

Architektura softwarových systémů, význam a součásti architektury, formy popisu architektury, architektonické styly.

Wednesday, May 29, 2013 4:56 PM

Klíčová rozhodnutí

Proč - vize

Co - požadavky

Architektura

- Technologie
- Struktura
- Pravidla, konvence

Jak - návrh

Význam

- Klíčové aspekty organizace
 - Jak je systém členěn, jak a proč dělá to co dělá
 - Technologie
 - Hrubé členění nasubsystemy, závislosti a rozhraní mezi nimi to spíš
- Aspekty prolínající celou implementací (konvence)
 - Jak systém navrhovat
 - Vzory návrhu, implementace
 - Mimofunkční aspekty...
 - RUP koncept "4+1" pohled
-
- **Co architektura dělá**
 - Architektura definuje konceptuální integritu systému.
 - Systém má vždy právě jednu architekturu (může integrovat více stylů)
 - Definice architektury je první krok návrhu
 - Umožňuje myšlenkové pochopení návrhu velmi složitých systémů
 - Stanovuje základní kameny návrhu a základní směry vývoje a údržby
- Kontext, principy
- Kontext(daný)
 - okolí systému => vazby, důvody pro některá rozhodnutí
 - Stakeholders => úhly pohledu => aspekty architektury
 - Navíc oproti vizi a požadavkům
 - Aspekty: logická struktura, procesní pohled, varianty nasazení, datová integrace, bezpečnost, provoz a podpora, provozní infrastruktura, rozhraní
- Principy architektury
 - Efektivita - runtime, vývoj
 - Jednotnost
 - Srozumitelnost
- Výběr technologie, omezení
 - Programovací jazyk, databáze, knihovny
 - Použití frameworků a již hotových komponent
 - Make vs buy decision
 - Faktory ovlivňující/omezující výběr technologie
 - Smluvní podmínky
 - Standardy
 - Kontext
 - Marketing
 - Technické znalosti
- Validace architektury
 - Nutno ověřit užitečnost návrhu
 - Mechanismy
 - Návrh na základě klíčových use cases a mimofunkčních požadavků
 - Referenční architektura
 - Proof of concept implementace
 - Oponentura
- Dokumentace architektury
 - Referenční architektura
 - Dokument
 - Kostra aplikace
 - Dokument
 - RUP artifact: software architecture document
 - Modely
 - UML: implementation view (koomponenty), logical view (třídy, balíky), process view (interakce, stavový model)
 - Adhoc diagramy - visio, tabule
- Funkční členění

Architektura softwarových systémů, význam a součásti architektury, formy popisu architektury, architektonické styly.

Proč – vize

Co – požadavky

Architektura

- Použité technologie
- Struktura
- Pravidla, konvence

Jak – návrh

Architektura definuje funkční členění systému, závislosti jednotlivých částí systému a použité technologie. Architektura systému je vždy jen jedna. Umožňuje myšlenkové pochopení návrhu velmi složitých systémů.

Kontext

- Okolí systému – vazby, důvody pro některá rozhodnutí

Výběr technologie

- Zkušenosti, existující komponenty, make vs buy decision, ...
- Validace architektury
- Nutnost ověřit užitečnost návrhu – proof of concept, oponentura, ...
- Dokumentace architektury
- Referenční architektura (MVC, MVVM)
- Dokument
- Model
 - Funkční členění architektury
- Subsystem, modul
- Modul je základní stavební jednotkou subsystemů. Subsystemy jsou funkčně soudržné celky často vázané na jednoho aktéra.

