motor globalizace, přístup k informacím, navazování vztahů na dálku
- Lidé mají stále méně času pro sebe své blízké
- Byrokracie roste
- IT nezabránilo opakování ekonomické krize v roce 2008
- Nejasné efekty pro vzdělanost, příliš snadno se opouští pravidla, dovednosti a o ověřené postupy, zapomíná se, že učitel má klíčovou roli
- Nečekané nepříznivé sociální efekty (terorismus)

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, není to ale bez rizik
 - Přehnaná interaktivnost
 - Každý ťuk se ověřuje v centru

Nový trend: dokumentová
rozhraní

DMS

Document management
systems

Příklad z praxe

Orchestrace dokumentově
orientovaných architektur v malé
SW firmě

Digitalizace a optimalizace sběru požadavků na zdravotnický materiál v nemocnici

- Udělat jako doplněk stávajícího ERP,
 - Důsledek: Udělat jako autonomní systém posílající zhromadněné požadavky jako objednávku na ERP,
- Digitalizovat stávající procesy, zpřesnit je.
- Zohlednit, že se schvalování zúčastní různé profese, nemají zájem měnit postupy když není jasný efekt

Struktura procesu

- Akreditovaní pracovníci zadají *žádanky* pro jednotlivé základních organizační jednotky
- Žádanky postupně na jednotlivých organizačních úrovních modifikují další pověřené osoby (vedoucí, manažeři, účtaři, případně externisté), potřebují mít přehled požadavků za celou svou org. jednotku (např. celou nemocnici). Používá se zhromadňování pomocí Excelu žádanky se upravují on-line
- Požadavky na jednotlivé materiály se zhromadí a odešlou do ERP

Pozorování

- Požadavky jsou spíše odhady (proces druhého modu v bimodálním pohledu), proto je důležité zhromadňování
- Je to celé potřeba koncipovat jako dvě autonomní entity podobné službám v SOA komunikující pomocí dokumentů, ta mohou být digitální i papírové, zprvu papírových
- Je nutné počítat s tím, že celý proces bude vždy zahrnovat lidi, i jako výkonné prvky
- Entita správy požadavků je navržena jako soubor specifických služeb komunikujících pomocí množin dokumentů. Je otevřená otázka do jaké míry bude třeba využít cloud a kdy fyzický transport dokumentů

Klíčové rozhodnutí

Dokumenty digitalizovat jako zapouzdřené spreadsheetové tabulky spravované systémem správy dokumentů (DMS)

- Lidé jsou zvyklí na oběh dokumentů na papírech
- Používají spreadsheet, potřebovali by správu (databázi) dokumentů (tabulek)
- V DMS jsou s tabulkami asociována org. data pomocí nichž lze vytvářet pohledy na celkové požadavky a využívat spolupráce zdravotníků s manažery i účtaři
- Použít DMS otevřený systém Alfresco a Excel

Postup realizace:

- Zavedení pořádku do ručního zpracování (registry, ...),
- Zavedení digitálních dokumentů v Excelu (jsou na Excel zvyklí)
 - Definování struktury dokumentu, operace tvorby žádanky, ukládání žádanky do DMS, papírový výstup
- Operace nad skupinami žádanek pomocí DMS Alfresco
 - Použito opakované REA, lze i papírový výstup
 - Zapouzdření dokumentů, metadata
- Zhromadnělé požadavky jsou odeslány přes standardní rozhraní do ERP, alternativně papírový výstup
- Současný stav: Rutinní provoz, naprosto hladká implementace, možnost použití na léky a v jiných nemocnicích, pro všechny pomoc, manažeři mají sklon do všeho strkat nos (vlastně šikana)

Otevřené otázky

- Kdy použít jen DMS, kdy lépe využít cloud
- Kdy dokumenty fyzicky transportovat a kdy je lépe digitalizované dokumenty vybavit metadaty a nechat je logicky „na jednom místě
- Jaké jsou meze zvoleného řešení
- Proč se rychle neupravilo i na léky
- Zřejmě se dá použít Adaptive Case Management pro etapu zhromadňování
- A leccos dalšího

