

Oborová 01

Uživatelské požadavky

- specifikace toho, co by mělo být implementováno
- co by měl systém dělat, nikoliv jak by to měl dělat
- dělení:
 - funkční (jaké chování bude systém nabízet)
 - nefunkční (vlastnosti nebo podmínky kladené na systém)
- chyby při sběru požadavků:
 - nepřesné nebo neúplné požadavky
 - navzájem si odporující
 - nevyčtené, přesto očekávané požadavky
 - přehnané požadavky
 - vytváření něčeho, co nikdo nechtěl

Definice funkcionality systému

Vztah mezi funkcionálními a nefunkcionálními požadavky na systémy

- funkční
 - definuje konkrétní funkci softwarového systému
 - funkce popsána jako množina vstupů, chování a výstupů
 - výpočty, manipulace s daty
 - plán implementace funkčních požadavků je upřesněn ve fázi návrhu
 - formulace: <id> <systém> musí/může/nesmí <popis funkce>
- nefunkční
 - podporují množinu funkčních požadavků
 - formulace: <id> <systém> musí/může/nesmí <popis vlastnosti/omezení>
 - některé typy:
 - dostupnost
 - robustnost
 - škálovatelnost
 - bezpečnost

Technická specifikace

- dokument popisující kompletní funkčnost vyvíjeného systému
- obsahuje všechny funkční i nefunkční požadavky

- funkční požadavky zapsány ve formě use case
- slouží pro rozhodnutí, co patří do projektu a co ne
- strukturovaný dokument:
 - úvod
 - popis
 - požadavky

Oborová 02

UML

- grafický modelovací jazyk
- vytváření grafických modelů
- není modelovací metodika
- dává grafickou syntaxi
- otevřený a standardizovaný jazyk

Oborová 03

Objektově orientovaný návrh software

- fáze návrhu
- postup
- koncepty OO

Metodiky objektového návrhu

- = metodika předepisuje CO, KDO, KDY, JAK má dělat
- architektonické vzory
- deméterovo pravidlo

Postupy rozvinutí technické specifikace na úrovni modulů do detailního objektového návrhu

- GRASP
- SOLID
- Top-Down
- Bottom-Up

Návrhové vzory

- creationals
- structural
- behavioral

Oborová 04

Interní a externí systémová rozhraní

- = umožnění komunikace s okolím
- konzolová aplikace
- desktopová aplikace
- webová aplikace
- webový server

Zásady návrhu rozhraní

- univerzalita
 - nezáleží na vnitřku
 - univerzální způsob, jak se systémem komunikovat
- zapouzdření
 - komunikace probíhá pouze přes rozhraní
- formální popis
 - např. WSDL

Komponentový přístup, jeho návrh a formalismy

- výhody
 - rozdělení práce mezi nezávislé týmy
 - možnost změnit modul bez zásahu do jiných
 - rozdělení do samostatných celků je přehlednější
 - znovupoužitelnost kódu
- komponentní programování
 - rozdíl oproti OOP:
 - v OOP se objekty snaží striktně odrážet realitu
 - nevyskytují se v něm pomocné objekty
- vlastnosti komponent:
 - nahraditelnost
 - opětované použití
 - soudružnost
 - low coupling

Integrace subsystémů

- řeší propojení více různých komponent či subsystémů
- metody:
 - vertikální
 - hvězdicová

- horizontální
- integrační vzory:
 - mediation
 - federation

Oborová 05

Synchronní a asynchronní distribuované systémy

- distribuovaný systém je takový, který pro své fungování využívá více nezávislých výpočetních jednotek
- místo jednoho superpočítače se využije více běžných počítačů
- jednotlivé uzly spolu komunikují (např. pomocí RMI)
- úloha se rozdělí na dílčí části, které se distribuují na další uzly
- mezivýsledky se složí dohromady

Specifika návrhu a vlastností distribuovaných systémů

Integrace externích datových zdrojů

Oborová 06

Techniky správy a organizace rozsáhlých softwarových projektů

Nástroje pro správu verzí a vývojových větví zdrojového kódu

- centrální repozitář
- pracovní kopie
- tag
- branch
- Git
- SVN

Nástroje pro automatické generování dokumentace a podporu orientace v rozsáhlých projektech