Formy popisu architektury

- UML (class diagram, component diagram,...), layer diagram

Architektonické styly

- Monolitická
- Dvouvrstvá (klient server)
- Třívrstvá – aplikační, prezentační a datová vrstva
- MVC – není třívrstvé, ale trojúhelníkové. Třívrstvá by byla MVVM.
- Vrstvená – více vrstev nad sebou
- SOA – service oriented architecture

- Subsystem, modul = skupina souvisejících prvků implementace tvořící funkční celek
 - Vícenásobně použitelné, volně vázané části
 - Celky vhodné pro vývoj a údržbu
- Subsystemy
 - Funkčně soudržné (lokalita změn)
 - Často vázané na jednoho aktéra
- Moduly
 - Základní stavební jednotka subsystémů
 - Snaha o vícenásobnou použitelnost
- Logická struktura subsystémů = moduly a jejich vnitřek, balíky tříd, třídy (low level architektura), jejich vztahy; Komunikace mezi moduly = ROZHRANÍ!!!!!!!!!!!!!!

Součásti

- konvence a politiky (pravidla pro návrh, dodržují všichni vývojáři)
- **Funkční, procesní, datová, aplikační**
- členění, doménová analýza:
- **logické členění** (např. do balíků)
 - balík – skupina souvisejících tříd, tvořící organizační celek, mapování do jazyka (balík vytváří jmenový prostor), hierarchické vnořování
 - třídy v balíku funkčně příbuzné
 - vhodné protože bude přehled o systému a snadné rozdělení implementace mezi členy týmu
 - analytický model tříd je příliš rozsáhlý -> lepší jej členit
- **funkční členění do subsystémů**
 - subsystém = skupina souvisejících balíků a/nebo tříd tvořící funkční celek
 - vhodné, protože monolitická aplikace není praktická
 - jak najít subsystémy?
 - buď je to dopředu zřejmé (jednoduché, architektonické styly)
 - na základě objektového modelu (nutno vidět všechny třídy a vazby, pak shluk těsně vázaných tříd je kandidátem)
 - na základě případů užití

Formy popisu architektury

Modely - UML, Layer diagramy, ...

UML, Unified Modeling Language je v [softwarovém inženýrství](#) grafický jazyk pro [vizualizaci](#), [specifikaci](#), navrhování a dokumentaci programových systémů. UML nabízí standardní způsob zápisu jak návrhů systému včetně konceptuálních prvků jako jsou [business procesy](#) a systémové funkce, tak konkrétních prvků jako jsou příkazy [programovacího jazyka](#), [databázová schémata](#) a znovupoužitelné programové komponenty.

UML podporuje objektově orientovaný přístup k analýze, návrhu a popisu programových systémů. UML neobsahuje způsob, jak se má používat, ani neobsahuje metodiku(y), jak analyzovat, specifikovat či navrhovat programové systémy. Standard UML definuje standardizační skupina [Object Management Group](#) (OMG).

From <http://cs.wikipedia.org/wiki/Unified_Modeling_Language>

Architektonické styly

- **Vrstvení** - funkce jsou uspořádány do několika vrstev tak, že funkce vyšší vrstvy mohou využívat pouze funkcí podřízených vrstev.
 - **Monolitická**
 - **Dvouvrstvá (lehký/těžký klient)**
 - **Třívrstvá** je nejběžnějším případem vícevrstvé architektury.
 - **Prezentační vrstva**
 - **Aplikační vrstva (též Business Logic)**
 - **Datová vrstva**
 - **Porovnání třívrstvé s architekturou MVC**
Model-view-controller má trojúhelníkovou topologii (ne třívrstvou) – pohled je obnovován (aktualizován) přímo modelem, na příkaz řadiče.
- Architektura distribuovaných systémů
 - Klient-server
 - Peer to peer
- Filosofický přístup k architektuře/ jednotlivé architektura: **Service-oriented architecture (SOA)**

Dle přednášek:

Základní architektonické styly

- **Klient server**
 - Tlustý klient
- **3-vrstvé a vícevrstvé**
 - Oddělení prezentace, business logiky a datové části
 - Dnes standard
- **Vrstvená**
 - Delegování na podřízené
 - Prezentace / řízení / doména / business služby / technická infrastruktura / knihovny třídy / systémové...
- **Další architektonické styly**
 - SOA - service oriented architecture viz SI
 - Pipes and filters
 - Blackboard
 - Broker

From <<https://d.docs.live.net/e3534876709763a3/Dokumenty/ZCU/Statnice/Statnice.docx>>