Čím se architekti 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,
- Musíme být schopni spolupráce s uživateli a také s jinými technickými profesemi, se kterými musíme spolupracovat, často společně s nimi budeme opravdovými partnery manažerů

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 – kódeři 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

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, krize IT 1980	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, krize IT 1990 (meze PC)	Sálové počítače s PC místo terminálů, kancelářské úlohy pro PC, datové báze
1990-2009	Interaktivní výpočty na síti, e-komerce, Internet, sociální SW 2002 krize, Internetová bublina 2008 krize, IT nepomohlo	Servery, počítačové sítě, Internet, grafika, vývojová prostředí, databáze, globalizace, webové služby, podpora sociálních sítí

Nejčastější profily činnosti menších SW firem

Instalatéři a provozáři, Vývoj od
začátku, Kódovací mašina,
Příštipkáři a Integrátoři, Leštiči bot
(rozhraní, webové stránky).

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 , web, Java
- ***Servisní orientace!!!***, po roce 2000 SOA
- Sociální sítě, web 2.0, globální IS
- Aspektové programování, agilní vývoj
- Cloud computing, gridy, (využití volných kapacit, balancování zátěže,)
- ***Dokumentová orientace***

Hlavní poznatky

- Krize ekonomiky a IT se vrací asi po deseti létech, může přijít znovu, asi znovu přijde
- Po krizi se vždy se vrátil zájem o informatiku, ale s podstatně jinými úkoly
- O české informatiky je stále zájem, především o špičkové kodéry a návrháře, kodovací manufaktury
- Nově globalizace vývoje SW, hlavně v kódování
- Není jasné, jak se projeví krize 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: témata výzkumu, změna zákon, Příklady 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**

