

## 01 - Úvod

### Různé technologie dospějí za různou dobu

- Fast Track -2-3 roky, SMS, instant messaging. Podpora velkých, užitečnost, jednoduchost použití, relativní snadnost implementace a nasazení (využití Ex. Infrastruktury).
- Standardní: 5-8 let
- Long fuse: až 20 let, i více vrcholů, nutnost nové infrastruktury, výzkum, zákony paradigmat. Internet, nnotechno, dataintegration, SOA

### Životní cyklus SW, „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: částí; testér: integrační, funkcí, systému)
- Předání
- Údržba
- Zrušení

Pokud se takto člení i projekt, mluvíme o metodě vodopádu. To nebývá dobře a proto existují modifikace vodopádu (iterace, inkrementy, prototypy)

### System

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

### 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
- Budeme se mu hodně věnovat

### Informační systémy

- Informační systém (IS) je systém umožňující ukládání, 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 (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

## **02 - Problémy životního cyklu SW**

### **Informační systémy**

- Informační systém (IS) je systém umožňující ukládání, získávání a prezentaci informací. IS je systém, tj. strukturovaný komplex technik, nástrojů, a zdrojů umožňující získávání, ukládání a poskytování informací uživatelům a jiným systémům. Výstupem IS mohou být přímo rozkazy osobám a signály procesům reálného světa (avionika letadla, reaktor, ...). IS nemusí využívat SW, my se 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

### **Systém**

- Zdroje (lidé, materiál, znalosti a dovednosti)
- Prostředky (stroje, nástroje)
- Vazby mezi komponentami
- Procesy umožňující za daných podmínek dosahovat určité cíle, u IS poskytovat informace, doporučovat opatření

**Informační systém je vždy součástí většího systému zahrnujícího i lidi a neautomatizované činnosti a jiné systémy**



## Proč jsou IS složité i když dělají jednoduché věci

- Mění stav světa
  - Změny nelze vrátit
  - Ohrožení ekonomické a na životech (kritičnost)
  - Prvky reálného času
  - Ovlivňují zájmy politiků a nakonec i manažerů
    - Sledování kvality škol, dá se využít IT na hodnocení úspěšnosti absolventů, pokud se to neprovede klesne zájem o IT ve společnosti
- Fakticky zahrnují i lidi (sociálně politická dimenze)
  - Týkají se jejich zájmů, často podvědomých, vyžadují změnu zvyků
  - Mají politické, ne vždy příznivé, důsledky - i celosvětové i místní
  - (Koncoví) uživatelé musí formulovat požadavky společně s vývojáři, to je náročné pro obě strany
    - Obě strany se musí naučit nové věci, je nutná spolupráce různých profesí
    - Potřeba víceoborových znalostí
  - Je nutné kombinovat sílu automatizace s intuicí lidí a balancovat automatizované i neautomatizované činnosti
- Jsou rozsáhlé, dynamické a otevřené

Někdy se výhody přeceňují, jindy nedoceňují

- Slepá víra ve výstupy IS bez ohledu na kvalitu použitých dat(hnůj tam hnůj ven) a chyby ve specifikacích, závislost a počítačích. Význam některých aspektů kvality dat je silně podceňován.
- Snaha o alibi (jinde přece SAP funguje), podceňování nutnosti ručit za svou práci
- Nedohlédnutí efektů nasazení

- Kvalifikace lidí
- Organizační změny
- Efekty jinde, než se čekalo, nevyužití synergie spolupráce lidí a softwarových systémů
- Efekty se objeví později a závisí na systémových změnách
- Zvýšení produktivity, možnost na některé práce používat méně kvalifikované síly
- Ztráta produktivity při nevhodném použití (mail, Internet)

#### Podceňování nevýhod používání IS

- Ztráta kontaktu s realitou (IS je velmi často založen na neúplných datech a nezahrnuje celou lidskou zkušenost, potřebu intuice a kritického myšlení)
- Podceňování mezilidské komunikace (v ní až 60% neverbální komunikace, která např. říká jsi mi sympatický/sympatická, city a osobní vazby)
- Ztráta dovedností improvizace a práce ručně při nečekaných při situacích, se kterými vývojáři nepočítali, z pohodlnění až ztráta pracovní inteligence (příklad z půjčovny aut v Texasu)
- Ergonomie a nemoci z povolání

### **Strojová byrokracie**

- Přísná hierarchie, „vojenská“
  - Pozice jmenováním
  - Vedoucí má poslední slova a nedílnou pravomoc ve své oblasti
  - Komunikace přes nejnižšího společného vedoucího
    - IS může být zkratkou, je ale třeba logovat

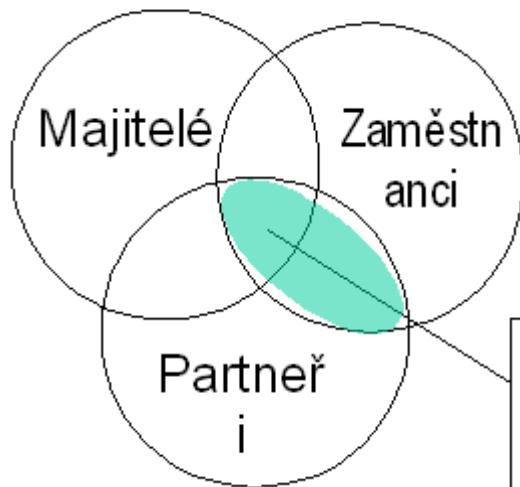
### **Profesní byrokracie**

- Pozice plyne ze vzdělání nebo z volby
  - Univerzita, státní správa, zdravotnictví
  - Vždy spojena se strojovou byrokracií

### **Podceňování lidské dimenze Koalice zainteresovaných v podniku**

- Majitelé
  - maximální zisk
- Zaměstnanci (i management)
  - Co nejvíce peněz za co nejméně práce
  - Management bývá i majitelem (vlastní akcie)
- Partneři v obchodě
  - Dodavatelé: měkké termíny, vysoká cena, nízká kvalita
  - Odběratelé: Tvrdé termíny, vysoká cena, vysoká kvalita
- Státní orgány (jen částečně)
  - Udržet zaměstnanost
  - Mít od koho vybírat daně

## Koalice v podniku



## Platí pro strojovou byrokracií

podřízeného jmenuje a  
odvolává nadřízený  
komunikace  
přes společného nařízeného

Společný zájem,  
dlouhodobá prosperita  
*IS musí být prospěšný  
pro všechny členy  
koalice*

39

## Společný zájem: dlouhodobá prosperita, to by se mělo promítnout do požadavků na IS

Vybalancovat zájmy skupin v koalici, jinak podnik zkrachuje a škodní budou z dlouhodobého hlediska všichni (pokud si management moc nenakrade)

- Vedení: Udržet dobré pracovníky – dobře platit, mimoplatové stimuly, ofensivní strategie na trhu, přednost má úspěch na trhu a až pak propouštění když už to opravdu jinak nejde
- Zaměstnanci: nežádat nereálné mzdy (viz příklad Baťa za krize)
- Snažit se vyjít vstříc partnerům, platit kvalitu
- Podpora strategie firmy (to ale závisí i na SW architektuře, která musí být otevřená)

## Klíčový důsledek

- Specifikace požadavků je ohrožena dvěma faktory
  - Neví se přesně co
    - Neznalost potřeb podniku
    - Neznalost možností IT
  - Vyvažování zájmů skupin v podniku nelze vždy předem odhadnout
    - Podpora odborného růstu, růstu mezd a zisků
    - Zlepšení postavení podniku a služeb

## Sedm S, požadavky

Informační systém ve strojové byrokracii by měl zlepšovat klíčové vlastnosti organizace charakterizované termínem 7S

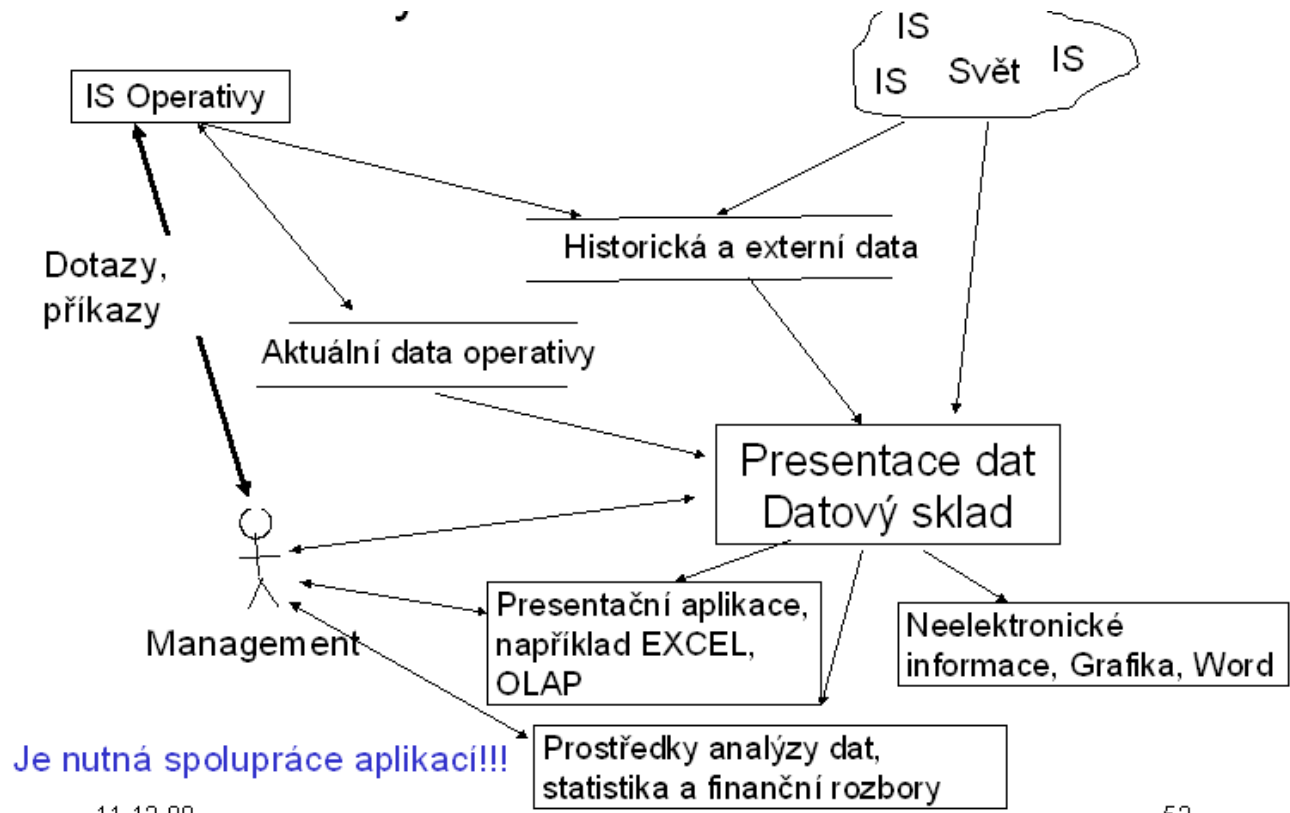
Platí to do značné míry i pro profesní byrokracii.

IS je v mnoha systém podporující nebo dokonce podmiňující chod jiných systémů (metasystém)

Informační systém by měl zlepšovat klíčové vlastnosti organizace charakterizované termínem 7S

1. Společné cíle
2. Strategie
3. Struktura a úlohy (centralizace versus decentralizace, procesy)
4. Styl managementu a podniková kultura (podpora spolupráce, identifikace s podnikem, podchycení iniciativ, využití znalostí, poskytování informací)
5. Systémy (kvalita, podpora spolupráce), sem patří IS, musí ale podporovat i jiné systémy
6. Spolupracovníci (kvalita, struktura pracovních týmů, spokojenost, kvalifikační růst)
7. Schopnosti a dovednosti, klíčové know-how, vývoj a výzkum

## Manažerský IS



Manažerský informační systém (MIS) integruje nebo využívá funkce mnoha aplikací/systémů.

