

Základy analýzy

autor

Jan Novotný

<http://blog.novoj.net/>

15. února 2007

V prezentaci jsou použity diagramy z:
Wikipedia, Sparx UML Tutorial, Argo UML

Metodiky vývoje

- Různé metodiky vývoje – vazba na fáze projektu
 - Vodopád
 - Iterace
 - Agilní metodiky
 - SCRUM
 - Extreme Programming

Obvyklé fáze vývojového cyklu

- Prvotní zadání
 - SMLOUVA NA ANALÝZU
 - Analýza
 - Hrubý návrh
 - SMLOUVA NA REALIZACI
 - Detailní návrh
 - Realizace
 - Testování
 - PŘEDÁNÍ ZÁKAZNÍKOVĚ
 - Akceptační testování a testovací provoz
 - Reálný provoz
-
-

Zadání

- Tvoří: obchodník ve spolupráci s business analytikem, někdy dokonce zákazník
 - Forma: excel Sheet, Textový dokument s odrážkami, v lepších případech UseCase diagram
 - Obsah: hrubé vymezení rozsahu systému, posbírané podklady od zákazníka, hrubý popis zákaznickových procesů (activity diagram)
 - Cíl: hrubý popis toho co a proč
-
-

Analýza

- Tvoří: business analytik
 - Forma: obvykle UML v nějakém UML editoru, obvykle na konci exportovaného do textového dokumentu
 - Obsah:
 - Identifikovat problém zákazníka a jeho potřeby
 - Definovat rozsah řešení a hranice systému
 - Popsat procesy, které se systému týkají (workflow)
 - Identifikovat rozhraní systému
 - Cíl: detailní popis toho co a proč
-
-

Hrubý návrh

- Tvoří: návrhář, senior programátor, vedoucí programátor
 - Forma: obvykle UML v nějakém UML editoru, obvykle na konci exportovaného do textového dokumentu
 - Obsah:
 - Rozhodnutí o technické platformě
 - Rozložení systému na subčásti
 - Zabezpečení systému
 - Deployment
 - Hrubý datový model
 - Logický model (diagram tříd)
 - PageFlow (+ grafický design)
 - Definice rozhraní s externími systémy
 - Akceptační testy
 - Projektový plán
 - Prototyping
-
-

Detailní návrh

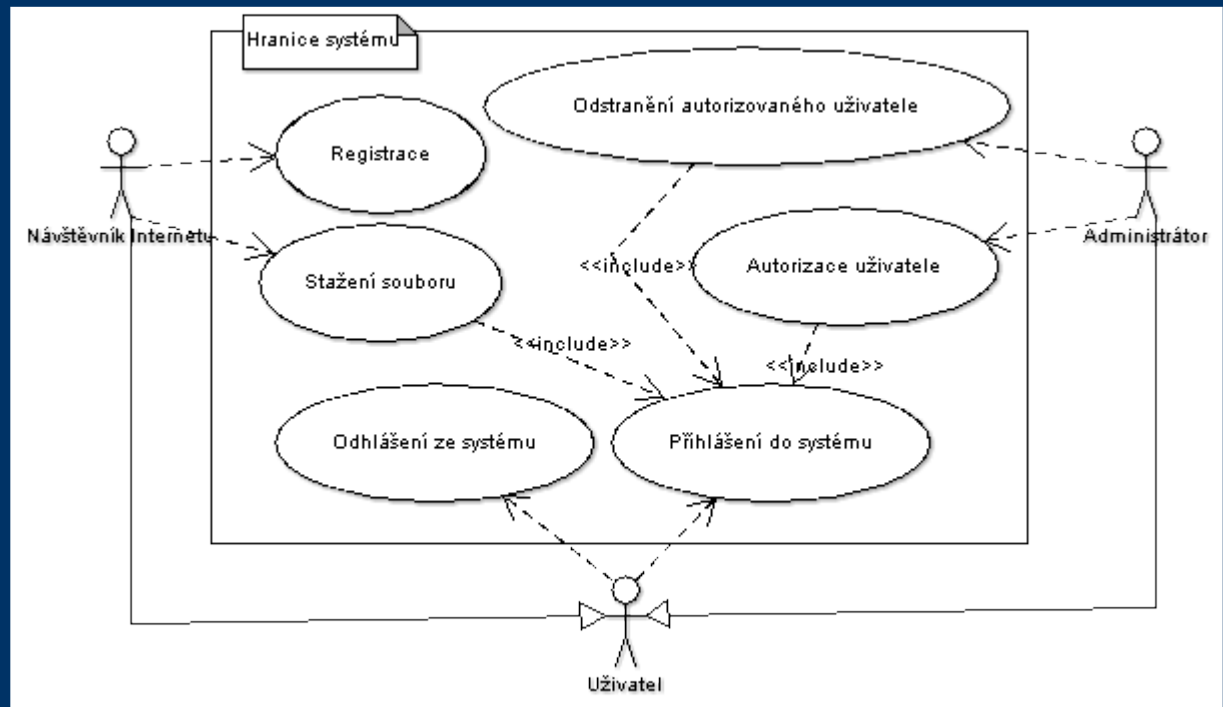
- Náročnost se odvíjí od velikosti projektu
 - Realizace detailních class diagramů (MDA?!) nebo již přímo interface kódu
 - Rozdělení práce mezi tým, zajištění flow
 - Příprava infrastruktury (source repository, task listy, implementace core tříd, base tříd testů, příprava buildování)
 - Hloubka se odvíjí od kvality týmu a přístupu organizace
 - U větších projektů se vyplatí spravovat projektovou site (Maven, Wiki)
-
-

