

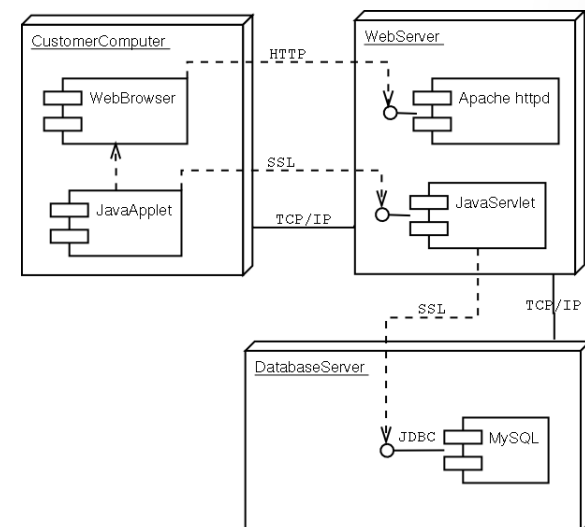
# Architektura softwarových systémů

## Definice, Strukturní a Procesní doporučení

Ing. Tomáš Černý, MSCS

# Pojem softwarové architektury (SA)

- Obvyklé způsoby vysvětlování pojmu SA
  - **komponenty a vazby**
    - celková struktura systému
    - diagram struktury systému



# Co dělá dobrou architekturu..

- Doporučení
  - procesní
  - strukturní

# *Procesní doporučení*

- **Architektura je produktem**
  - jednoho architekta
  - malé skupiny architektů s definovaným vedoucím
- **Architekt**
  - k dispozici technické požadavky na systém
  - sepsaný prioritní list kvalitativních vlastností
- **Architektura by měla být**
  - dobře dokumentovaná
  - používat domluvenou notaci (porozumění)

# Procesní doporučení

- Architektura by se měla
  - opírat o **lidské účastníky**,
    - ti se aktivně podílí na revizi
- Architektura by měla být
  - analyzována pro **kvantitativní** měření
    - **(propustnost)**
  - formálně revidována pro **kvalitativní** vlastnosti
    - **(modifikovatelnost)**
- Architektura by měla vézt k
  - implementaci přes vytvoření **kostry** systému
    - naplňuje jen minimum požadavků.
    - kostra je inkrementálně rozšiřována
      - usnadňující integraci a testování

# Procesní doporučení

- **Architektura by měla**
  - vyústit ve specifickou množinu možných částí
  - řešení jsou pak dobře specifikována, cirkulována a udržována
- **Př. Network utilization area**
  - Architekt produkuje doporučení pro každý team, tak aby se dosáhlo minimální zahlcení sítě
    - „network traffic“
- **Př. Výkonnostní zájem**
  - Architekt vyprodukuje časové úvahy pro hlavní vlákna

# Strukturní doporučení

- Architekt by měl navrhnout funkční moduly:
  - využívající zapouzdření informací (information hiding)
  - oddělení zájmů (separation of concerns)
  - každý modul by měl:
    - mít rozhraní vynášející jen důležité metody
    - schovávat měnitelné aspekty (implementační strategie) od ostatního SW využívajícího služeb daného modulu.
- Př. GRASP
- Př. Abstrakce a implementace
- Př. Dependency Injection (DI)

# Strukturní doporučení

Moduly by měly reflektovat oddělení zájmů

- (separation of concerns)
  - tak že je umožněno jejich rozdělení do týmů
  - týmy pak pracují nezávisle na dalších

Moduly schovávající informace by měly

- zahrnovat ty co zapouzdřují výpočetní infrastrukturu
- tedy izolují podstatnou část SW
  - od změn, které vyplynou ze změn platformy.



# Strukturní doporučení

- Architektura by neměla záviset na konkrétní verzi komerčního produktu či nástroje.
  - Hibernate 2.6 a search pro many to many fungoval ale nebyl popsán v dokumentaci
  - Hibernate 3.0 již neimplementoval tuto funkčnost
    - Př. SW postaven na 2.6 využívající search
      - Změna kódu pro tuto funkčnost = implementace nového systému
- Pokud arch. závisí na komerčním produktu, pak
  - by měl být systém strukturován tak, že změna na jiný produkt je snadná a nenákladná..
    - (diskuze ???)
      - Bridge GOF pattern

# *Strukturní doporučení*

- Moduly **produkující** data by měly být **odděleny** od modulů **konzumujících** data.
  - toto má za následek **snadnou modifikovatelnost**
    - změny jsou často buď pro produkci dat
    - nebo pro konzumaci dat
  - pokud jsou přidána nová data, potom bude změna na obou stranách
  - Separace umožní inkrementální upgrade
  - Export XLS vs. Import XLS

# *Strukturní doporučení*

- Pro paralelní systémy
  - architektura rozvrhne procesy a úkoly, tak že
    - nemusí nutně odrážet strukturu modulů.
      - proces se může rozprostírat přes více modulů.
      - modul může poskytovat služby více procesům.
- Úkol by měl být napsán tak,
  - že přiřazení specifickému procesoru může být snadno změněno.
    - Možná i v run-time.

# Strukturní doporučení

- **Architektura by měla**
  - používat malé množství vzorů interakce
    - design patterns
- **System by měl dělat stejné věci stejným způsobem.**
  - (Jak navenek tak v implementaci).
  - Lepší srozumitelnost
  - Nižší vývojový čas
  - Lepší čitelnost
  - Lepší modifikovatelnost
  - Také umožní konceptuální integritu v architektuře
    - tj. Snadnější vývoj
  - **Zamezení duplikace kódu!!!!**

# Definice softwarové architektury

“Architektura softwarového systému představuje definici struktury systému.

Tuto strukturu tvoří softwarové **komponenty**, ze kterých se systém skládá, z vnějšku viditelné **vlastnosti** těchto komponent a **vztahy** mezi nimi.”

*Bass, Clements, Kazman.*

*Software Architecture in Practice, Addison-Wesley 1997*

# Definice softwarové architektury

“Architektura je organizační **struktura** systému. Architektura může být rekurzivní dekomponována na **části**, které **komunikují** prostřednictvím **rozhraní**, na **vztahy** propojující jednotlivé části a **omezení**, která na jsou na jednotlivé části kladena. Komunikující složky zahrnují třídy, komponenty a subsystémy.”

# Co plyne z definice

„Mít architekturu“ se liší od  
„Mít známou architekturu“

- architektura
- vs. specifikace architektury