Je žádoucí, aby MIS spolupracoval s obdobnými externími systémy.

Manažeři vyžadují podporu své intuice.

Nelze od nich žádat hlubší znalosti IT (mají svých starostí dost).

- To neplatí pro funkce IS, IS hlubší a širší znalosti vývojářů potřebuje

## Podpora strategie firmy

- Teorie omezení (TO) říká, že obvykle existuje jediné úzké místo. Úzké místo je nějaká charakteristika systému. Úzké místo podle TO má tu vlastnost, že pokud je nevyřešíme, nedojde i při libovolně velkých investicích ke zlepšení chování systému
  - (př. tunel Mrázovka neřeší úzké místo pražské dopravy, jeho otevřením se dopravní situace nezlepšila, totéž platí o pisáreckém tunelu v Brně)
- Úzké místo může být stejně dobře kvalita lidí, jako kvalita výrobků, kapacita výroby, vývoj nových výrobků nebo marketing.
- Určení úzkého místa je obvykle problém rozhodování s neúplnou informací, je to tedy úloha

- vyžadující zkušenost a intuici.
- Úzké místo může být jiné pro různé úkoly pro různé doby výhledu
- V české ekonomice jsou kandidáti na úzké místo pro výhled roků sociální systém, právní systém a daně, ve výhledu desetiletí demografie, především porodnost a kvalita vzdělávání
- IS by měl usnadňovat hledání úzkých míst a pomáhat je řešit

## Vývoj od začátku versus nákup (customizace)

- Vývoj od začátku „na míru“
  - Převažují nově vyvinuté moduly
  - Požadavky nemusí být omezovány tím, jaký SW máme k dispozici
- Customizace - Nákup softwaru a jeho přizpůsobení konkrétním podmínkám (přesnost dat, formáty, účetní schéma) - U velkých firem dnes customizace převažuje, trend se ale možná obrací
- SOA může kombinovat obojí

## Životní cyklus softwaru, vývoj od začátku pro monolity

- Marketing a specifikace cílů (formulace problému), hledání odpovědí na otázky proč a rámcově co. Výstupem je vize či specifikace cílů
- Specifikace požadavků, formulace přesné odpovědi na otázku co, zčásti na otázku jak.  
Výstupy
  - Model systému, diagramy a (formalizované) požadavky,
    - závisí na architektuře SW modelovacích prostředcích a použitém paradigmatu (např. objektovém), musí být srozumitelné oběma stranám
  - Termíny řešení, náklady, zdroje
 Oponentura: feasibility study (studium uskutečnitelnosti)
- Návrh systému. Řešení technických otázek dekompozice, návrhy rozhraní, struktury dat a algoritmů a volba softwarových vývojových prostředků a systémového softwaru. Dostatečně podrobná dokumentace (u agilního vývoje nemusí být rozsáhlá a obvykle to platí i pro SOA). Návrh celku může dělat specialista  
Oponentura návrhu
- Kódování (programování) částí.  
Čtení kódu
- Testování: částí – unit tests (testy částí, to dělají kódéři), integrační, funkcí, systému a předávací.
- Oživení a předání: instalace hardwaru a základního softwaru, instalace systému, předávací testy, zkušební provoz. Někdy zkušební provoz.
- Provoz a údržba.
  - odstraňování chyb zjištěných za provozu,
  - přizpůsobování novému hardwaru (HW) a přizpůsobování změnám v použitém základním (systémovém) softwaru (ZSW), jako jsou databázové a operační systémy, a konečně
  - vylepšování funkcí.
- Stažení z provozu.

## Customizace

- Generace systému z polotovarů zadáváním parametrů a jiných informací pro generaci.
- Předobraz: generace operačních systémů mainframů

- Customizace vyžaduje vysokou kvalifikaci těch, co ji provádějí

## **Životní cyklus softwaru, customizace**

1. Marketing a specifikace cílů (formulace problému), hledání odpovědí na otázky proč a rámcově co. Výběr dodavatele.
2. Specifikace požadavků, formulace přesné odpovědi na otázku co, zčásti na otázku jak od daného dodavatele.
  - Model systému, diagramy, závisí pravidlech dodavatele modelovacích prostředcích a použitém softwaru,
  - Výběr modulů a funkcí
  - Termíny řešení, náklady, zdroje
 Oponentura feasibility study
3. Customizace systému (generace systému). Zadávání funkcí, jejich vazeb a parametrů (např. přesnost, formáty dat). Tato etapa je poměrně pracná a náročná na práci expertů
4. Kódování (programování) nezbytných doplňků. Obvykle provádí dodavatel, kódování není mnohdy nutné
5. Testování: testy systému a předávací.
6. Oživení a předání: instalace hardwaru a základního softwaru, instalace systému, předávací testy, zkušební provoz.
7. Provoz a údržba. odstraňování chyb zjištěných za provozu, přizpůsobování novému hardwaru (HW) a změnám v použitém základním (systémovém) softwaru (ZSW), jako jsou databázové a operační systémy, a konečně úpravy systému. Většinu provádí dodavatel. Často spojeno i se zajišťováním provozu
8. Stažení z provozu.

## **Výhody customizace**

- Ověřený dodavatel, zná obor, ověřené techniky specifikací a oživování, know-how z mnoha instalací
- Dobré reference,
- Alibi pro management (jinde to přece fungovalo)
- Menší nebezpečí selhání projektu a toho, že dodavatel opustí trh
- Úspory-cena (má to ale háček, viz níže), hlavní úspora je u údržby
- Velká nabídka funkcí (ale nebezpečí, že se koupí i zbytečnosti a že cena proto bude zbytečně vysoká)
- Rychlejší realizace (ne závratně)
  - Osvědčuje díky malé pravděpodobnosti totálního selhání projektu,, plný úspěch také nebývá častý

## **Nevýhody customizace**

- Kupuje se vlastně něco jako konfekce
  - Může znamenat zbytečné organizační změny a tím značné zvýšení nákladů,
  - Může blokovat žádoucí organizační změny
  - Nemusí vyhovovat daným podmínkám, což může vyvolat další ztráty
  - Často se používají zastaralá řešení s dopady na funkce
  - Často se nakupují zbytečnosti – větší náklady při nákupu a při provozu, zbytečnosti mohou při provozu i překážet



- Nedostatečná lokalizace (dnes spíše jiné než jazykové problémy, např. nedostatečná implementace legislativy a nedostatečné zohlednění místní kultury)
- Ztráta vlastních znalostí a kvalifikace a nezdravá závislost na dodavateli, často nemožnost používat i svoje řešení a řešení třetích stran (řešení – servisní orientace)
- Odstraňuje spíše konkurenční nevýhodu než přináší výhodu. Vhodné spíše pro operativu.

## Pozor

- Customizace ušetří jen max. 50% času a 50-70% nákladů (specifikace zůstává pracná)
- Hlavní úspora ze sdílení údržby
- Konfekce, ale dá se nosit
- Dlouhodobé zkušenosti dodavatele
- Výhoda nového programovacího jazyka není ve zrychlení programování, přínos jsou nástroje a nové úlohy.

## Kde je úzké místo při vývoji SW

- 56% chyb vzniká ve specifikacích, stojí více než 80% nákladů a času na odstranění, stojí za většinou neúspěchů projektů a za nespokojeností při provozu
- Cena odstranění chyb
  - 82% ze specifikace požadavků, 1% kódování
- Vlastní zkušenost přednášejícího
  - Hlavním zdrojem problémů jsou nedokumentované a hlavně opomenuté předpoklady, nejasné požadavky a požadavky u nichž se neví, jak se k nim došlo (nelze je vystopovat). Ty vedou i k neúplným a neadekvátním specifikacím a jsou odpovědné i za nekonzistence a ztrátu znalostí o tom, proč se věci řeší tak a ne jinak (traceability). Tj. podobně jako v předchozím slajdu odpovídají za 80% problémů ve specifikacích.

## Růst cenu opravy

- Cena opravy se ztrojnásobí až zpětinásobí na každou etapu, kterou daný defekt projde aniž je detekován.
- 1,3,9,27,81
- 1,4,16,64,256
- Řešení“ provádět opomenuty po každé etapě

## Doporučované varianty testování

1. Testy připravuje a provádí programátor
  - Po napsání programů (kromě testů částí se to nedoporučuje)
  - Před napsáním programů (agilní vývoj, návrh testů, oponentury, vhodné)
2. Testy připravují a provádí testéři (s možnou výjimkou testů částí) a systém testují jako černou skříňku
  - Nutné a velkých a u kritických systémů
  - Pracné ale účinné
  - Je žádoucí opakování testů automatizovat. Automatizace testování je podle teorie složitosti (např. test na mrtvý kód, Churchova téze) nereálná
3. I varianta s testéry má dvě možnosti

- Testování černých skříněk (pracné, účinnější), testér zná jen rozhraní
- Testování bílých skříněk

Hlavní problém testů: dokazovat sám sobě, že jsem chyboval – proto testěři.  
Být dobrým testérem vyžaduje specifický talent.

## Trochu terminologie

- Validace: Ověření správnosti systému nebo jeho funkčního modelu pokusem
  - Standardní test (věc vývojářů)
  - Předvádění uživatelům (tak to chápe ISO 9000)

V obou případech obdobné techniky.

- Předvedení bývá spojeno s testy systému a samozřejmě s předáváním.
- Verifikace: Ověření správnosti nějakého dokumentu důkazem či oponenturou.
  - Často výstupního dokumentu etapy proti zadávajícímu dokumentu etapy (vize/cíle \* specifikace požadavků, specifikace požadavků \* návrh, ...)

Verifikace: Ověření správnosti nějakého dokumentu důkazem či oponenturou.

- Revize (review): Verifikace většího celku formou blízkou oponentuře v běžném smyslu. Používá se i u feasibility study (studie uskutečnitelnosti)
- Inspekce (inspection): Přísně formalizované oponentní řízení pro menší dokumenty s řadou činností a rolí.
- Walkthroughs: poloformální pročitání dokumentu v malé skupině, varianta je čtení kódu prováděné obvykle ve dvojici

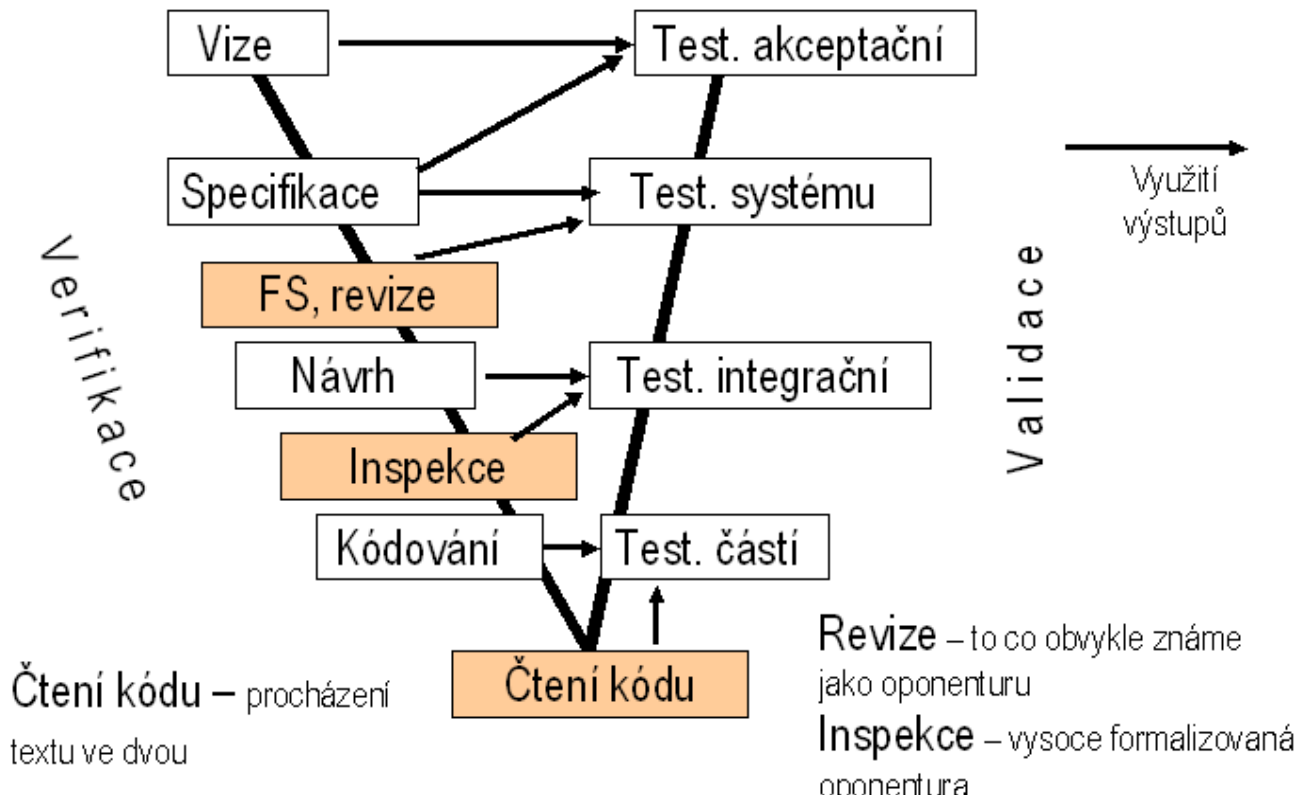
Dobře provedená verifikace je velmi účinná - až 80%, odhalí i problémy obtížně detekovatelné testováním