Výrazové prostředky analytika a návrháře

- UML Diagramy
 - Popis chování (obvykle součástí analýzy)
 - UseCase diagram
 - Activity diagram
 - State Machine diagram
 - Popis struktury (obvykle součástí návrhu)
 - Class diagram
 - Component diagram
 - Deployment diagram
 - a další méně používané
 - Popis vztahů a komunikace (obvykle součástí návrhu)
 - Sequence diagram
 - Collaboration diagram
-
-

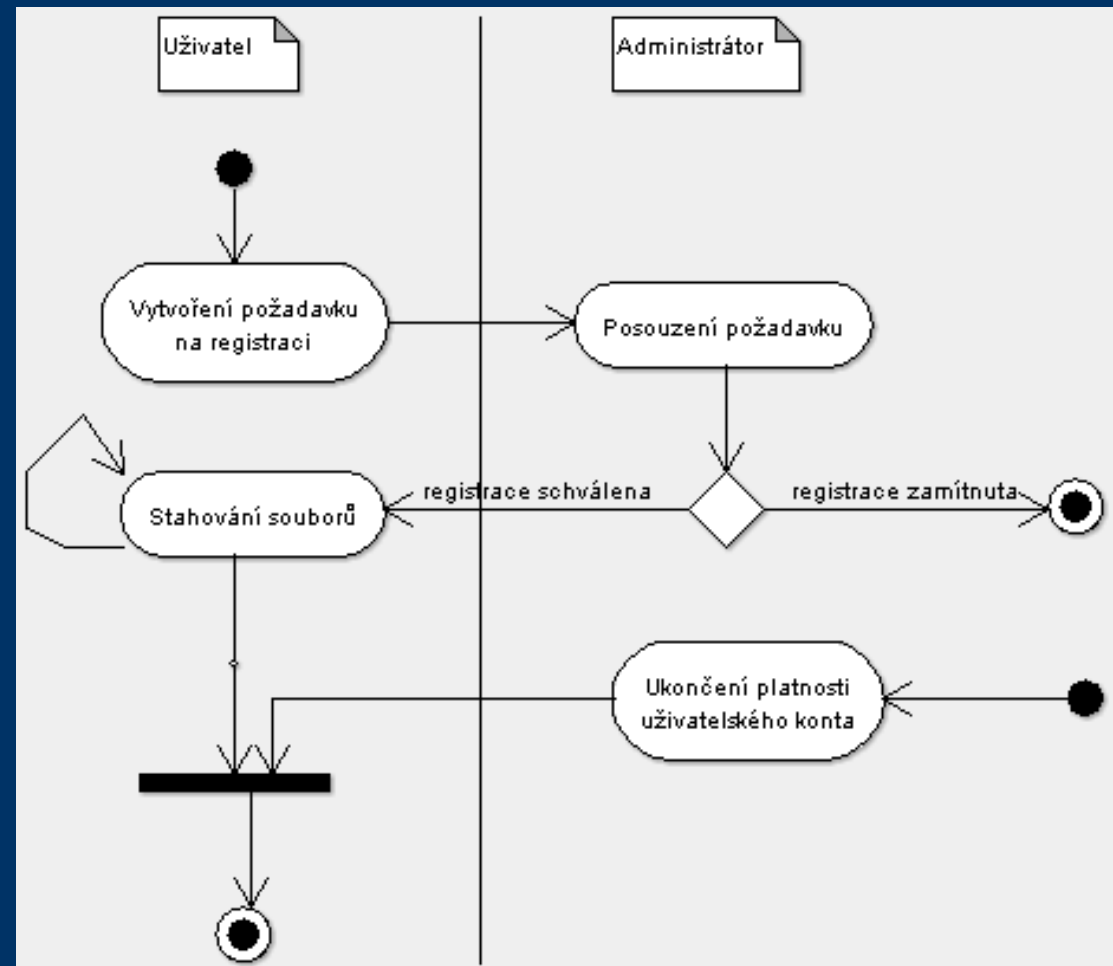