- JavaDoc
- Doxygen
- Enterprise Architect

Infrastruktury zajišťující spolupráci mezi vývojáři navzájem a i s uživateli

- repozitář
- wiki
- správa úkolů
- sledování chyb

Systémy pro sledování a řešení chyb a uživatelskou podporu

- Bugzilla

Oborová 07

Požadavky a pravidla pro tvorbu přenositelného kódu

- psát čistě a jen to, co je požadováno
- používat standardizované API (Posix, ...)
- nepředpokládat pořadí bytů ve slově (little / big endian)
- nepředpokládat počet bytů v adresační jednotce
- psát kód proti knihovnám, nikoliv volat přímo služby OS

Organizace projektů a operačních systémů pro splnění přenositelnosti mezi různými architekturami CPU

- kompilace balíků
- portace kódu a křížový překlad

Konverze vnitřních a vnějších/sít'ových formátů dat

Oborová 08

Ekonomické aspekty informačních systémů

- produkční funkce $Q = f(K, L)$
 - f ... spojitá funkce
 - K ... kapitál
 - L ... náklady
- izokvanta = křivka, která je tvořena všemi kombinacemi vstupů, které umožňují vyrobit stejné množství produkce Q

Celkové náklady vlastnictví systémů

- zahrnuje veškeré náklady na provoz za určité období
- počítá se za určitou periodu, aby bylo možné porovnávat různé varianty IT řešení
- zohledňuje časovou hodnotu peněz v delším časovém horizontu (dopředu počítá s inflací)
- $TC = w \cdot L + r \cdot K$
 - TC ... celkové náklady
 - w ... cena práce
 - r ... cena kapitálu
 - izokosta = kombinace výrobních faktorů, které jsou z pohledu firmy stejně nákladné
- optimum firmy = bod, kde se dotýká izokvanta a izokosta
- TCO zahrnuje:
 - celková cena
 - provozní náklady
 - dlouhodobé náklady

Návratnost investic

-

Náklady spojené s výpadky, kapacitními problémy a chybami

- BCP (Business Continuity Planning)



Oborová 09

Průmyslové aspekty IS

- funkčnost
- bezporuchovost
- použitelnost
- účinnost
- udržitelnost
- přenositelnost

Nefunkcionální požadavky na systém podle ISO 9126

•

Kapacitní plánování systému

- požadavky na systém
- porozumění prostředí
- nastavení systému (implementace monitorovacích nástrojů)
- monitoring (sběr dat)
- identifikace metrik
- tvorba predikčních modelů
- analýza, simulace, měření <-> tuning, sizing
- pokud cena a výkon ok, tak na monitoring, jinak:
 - změna parametrů a konfigurací -> analýza

Zajištění odezvy a škálovatelnosti systému

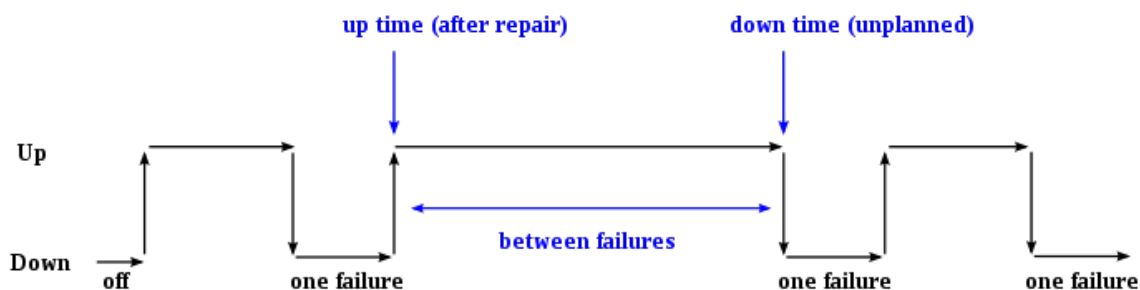
Oborová 10

Stabilita a spolehlivost systému

- spolehlivost = pravděpodobnost, že systém bude za daných podmínek vykonávat svoji funkci po určenou dobu
 - $R(T) = e^{-\frac{T}{MTBF}}$
- bezpečnost = schopnost pracovat správně nebo se ukončit tak, aby nebyla narušena činnost jiného systému
- spolehlivostní modely:
 - blokové
 - sériové
 - paralelní
- názvosloví:
 - pochybení
 - vada
 - selhání
 - chyba