Vyžaduje to ale individuální nasazení, disciplinu a koncentraci, které se obtížně kontrolují a ne všichni jsou jich schopni

Validace: Ověření správnosti systému nebo jeho funkčního modelu (prototypu) pokusem (testem), někdy virtuální testování

- Předvedení prototypu (= částečně funkčního modelu systému)
- Testy částí (programátor)
- Testy integrační (testér)
- Testy systému (testér, koncový uživatel)
- Regresní testování (opakované provedení testů na používaném systému)
- Test funkcí (testuje se nově integrovaná funkce)
- Pochybení (error, fault) - mentální selhání člověka
- Defekt (defect) – místo v nějakém dokumentu, které není správně a může případně v kombinaci s jinými defekty způsobit špatnou funkci systému, ať existujícího tak projektovaného.
  - Selhání (failure) – případ, kdy systém nepracuje správně.
    - Pochybení -> defekty -> selhání
- Jak by se to popsaloby konceptuálně?
- Chyba bude označovat pochybení nebo defekt nebo selhání

## V-schema



## Architektura

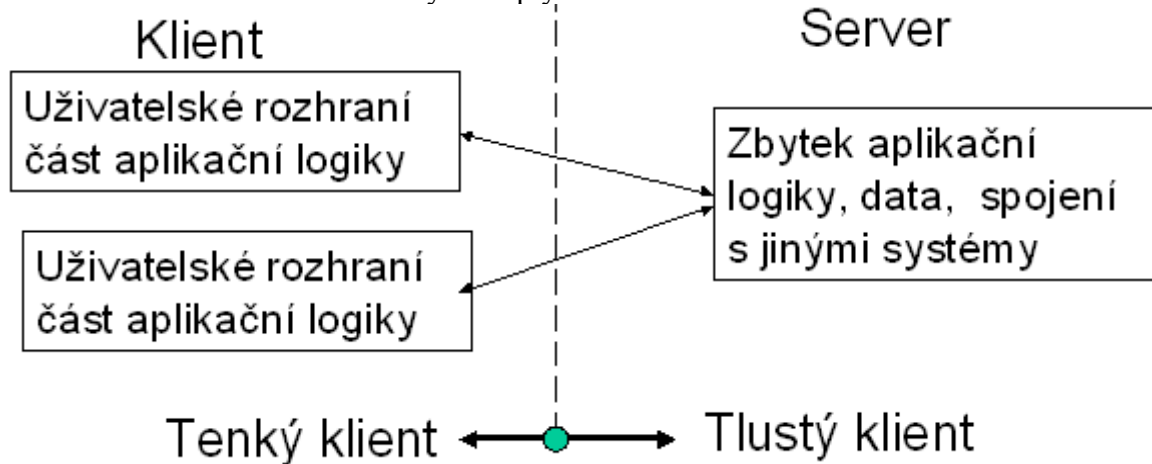
- Struktura systému ve velkém.
  - Komponenty, jejich vlastnosti a spolupráce a rozhraní navenek
- Jednotící idea/filosofie
- Základní vlastnosti rozhraní systému
- Systém může mít různé architektury z hlediska HW, logiky a fyzické dekompozice softwaru
- Architektura ovlivňuje uživatelské vlastnosti systému, dostupné operace, techniky specifikací a metodologii vývoje

## Architektura, k čemu

- Celková koncepce a návrh, jednotící idea, porozumění celku
- Usnadňuje zobrazení celku (dílenský výkres)
- Většinou prostředek dekompozice, porozumění, analýzy (např. konzistentnost a kvalita návrhu)
- Důležité pro
  - Zvládnutí vývoje
  - Znovupoužitelnost částí
  - Údržbu
  - Dostupnost některých manažerských operací
  - Podporu decentralizace
  - .....
- Usnadňuje řízení (procesy, malé etapy, inkrementální vývoj, minimální rozsah, rozšiřování)

## Architektura klient – server

- Společná správa dat a aplikací,
- Úspora paměti a nákladů koordinace (kód a společná data pouze jednou nebo málokrát, opravy na jediném místě)
- Úspora výkonu – vysvětlení – server jako služba, výkon součet nezávislých n.v. s velmi malou střední hodnotou a velkým rozptylem



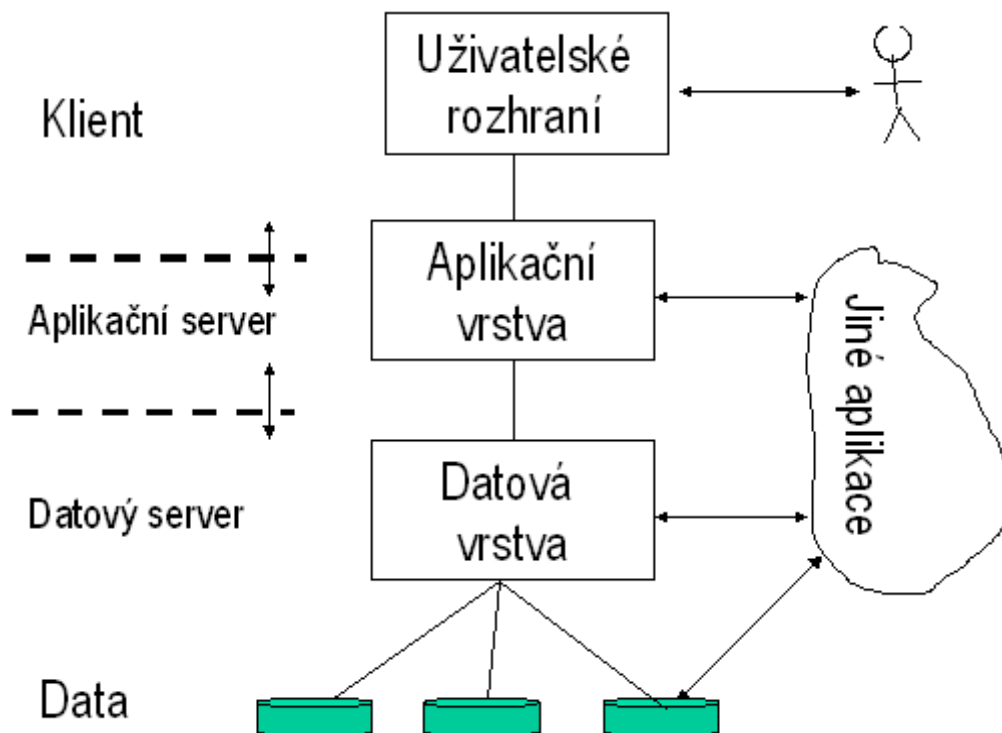
### Výhody tenkého klienta

- Úspory nákladů na hardware klientů
- Snazší správa
  - Bezpečnost
  - Méně záplat
  - Méně příležitostí k flákání hraním her
  - Lepší kontrola práce (nesmí se ale přehánět)
  - Snazší upgrade

### Nevýhody tenkého klienta

- Výpadek centrálních služeb znamená totální výpadek systému
- Některé služby mohou být lépe proveditelné lokálně.
- Neochota zvládat nové a cena přechodu na novou architekturu
- Ztráta lokální prestiže (odpovědnost je na centru)
- Obtížnější outsourcing (viz ale servisní orientaci)
- U systémů s několika málo klienty neefektivní

## Třívrstvá architektura



## 03 - Pravidla pro formulaci cílů

### Co má stanovovat cíl: zformulovat proč se IS vyvíjí

- Přispět k zlepšení poslání organizace, tj. vyhovění důvodů, proč organizace existuje (jejího poslání)
- Podnik – vydělávání peněz prvotní důvod existence
- Organizace státu – zlepšit službu občanům (s co nejmenšími náklady)

### Čeho se vyvarovat

- IS není hlavní nástroj zlepšení nefunkční organizace a odstranění jiných nedostatků podniku
  - IS sám nevytvoří koncepci podniku a nevymyslí nové výrobky a ani sám o sobě nezlepší marketing podniku
  - Počítač je zesilovač, zesiluje pořádek ale také nepořádek
  - Je nebezpečné měnit současně organizaci podniku a zavádět IS, razantní restrukturalizaci podnikových procesů (BPR) provádět opatrně, zvláště v nižších patrech hierarchie, import procesů může být kontraproduktivní

Je žádoucí podnik nejdříve uvést do funkčního stavu; jiný postup je někdy nutný, ale vždy riskantní

## Psychologický kontrakt při uzavírání pracovní smlouvy

- Lidé, někdy podvědomě, uzavírají sociálně psychologický kontrakt který má dva aspekty:
  - Ekonomický (peníze, vedlejší výhody jako rekreace, pracovní podmínky, pracovní doba, postavení, kariérní růst)

- Sociální (dobrý kolektiv, jistota zaměstnání, pořádek a práce bez stresu, prestiž firmy, zajímavá práce a někdy odborný růst)
- Při nástupu do zaměstnání je dobré vědomě uzavírat smlouvu s vědomím, jak dalece pokrývá všechny aspekty psychologické smlouvy

## **Psychologický kontrakt**

IS by měl vytvářet podmínky pro psychologický kontrakt

- Zlepšení pracovních podmínek, pořádek, menší stres
- Vlastní počítač a přístup na internet
- Vyplatí se nebránit pracovníkům v zábavě na internetu, pokud je to jen v rozsahu menší přestávky
- Může to být výhodné pro efektivnost práce a zdraví

## **Psychologický kontrakt při uzavírání smlouvy na IS**

- Všechny aspekty psychologické smlouvy je žádoucí zohlednit při vývoji IS
- Hledat spojence mezi těmi, jimž mohu nabídnout výhodný psychologický kontrakt.