UseCase diagram

- Popisuje aktéry systému a užité případy
- Definuje rozsah – tzn. co se řeší a co nikoliv
- Základní entity:
 - Use case
 - zlaté UC
 - Actor
 - Boundary
 - Association
 - Include
 - Extends



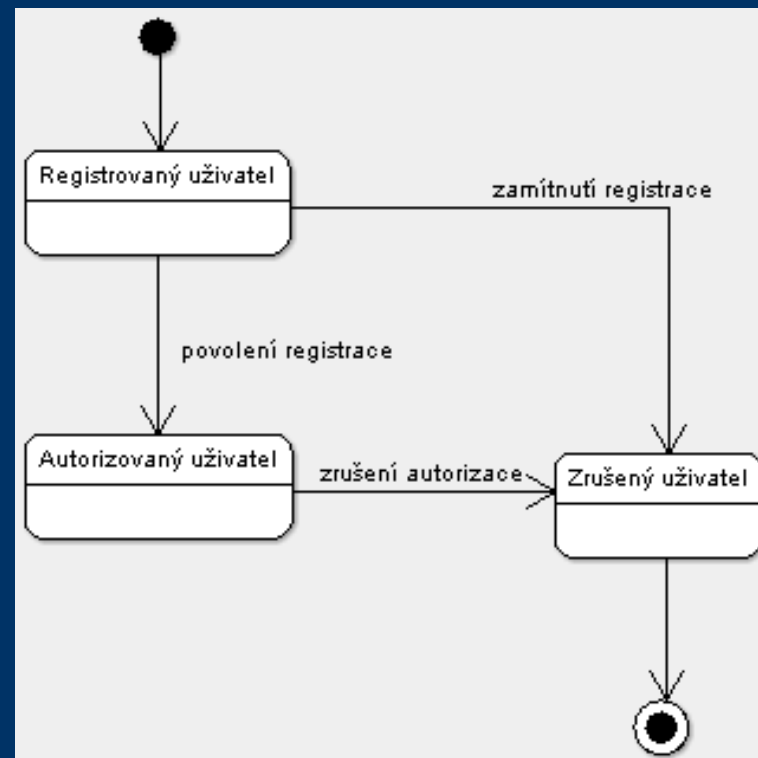
Activity diagram

- Popisuje workflow procesy
- Základní entity:
 - Start / end point
 - Activity
 - Flow (transition)
 - Fork / Join
 - Condition
 - Swimlanes



