

# **Disciplined Agile Delivery (DAD) framework**

## **(Podpora DAD v rámci platformy Jazz)**

**(SEMESTRÁLNÍ PRÁCE Z PŘEDMĚTU 4IT421)**

Vypracoval: Bc. Tomáš Svoboda (xsvot13)

Datum: prosinec 2011

# Obsah

1 Úvod.....	3
2 Platforma Jazz .....	3
2.1 Architektura platformy Jazz .....	3
2.2 Základní komponenty Serveru Jazz .....	4
2.3 Rational Team Concert .....	5
3 Instalace Jazz Team Serveru .....	5
3.1 Spuštění serveru .....	7
3.2 Nastavení serveru .....	8
3.3 Správa licenčních klíčů .....	9
3.3.1 Contributor.....	9
3.3.2 Developer.....	9
3.3.3 Developer for IBM Enterprise Platforms .....	9
3.3.4 Stakeholder .....	9
3.3.5 Licence zdarma.....	10
3.4 Zhodnocení.....	10
4 Rational Team Concert Eclipse klient.....	10
4.1 Šablony procesů .....	11
4.2 Přidání šablony pro DAD.....	11
4.3 Využití šablon při vytváření projektu .....	12
4.4 Uživatelé a jejich role.....	13
4.5 Plánování projektu.....	14
4.6 Pracovní položky.....	15
4.7 Další funkce.....	15
4.8 Zhodnocení.....	15
5 Závěr .....	16
6 Zdroje.....	17

# 1 Úvod

Tato práce se zabývá softwarovou podporou Disciplined Agile Delivery (dále jen DAD) frameworku od společnosti IBM. Cílem této práce je popsat platformu pro podporu agilních metodik, která je společností IBM vyvíjena, a to včetně postupu instalace potřebného software. Dalším cílem je zhodnocení samotné práce s tímto softwarem z pohledu nezaujatého uživatele. Je kladen důraz především na podporu frameworku DAD. Práce se nezabývá teoretickým popisem DAD frameworku.

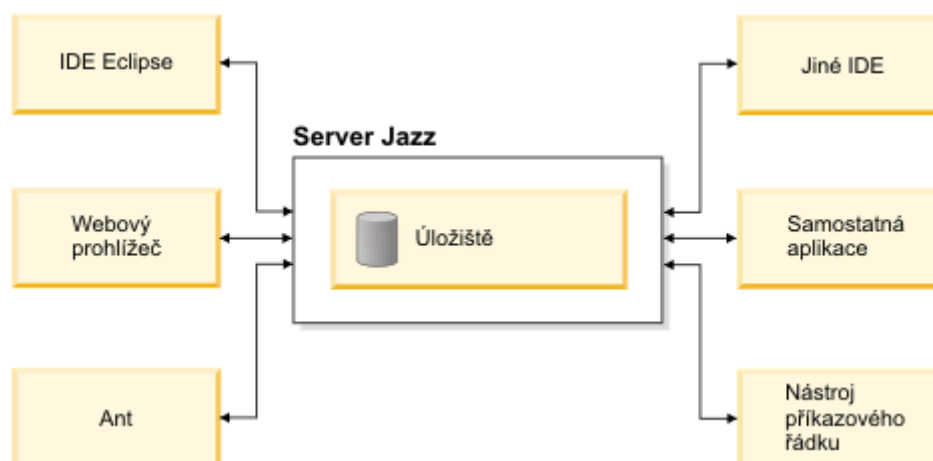
## 2 Platforma Jazz

IBM pro podporu svých agilních metodik a frameworků využívá platformu Jazz. Ta si klade za cíl dosáhnout lepší spolupráce v životním cyklu software. Technologická platforma Jazz má tyto základní cíle:

- ✦ Podpořit hladkou integraci úloh v rámci životního cyklu softwaru.
- ✦ Napomoci týmové spolupráci a koordinaci napříč životním cyklem softwaru.
- ✦ Pomoci týmům efektivněji sestavovat software.
- ✦ Podpořit globálně rozptýlené vývojové týmy.
- ✦ Poskytovat přizpůsobitelná řešení od malých týmů až po velké podniky.
- ✦ Zajišťovat údržbu záznamů pro audit a automatizaci evidence.
- ✦ Podpořit integraci uživatelských rozhraní (IDE, webový prohlížeč atd.), která vyhovují potřebám zákazníků.
- ✦ Pěstovat široký ekosystém poskytovatelů nástrojů, včetně nezávislých dodavatelů softwaru.
- ✦ Učinit vývoj softwaru zábavnější

### 2.1 Architektura platformy Jazz

Platforma Jazz je založená na modelu Klient-server. Komunikace probíhá pomocí protokolu HTTP na základě RESTu. Pro platformu je typické, že Server může obsluhovat mnoho různých klientů.



