

Novinky v UML 2.5 a agilní modelování

Marek Rychlý

Vysoké učení technické v Brně
Fakulta informačních technologií
Ústav informačních systémů

Přednáška pro AIS
17. října 2019



- 1 Novinky v UML 2.5 a doménově specifické jazyky
 - Cíl UML 2.5 a nové diagramy
 - Ukázka nových diagramů
 - Doménově specifické modelovací jazyky

- 2 Agilní modelování
 - Best Practices
 - Agilní vývoj

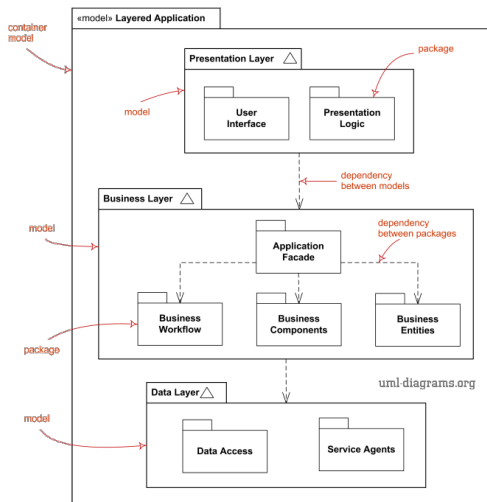


UML 2.5

- UML 2.5 bylo oficiálně vydáno v červnu 2015.
(ale vyšlo již v říjnu 2012 jako „pracovní verze“)
- Zjednodušuje dokument, implementaci a spolupráci UML nástrojů.
(specifikační dokument skutečně jednodušší, jinak není zjednodušení moc vidět)
- Přibýly tři nové strukturální diagramy.
(navýšení z 16 na 19 diagramů mezi UML 2.0 a 2.5)
- Nové diagramy v podstatě upřesňují způsob aplikace některých stávajících diagramů pro konkrétní oblasti návrhu.
 - diagram modelů – seskupuje balíčky do modelů v diagramu balíčků
(např. pro popis vysokoúrovňové architektury seskupením balíčků do vrstev)
 - diagram manifestací – popisuje impl. komponent pomocí artefaktů
(mezi diagramy komponent a nasazení, fyzická „manifestace“ komponent)
 - diagram síťové architektury – popisuje log. a fyz. síťovou arch.
(vychází z diagramu nasazení, orientován na síťovou infrastrukturu)



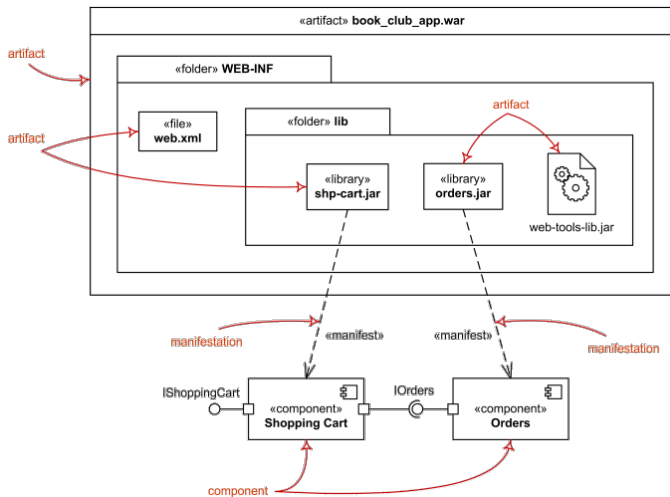
Ukázka diagramu modelů (Model Diagram)



(převzato z „UML Package Diagrams Overview“)



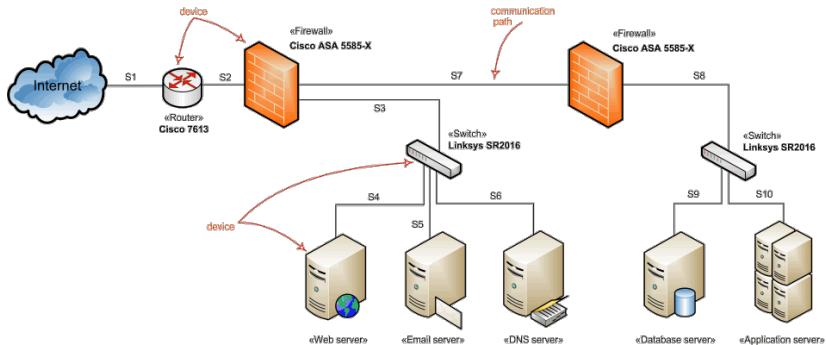
Ukázka diagramu manifestací (Manifestation Diagram)