## **Atributy technické složitosti úkolu**

Systém, který implementujeme nesmí být technicky podstatně náročnější na vývoj, než systémy, se kterými jsme dosud měli zkušenosti. Atributy složitosti:

- Množství funkcí, velikost systému
- Rozsah a kvalita dat,
- Interaktivnost, počet koncových uživatelů a počet rolí koncových uživatelů
- Kritičnost aplikace (riziko ztrát při špatné funkci)
- Rozsah zabezpečení
- Potřeba nových metod a nástrojů vývoje
- Nový typ úkolu, nový typ uživatele
- Příliš krátký termín
- Modifikovatelnost, otevřenost IS

## Tabulka pro hodnocení rizika neúspěchu pro vůdčí aspekty složitosti

Aspekt	0	1	2	3
Interaktivnost	Dávka	Dotazovací systém	Soft real-time	Hard Real-time
Počet on-line uživatelů	1	Desítky	Stovky až tisíce	
Rozsah dat	Gigabyty	Terabyty		
Kritičnost	Prosté IS	Ekonomické ztráty	Ohrožení životů	Vyvolání katastrofy
Velikost	Běžná	Pětikrát větší než obvyklá	Alespoň 30krát větší	Alespoň stokrát větší
Rozsah použití	Jediný uživatel	Více uživatelů	Hromadné použití	
Zabezpečení	Nízká úroveň	Běžná ochrana	Vysoká ochrana	

## 04 - Architektury SW a servisní orientace

### Servisně orientovaná architektura

- Virtuální p2p síť volně vázaných komponent
- Komponenty spolupracují předáváním zpráv
- Základní styl komunikace je asynchronní , sekvenční je nadstavba

#### Konfederace a aliance

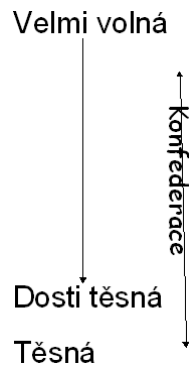
- ALIANCE
  - Na začátku komunikace se partner musí vyhledat. Typické pro e-komerci, reservační systémy atp. Nutnost standardů
- KONFEDERACE
  - partneři jsou zpravidla známi. Široká paleta aplikací, historicky nejstarší SOA systémy (soft RT systémy, e-government, řízení globálních společností, výrobní systémy, koalice podniků a zdrav. zařízení). Protokoly komunikace mohou být proprietární
- Existují i mezistupně mezi konfederacemi a aliancemi, nejsou ale časté. Často je jádro konfederace a periferie aliance. Konfederace může využívat služby a funkce vyvinuté pro aliance a naopak. Převládají unie (platí pro jádra systémů)
- Konfederace mohou mít specifické vlastnosti

## Konfederace otevřené a uzavřené

- Konfederace mohou být různého typu
  - IS podniku, části musí poslouchat centrum a systém není extrémně velký
  - IS státní správy nebo globálního podniku, části jsou téměř nezávislé - unie
  - Řídicí systém menší technologie
    - Menší komponenty, programátorské rozhraní, uzavřenost
- Klíčová výhoda konfederací – rozhraní služeb může být uživatelsky orientováno, v uniích musí být uživatelské

## Těsnost vazeb

- e-komerce
- Informační systém státní správy
- Spolupráce zdravotnických zařízení
- Možná implementace CRM, SCM
- Informační systém globálního podniku
- Řízení procesů (soft real-time)
- OO aplikace



## Pozorování

- Spolupráce SW komponent může být dávková
  - exportem a importem dat
  - pomocí společného (virtuálního) datového úložiště plněného dávkově, je-li třeba
- Spolupráce může být interaktivní, primárně asynchronní
  - Přímá výměna zpráv
  - Aktivní databáze
- Některé služby mohou být dávkové (Excel)

Systémy mezi sebou spolupracují. Aby mohly být používány rozumně, musí spolupracovat podobně jako služby reálného světa, být většinou stále k dispozici, (asynchronně) reagovat na požadavky z různých zdrojů a být používány jako černé skříňky. To reálně lze dosáhnout jen, tvoří-li virtuální p2p síť.

## Servisně orientovaná architektura v naší interpretaci

- Soubor softwarových komponent spolupracujících obdobně jako služby reálného světa
  - Všechny služby jsou stále aktivní
  - Vyřizují požadavky asynchronně
  - Konfederace – služby se znají - i aliance, kdy je službu nutno vyhledat
- Služby a infrastruktura, zajišťující jejich spolupráci (middleware), včetně možnosti čekání na odpověď
- V prakticky použitelných systémech se pravidla pro SOA vztahují pouze na část (jádro) systému. Jeho volnou součástí jsou dávkové části (export dat, hromadné výpočty, atd.)

## Kdy použít SO

- Jako prostředek podpory distribuovanosti



- Jako prostředek dekompozice na daném místě, v daném počítači
- Jako prostředek sdružování kapacit (gridovské systémy, viz projekt CETI)
- Podpora inkrementálního vývoje a jiné SW inženýrské výhody, např. znovupoužitelnost, kombinace produktů různého původu
- Podpora otevřenosti, CRM a SCM
- Aby to vůbec šlo napsat (Neexistence SOA v sedmdesátých letech byla jedním z důvodů krachu projektu protiraketové obrany SALT)

## **Výhody servisně orientované architektury**

- Chyby zůstanou lokalizované, modernizaci lze provádět po částech
- Stávající systémy mohou sloužit nadále bez podstatnějších změn, lze integrovat produkty třetích stran
- Flexibilita: Výše uvedený registr se dá snadno upravit na kontrolu dluhů a případně majetků
- Škálovatelnost: Lze snadno doplňovat nové služby a také je klonovat
- Systém je možné zprovoznit v rozumném čase, s rozumnými náklady
- Chyby zůstanou lokalizované
- Stávající systémy mohou sloužit nadále bez podstatnějších změn, lze integrovat produkty třetích stran
- Flexibilita: Výše uvedený registr se dá snadno upravit na kontrolu dluhů a případně majetků
- Autonomie částí zvětšuje odolnost systému proti výpadkům
- Dosavadní uživatelé existujících služeb jim mohou věřit, protože se v podstatě nemění, Nemusí se učit příliš nového a mohou důvěřovat i datům.
- Úspory nákladů z důvodů znovupoužitelnosti, použití produktů třetích stran a snadnějšího vývoje všeobecně

## **Důsledky p2p architektury**

- Stabilita systému zásadním způsobem závisí na stabilitě (neměnnosti) rozhraní
- Rozhraní se bude málo měnit, pokud nebude záviset na implementačních detailech a změnách úrovně technologie a bude ověřena používáním (je-li např. deklarativní a uživatelsky orientované, uživatelé v operativě používají léty ověřené postupy, viz účetnictví)
- P2p systémy se málo kamarádí s centralizovanými službami, i jen funkčně centralizovanými.
  - Takové služby lze sice někdy používat, lze však očekávat problémy
- Přidání/ubrání služeb je relativně snadné, pokud mají vhodné rozhraní
- XML a www výhodou, ne však podmínkou

## **Pattern a antipattern**

- Pattern – osvědčené řešení nějakého častého problému
- Antipattern – neefektivní až špatné řešení nějakého častého úkolu

## **Antipatterns v OO, které jsou patterns v SO**

- Použití existujících aplikací (legacy systems),
- Ostrovy automatizace (viz automatizace dílen),
- Funkční dekompozice, používání datových úložišť (DFD)

## Snadná prevence následujících antipatterns

- Plánování bez konce (lze prosadit rychlou realizaci jádra systému výběrem nejužitečnějších služeb ),
- Obtížný člen týmu (dáme mu vyvinout autonomní službu)
- Dlouhé termíny než je systém funkční (inkrementální vývoj),
- Návrh výborem který fakticky za nic neručí (nelze se se vymlouvat na fakt, že ještě nic nefunguje),
- Znovu vynalézat kolo (snadno používáme existující aplikace),
- Vendor lock-in - závislost na jednom dodavateli (lze doplňovat produkty třetích stran a nově vyvinuté komponenty)

## 05 - Vize a smlouva

### Jak formulovat otázku PROČ

Měla by vycházet z hodnocení nedostatků, příležitostí a rezerv v tom, co uživatel dělá (týká se jeho byznysu)

1. Zlepšení pozice na trhu (strategie)
2. Zefektivnění podnikových procesů (taktika a operativa), úspora zdrojů
3. Odhad skutečných přínosů plánovaného řešení nutný!!!

Mělo by být zaměřeno na plnění poslání podniku nebo organizace

- podnik – vydělávání peněz
- úřad – služby klientům/občanům
- podpora činností neziskových organizací – úspory, lepší služby, zdroje pro činnost

### Důvody zavádění IS, strategie

- Přínos IS by měl být především v oblasti strategických výhod. Využití IS je cesta jak se vyrovnat s rostoucí složitostí rozhodovacích procesů spojených s
  - s rostoucím počtem skutečností, které je při rozhodování nutné brát do úvahy,
  - se zkracováním doby na rozhodnutí,
  - s růstem rizik z opožděného či chybného rozhodnutí,
- Využití IS je cesta jak se vyrovnat s rostoucí složitostí rozhodovacích procesů spojených
  - s důsledky rostoucí migrace pracovníků. To mj. vyžaduje, aby podnik nebyl závislý na žádném pracovníku a na informacích které jsou známy jen jemu,
  - se zrychlením inovací,
  - s potřebou oběhu informací mezi všemi oprávněnými osobami nutnou pro horizontální spolupráci a zrychlení podnikových procesů.

### Důvody zavádění IS, marketing

Z marketingového hlediska jsou důležité následující potenciální možnosti IS.

- Lepší porozumění vývoje na trhu a potřebám zákazníků.
- Zrychlení inovací výrobků a služeb. Tím dosáhnout předstihu nabídky před konkurencí.

- Pro zrychlení inovací je možné zkrátit dobu potřebnou k tomu, aby se správně rozhodlo, zda je inovace nutná a jaká by měla být, zkrátit dobu vývoje a náběhu výroby.

Z marketingového hlediska IS umožňuje

- Sledování obchodních charakteristik (trendy, rozložení poptávky podle kategorie a druhů zboží atd.).
- Využívání získaných informací pro optimalizaci (např. optimalizace výrobních dávek, optimální řízení zásob atd.).
- IS umožňuje zachytit i méně zjevné či téměř skryté skutečnosti, např. fakt že existuje rezerva v prodeji u prodejce s vysokým obratem (zásobuje více velkých zákazníků než ostatní).

## **Překážky přínosů IS**

Skrytým zdrojem růstu nákladů na IS bývá

- nutnost příliš velkých organizačních změn (prodlužuje to dobu zavádění IS a snižuje po jistou dobu výkon, zvyšuje rizika, zvyšuje náklady),
- snaha o naprostou úplnost a dokonalost oproti včasnosti. Specifikace mohou být, jak víme, dokonalé jen postupně při postupném budování systému využíváním zkušeností s dosavadním provozem.
  - Přesné postupy mají smysl jen pro kvalitní data, jinak jsou kontraproduktivní
- IS je drahé zboží, které poměrně rychle zastarává.
- Je tedy důležité, aby IS byl uvedeno do provozu včas i za tu cenu, že budou zprvu zprovozněny jen hlavní funkce.
- Cenu IS při metodě velkého třesku (všechno naráz) zvyšují ztráty vzniklé tím, že IS nepracuje během vývoje a uvádění do provozu a že vývoj nelze příliš zkracovat..
- Dlouhá doba zavádění IS do provozu zkracuje vlastně i dobu, kdy bude IS v provozu (od optimální doby provozu je nutné odečíst dobu zavádění). Je tedy důležité IS oživit co nejrychleji.
- Jsou známy případy, kdy odkládání uvedení IS do provozu pro nepodstatné maličkosti způsobilo ztráty z přínosů ve výši několikanásobně převyšující cenu IS. Stalo se dokonce, že kvůli odkladům nebyl vcelku vyhovující IS vůbec uveden do provozu.