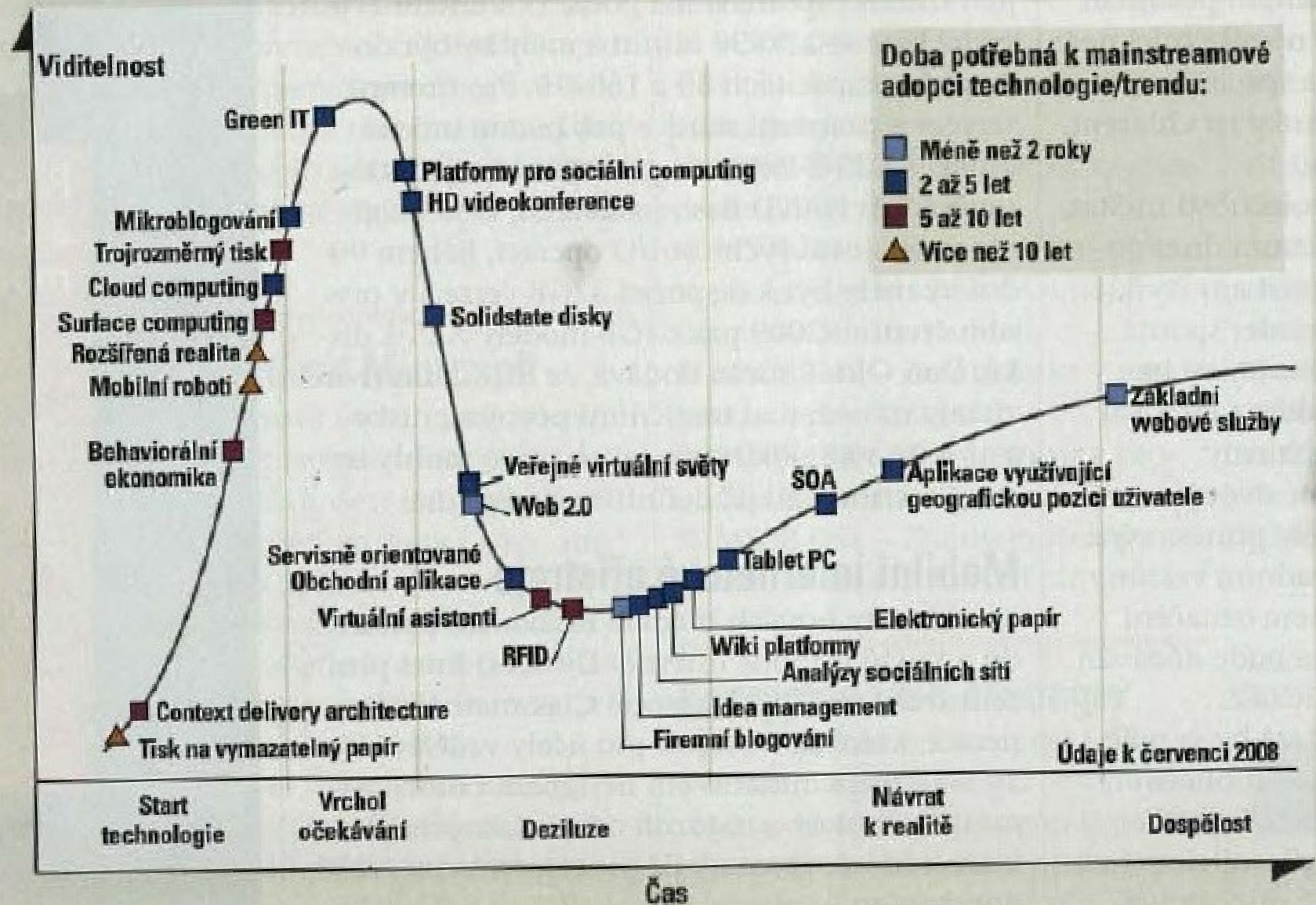
Příklady na hype křivce

- 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

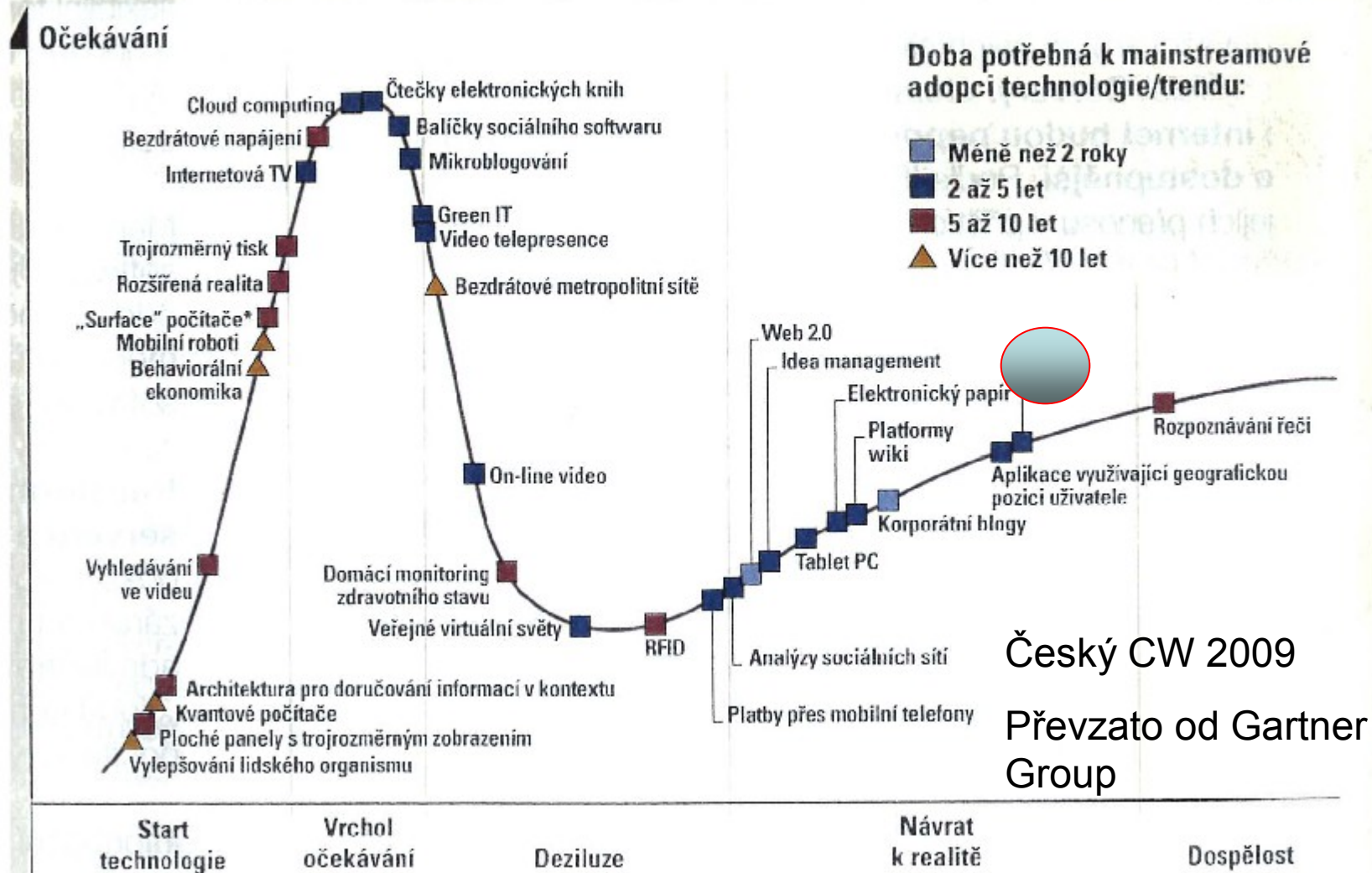
1. Velká stále rostoucí popularita, **líbánky**
2. Stále bolestnější zjišťování mezí možností dané technologie a proto **desiluze** čili vystřízlivění a cosi jako kocovina.
3. **Krize**, objevení či neobjevení hlavního užitku a optimálního využití
4. Rozvod, nebo soužití - plató těch, co to dokázali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