(převzato z „UML Package Diagrams Overview“)



Ukázka diag. síťové arch. (Network Arch. Diagram)



(převzato z „Network Architecture Diagrams“)

Specifikace UML 2.5 nestanovuje grafickou notaci, jak je uvedeno výše; grafická podoba by dle specifikace byla podobná prvkům diagramu nasazení.



Doménově specifické modelovací jazyky

- UML není „jediný správný modelovací jazyk“.
(návrhář/vývojář musí znát i jiné jazyky, UML není vhodné na vše)
- Mnoho alternativ/doplňků UML je navázáno na metodiky vývoje.
(např. modely Agile Modeling či ERD/DFD v Coad-Youronově strukt. analýze)
- Navíc, použitím UML pro specifické účely, vznikají UML profily.
- Některé z UML profilů se osamostatňují, např.
 - Systems Engineering Modeling Language (SysML)
(pro modelování HW systémů, vč. metodiky specifikace, analýzy a vývoje)
 - UML Testing Profile (UTP)
(pro specifikaci, analýzu, návrh a realizaci artefaktů pro testování)
 - Modeling and Analysis of Real-Time and Embedded Sys. (MARTE)
(pro real-time zabudované systémy, především pro návrh jejich software)

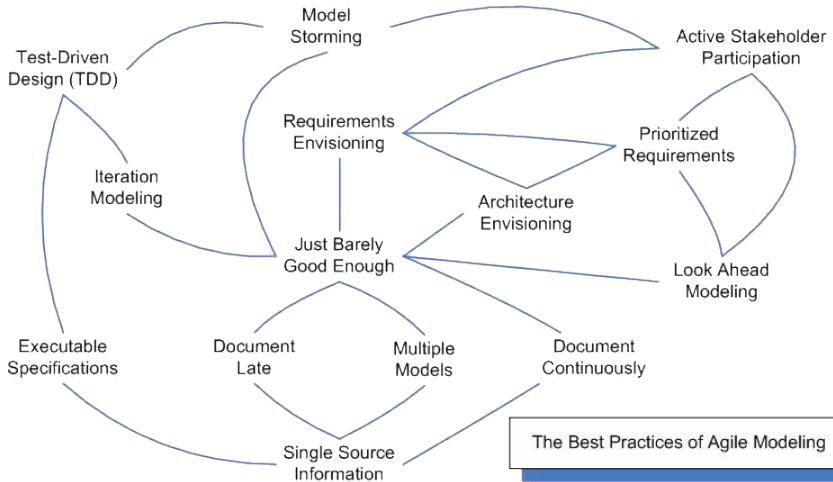


Agilní modelování (Agile Modeling, AM)

- Soubor „best-practices“ pro modelování a dokumentaci software.
(z klasického RUP, agilního extrémního progr. i moderních metodik, jako je Scrum)
- Definuje hodnoty, principy a praktiky pro modelování software.
(5 hodnot, 11+2 principů a 13+5 praktik pro prosazování principů)
- Popisuje jak jsou výše uvedené zapojeny do vývoje software.
(Agile Model Driven Development, AMDD, přístup k vývoji software)
- Konkrétně definuje AM následující hodnoty
 - komunikace
(komunikace je důležitá a častá, uvnitř týmů i ven mezi týmem a zákazníkem)
 - jednoduchost
(tvorba jednoduchých modelů, spíše pro porozumění, než pro dokumentaci)
 - zpětná vazba
(rychlá a častá zpětná vazba, reakce na předložené modely a interakce)
 - odvaha
(dělat rychlá rozhodnutí, zkoušet nové, zahazovat či měnit stávající)
 - pokora/respekt
(naslouchat ostatním vývojářům i zákazníkům, přijímat jejich nápady)