*Ilustrace 1: Architektura platformy Jazz; Zdroj:*

Jak lze vidět z Ilustrace 1, tak k Serveru lze přistupovat různými způsoby. Nejobvyklejší je přístup pomocí IDE nástrojů a Webového prohlížeče. Nicméně je umožněna komunikace i pomocí automatizovaných Ant skriptů a pomocí specifických příkazů zadávaných přímo do příkazové řádky. Samozřejmě je také možné si vytvořit vlastní aplikaci, která bude komunikovat se serverem.

Samotný server je naprogramován v jazyce Java a je postaven na standardu Java EE. Pro svůj správný běh je tedy nutné mít správně nainstalován vybraný aplikační server. Současná verze Jazz Team Serveru podporuje server Apache Tomcat a IBM Websphere.

## **2.2 Základní komponenty Serveru Jazz**

Jazz Team Server je složen ze dvou základních komponent, kterými je úložiště a komponenta týmových procesů.

Většina nástrojů životního cyklu aplikace ukládá informace specifické pro daný nástroj na centrálním místě, kde mohou být zpřístupněny všem členům týmu. Platforma Jazz poskytuje tento mechanismus v rozšiřitelném úložišti. Komponenta Úložiště se nachází v jádře, takže její mechanismy jsou dostupné jiným komponentám ve všech konfiguracích klienta a serveru.

Každá položka v úložišti má jedinečné ID položky, které lze použít jako klíč k jejímu načtení. Ve skutečnosti jsou ID položek univerzálně jedinečnými identifikátory (UUID), takže položku vytvořenou v jednom úložišti lze replikovat do jiného úložiště a zachovat přitom její identitu. Podobně jsou univerzálně jedinečné identifikátory přidruženy také ke stavům položek a hodnotám obsahu, takže lze replikovat libovolný stav položky do jiných úložišť a přitom zachovat jeho identitu.

Operace čtení z úložiště a zápisu do úložiště jsou atomické. Rozhraní API na straně serveru rovněž obsahují operaci pro explicitní uzavření libovolné posloupnosti operací čtení a zápisu do jediné

atomické transakce. Platforma Jazz obsahuje rozhraní API na straně serveru pro streaming objektů obsahu do úložiště a z úložiště navržené tak, aby streaming mohl být realizován bezpečně mimo transakci úložiště.

Komponenta Týmových procesů slouží k vytváření a ukládání procesů týmu. V rámci platformy Jazz je proces chápán jako kolekce postupů, pravidel, pokynů a konvencí, které tým používá k uspořádání své práce. Důležité je podotknout, že úroveň popisu procesů se liší v závislosti na velikosti týmu a zvyklostem. Typicky platí, že u malého týmu bývá proces obvykle neformální a nedokumentovaný. Zatímco s růstem velikosti týmu se proces stává více formalizovaný.

Platforma Jazz podporuje proces následujícími způsoby:

- ⤴ Technologická Platforma Jazz zohledňuje procesy. Procesy týmu mohou obecně ovlivnit libovolný aspekt práce, kterou člen týmu provádí. Myšlenka procesu je klíčovým prvkem návrhu technologické platformy Jazz. Proces týmu je reprezentován explicitně a veškerá práce, kterou členové týmu v rámci systému provádějí, se děje v rámci kontextu procesu.
- ⤴ Komponenty Jazz povolují procesy. Všechny komponenty platformy Jazz včetně počátečního jádra, byly navrženy tak, aby umožnily platnému procesu uplatnit vliv nad položkami a operacemi komponenty.
- ⤴ Technologická platforma Jazz reprezentuje proces pomocí standardních položek platformy Jazz, které jsou uloženy v úložišti. Tato reprezentace procesu dokáže popsat procesy všech druhů a může zachytit a ustanovit proces týmu v přiměřeně velkém (nebo malém) rozsahu.
- ⤴ Technologická platforma Jazz je z pohledu procesů neutrální. Samotná platforma Jazz nemá žádný vestavěný proces, který by vynucoval, co se provádí v libovolné konkrétní situaci. Je na týmu, aby definoval příslušný řídicí proces za účelem provádění takovýchto rozhodnutí v rámci jeho projektu. Verze 3 obsahuje několik šablon procesu, včetně verze šablony zapouzdřující proces použitý samotným projektovým týmem platformy Jazz.