MTBF

- $\Sigma \frac{(\text{downtime} - \text{uptime})}{\text{počet chyb}}$
- střední doba mezi poruchami
- slouží k ohodnocení spolehlivosti systému



$$\text{Time Between Failures} = \{ \text{down time} - \text{up time} \}$$

- MTTF (mean time to failure) - pro zařízení, která se neopravují
- MDT (mean down time) - střední doba výpadku
- MTTR (mean time to repair) - střední doba opravy

Dostupnost systému

- pravděpodobnost, že se systém nachází v spolehlivém stavu
- výpočet: $MTBF / (MTBF + MTTR)$

Redundance

- výsledná spolehlivost je určena:
 - hardwarovou spolehlivostí
 - softwarovou spolehlivostí
 - informační spolehlivostí (přenos a uchovávání informací)
 - spolehlivostí lidského činitele
- hardwarová redundance
- softwarová redundance
- DMR (dual modular redundand)
- TMR (triple modular redundand)

Návrh vysoce robustních systémů

- redundance
- zálohování
 - substituční - existuje záložní prvek
 - sdílení zátěže - prvky pracují napůl a v případě poruchy jeden přejímá všechnu zátěž
 - zálohování s obecnější rekonfigurací struktury
- škálovatelnost
- zajištění odezvy systému (RT systémy)

Oborová 11

Data mining

- získávání informací z dat a souvislostí mezi nimi
- např.:
 - analýza nákupního košíku
 - automatické párování uživatelských dotazů a odpovědí na ně

Řízení rizika

- nedílnou součástí správného řízení projektu
- postupný, neustále se opakující proces zlepšování
- začlenění do existujících praktik

- vybudování vhodné infrastruktury ke:
 - zjištění souvislostí
 - identifikaci
 - analýze
 - vyhodnocení
 - zvládání
 - sledování a hlášení rizik
- zahrnuje:
 - komunikace rizik
 - definice prostředí
 - identifikace rizik
 - analýza rizik
 - vyhodnocení rizik
 - zvládání rizik
 - akceptace rizik
 - monitorování rizik

Enterprise Resource Planning systémy

- software zvládající alespoň 2 ucelené podnikové agendy
- komplexní informační systémy zajišťující:
 - výroba
 - finance
 - účetnictví
 - dodavatelé
 - zákazníci
 - lidské zdroje
- funkce
 - evidence dat
 - reporting
 - organizování procesů
 - kontrola procesů
- komponenty
 - obecné, které lze nasadit v každé firmě (účetnictví)
 - speciální, které využije konkrétní firma
 - odvětvové, které slouží pro dané odvětví

Customer Relationship Management systémy

- systém podporující péči o zákazníka
- shromažďování, zpracování a využití informací o zákazníkovi
- umožňuje poznat, pochopit a předvídat potřeby zákazníka
- cíle:
 - tvorba marketingové strategie jak
 - udržet zákazníka
 - oslovit nového zákazníka
 - získat zákazníka
 - porozumět zákazníkovi
 - identifikace klíčových procesů

- zvyšování spokojenosti zákazníka
- typy:
 - operativní - podpora business procesů
 - marketingové kampaně
 - prodejní procesy
 - analytické - analýza zákaznických dat
 - kolaborativní - komunikace společnosti se zákazníkem

Oborová 16

Kategorizace SW chyb

- softwarová chyba = prezentace toho, že program nedělá něco, co koncový uživatel předpokládal
- typy:
 - pochybení - akce člověka, která generuje nesprávný výsledek
 - vada - nesprávný krok v počítačovém programu
 - selhání - nesprávný výsledek
 - chyba - kvantitativní vyjádření toho, na kolik je výsledek nesprávný
- kategorie:
 - chyby uživatelského rozhraní
 - chyby omezení
 - procesní chyby
 - chyby vedení
 - chyby požadavků, vlastností a funkčnosti
 - strukturální chyby
 - datové chyby
 - chyby implementace

Kritéria korektnosti a použitelnosti

- obecné ukazatele kvality SW:
 - bugs per line of code
 - code coverage
 - cohesion
 - coupling
 - number of classes
 - LOC
 - operational complexity
- korektnost a použitelnost
 - intenzita defektů
 - bezpečnost proti útokům
 - použitelnost
 - udržitelnost