„Best Practices“ v agilním modelování



(převzato z „Agile Modeling Best Practices“)

Copyright 2005-2011 Scott W. Ambler



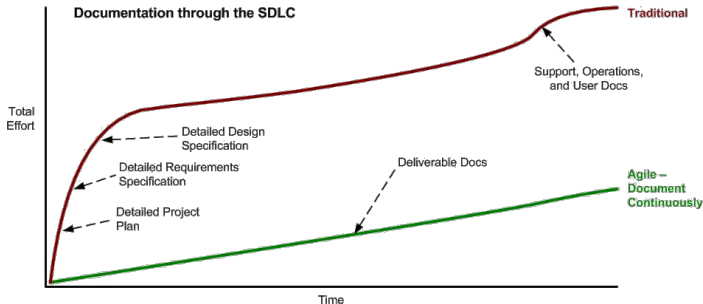
Best Practices: Active Stakeholder Participation

- „Stakeholders“ (SH) jsou všichni účastníci projektu.
(příímí/nepříímí uživatelé, jejich vedoucí, tech. podpora, investoři, auditoři, atd.)
- Potřeba je zapojit do řešení, mají informace, které vývojářům chybí.
(nestačí komunikovat jen s uživateli, nutné co nejvíce zapojit každého)
- Různé způsoby jak je zapojit do řešení
 - časté diskuze – SH musí být připraveni a ochotni se jich účastnit
(vývojáři by neměli plýtvat jejich časem, jinak ztratí důvěru)
 - „inclusive modelling“ – SH mohou používat snadné model. techniky
(tvorba návrhů uživatelských rozhraní, popisů scénářů, atp; zvyšuje motivaci)
 - vzdělávání v IT – naučit SH vidět pohledem IT, příležitosti a hrozby
(musí mít zájem se takto vzdělávat, musí vidět, jak jim to pomůže v růstu)
 - časné zapojení uživatelů/podpory – snižuje potřebu je školit
(SH pak lépe přijmou systém, mohou pomoci vzdělat další uživatele)
- Vývojáři musí porozumět potřebám SH a pracovat s nimi.



Best Practices: Document Continuously

- Dokumentace by měla vznikat spolu se vznikem finálního kódu.
(to se týká i modelů, které s tím kódem souvisí; např. diagramů tříd, balíčků, atp.)
- Modeluje se již před kódem, pro porozumění, ne pro dokumentaci.
(model je tvořen pro porozumění problematice, až potom prostředek dokumentace)
- Často se kombinuje s „just-in-time“ a „executable specification“.
(modelování až, když je potřeba; vývoj řízený testem, Test-driven Development)



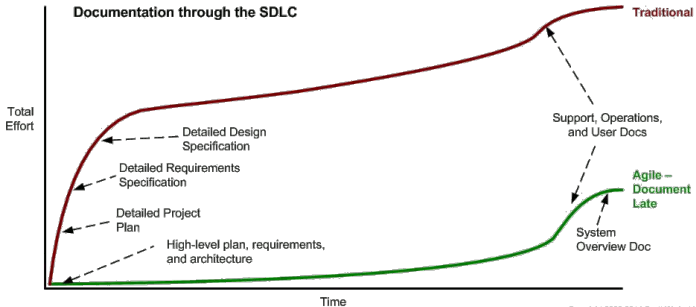
Copyright 2006-2014 Scott W. Ambler

(převzato z „Document Continuously: An Agile Best Practice“)



Best Practices: Document Late

- Dokumentace se musí udržovat, je lepší ji připravit až na konec. (počáteční/průběžné modely se většinou zahodí; dok. musí být vždy aktuální)
- Odložená tvorba dokumentace zvyšuje flexibilitu, ale i rizika. (může chybět, když je potřeba, např. při předání rozpracovaného; odkládá náklady)
- Často se kombinuje s „just-in-time“ a „executable specification“. (modelování až, když je potřeba; vývoj řízený testem, Test-driven Development)

