

Téma 25: Softwarové inženýrství

Standart IEEE 1993 – **softwarové inženýrství** je systematický disciplinovaný a kvalifikovaný přístup k vývoji, tvorbě a údržbě software (tzn. Jak slušně vyvinout, naprogramovat a udržovat software).

Softwarová krize měla dva hlavní problémy:

- Rozsáhlost a komplexnost software – počítače se rozmohly a měli k nim přístup i normální lidé. Software se začal zvětšovat a přestalo být možné, aby jeden člověk naprogramoval rozsáhlý software sám. Tzn., vzniká spolupráce týmu.
- Opakování chyb – problémy se pořád opakují. Mění se pouze doba jazyky a hardware. Pokud tedy někdo v historii vymyslel řešení daného problému a byl ochoten se o svoji myšlenku podělit, není důvod proč ji nevyužít. Vznikly tzv. Návrhové, Architektonické vzory.

Unifikovaný modelovací jazyk (UML) je množina diagramů, které souží pro vývojáře, návrháře nebo i údržbu, k domluvě na projektu. Nezávisle na programovacím jazyce. Existuje software, ve kterém, když namalujeme diagram v UML, dokáže vygenerovat kód daného jazyka (např. BlueJ který dokáže vygenerovat třídy, ale samotnou vnitřní implementaci (logiku) musíme dokončit sami).

Pohled 4 + 1 obsahuje:

- **Požadavky uživatele** – používá se diagram požadavků. Z pohledu běžného koncového uživatele, který počítá, že daný software bude něco umět (přihlásit, odhlásit apod.)
- **Struktura** – zachycuje z jakých částí, se daný systém bude skládat. Popis jednotlivých komponent
- **Chování** – pohled z dynamického pohledu, co všechno se musí udělat, než bude akce dokončena
- **Implementace** – pohled programátora. Popis v konkrétním programovacím jazyce. Komponenty, které se musí programovat.
- **Prostředí** – z pohledu správce systému, na jaké technologii software poběží



Pro každý z pohledů má UML definované svoje diagramy.

Softwarové vzory jsou popsány, ověřené a funkční řešení, nějakého problému.

- **Architektonické** – zobrazují z jakých podsystémů, se daný systém bude skládat, a jak budou mezi sebou komunikovat (ISO/OSI, TABULE – komponenty řeší společný problém na tabuli)
- **Návrhové** – popis podsystému
 - Vytvářecí – např., SINGLETON (jedináček) – jediná instance dané třídy. Přístup je globální
 - Strukturální – např., ADAPTER – přizpůsobení mezi třídami, aby spolu mohly komunikovat.
 - Chování – např., OBSERVER – pozorovatel, pouze bdí nad systémem
- **Idiomy** – platformě závislé, konkrétní řešení popsáno pomocí idiomů