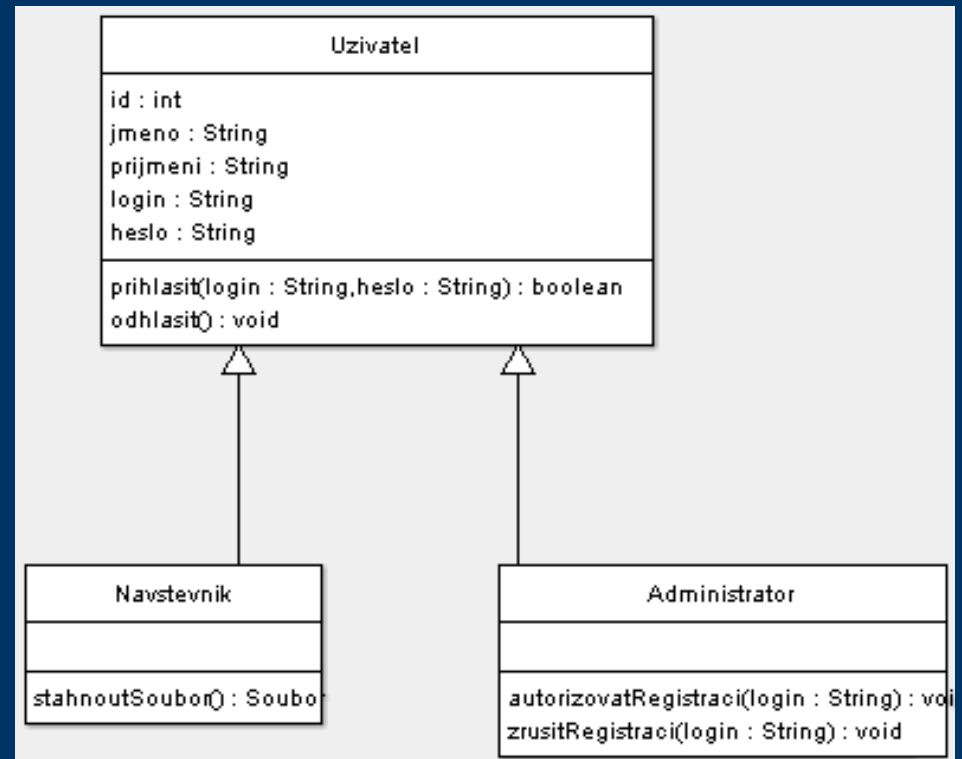
State diagram

- Popisuje stavy jednotlivých objektů
- Základní entity:
 - Start / end point
 - State
 - Transition
 - Fork / Join