## **Stupně k uzavření smlouvy na straně uživatele**

- Uvědomění si základních potřeb a hlavní body vize (proč, zčásti co)
- Jednání o vizi a cílech projektu, rozhodnutí o servisní orientaci
- Poloformální písemná analýza potřeb
- Vyhledání potenciálních partnerů (výběrová řízení) s případnou pomocí poradenské firmy
- Rozhodnutí o míře outsourcingu a rozsahu nákupu customizovaných systémů
- Výběr partnera (partnerů)
- Uzavření (rámcové) smlouvy
- Uzavírání smluv na jednotlivé etapy a realizace etap

## **Uvědomění si základních potřeb**

- Situace na trhu: Existují (potenciální) zákazníci
  - Kteří, kde, kolik jich je
- kteří potřebují něco z toho, co umíme poskytnout

- Co
- Jak moc
- Proč právě od nás
- Proč to chtějí a co si na tom cení
- Jak dobře to děláme a za kolik
- Co jsou ochotni zaplatit
- Jak dosáhnout toho, abychom to dělali stále lépe
- V čem bychom se měli zlepšit a jaká jsou rizika

## Tři dimenze projektu

- Při uzavírání smlouvy je nutné sladit dimenze:
  1. Termín dokončení (čas řešení)
  2. Funkcionalitu (počet a komplexnost funkcí)
  3. Peníze
- Je dobré si ujasnit, který aspekt je rozhodující (zda je nutné dodržet především termín, lze ale překročit rozpočet nebo omezit funkcionalitu). Snaha vyhovět všem třem aspektům vede často k potížím.
- Realistický termín nelze při pevném rozsahu funkcí zkracovat o více než třetinu

## Nejčastější prohřešky

- Syndrom pejska a kočičky (vše co by se mohlo hodit), typické pro customizovaný SW, nevyhýbá se ani vývoji od počátku
  - To by se snad mohlo hodit, tak to koupím. Nepotřebné ale překáží, nejen v domácnosti
    - Všechno hned (velký třesk), neuzavírat rámcovou smlouvu i když to architektura SW umožňuje v domnění, že dodavatel pak neodejde od nedokončeného díla, podceňovat problémy se zaváděním systému naráz
- Nezvažovat, která řešení jsou důležitější a která méně a podle toho řídit projekt i nákup a oživování
- Nadměrná snaha mít SW od jednoho výrobce (antipattern Vendor Lock In)
- Nezajištění správné spolupráci mezi vývojáři a uživateli

## Minimalizace rozsahu

- Začínat od co nejmenšího již užitečného systému pokrývajícího nejurgentnější potřeby,
  - Při rozšiřování systému se zaměřit na nejvýznamnější dosud nepokryté potřeby aktualizací odpovědi na otázku PROČ
  - Přijímat požadavky až po oponentuře daného proč;
  - Vždy vážit, zda použít hotovou komponentu a jak nejlépe kombinovat ruční a automatizované prostředky a zda není ruční práce efektivnější
- Minimální řešení by mělo být zaměřeno na rozhodující omezení ve smyslu Goldratta, pokud se ho podaří identifikovat, jinak alespoň na několik málo problémů s největším dopadem
- Problém: je nutný inkrementální nebo iterativní vývoj a tedy je nutná rámcová smlouva s ne zcela jasně specifikovaným předmětem dodávky. To vyžaduje důvěru mezi partnery. Tento problém je ještě vyhraněnější pro agilní formy vývoje

## Je rozumné začínat od co nejmenšího použitelného jádra?

- Námitka:
  - Zákazníkovi bude stačit to jádro a já přijdu o kšeft
- Řešení:
  - Dobře fungující systém inspiruje zákazníka k novým rozumným požadavkům tím spíše, že nedošlo k hororům při zavádění systému
  - Některé funkce jsou možné až po oživení jádra, např. data pro management podniku. Až tehdy management zjistí, co může rozumně chtít. A bude za to ochoten platit (pokud není na hlavičku)

## Vyloučit rizika ztráty znalostí

- Přílišná závislost na počítači
  - Ztráta kontaktu s realitou
    - Co je ještě OK, závislost na datech opomíjena (praktik okamžitě poznal, že hala je špatně navrhnutá, ostatní slepě věřili počítači, který ale počítal pro nesprávná data), IS vychází z modelu, který nemusí být OK
    - Přílišná víra ve výstupy (kvalita dat a meze výpočtů)
- Ztráta schopnosti improvizovat
- Ztráta znalostí lidí
- Ztráta výhod mezilidské komunikace (komunikace mezi lidmi může být i neverbální)

## Užitečná pravidla pro formulaci cílů (vizí) projektu

Čeho se vyvarovat

- IS není hlavní nástroj zlepšení nefunkční organizace a odstranění jiných nedostatků podniku
  - IS sám nevytvoří koncepci podniku a nevymyslí nové výrobky a ani sám o sobě nezlepší marketing podniku
  - Počítač je zesilovač, zesiluje pořádek ale také nepořádek
  - Je nebezpečné měnit současně organizaci podniku a zavádět IS, razantní restrukturalizaci podnikových procesů (BPR) provádět opatrně, zvláště v nižších patrech hierarchie, import procesů může být kontraproduktivní
- Je žádoucí podnik nejdříve uvést do funkčního stavu; jiný postup je někdy nutný, ale vždy riskantní

Strategické cíle mají přednost i když i zlepšení operativy je významné, nebývá rozhodující. Co hlavně zohledňovat

- Chování na trhu (marketing, CRM, SCM,...) je klíčové, úspory uvnitř podniku (lidé, zásoby) jsou důležité, ale ne zásadní
- Musíme zohledňovat zájmy všech členů koalice v podniku
- Informace jsou dostupné každému, kdo je potřebuje
- Zlepšování sociálního kapitálu (ztotožnění s podnikem, spokojenost, dobré klima, menší stres), sociální kapitál může mít efekty srovnatelné s investovaným kapitálem
- Uplatňovat principy učící se organizace (znalosti se uchovávají a postupy zdokonalují, viz CMM)
- Cíle by měly být stanoveny kvantitativně, ne však na úkor intuice
- Minimalizovat okamžité organizační změny

Strategické cíle mají přednost (pokud není nebezpečí z prodlení), pro stanovení strategických cílů

je nejdůležitější intuice a schopnosti managementu, IS má jen podpůrnou úlohu při formulaci cílů byznysu, ale zásadní význam při implementaci změn a při kontrole efektů

## Výběrová řízení

- Cíl: najít nejlepšího dodavatele (někdy partnera) veřejnou poptávkou. Ve státních orgánech a někdy i pro přiznání dotací ze zákona.
- Jednostupňová varianta
  - Veřejná poptávka
  - Příjem a vyhodnocení nabídek
  - Výběr nejlepšího
  - Uzavření smlouvy
- Vícestupňová varianta
  - Veřejná poptávka
  - Příjem a vyhodnocení nabídek
  - Výběr nejlepších dvou-tří nabídek
  - Vybraní vypracují za úplaty podrobné nabídky
  - Výběr nejlepší nabídky,
  - Uzavření smlouvy
    - Je možné požadovat integraci toho nejlepšího ze všech nabídek
- Některé špinavé triky
  - Nedostatečná publicita (např. v regionálním listu v druhé části republiky)
  - Odmítnutí nabídky lpěním na nepodstatných formálních maličkostech s tím, že někteří žadatelé nejsou požádáni o nápravu (indoš, mýto)
  - Stanovení nesmyslných podmínek tak, aby jim vyhověl jediný žadatel, př. Mýtné.
- Obrana: Alespoň dva posuzování, audit průběhu, vícestupňové řízení, dohled odpovědných orgánů, sledování možnosti korupce
- Bohužel je to věcí orgánů, jejichž činnost máme jen malou možnost ovlivnit. Lze na to upozorňovat

## Spolupráce s poradci

- Jsou vlivní, ale za výsledek přímo neodpovídají, často poskytují hlavně alibi pro management, ručí přes renomé své firmy (dá se zneužít: Andersen Consulting a Enron)
- Mají rozsáhlé znalosti, viděli leccos, vymakané postupy
- Umí vyhledat lidi s rozhodujícími znalostmi a vlivem (šedé eminence) v organizaci i mimo ni

## Problém volby partnera, podmínky:

- Oboustranný prospěch je možný (strategie vyjednávání vítěz-vítěz, získají oba)
  - V dlouhodobé spolupráci jinak nelze, jinak trátí oba
  - Při vývoji IS je typická dlouhodobá spolupráce
- Hodí se do portfolia našich zákazníků
  - Dodavatel a odběratel by se neměli výrazně lišit co do velikosti
  - Odběratel požaduje něco, co umím

- Je ekonomicky zdatný (zaplatí, asi nespadne, bude asi dobrý při jednání a poněvadž jako úspěšný bude pravděpodobně vědět, co je potřeba a co má chtít).
  - Indikátory: Růst, zisk a růst obrátu, pověst na trhu, předchozí zkušenosti s daným partnerem, zkušenosti jiných s daným partnerem (pozor na „restart“), požaduje to, co umím
- Dobré reference (vysoký počet referencí na partnera ale může znamenat, že se nebude vývoji IS věnovat dostatečná pozornost, neboť zákazník netlačí bota a tak neví, proč by se měl důkladně zabývat něčím takovým, jako je IS)
- Vstřícnost
  - Podpora managementu a vstřícnost partnerů
    - Dodržování termínů, dochvilnost, čas na jednání
    - Vstřícnost při jednání (vítěz-vítěz), snaha najít oboustranně přijatelná řešení
    - Zajištění účasti a dostupnost koncových uživatelů bez ohledu na pozici, včetně odměn a vytváření časového prostoru a odměn za vícepráci spojenou s vývojem IS, styčný důstojník při běžném vývoji, spolupráce při agilním programování
    - Zajištění zdrojů (místo, dokumenty, jiné prostředky)
    - Přístup do pracovního procesu a k potřebným pracovníkům
- Dojem
  - Pověst, reference
  - Pořádek na pracovištích i mimo ně,
  - Pravidelný pracovní rytmus,
  - Kvalita a pořádek na sociálních zařízeních.
    - Tento indikátor je kupodivu velmi významný, teď ale již méně než kdysi
- Má parametry, na které jsem zvyklý
- Velikost: není podstatně větší než já (velký asi raději použije SAP, velký se s malým těžko domlouvá – mají jiný pohled na svět, př. Berit Brno),
  - technický důvod: Je dobré řešit několik zakázek současně, aby problémy s některou zakázkou, nebo její ukončení neohrozilo firmu
- Je činný v oblasti, se kterou mám zkušenosti
- Nebudu muset podstatněji měnit podnikovou kulturu (někdy ale nutné, např. když podnik ovládly soupeřící kliky)
- Používá typ organizace, se kterou mám zkušenosti

## Organizační typy (teorie organizace), opakování

- Jednoduchá struktura.
  - Nejsou explicitě stanoveny role, vše na okamžité dohodě (jako tým je známo pod jménem horda)
- Ad-hoc-kracie
  - Role se na jistou dobu dohodnou (demokratický tým)
- Strojová byrokracie
  - Vše přes společného šéfa, přísně stanovené role, typické pro vojenské jednotky
- Profesní byrokracie
  - Pozice plyne z profesní způsobilosti (lékaři, akademická sféra, státní správa )
- Divizní struktura (decentralizace)

## Ad-hoc-kracie

- Role se na jistou dobu dohodnou (demokratický tým)
- Nízká organizační pyramida
- Dobře vyvinutá horizontální specializace (jaké typy činností kdo dělá), málo vertikální (kdo koho řídí), vysoká flexibilita
- Vhodná
  - Pro nová nikoliv extrémně velká řešení
  - Pro situace vyžadující přímou, častou a rozsáhlou komunikaci mezi pracovníky
  - Pro vývoj a výzkum, jsou-li k dispozici kvalitní lidé, pro agilní formy vývoje
  - Pro virtuální týmy spolupracující přes internet (GNU, virtuální ad-hoc-kracie)
- Podmínky:
  - Nekritické aplikace
  - Nepříliš tvrdé termíny
  - Kvalitní pracovníci
  - Nová řešení
- Demokratický tým je typický pro menší SW firmy a SW týmy
- Tým založený na jednoduché struktuře se v softwarovém inženýrství často nazývá HORDA

