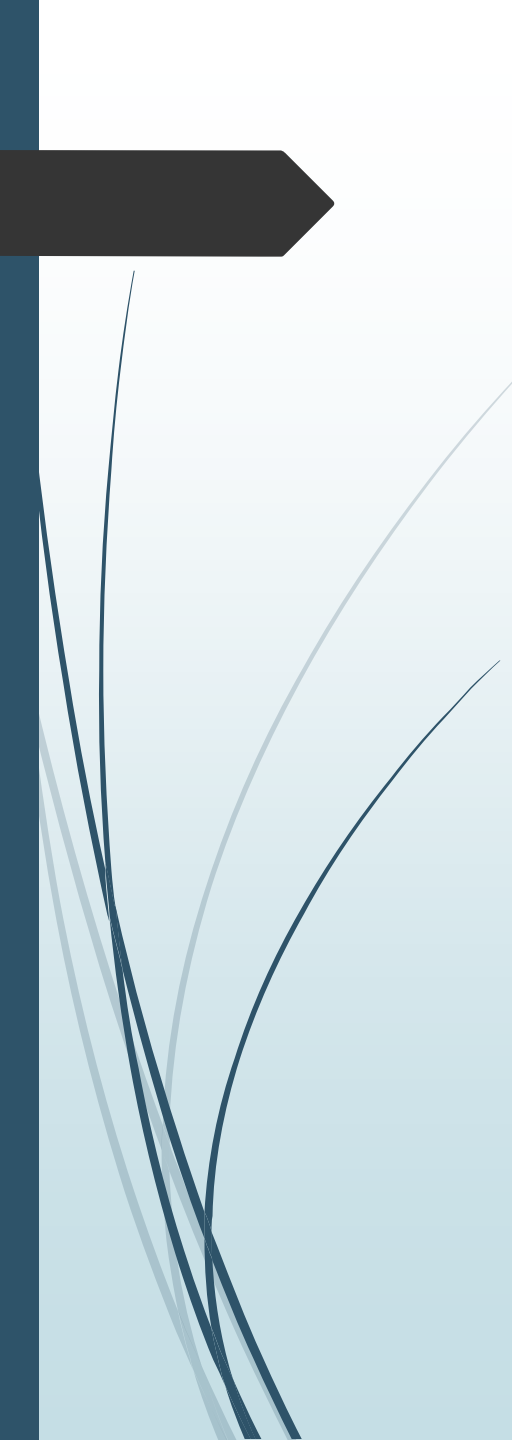




Programování

Od myšlenky k aplikaci



Co stojí za tvorbou aplikace z pohledu programátora?

- Specifikace zadání
- Zjištění řešitelnosti
- Identifikace vstupů a výstupů
- Rozdělení funkcionality na menší, lépe řešitelné části
- Návrh modelů – databáze, třídy, struktura komunikace
- Volba technologií použité pro tvorbu aplikace
- Návrh algoritmů pro funkční části
- Implementace
- Testování aplikace – ověření, zda splňuje zadané požadavky

Specifikace zadání

- Ujasnění požadavků na výsledek aplikace
 - Co „zákazník“ požaduje
 - Stanovení funkčních a nefunkčních požadavků
- Velmi kritická část při samotném procesu
 - Špatně pochopené zadání je postupem času velmi drahé
 - Uživatel/zákazník potřebuje jasně vymezit, co má aplikace splňovat
- V praxi se o shromažďování těchto specifikací a uživatelských scénářů stará tzv. **PRODUCT OWNER**



Zjištění řešitelnosti

- Částečně je samotná řešitelnost vyřešena již ve specifikaci
- Cílem této fáze je dopátrat se, zda existuje řešení na daný problém
- Aplikace, kterou tvoříme v zásadě řeší nějaký problém
- Řešitelnost potřebujeme posoudit nejen z pohledu algoritmu, ale i technického



Identifikace vstupů a výstupů

- Zásadní část nutná pro konstrukci algoritmu
- Tato část je rovněž naznačena při specifikaci problému
- Potřeba znát dostupné vstupy, se kterými lze pracovat
- Očekávaný výstup na základě vstupních dat
- V rámci identifikace vstupů řešíme i okrajové podmínky, které mohou nastat
 - Rozsah vstupních hodnot a určení jeho typu
 - Rozlišení platných a neplatných vstupů
- Jaké jsou požadavky na výstup – zejména jejich forma a jak je budeme zobrazovat



Návrhy modelů

- V praxi tuto činnost zajišťují SW architekti a analytici
- Během návrhu modelů řešíme:
 - Specifikací rolí – uživatel, administrátor, super admin, ...
 - Určení tříd v databázovém modelu
 - Určení tříd pro strukturu aplikace
 - Specifikace akcí navazujících na konkrétní události
 - Popsání toku dat v aplikaci – co s čím komunikuje



Návrh algoritmů

- Algoritmy v našich aplikacích jsou ve výsledku malé aplikace v jedné velké aplikaci
- Postup návrhu algoritmu je totožný s celým životním cyklem samotné aplikace
- Analýza řešitelnosti -> identifikace vstupů a výstupů -> návrh -> implementace -> ladění
- Výstup jednoho dílčího algoritmu může být vstupem pro následující algoritmus – utváříme posloupnost akcí
 - Příkladem je zpracování vstupních dat do formy, se kterou umíme pracovat



Implementace

- Jedna z nejnákladnějších fází při vývoji aplikace
- Odstranění chyby, které se v průběhu implementace objeví je časově náročná – často se musíme vracet zpět do prvotních fází vývoje aplikace
- Jakýkoliv zásah do navržené struktury může dominovým efektem poškodit další části naší aplikace
- K implementaci se uchylujeme v momentě, kdy jsme si odsouhlasili výsledky v předešlých fázích
- Samotná implementace se skládá z převedení myšlenek, které jsme vytvořili při modelování a návrhů algoritmů, do kódu v příslušném programovacím jazyce



Testování aplikace

- V mnoha případech přehlížená fáze při vývoji
- Testování aplikace neslouží pouze k ověření, že funkcionality stanovená při specifikaci je naplněna
- Na testování aplikací koukáme na více úrovních:
 - Unit testování
 - Integrovaní testování
 - Behaviorální testování
- V ideálním případě jsou lidé zodpovídající za testování přítomní v nějaké míře v každé fázi a dohlíží na kvalitu doručené práce - **QA**

Životní cyklus aplikace