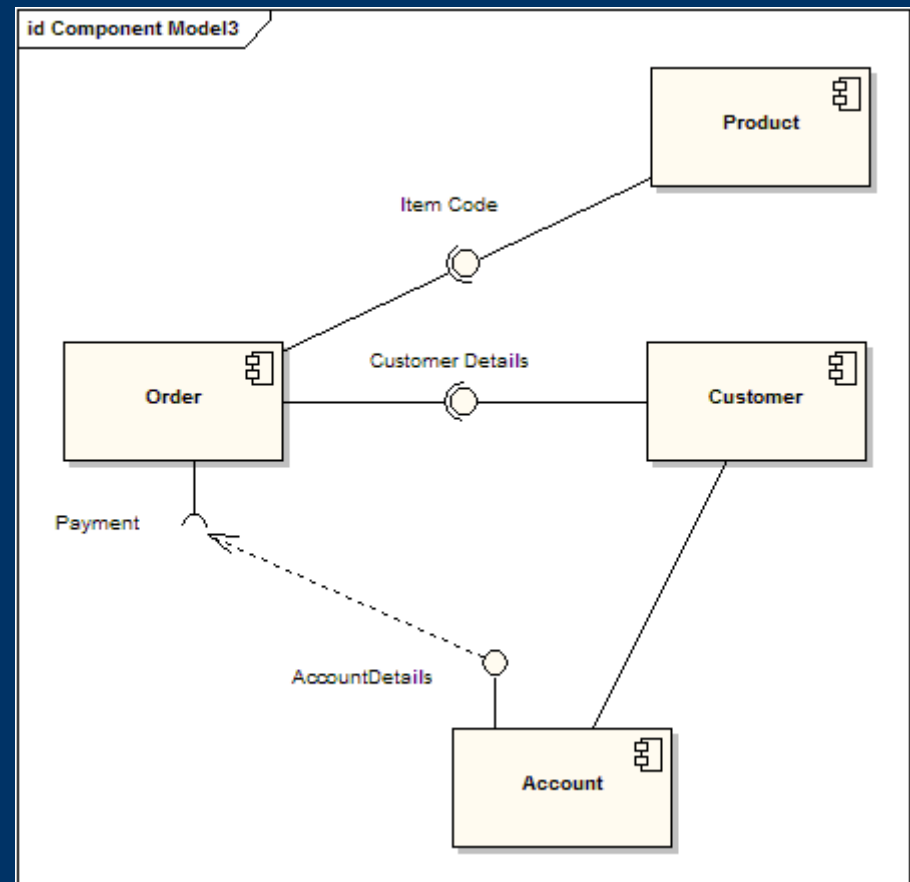
Class Diagram

- Popisuje strukturu objektů
- Základní entity:
 - Classes
 - Fields
 - Methods
 - Agregation
 - Composition
 - Association
 - Generalization
Specialization
 - Multiplicity



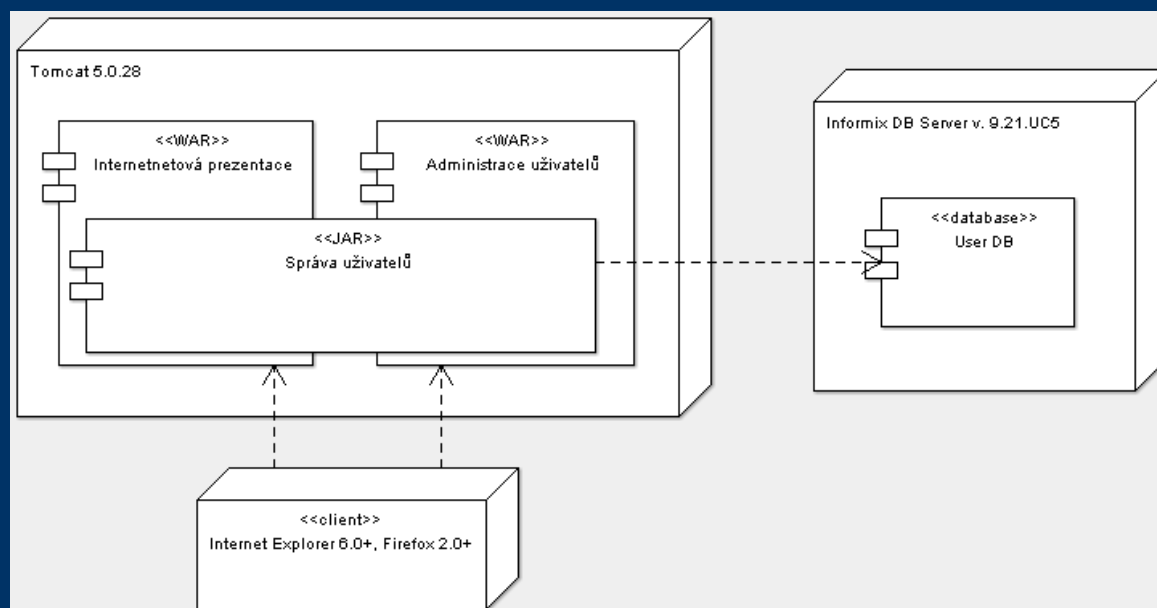
Component diagram

- Popisuje rozložení systému na podčásti
- Základní entity:
 - Component
 - Interface (sedátko)
 - Port (lízátko)