Spolehlivost SW

- pravděpodobnost, že systém bude svou funkci v daných podmínkách vykonávat po specifikované době
- sleduje se počet defektů za čas nebo čas mezi dvěma defekty
- modely spolehlivosti
 - používají se k odhadu spolehlivosti
 - důvody použití:
 - vyjádření kvality produktu
 - plánování zdrojů pro údržbu
 - statický model
 -
 - dynamický model

Oborová 17

Testování metodami bílé a černé skříňky

- ortogonální pole
- latinské čtverce

Strukturální analýza

- hledání cest
 - cesta = posloupnost uzlů od začátku do konce programu
 - jednoduchá cesta = posloupnost uzlů od i do j bez opakování uzlů
 - hlavní cesta = jednoduchá cesta, která není podcestou
- řídicí toky
- datové toky

Statická analýza

- model-checking
 - UPPAAL
- testování konečného automatu
 - pojmy:
 - automat
 - přechodová funkce
 - množina stavů
 - počáteční, konečný stav
 - přechod
 - vstup
 - výstup

- přechodová tabulka
- kontrola modelu:
 - úplnost a konzistence
 - jednoznačné kódování vstupů
 - model by měl být silně souvislý
- odhalení chybějících vstupů
- pokrytí stavů
 - minimální množina vstup, při jejichž přijetí projde automat všemi stavy
- pokrytí přechodů
 - minimální množina vstup, při jejich přijetí projde automat všechny přechody

Dynamická analýza

- testování za běhu

Zátěžové testy

Oborová 18

Formální specifikace programu

- formální verifikace
 - technika založená na formálních metodách
 - využívá matematicky založené jazyky, které umožňují specifikaci a verifikaci systémů
 - specifikace = zapsání požadavků na systém v matematickém jazyce
 - verifikace = formální důkaz toho, že splňuje požadavky
- vstupy:
 - matematický model systému
 - specifikace požadavků
- techniky:
 - statická analýza
 - abstraktní statická analýza
 - ověřování modelů - úplné procházení dosažitelných stavů
 - omezené ověřování modelů
 - dokazování vět - nalezení důkazu, že systém a jeho vlastnosti jsou vyjádřeny jako formule v nějaké logice

Model-checking

Oborová 19

Architektura webové aplikace

Funkce jednotlivých vrstev

Životní cyklus standardizovaných komponent Java EE

Servlety

JSP

- textový dokument obsahující (X)HTML tagy a JSP tagy
- JSP tagy mohou generovat dynamický obsah
- umožňuje používat kusy Java kódu v HTML stránce (jako PHP)
- při prvním použití se zkompiluje a vytvoří servlet

Frameworky

Návrhové vzory

Oborová 20

Vhodnost nasazení jednotlivých webových architektur

- třídy aplikací
 - desktopová
 - webová na aplikačním serveru
 - cloudová

Sdílení dat

Persistence

Webové služby a REST

Asynchronnost

Messaging

Oborová 21

Cloud architektura

- Infrastructure as a Service
 - poskytování hardware
 - storage
 - servery
 - hardware
 - síťové prvky
 - poskytovatel je zodpovědný za běh a údržbu
 - např.: Amazon S3
- Platform as a Service
- Software as a Service

Virtualizace

- = postupy a techniky, díky kterým je možné přistupovat k dostupným zdrojům jiným způsobem, než kterým jsou fyzicky propojeny
- úrovně:
 - celý počítač (virtuální stroj)
 - jednotlivé hardwarové komponenty (procesory, paměť, ...)
 - softwarové prostředí (virtualizace operačního systému)
- způsoby:
 - emulace nebo simulace
 - virtuální stroj simuluje celý hardware
 - dovoluje běh neupraveného OS na zcela odlišném hardware
 - nativní (plná) virtualizace
 - virtuální stroj simuluje dostatečné množství komponent, aby umožnil běh neupraveného OS
 - např.: VMware, Microsoft Hyper-V, VirtualBox
 - částečná virtualizace
 - virtuální stroj simuluje instance mnoha prostředí, na kterém běží hostitel
 - paravirtualizace
 - virtuální stroj nabízí API (hypercall), přes které se komunikuje
 - např.: Xen
 - virtualizace na úrovni operačního systému
 - aplikační virtualizace

Pojetí cloudu různými dodavateli

Omezení cloudových aplikací

Náklady na provoz

Třída aplikací vhodných pro nasazení v jednotlivých cloudových technologiích