## Strojová byrokracie

- „vojenská organizace“
- Centralizace: Vysoká org. pyramida, veškeré dohody přes nejnižšího společného šéfa, jasné vymezení rolí a nadřízeností a podřízeností
- Náplň práce je do značné míry určena postavením v pyramidě
- Vhodné pro rychlou předem „nacvičenou“ reakci na známé situace a pro spíše rutinní činnosti
  - Nutná v armádě. Často nutná u velkých organizací, pokud se nemohou, nebo nechtějí decentralizovat
- Flexibilita závisí od kvality vedení, nebývá nikdy vysoká
- Žádoucí podporovat iniciativu a horizontální vazby

## Strojová byrokracie a IT

- Předpoklad, že IT sníží výšku pyramidy se potvrdil jen zčásti, spíše usnadnil decentralizaci.
- IT usnadňuje vedení opravdu velkých organizací, podporuje horizontální spolupráci urychlením cest přes šéfy, často ale neformálně i přímo (horizontálně)
- Je nutno pro IT získat šéfy. Je ale riziko, že se nedostaneme ke koncovým uživatelům.
- IT často ohrožuje střední management, bez něhož ale nelze systém specifikovat a uvést do provozu, jsou nutné kompromisy (jen taková řešení, která ty střední manažery, které potřebujeme, ohrozí jen málo)
- Podniky se strojovou byrokracií bývají konzervativní. Často čekají, jak počínající podniky založené na ad-hoc-kracii uspějí a pak je koupí nebo napodobí (ESB, Sonic software a Progress Software, do značné míry i rozhraní Windows)

## Profesní byrokracie, opakování

- Pozice a pracovní náplň plyne z profesní způsobilosti nebo z politického pověření - z



- „papíru“ (lékaři, akademická sféra, státní byrokracie)
- V části organizace je vždy nutná strojová byrokracie (důvodem je potřeba kontinuity a profesionality), s tou je vhodné domlouvat detaily IS.
- Vedoucí bývají pověřeni vedením nebo zvoleni jen na určitou dobu. Není proto výjimkou, že nemají zájem o dlouhodobé projekty a často ani pro ekonomiku provozu

Při budování IS je nutné balancovat mezi šéfy volenými a byrokraty.

- Rizika změny cílů na začátku dalšího volebního období.
- Bývají nejasnosti v dělbě pravomocí
- Nebývá ochota efektivně spolupracovat, tendence ke všimnému. Dosti obtížná varianta spolupráce.

## **Divizní struktura (decentralizace)**

- V globalizovaném světě stále častější
- Víceméně nutná pro opravdu velké organizace
- Centrum stanovuje rámcové podmínky a alokuje zdroje, o které divize fakticky bojují na základě svých výsledků, vizí a někdy i známostí. Divize si tedy mohou i tak trochu konkurovat (viz VW a Škoda Mladá Boleslav)
- Přechází až do spolupráce nezávislých (dceřinných) organizací
- Šetří náklady, zvyšuje otevřenost, zvyšuje flexibilitu včetně outsourcingu a insourcingu, prodeje a nákupu částí
- Divizní struktura je hlavní oblast uplatnění SW konfederací. IS budovány často zdola metodou pokusů a někdy i omylů. Důležitost vizí. IS usnadňují decentralizaci a organizační změny. Usnadňuje budování virtuálních sítí podniků

## **Zásady mezilidské komunikace**

- Schopnost ptát se
  - Dobře formulovat otázku
  - Střídat uzavřené otázky (odpověď ano/ne nebo volby varianty odpovědi) s otevřenými
  - Při prodeji má během vyjednávání mluvit hlavně kupující
- Umění naslouchat
  - Slušně se zeptat se, mám-li pochybnosti zda rozumím
  - Neskákat do řeči, dávat najevo, že sleduji (hm, gesta, pohled)
  - Parafrázovat komentáře kolegů, ale slušně a je-li obava z nedorozumění
- Umění kritiky
  - Věcnost, ne emocionalita. Ne ale záporné emoce k partnerovi (nedávat je alespoň najevo)
  - Začínat pochvalou toho, co je dobré, je-li co, nebo alespoň něco ve smyslu, že to přispělo k porozumění problému
- Umění uznat přínos
  - Co nejdříve po tom, co je k pochvle důvod
  - Přesně říci a shrnout, co se skutečně podařilo a kde to lze využít
- Umění využívat nápady
  - Nápady se snažit využít a dobré rozvíjet, hledání nápadů není sportovní intelektuální soutěž, neprosazovat své za každou cenu
- Umění předávat instrukce
  - Jasnost, jednoznačnost

- Adresáti měli svými slovy zopakovat obsah instrukce (test porozumění)
- Umění být struční a včas ukončit řeč či příspěvek do diskuse
- Umění zvládání konfliktů
  - Zviditelňovat skryté potřeby a motivace stran
  - Sladit zájmy a cíle stran, i v diskusi je dobré používat zásadu vítězí oba (zde ve smyslu, že se nikdo nedostane do nepříjemné pozice)
- Asertivita
  - Názory a pocity vyjadřovat neútočně a klidně
  - Budit důvěru a šířit dobrou náladu a optimizmus
- Umění řeči těla
  - Gestikulací a postojem budit dojem otevřenosti znalostí a talentu
  - Nepotlačovat mimiku vyjadřující pozitivní emoce
  - Budit dojem přímosti a přiměřené sebejistoty, nežmoulat ruce, se nedívat stranou a také necivět upřeně do očí atd.

## Základy vyjednávání

- Cílem vyjednávání je dosažení dohody (v podstatě formální nebo neformální formulace nějaké smlouvy) vyjadřující zájem jednajících stran změnit ve vzájemné spolupráci stávající stav.
- Cílem jednajících stran je dosažení stavu, který bude pro ně „výhodnější“ (žádoucnější) než stav stávající (já chci jablko a je to pro mne důležitější, než peníze, které budu muset zaplatit, u prodejce je to naopak)
- Dosažení cílového stavu budu muset něčím „platit“, něco pro to obětovat (peníze, práci, ...), výši platby chci minimalizovat.
- Vyjednávání proto obsahuje stanovení předmětu smlouvy (koupě jablka), a (abstraktních) nákladů – (jablko a peníze), které musí strany „investovat“.

## Tvorba koláče a licitace

- Licitace je při jednání obvykle skutečnou licitací v běžném smyslu toho slova
  - partneři se vstoupí do licitace s návrhy, které jsou pro ně velmi výhodné, o nichž si však myslí, že ještě stále nejsou tak nevýhodné pro partnera, aby hned odstoupil od vyjednávání
  - Partneři se pak snaží postupným slevováním ze svých požadavků usmlouvat vzájemně přijatelné podmínky
  - Je nutný talent jednat a schopnost najít ještě přijatelnou výchozí nabídku a rozumně „ustupovat“



- Z pozorování a z prosté úvahy lze učinit závěr, že pokud si partneři důvěřují a nemusí se proto všelijak jistit a skrývat citlivé informace, je možné dosáhnout stavu, kdy je smlouva výrazně výhodnější pro obě strany, než by tomu bylo v případě vzájemné nedůvěry (viz. příklady z Lax). To je typické zvláště pro dlouhodobou spolupráci.
- Mělo by být snahou obou stran, aby smlouva byla vyrovnaná a přinesla maximální

uspokojení oběma stranám. To je podstata taktiky vítěz-vítěz. Teoretici vyjednávání ukazují, že taktika, kdy jedna strana ztrácí, vede dlouhodobě ke ztrátám všech.

## **Proč i jednostranný vítěz, i když přímo nekrade, ve schématu vítěz-poražený trátí**

- Ztratí zákazníka, neboť ten např.
  - Spadne
  - Nebude mít na další investice
  - Půjde ke konkurenci
- Nezíská další zákazníky (pověst)
- Tratí
  - Zleniví, neprovede modernizaci (Telekom)
  - Přispěje k devastaci trhu (nedůvěra při vyjednávání a proto ztráty, špinavé praktiky, málo peněz)
  - Zhorší politické podmínky a zvýší šance extrémních politických stran
  - Přispěje k zhoršování ekonomických podmínek

## **Tvorba koláče a licitace u IS**

Zvláštnosti pro IS

- Vyjednáváním lze dosáhnout toho, že se za dané peníze nebo s jen málo většími náklady dodá podstatně užitečnější systém (zvětší se podstatně koláč)
- Obvykle je nutná dlouhodobá spolupráce,
  - Je potřeba se chovat slušně a nevyužívat krátkodobé výhody, např. svoji převahu ve znalosti IT. Z dlouhodobého hlediska se to nevyplatí. Je proto nutný přístup vítěz-vítěz
- Předmět dohody se může měnit (vše naráz nebo postupně, výběr funkcí), odběratel nemusí mít jasno v tom co vlastně chce.
  - Při daných nákladech může dodavatel optimalizovat obsah IS z hlediska přínosů pro uživatele (a sám z toho trochu získat)
  - Modifikace předmětu uzavírané smlouvy se může kombinovat s cenovými nabídkami (licitací)
  - Výhodné je používat dobré poradce
- Zkušenosti ukazují, že optimalizovaná nabídka může po realizaci zvýšit zájem odběratele o další spolupráci a o rozšiřování systému – princip vítěz-vítěz nutností

## **Vyjednávání a společenské chování**

- Důležité jsou obecné zásady slušného chování a společenského postavení
  - Dochvilnost, ceremoniální zvyky
  - Oblečení odpovídající situaci (za ředitelem nepůjdu v džínách, za kovářem ve fraku), pohodlné (nepohodlnost může vyvolat dojem nejistoty v předmětu jednání)
  - První dojem bývá důležitý
- Organizační podmínky úspěšného vyjednávání
  - Dost času na jednání
  - Příjemné prostředí
  - Dostupnost účastníků a expertů a budoucích koncových uživatelů (to závisí na správné volbě termínu, s účetními je lépe nevyjednávat v době uzávěrek)

## Organizace vyjednávání

- Přípravenost k jednání (vše, co se dohodlo, si pamatují, vše, co jsem si mohl sám zjistit, znám)
- Dokonalé zajištění lidí a ostatních podmínek vyjednávání
- Dohoda o formátu a způsobu vyjednávání (dvojice, porada,...); vyjednávání na dálku (mail, telefon) je možné, ale spíše na doladování detailů a přípravu vyjednávání tváří v tvář,
  - při vyjednávání na dálku chybí řeč těla (až 60% komunikace)

### Místo konání

- Snadno dostupné pro obě strany
- Je lépe, když je to v místě odběratele, neboť jsou dosažitelní lidé, jejichž účast se ukáže potřebná až v průběhu jednání, má to však nevýhodu, že účastníci budou odvolávání k problémům z pracovního procesu
- Jednání v místě může umožnit hostujícímu partnerovi nepříjemný vhled a poskytnout i informace, které by mohlo být pro zákazníka lépe nechat pod pokličkou

## Řeč těla závisí také na kulturních zvyklostech jako součást společenských ceremoniálů

Svraštění čela

Zvednuté obočí

Záklon

Předklon

Pohled na partnera (nepříliš upřený)

Pohled stranou

Hraní si s tužkou, prsty

Křečovitě sepnutí rukou

Nohy zkřížené nad kotníky, ruce u těla

Pohodlné sezení, zkřížené nohy nebo nohy od sebe směrem k partnerovi

Přemýšlení, rozhořčení

Překvapení, nesouhlas

Nezájem, odmítání, pauza na přemýšlení

Zájem, přestávka by bodla

Zájem, opravdu?

