

# Architektura softwarových systémů (2)

= skládání (libovolně složitých) struktur

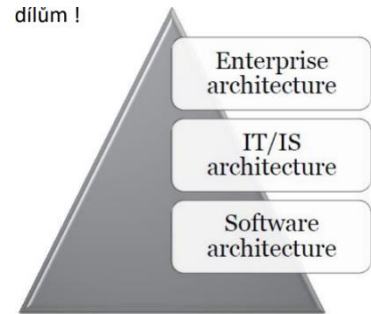
- u SW se dělí na: software, hardware, databáze, informační systém
- slouží pro pochopení, evoluci myšlenek, dokumentaci, validaci,...
- neopakování chyb, využívání nalezených řešení, popsání různých úhlů pohledu

## Softwarová architektura (SA) (lvl 1)

= struktura skládající se z komponent a vztahů mezi nimi

- (= **komponenty, konektory, konfigurace**)
- není zcela libovolná, je omezená pravidly, účelem a prostředím
- určena různým aktorům – analytik, management, tester,....)
- **aktor** = osoba zainteresovaná do procesu tvorby SW
- **příklady:**
  - o SoC (Separation of Concerns)
  - o objektové paradigma
  - o klient-server
  - o roura (pipes and filters)
  - o vzory
- **Jak popsat architekturu?**
  - o pomocí jazyků, pohledů a frameworků
  - o jazyky: ADL
  - o pohledy: funkční, datový, fyzický, uživatelský,...
  - o frameworky: TOGAF

dílům !



## Informační systém (lvl 2)

= systém složený z osob, datových záznamů a aktivit, které zpracovávají data a informace v organizaci (jak manuální, tak automatizované procesy)

- model zahrnující všechny aspekty IS dané organizace (SW, HW, procesy,...)
- samotné UML není dostačující pro tuto abstrakci
- nejsou to jen PC, ale celá org. získávání, zpracování a distribuce údajů k uživatelům
- vyšší abstrakce – IA (architektura IS = SA + hardware, procesy,...)

## Enterprise architektura (EA) (lvl 3)

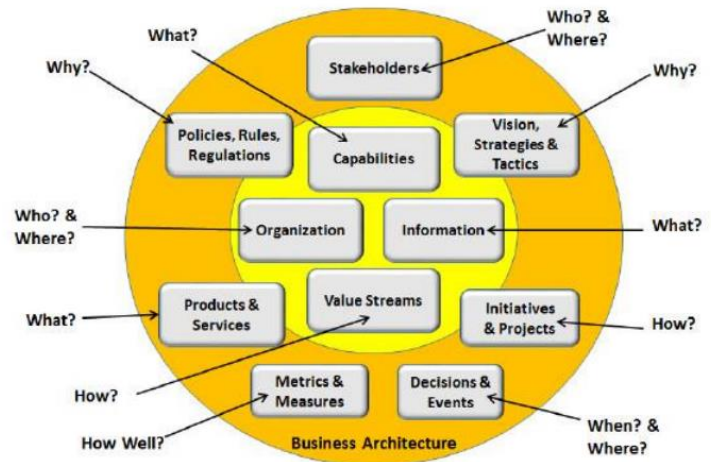
= organizační logika podnikových procesů a IT infrastruktury, odráží integraci a standardizaci požadavků operačního modelu

- EA je architektura systému na nejvyšší úrovni
- její součástí (nervovou soustavou) je informační architektura
- architektura SW komponent, ze kterých se IS podniku skládá = SA

## Business architektura (BA)

= model popisující tok informací a práce mezi autoritami

- založená na business strategii dané firmy
- využívá různé principy a frameworky
  - o UML, MDA, BPMN,...
- **musí nám odpovědět na základní otázky:**
  - o **Co** děláme?
  - o **Kdo** to dělá?
  - o **Kterou** informaci?
  - o **Kde** je to uděláno?



## Další dělení architektur

### Statická

= nemění se za běhu

- zachycuje pevnou danou strukturu SW
- je potřeba znát kontext při analýze
- nelze spoléhat na učení za běhu (machine learning apod.)

### Dynamická

= mění se (vznik a zánik)

- podporuje vznik a zánik komponent a vazeb za běhu systému (dle pravidel určených při návrhu)
- struktura systému se dynamicky mění (např. distribuovaná aplikace, která si alokuje dostupné zdroje během výpočtu)
- tj. architektura se sice mění, ale je dopředu daná

### Mobilní

= mění se (vznik, zánik + pohyb)

- rozšiřuje dynamickou arch. o mobilní prvky, kdy se komponenty a vazby přesouvají za běhu systému dle stavu výpočtu (např. aplikace s multiagentní architekturou)
- architektura se mění a není dopředu předvídatelná

## ADL (Architecture Description Language)

= jazyk pro popis softwarové, technické nebo systémové architektury

- **systémová architektura** = konceptuální (pojmový) popis struktury a chování systému
- **technická architektura** = struktura HW komponent systému a SW komponent na nich
- **SW architektura** = struktura systému složeného ze SW komponent, jejich vlastností a vztahů

- má větší potenciál než samotné UML (ale podobné vlastnosti)
- je to prostředek pro vyjádření SW architektury, srozumitelný pro všechny aktory
- musí umět vyjadřovat **komponenty**, **konektory** a **konfigurace** systému
- podpora grafické formy a generování kódu
- **základní elementy architektury:**
  - **komponenty** = části dekomponovaného systému s daným rozhraním
  - **konektory** = komunikační kanály pro propojení komponent s daným rozhraním
  - **konfigurace** = konkr. způsob vzájemného propojení komponent pomocí konektorů
- u dynamických a mobilních musí také umět vyjadřovat:
  - **události**, na které systém reaguje (vnější podněty)
  - **akce**, které systém vykonává (reakce na podněty)
  - **vztahy** mezi akcemi (jaké události → jaké akce)
- chování systému (jak se mění SA v důsledku akcí) je dáno vztahy mezi událostmi a akcemi
- **dělení ADL jazyků:**
  - **formální**
    - využívají matematické/algebraické nástroje, grafy apod.
    - jsou přesné
    - horší na používání a pochopení
    - např. Wright
  - **neformální/obecné**
    - čitelnější
    - některé omezenější, např. jen embedded systémy
    - např. ACME, AADL, Archimate