Základní zásady

- Práce ve dvou
 - Kontrola, oponentura
 - Dělbba práce podle talentu
 - Především omezení ztrát souvisejících s možným odchodem
 - Předávání dovedností
- Velmi často se zanedbává integrace uživatelů do týmu

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 visting 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

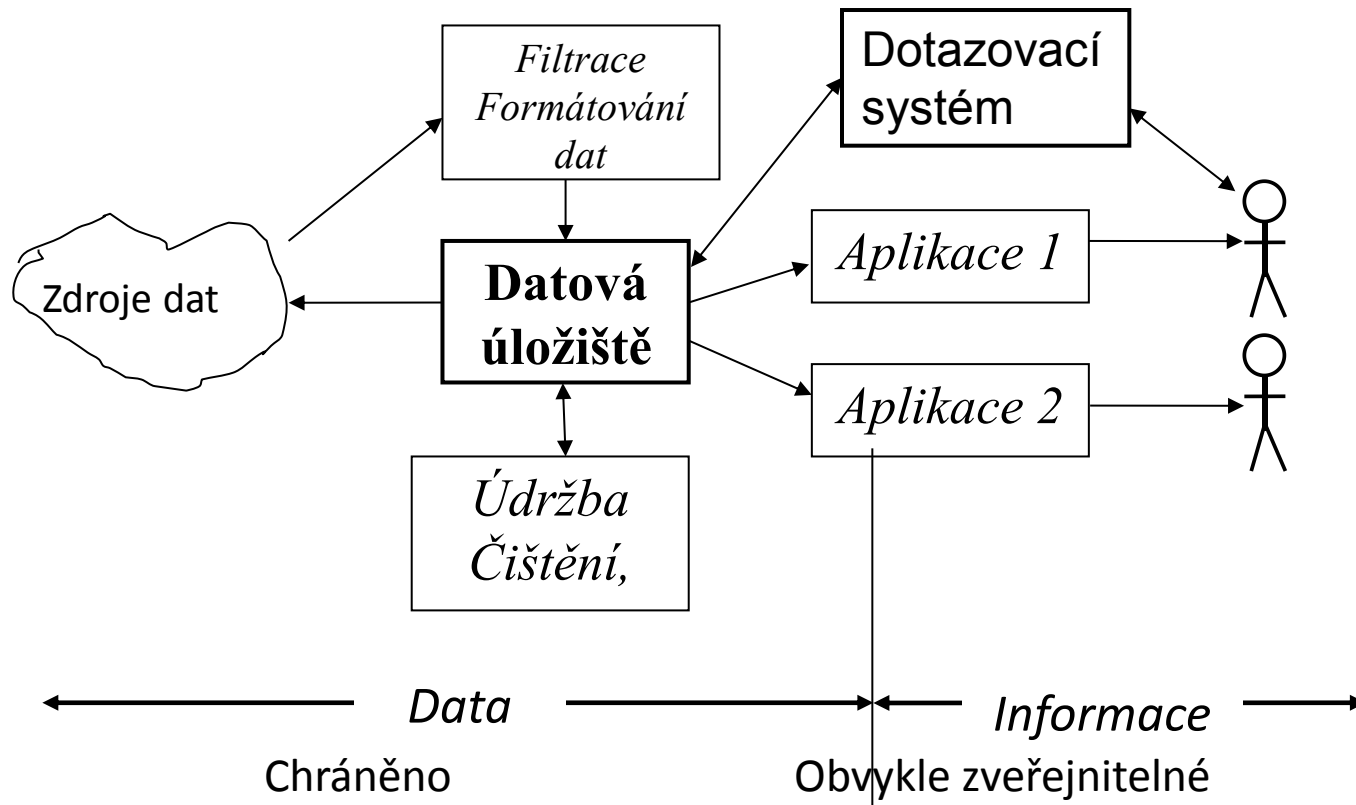
- Dosti rozšířený a asi správný pocit, že se naše školství vyvíjí špatným směrem
 - Opomíjení STEM, To pocítujeme i u nás, je problém v podnicích (průzkum Manpower)
 - Opomíjení tréninku dovedností. Včetně trivia a cizích jazyků, pologramotnost
 - Nerovnoprávné postavení učitelů
 - Za stížnosti žáků 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 efektivity datové skartace