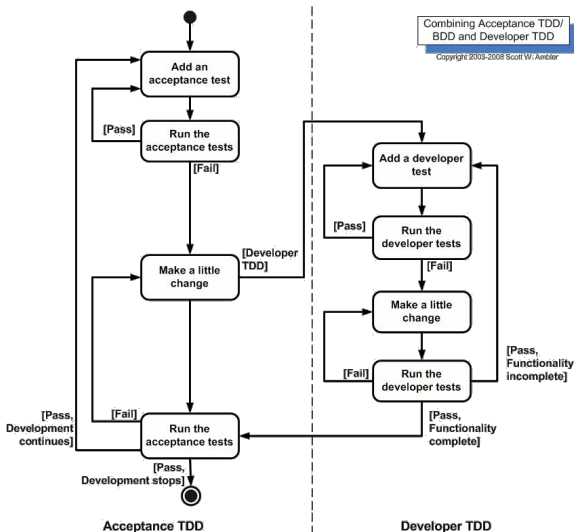


Copyright 2006-2014 Scott W. Ambler

(adopted from „Document Late: An Agile Best Practice“)



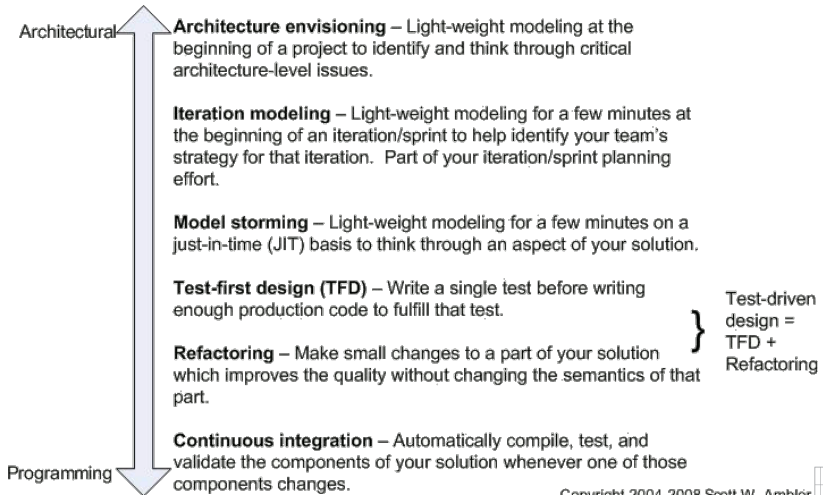
Best Practices: Test-driven Development (TDD)



(převzato z „Introduction to Test Driven Development (TDD)“)



Agile Design Practices



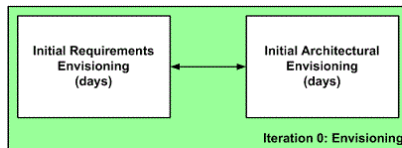
Copyright 2004-2008 Scott W. Ambler

(převzato z „Agile Design“)

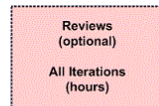
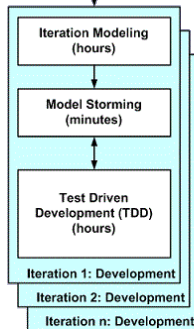


Agile Model Driven Development (AMDD)

- Identify the high-level scope
- Identify initial "requirements stack"
- Identify an architectural vision



- Modeling is part of iteration planning effort
- Need to model enough to give good estimates
- Need to plan the work for the iteration
- Work through specific issues on a JIT manner
- Stakeholders actively participate
- Requirements evolve throughout project
- Model just enough for now, you can always come back later
- Develop working software via a test-first approach
- Details captured in the form of executable specifications



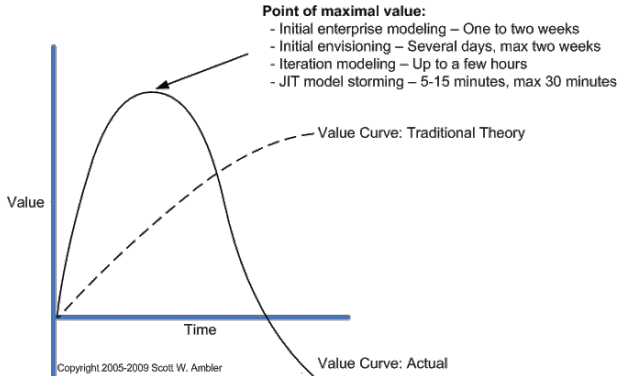
Copyright 2003-2007
Scott W. Ambler



(převzato z „Agile Model Driven Development (AMDD): The Key to Scaling Agile Software Dev.“)