Nejistota, ztráta koncentrace

Obavy, nervozita

Strach, nedůvěra

Nejistota

Pohoda, sympatie, důvěra, trochu i sebedůvěra

## Cíle a účel porad

1. Rychlé předávání informací a příkazů s podporou osobních kontaktů.
2. Rozvoj osobních vztahů, týmové loajality a týmového ducha
3. Synergické efekty týmového hledání návrhů a řešení (více lidí více ví a vzájemně se stimulují), někdy podpořeno večírky
4. Oponentury a posuzování návrhů a možností,
5. Integrace dílčích řešení

## Obecné zásady vedení porad

- Příprava předem
  - Zajištění účasti a rezervace místa
  - Je vhodné rozeslat materiály k poradě předem
  - Dohodnout styl debat (někdy i školením)
- Těsně před zahájením

- Prověření místnosti, presentační prostředky (flipchart, dataprojektor atd.)
- Rozestavení židlí
- Přístup do DB a na internet, je-li třeba (pozor, aby to neblokovalo synergii spolupráce lidí)
- Při zahájení
  - Zopakovat pravidla hry (program, volba témat a úhlů pohledu, pravidla přihlašování do diskuse a psaní na flipchart, délka příspěvků)

Doba trvání nejlépe do hodiny, max. 3 hodiny. Přestávky po hodině, krátké (historky..) alespoň po 30 min.

Má být stanoven vedoucí (moderátor) porady, tato role může být rozdělena, pak administrativu spojenou s poradou dělá zástupce vedoucího, u brainstormingu může role moderátora chybět a pokud nechybí má být neformální.

- Stimulující vedení (moderování) porady
  - Zahájení se shrnutím cílů porady
  - Pokud možno nenápadné vedení diskuse, hlavně aby se dostalo na všechny, dodržování fair-play (vítězi všichni)
  - Formulace poznatků, pokud s tím mají účastníci problémy
  - Vedoucí de facto (všichni ho uznávají)
  - Doba trvání do dvou hodin, s velkou přestávkou do 3 hodin, u brainstormingu raději do 90 min. Krátké přestávky po půlhodinách
- Nápady a řešení (i zdánlivě bláznivé) se ihned zapisují a to tak, aby byly stále na očích všech (např. na flipcharty, lze navíc zapisovat i do databáze), to platí především u brainstormingu, je vhodné i jindy. Zápis na flipchart (lepší než do počítače):
  - Zvyšuje to pocit závažnosti, objektivnosti,
  - Zpřesňuje formulaci myšlenek,
  - Zbaví zápisy osobních ostnů
  - Nic se neztratí, nápady nezapadnou
  - Zlepšuje to komunikaci a brání zacyklení diskuse
  - Je nutné pro paralelní formy vedení porad
- Je vhodné, aby zápisy dělal pověřený člen skupiny - zapisovatel

Výstupem porad je zápis o jednání a obvykle i několik listů flipchartu.

- Zápisy na flipchartu je třeba konzolidovat a zredigovat
- doplnění odkazů,
  - vyloučení duplicit a nejasných formulací,
  - řešení kontroverzí a rozporů,....
  - a zanést do databáze projektu, je-li třeba a je-li kam, využít stránku projektu
- Následuje po ukončení porady
  - Vymezení úkolů,
  - Kontrola využití závěrů
  - Případné naplánování následné porady

## Použití porad při hledání nových cest

1. Na začátku je třeba se rozhlédnout a zjistit, jaké věci by se měly řešit, v čem je problém, co se o věci ví ve světě a co stojí za to, bychom se měli věcí zabývat. Přitom je velmi potřeba

- pozorně vyslechnout lidi ve vlastní organizaci a zcela na počátku požádat o pomoc manažery firmy. To řeší úvodní porady
2. Pak je nutno ve spolupráci se spolupracovníky vytipovat problémy a najít cesty jak řešit vytypované problémy. Řešitelské porady, těch se mají účastnit i uživatelé i vývojáři
  3. Nakonec je třeba přijatá řešení podrobit formální oponentuře. Oponentury by měly končit zápisem, v jehož důsledku by se mělo stanovit, jaký je výsledek oponentury (obhájeno?) a jaká jsou následná opatření

## **Jaká rizika jsou spojena s poradami**

- Špatné rozhodnutí z nedostatku informací
  - Přehlédnutí důležitých skutečností
  - Převáží názory řvounů, opomenou se nesmělí
  - Nepostřehne skryté emoce
  - Důležité informace se díky špatné organizaci porady vůbec nevynoří
- Lidé závěry nevezmou za své
  - Mají důvod je považovat za chybné nebo pro ně nevýhodné
  - Mají pocit, že nebyly respektovány jejich znalosti (minule se to přece takto úspěšně řešilo)
  - Mají pocit, že se bezdůvodně jejich názor neprosadil proto, že ani oni nejsou bráni dostatečně vážně
  - Nemohli vyjádřit svoje intuitivní pocity

### **Zabíjáci smysluplné debaty**

- Dlouhé monology, odbíhání od tématu, nedodržování pravidel
- Osobní výpady nebo jen pocit, že jde o snížení prestiže
- Zatvrzelost při hájení stanovisek
- Neschopnost mezilidské komunikace

### **Obrana proti zabijákům**

- Dodržovat pravidla slušnosti při dialogu a zásad mezilidské komunikace
- Vedení diskuse zkušeným moderátorem
- Stručně k věci
- Nezačínat negativně, např. vždy konkrétně pochválit, je-li co, a to tak, že není pochyb, že to myslíte vážně (je dobré hledět při pochvale přímo do očí). Nesouhlas skrýt do návrhu jak nápad vylepšit s využitím toho, co je využitelné v tom, s čím nesouhlasíte.
- Není-li zbylí, je nutno nesouhlas vyjádřit přímo, ale neurážlivě a fundovaně (př. Marek Eben)
- Neospravedlňovat se, nepotvrzovat se

## **Techniky porad**

- Schůze, oponentura
  - V SW se používají různé varianty oponentur (revize, procházení, inspekce)
- Brainstorming
- Workshop
- Interview
- Audit

## Schůze

- Presentace dokumentu, úkolu nebo záměru (referát),
  - text je většinou vhodné ho rozdat účastníkům předem
- Krátká diskusní vystoupení,
- Závěr
- Varianty
  - Seznamovací
  - Kontrolní
  - Oponentury/revize (review)
- Zřídka hledání nových idejí

## Brainstorming

- Neformální porada s cílem najít nová řešení, vize a myšlenky
  - Krátký úvod, kde nás bota tlačí, možné využití podnětů managementů
  - Okamžité nápady se hned zapisují na flipcharty a obvykle neformálně hodnotí, i bláznivé nápady se neztracují a zapisují, zápisy do 2 řádek,
  - zapisovatel
  - Vše, co se napsalo má být na očích
  - Vyhodnocení a koordinace nápadů již nebývá součástí porady
  - Není jasné, zda se má vždy formalizovat postavení moderátora,
  - Používají se různé techniky řízeného přemýšlení a formulace strategických záměrů

## Workshop

- Pro hodnocení a kontrolu průběhu prací, získání přehledu o stavu prací
  - Úvod
  - Řada kratších vystoupení – prezentací výsledků s diskusí
  - Shrnutí a závěr

## Interview

- Varianta porady, kde se jedni převážně ptají a druzí odpovídají
  - Ve dvou (moderátor a respondent)
  - Skupinové
  - Probereme podrobně později

## Audit

- Varianta porady, kdy se má zjistit, zda se řešení (ekonomicky) neodchyluje od plánu a zda je naděje na dosažení cílů co do obsahu i termínů
  - Často za účasti managementu a vnějších auditorů
  - V komplikované formě známé jako kontrolní den

## SWOT tabulka

- Velmi používaná technika volby cílů používající následující atributy:

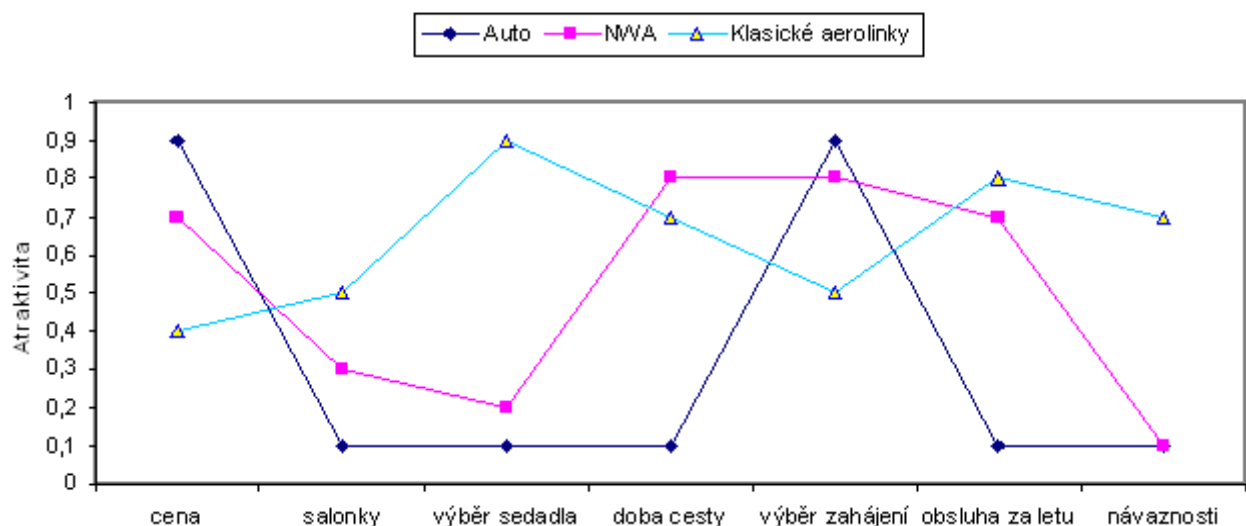
- Externí
  - O opportunities, příležitosti
  - T threats, hrozby
- Interní
  - S strengths, přednosti
  - W weakness, nedostatky, slabiny

		O	T
		Příležitosti pro naše přednosti	Hrozby, a překážky naší činnosti
S	V čem jsme dobří, jaké jsou naše přednosti	Jak uplatnit přednosti, které máme <b>Útok</b>	Jak odvrátit hrozby využitím našich silných stránek <b>Obrana</b>
W	Jaké jsou naše slabiny	Jak se zbavit našich nedostatků, abychom využili příležitost <b>Posílení</b>	Krizový plán, jak snížit škody, kterým se neubráníme <b>Krize</b>

## Strategické plátno

Atraktivitu jednotlivých obchodních atributů produktu (cena, občerstvení, rezervace, ....) vlastních produktů a konkurenčních produktů se hodnotí známkou od 0.0 (nejhorší) do 1 a vynesou do čárových grafů (lze použít excel). Uvedme příklad pro aerolinky podle W. Chan Kim a R. Mauborgne, letadlo v ceně auta. Má podobné hodnocení atributů, zčásti i ceny

**Letadlo v ceně auta, NWA**



## Strategická matice

- Matice
  - Řádky jsou podúkoly cíle/akce
  - Sloupce obsahují úkoly pro oddělení nebo pro nástroje, např. poradu