Deployment diagram

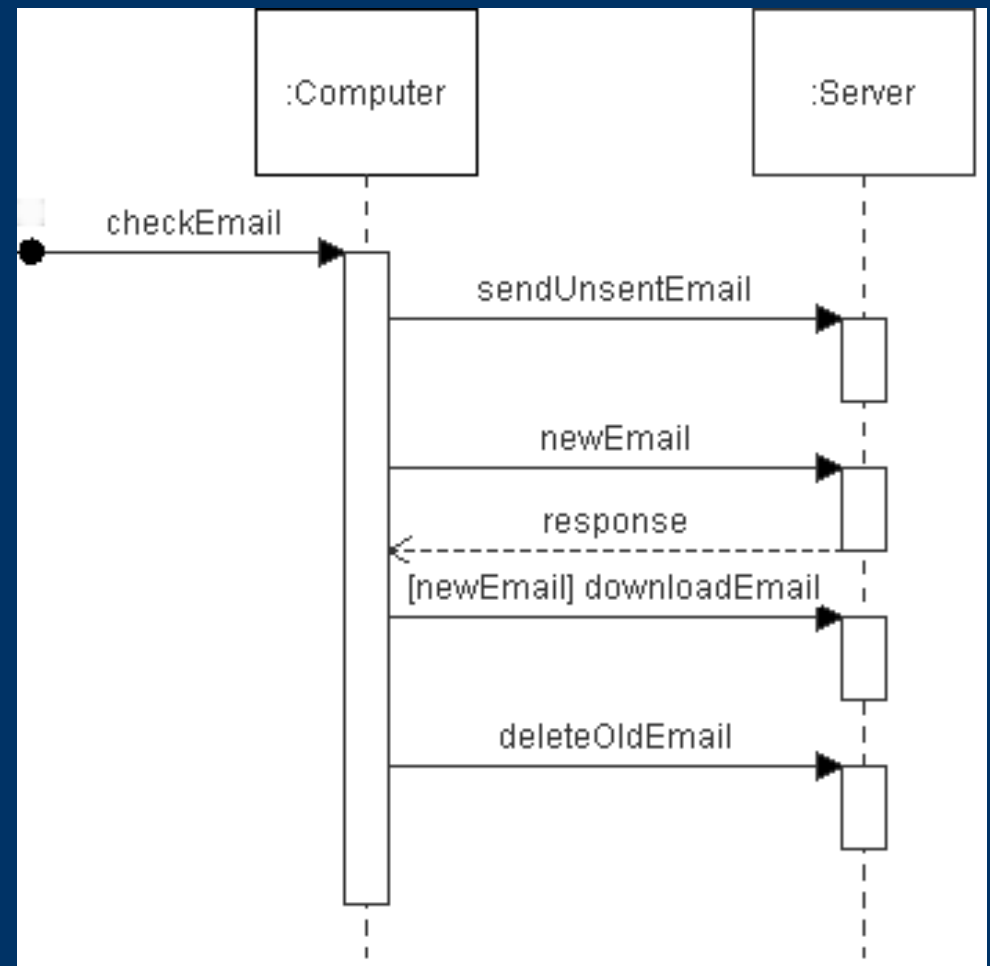
- Popisuje fyzické části aplikace při nasazení
- Základní entity:
 - Node
 - Component
 - Dependency