Best Practices: Just Barely Good Enough / ... In Time

- Modely je vhodné tvořit, když, jak a dokud jsou potřeba.
(časné či velké úsilí věnované tvorbě modelů je zbytečné; používejte papír/tabuli!)
- Hodnota modelů podle toho, jak nám pomohou k lepšímu SW.
(tj. užitečnějšímu a bezchybnému, ale i k lépe udržovatelnému software)

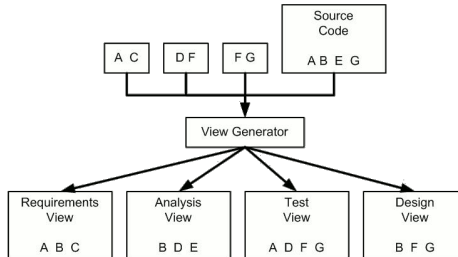


(převzato z „Just Barely Good Enough Models and Documents: An Agile Best Practice“)



Best Practices: Multiple Models & Single Source Info.

- Každý druh modelu/diagramu má své specifické použití.
(je potřeba mít přehled, znát, na co který model (ne)použít, nejen v UML)
- Pokud se něco nedaří namodelovat, je asi chybný druh modelu.
(často zkoušet různé druhy modelů/diagramů, tedy různé pohledy na problém)
- Modely provázané, různé pohledy na tytéž entity, ne jejich kopie.
(použít modelovací prostředí, které udržuje modely (a kód) synchronizované)

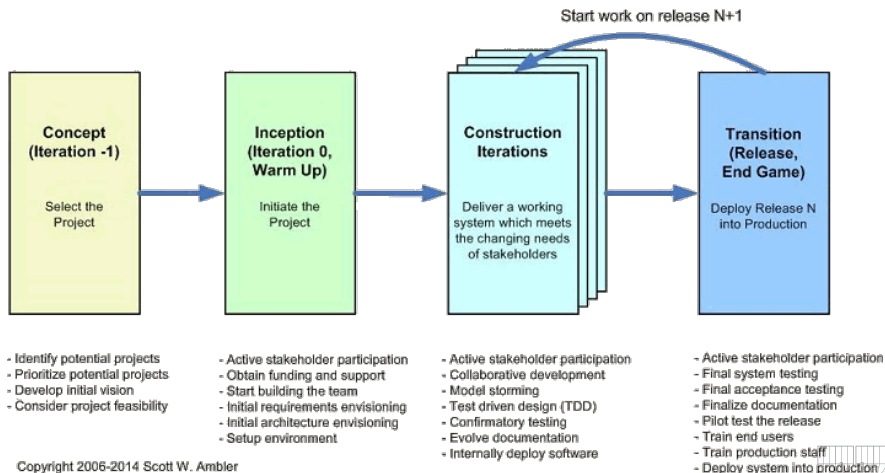


Copyright 2005 Scott W. Ambler

(převzato z „Single Source Information: An Agile Best Practice for Effective Documentation“)



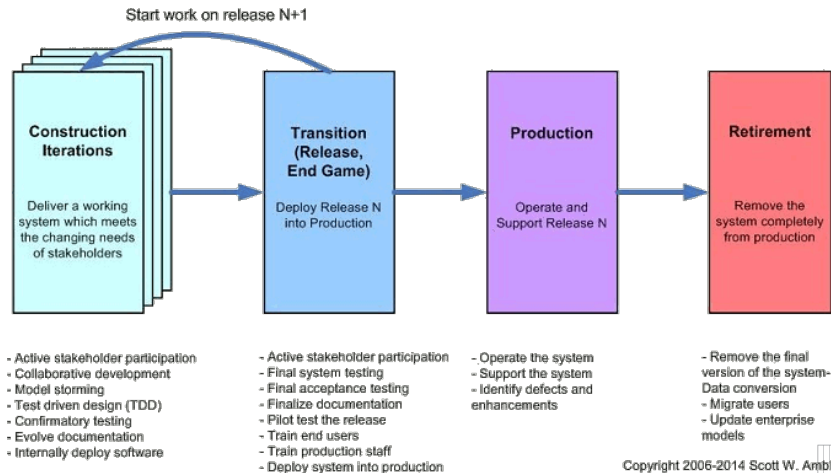
Agile Development (I)