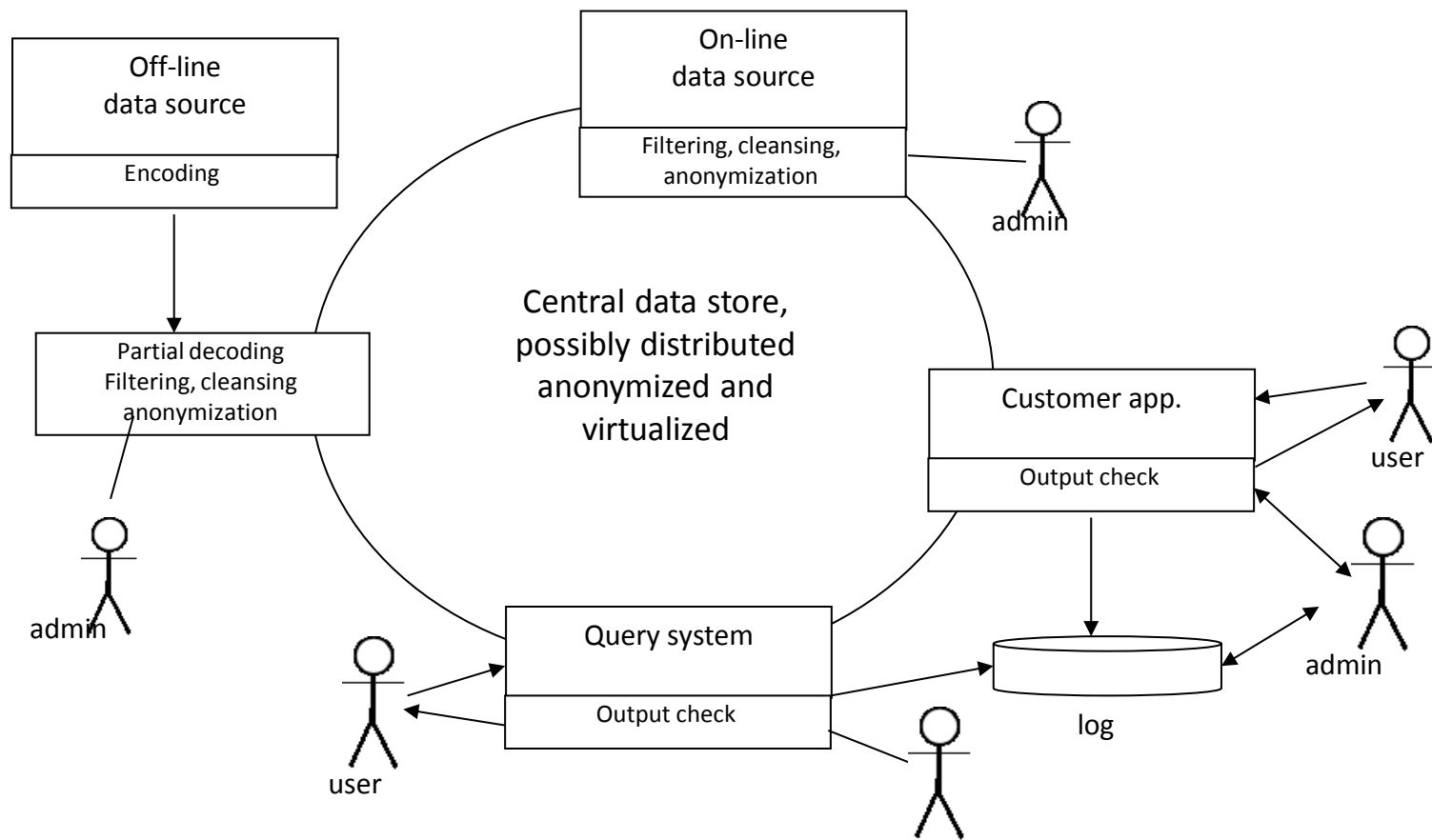
Virtuální skartace je v principu neúčinná, neboť nemůže zásadně omezit únik dat (vlastně těch nejdůležitějších)

- 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



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, údržbáře a obchodníky
- 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 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

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

Co je informatika

Tvůrčí obor?

Ano, ale ve smyslu hledání vynálezů a vývoje systémů. To jsme si zpočátku nemysleli.

Specifický Inženýrský obor!

Obtíž formulovat správné otázky, na správné otázky jsou často snadné odpovědi, ale správné otázky chybí

Vytváří artefakty, jsou sice nehmotné, jsou ale obvykle zbožím a mají překvapivě mnoho společného s jinými technickými artefakty

- závisíme na obchodnících
- vlastnosti lidí jsou v informatice ještě důležitější, než u jiných inženýrských oborů, informatika je více ovlivňuje jako uživatele,
 - ohrožuje jejich pracovní s mocenské pozice, mění náplň jejich profesí, platí to i pro vývojáře SW systémů
- kvalita služeb IT, především těch klíčových, závisí i na právním prostředí a společenské situaci
 - Co je dovoleno, co se vyžaduje
- některé činnosti a úkoly vyžadují dlouhodobé zkušenosti
- Silné prvky mezioborovosti

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?

Ochrana osobních dat

- **Privacy-preserving data publishing: A survey of recent developments**
- [Benjamin C. M. Fung](#) Concordia University, Montreal, Montreal, QC, Canada [Ke Wang](#) Simon Fraser University, Burnaby [Rui Chen](#) Concordia University, Montreal, Montreal, QC, Canada [Philip S. Yu](#) University of Illinois at Chicago
- ACM Computing Surveys (CSUR),
Volume 42 , Issue 4 (June 2010), pp 1-53

Od dostupnosti k nedosažitelnosti

Konec copyrightu?