- Výhodné použití stereotypů
- Důležité zafixovat čísla verzí SW

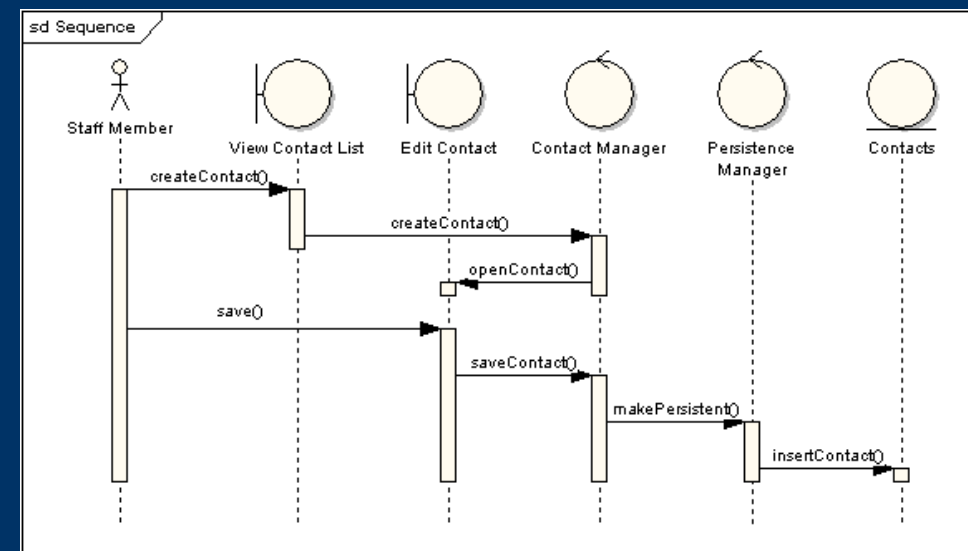
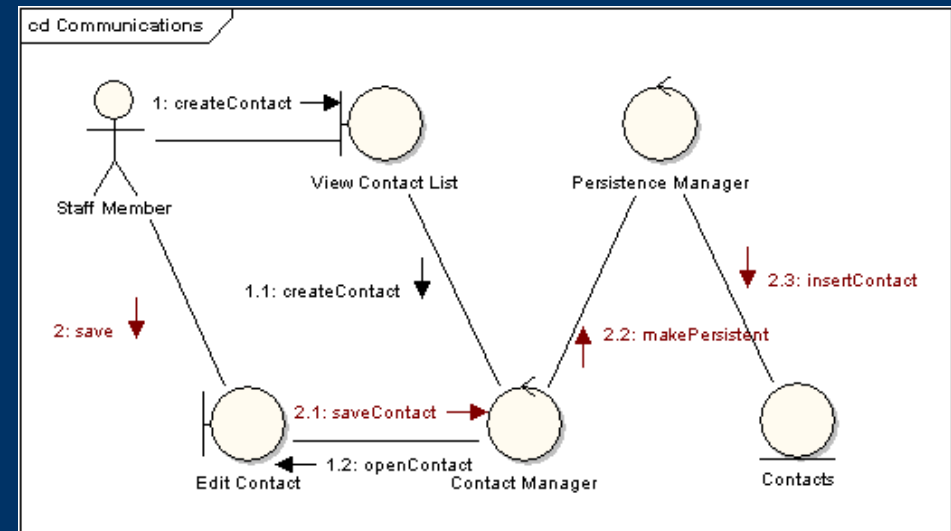
Sequence diagram

- Detailně popisuje sekvenci volání mezi objekty (algoritmus)
- Úzce propojen s class diagramem (blízký programování)
- Základní entity:
 - Instance : class
 - Lifeline
 - Messages
 - Loop



Collaboration diagram

- Podobný sequence diagramu
- Orientuje se na popis vztahů nikoliv na přesný popis návaznosti volání
- Základní entity jsou stejné jako u sequence diagramu



Odhady a plánování

- Bolest většiny projektů, nutné nezapomenout na testy, řízení, dokumentaci, deployment
 - Zajímavé řešení v agilních metodikách (SCRUM, XP)
 - Více lidí = více nedorozumění vs. více nápadů
 - Více lidí = více „projedeného času“ vs. výsledek za kratší dobu
 - Pro každého nového člověka je nutné započítat dobu na „vejítí do projektu“
 - Vhodné započíst 2 druhy rezerv:
 - Realizační (na chybu v odhadech) 10 – 30% (v případě potřeby větší rezervy klást větší důraz na prototypy)
 - Termínovou (řeší rizika onemocnění atd.)
-
-

Tipy a triky

- Nikdy neakceptujte bílá místa
 - Používání stereotypů
 - Komentáře – každý diagram je kryptogram
 - Oponentura (návrhář analytikovi, vedoucí programátor návrhářovi – 2. fázově)
 - respekt k osobnosti návrháře
 - disciplína oponentů – výsledky ve formě návrhů a doporučení
 - Méně je často více
 - Verbální interpretace zákazníkovi
 - Šablony vám umožňují soustředit se na řešený problém
 - Pravidlo jednoho guru projektu (komponenty)
 - Code revision (neslouží k buzeraci programátorů)
-
-

Děkuji za pozornost