(převzato z „Agile Design“)



Agile Development (II)



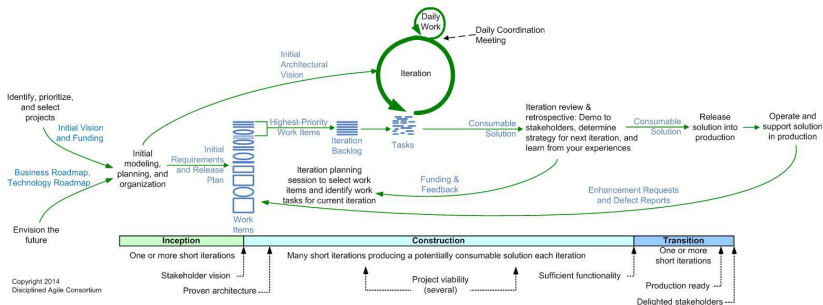
Copyright 2006-2014 Scott W. Ambler



(převzato z „Agile Design“)

Disciplined Agile 2.0: Agile Delivery

- Iterativní, částečně založené na Scrum, ale s jinou terminologií. (vychází ze skutečnosti, že v praxi se čistý Scrum moc nepoužívá)
- Pracuje se externími vstupy, úkoly v prioritní frontě a milníky. (zpracovávají se vždy úkoly s nejvyšší prioritou, vhodné i pro nezkušené týmy)



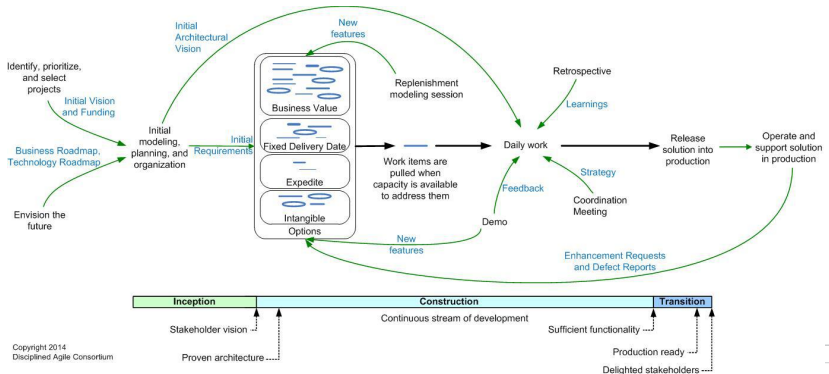
Copyright 2014
Disciplined Agile Consortium

(převzato z „The Agile/Basic Lifecycle: Extending Scrum“)



Disciplined Agile 2.0: Lean & Continuous Delivery

- Jen zkušené týmy si mohou dovolit „lean“ přístup.
- Není fronta úkolů, ale tyto jsou zařazeny do čtyř kategorií.
(zpracovávají se úkoly, co mají blízký termín či je tým sám chce řešit)



Copyright 2014
Disciplined Agile Consortium

(převzato z „The Advanced/Lean DAD Lifecycle“)



Shrnutí a závěr

- UML 2.5 zjednodušuje a zavádí tři nové strukturální diagramy.
(diagram modelů, diagram manifestací, diagram síťové architektury)
- Agilní modelování popisuje „best-practices“ v oboru.
(definuje hodnoty, principy a praktiky pro modelování software)
- Vychází, využívá a je navázáno na moderní metody vývoje SW.
(z klasického RUP, Scrum, XP, Agile Data, ...)



Literatura



Ambler, S. (2002).

Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process.

John Wiley & Sons, Inc., New York, NY, USA.



Lines, M. and Ambler, S. W. (2015).

Introduction to Disciplined Agile Delivery: A Small Agile Team's Journey from Scrum to Continuous Delivery.

CreateSpace Independent Publishing Platform, USA, 1st edition.