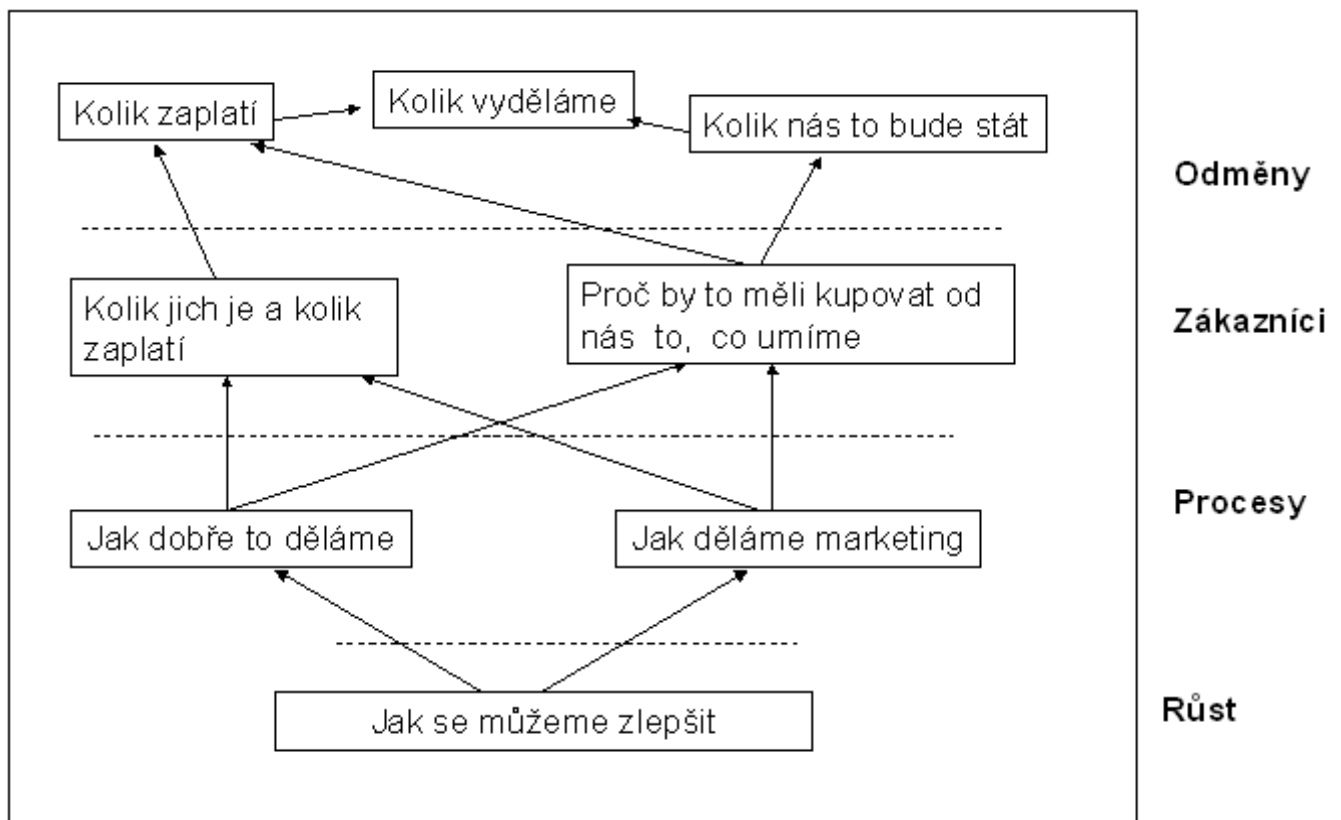


- Prvek i,j: Co se má udělat pro úkol i v oddělení (nástroji) j.
- Použití
  - Horizontálně: Jsou uvedeny všechny potřebné akce pro daný cíl/úkol
  - Vertikálně: Je oddělení vytíženo? Je schopno úkoly zajistit?
- Podobné použití: Delegování pracovníků na úkoly v týmu

	Kuchař, vedoucí	Číšník
Domácí a krajová jídla	Místní názvy jídel. Krajová jídla Vaření neobvyklých ale chutných jídel. Moravská vína	Vysvětlí názvy jídel a doplní historky o kraji.
Informace o kraji	Mapy. Pověsti. Historky o jídlech Výstava místních kuriosit	Doporučí vhodné cíle na další den, kulturní akce
Přísun turistů	Zvláštní menu pro průvodce	Umí cizí jazyky. Navazuje kontakty s cestovkami

## Strategická mapa

- Oblasti:
  - Růst
  - Procesy
  - Zákazníci
  - Odměny
- Výhody:
  - Orientace na klíčové prvky
  - Lze doplnit kvantitativní údaje (metriky)
- Sledovat cykly (doporučení firmy Anima Praha)
  - Snížit počet reklamací- Vyšetřit příčiny reklamací – Odstranit příčiny reklamací – Snížit počet reklamací....
  - Každá akce by měla být na nějakém cyklu (nebo alespoň cestě od zdola až nahoru)
    - Akce se lépe formulují
    - Odhalí se chybějící články
    - Cyklus podporuje stálé zlepšování
  - Funguje i když nejsou k dispozici metriky, účinnost cyklů se ale lépe prověří na metrikách
- Balanced score card: strategická mapa doplněná o metriky



## Budování strategické mapy

1. Nejprve se vyhledají problémy k řešení pomocí tipů od manažerů, a porad, především brainstormingů, případně interview
2. Vyberou se ty, o kterých se předpokládá, že jsou nejdůležitější (asi by jich nemělo být mnoho, tak pět až deset) a ty se umístí do strategické mapy
3. Najdou se (doplň) sousedi daných problémů v mapě
  1. Bezprostřední předchůdci opatření která jsou pro dané opatření nutná v dané a v nižší vrstvě a
  2. Bezprostřední následníci - opatření, která na dané opatření navazují v dané a vyšší vrstvě
4. Musí se najít následníci ve všech vyšších vrstvách a předchůdci ve všech nižších vrstvách (podmínka cyklů) tím, že se hledají sousedé sousedů

## Nejčastější účely porad, zpřesnění

- Vstupní porada
  - Seznámení lidí, seznámení s úkolem, formulace vize
  - Forma – klasická schůze
- Investigativní
  - Hledání nových řešení
  - Brainstorming, řízené přemýšlení, interview
- Integrační
  - Workshop, schůze, revize
- Kontrolní porada
  - Oponentura, revize (schůze)
  - Akceptační (schůze)
  - Inspekce (silně formalizovaná oponentura)

- Kontrolní den (i oponentury)
- Kontrolní porada (průběh testů, schůze či workshop)
- Závěrečná porada
- Audit
  - kontrolní den (ověřování výdajů, dodržování cílů, kontrola artefaktů),
  - schůze s prvky workshopu (hlášení týmů o postupu prací)

## Optimální řešení

- Při servisní orientaci mám poměrně snadnou možnost kombinovat customizované a nově vyvíjené komponenty, to je třeba využít s cílem dosáhnout optimálního výsledku
- Dodavatelé se ale s tím těžko smířují a bojují pomocí požadavků na přehnanou standardizaci a kejklů s ručením a dalšími triky

## Hlavní zadrhele na počátku projektu

- Podceňování počátečních etap a předčasná formalizace či programování.
  - V dokumentech nejde jen o „řeči“, hr na programování (programuje se i ve specifikačních jazycích)
  - Nereálné představy
  - Špatná spolupráce
  - Obtížné testování – používat oponentury, hlavně typu inspekce!
- Nedostatečné vedení projektu
  - Nemá vedení týmu, chyby v týmové práci, neobsazeny důležité role
  - Nemá formální (tj. podle přesně formulovaných pravidel) stanovování úkolů, jejich kontroly a přebírání
  - Mnohé důležité informace nejsou v písemné formě
- Nepostupuje se metodou maximální možné lenosti – předčasné řešení technických detailů
- IS jako náhrada invence při hledání strategie podniku a podnikatelského záměru,
  - Nejasné cíle
- Syndrom pejska a kočky
- Přesnost požadována větší, než umožňují data,
- Zbytečně ostré požadavky na interaktivnost, neostatečná kvalita dat
- Zamčené předpoklady a souvislosti
- Nedostatečná kvantifikace cílů
- Nereálné termíny, nezáměr managementu, chybějící zdroje
- Neochota budovat systém postupně (velký třesk).
- Standish – nezapojení uživatelů, nekvalitní management

## Jazyk dokumentů počátečních etap budování IS

- Při specifikaci požadavků je ovšem potřeba ověřovat u budoucích uživatelů, zda to, co navrhujeme, pokrývá potřeby.
- Zde je problém společného jazyka partnerů, protože způsob vyjadřování (jazyk) různých profesí (programátor a např. ekonom) bývá velmi odlišný.
- Navíc mívají různé profese různé požadavky na přesnost vyjadřování a různá kritéria pro to, co je pro řešení problému důležité (a co je třeba) a co nikoliv - mají různá základní paradigmatata

- Vzhledem k problému společného jazyka partnerů bývá nejdostupnější formou specifikace požadavků formulace požadavků na IS formou odborného článku (tj. formou používanou ve vědeckých a technických publikacích znalostního oboru uživatele s menšími doplňky, např. diagramy), tj. specifikace požadavků formou založenou na přirozeném jazyce.

## Texty jsou výhodné

- Nejlépe je, když jsou všechny dokumenty textové
- Snáze se skládají
- Snáze se opravují
- Syntaxí řízené transformace

## Specifikační jazyky

- Specifikační jazyk je v mnoha směrech jazykem programovacím (viz možnost generace prototypů). Požadavek používání specifikačního jazyka tedy může obsahovat požadavek zvládnutí základů programování (nebo přinejmenším požadavek přesného vyjadřování v oblasti, kde není zadavatel odborník). Také to může omezit intuici a vnucení myšlenkových modelů specifikačního jazyka.
- Spolupráci se zadavatelem může někdy usnadnit takový formální specifikační jazyk, který je "šit na míru" řešené problematice.
- Kombinovat s diagramy (dnes UML, to je také formalizace). Používají se případy použití, sekvenční diagramy a digramy aktivit.
- Formalizované specifikační (tj. téměř programovací) jazyky jsou obecně schopny vyjádřit požadavky, a snížit tedy nebezpečí nedorozumění.
- Tento efekt však přinesou jen tehdy, když jsou opravdu zvládnuty všemi zúčastněnými. V opačném případě mohou být výsledky horší než při specifikacích formou odborného článku.
- Formální metody mohou být používány jen velmi kvalifikovanými týmy a jsou vhodnější pro software, při jehož vývoji není nutná spolupráce s uživateli.

## Specifikační jazyky, výhody

- Shrňme-li, mají formalizované specifikační postupy následující výhody:
  1. Umožňují (v principu) přesnější vyjadřování a snáze se při nich dodržuje disciplína.
  2. Lze pro ně provést snáze důkaz správnosti.
  3. Někdy mohou být formálně (automaticky) transformovány na prototypové řešení (a tím usnadňují prověření správnosti specifikací).
  4. Lze snáze zkontrolovat, zda výsledné programy odpovídají specifikacím.

## Specifikační jazyky, nevýhody

- Specifikační jazyky mají i nevýhody:
  1. V případě IS je obvykle nutné, aby specifikačnímu jazyku rozuměli všichni zúčastnění. Pokud tomu tak není, nelze očekávat dobré výsledky. To je většinou případ formálních specifikací.
  2. Zápis požadavků ve specifikačním jazyku je složitější a pracnější než u specifikací formou odborného článku.
  3. U těch částí specifikace cílů, které nelze testovat na prototypch, je větší nebezpečí, že

specifikujeme (zdánlivě přesně) něco jiného než chceme (podobnost procesu specifikování s programováním znamená, že začínáme de facto programovat příliš brzy).

## **Proč nemůže specifikovat zákazník sám**

1. Uživatel nebývá schopen omezit požadavky na rozumné minimum. Nevylučují se požadavky méně potřebné až neužitečné (syndrom dortu pejska a kočičky). Může být nepříznivě ovlivněno těmi „co jsou při tom“
  2. Uživatelé jsou si zřídka plně vědomi vlastností, možností a omezení moderních informačních technologií (zvláště u customizovaných IS).
  3. Uživatele nemají dostatek zkušeností pro vypracování uceleného systému požadavků (jako slohové práce). Zřídka jim chybí komplexní znalosti o fungování vlastního podniku na mikroúrovni a top úrovni. Nemají cit pro to, co lze použít, co lze snadno realizovat a kdy je realizace ohrožena.
  4. Je větší nebezpečí, že se do požadavků prosadí zájmy těch "co jsou u toho" a budou opomenuti ostatní budoucí uživatelé systému.
- Dokument "Specifikace požadavků" by měl proto být vypracován ve spolupráci pracovníků dodavatele s pracovníky uživatele a oponován managementem uživatele ale především budoucími koncovými uživateli systému.

## **Základní vlastnosti specifikací požadavků**

- úplné
- ověřitelné (testované)
- bezrozporné
- konzistentní
- modifikovatelné
- vystopovatelné
- použitelná i během provozu systému
- stabilní