- Existují pokusy zavést obdobu licenci podobných licencím otevřeného SW. Diskusi různých aspektů řešení problému
 - Lawrence Lessig: *Free Culture, How Big Media Uses Technology and the Law to Lock Down Culture and Control Creativity*,
 - český překlad lze nalézt na <http://wiki.root.cz/Main/FreeCulture>
- Současná pravidla dávají konkurenční výhodu takovým zemím, jako je Čína

Problémy jsou i důsledkem používání IT, IT se musí uplatnit při jejich řešení.

Jádro problému je mimo informatiku

Jiné hrozby

- Finanční operace v milisekundách a finanční bubliny, nedostupnost základních ekonomických info
- Sobecké zájmy velkých hráčů
 - Obdoba starověkých nomádů ničících staré civilizace?
- Obrovská koncentrace na trhu SW
- Globální aktivity bez globálního dohledu
 - I zde hraje roli IT
- Přesun kódování do mzdových rájů
- Specifikace požadavků bude asi muset zahrnout i specifikaci požadavků na změnu prostředí

Neřeší se problém kvality dat

- Aktualizace
 - povinnost aktualizovat
- Věrohodnost
- Konsistence
- Odpovědnost
- Postih za zničení dat

Hlavní výzvy pro ajťáky

- Zdá se, že klíčový problém IT (bottleneck) je spojen s pravidly (ne)dostupnosti dat a nekompetentností uživatelů
 - Jak na věc_
 - ?
- Úspěch SW závisí na tom, zda se podaří ovlivnit legislativní prostředí,
- Budování informačních systémů a SW systémů obecně je stále více mezioborový a společenský problém a IT se rychle mění
 - Jak to zohlednit ve vzdělávání, jak upravit studium (i to celoživotní), jak se sám vzdělávat
 - IS pro kontrolu kvality vzdělávacích institucí (samo o sobě nestačí), asi bychom se měli více zajímat o zkušenosti dřívějších absolventů
- Jak se prosadit v malých firmách proti velkým hráčům
- Potřeba kontaktů na praktické problémy a spolupráce s firmami

Hlavní výzvy pro ajťáky

- Využití dat je možné jen, jsou-li dostupné a jinak kvalitní, to není zdaleka splněno
 - Změna vyžaduje zásah do společenského prostředí
 - Předsudky veřejnosti
 - Legislativa
- Kombinace oborů
 - Kombinace měkkých a tvrdých znalostí

Hlavní výzvy pro ajťáky

Teorie není jen pro akademiky

- Využívání dat je možné jen s použitím matematické statistiky
- Mnohé inženýrské vlastnosti jsou důsledkem abstraktní matematiky a matematické statistiky
- Ajťáci matematice moc nedají
- Roste význam statistiky a ta je ajťákům protivná
 - Je nutná v business intelligence
 - Specifikace často spojena se statistickou analýzou
 - Způsob myšlení
- Abstrakce jen když je vhodná, trivální, přesto se nedodrží

Kdy se nevyplatí vyrobit úplně dokonalý SW produkt

- Systém má být použitelný, ale musí se upravovat (i z důvodů změn prostředí), co je pak dokonalý systém
- Uživatel je pak obvykle na dodavateli závislý a nemůže jednoduše přejít k jinému
 - Opravovat musí stále stejný dodavatel
- Úplatky, o přechod není zájem.
 - Nedokonalosti se za úplatu odstraňují a systém se „zdokonaluje“ i když to nemusí být potřeba
- Legitimní důvod: Bylo by to příliš drahé nebo příliš pozdě, někdy ani nelze (, není jasné co, faktor času, reorg cycle – v době kdy předávám je to už zastaralé)

Vyhlídky kódérů

- Vysoce kvalifikované řemeslo, z VŠ velmi kvalitní příprava, nebývá výhodné pro vůdčí pozice
 - Existence superpogramátorů (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), cloudy, správa dokumentů, dokumentová orientace, bimodální systémy

Závěr

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