## **2.3 Rational Team Concert**

Základním nástrojem, který je postaven na platformě Jazz je Rational Team Concert (dále jen RTC). Tento nástroj slouží ke spolupráci na vývoji software v celém jeho životním cyklu. Klientské rozhraní je v současnosti vytvořeno pro IDE Eclipse, IDE Microsoft Visual Studio a také přes webové rozhraní.

### 3 Instalace Jazz Team Serveru

Všechny nástroje platformy Jazz lze stáhnout ze stránek Jazz.net. Ke stažení je však nutná registrace, která je ale zdarma. Každý registrovaný uživatel má možnost zasahovat do vývoje platformy pomocí kladení požadavků na vývoj.

Pro účely podpory frameworku DAD budeme potřebovat nainstalovat Jazz Team Server a Rational Team Concert. Je nutné si tedy stáhnout instalační balíček Rational Team Concertu ze serveru Jazz.net. Samotná instalace je velmi snadná a probíhá pomocí průvodce, který je lokalizován do češtiny, proto jí zde nebudu kompletně popisovat. Pro účely menších týmu je vhodné nainstalovat Jazz Team server společně s aplikacemi na stejný server. Dále popisovaná instalace odpovídá instalace dle Ilustrace 2 níže. Pokud instalujete Server na operační systém Windows 7 nebo Vista, tak pravděpodobně budete upozorněni, že tento typ OS není podporován. Nicméně toto upozornění lze ignorovat. Jako instalační adresář bych doporučil složku *C:\IBM\*, protože při instalaci do složky *Program files* může docházet k chybám.

## Rational Team Concert



**Plán a příprava**  
Chcete-li rychle zahájit malou implementaci s použitím výchozí topologie a databází **Jazz Team Server a aplikace**. Implementace databáze Derby je omezena na 10 uživatele. přečtete **Interaktivní instalační příručku (Interactive Installation Guide)** nebo **(Interactive Upgrade Guide)**, která vám pomůže s plánováním implementace nebo **Volitelné programy** na adrese jazz.net, kde získáte programy, jako DB2, WebSphere Development Intelligence.

**Nainstalujte server Jazz Team Server a aplikace**  
Rational Team Concert, Rational Quality Manager a Rational Requirements Composer nainstalujete Jazz Team Server a aplikace, můžete povolit funkčnost pro libovolnou konfiguraci a aktivaci odpovídajících licenčních klíčů produktů. Zkušební klíče jsou součástí produktových stránek administrace serveru Jazz Team Server.

**Nainstalovat Jazz Team Server spolu s aplikacemi na jeden aplikační server. Tuto volbu zvolte pro malé implementace.**

**Jazz Team Server s nezbytnými základními klíči, včetně zkušebních, a aplikace CCM, QM anebo RM**

Nainstalovat Jazz Team Server a aplikace rozdělené na různé aplikační servery. Tuto volbu zvolte pro velké implementace.

**Návod k nastavení a konfiguraci serveru viz část **Poinstalační úlohy**.**

**Instalovat volitelné nástroje**  
Nainstalujte vývojářské klienty založené na prostředí IDE

**Rational Team Concert - Client for Eclipse IDE**  
**Rational Team Concert - Client for Microsoft Visual Studio IDE**

Nainstalujte sadu nástrojů sestavovacího systému pro spouštění sestavení Jazz

**Rational Team Concert - Build System Toolkit**

Nainstalujte synchronizátory pro integraci s jinými systémy

**Rational Team Concert - ClearCase Synchronizer**  
**Rational Team Concert - ClearQuest Synchronizer**

Ilustrace 2: Úvodní obrazovka instalace Rational Team Concert

### 3.1 Spuštění serveru

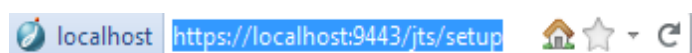
Jako úložiště využívá Jazz Team Server standardně databázi Derby. Tato databáze je však vhodná pouze pro malé týmy do 10 uživatelů. V případě, že je potřeba zapojit více lidí je nutné využít jiný databázový server. Jazz podporuje databáze DB2, MS SQL Server a Oracle. Pro účely této práce je využita databáze Derby, která je nainstalována automaticky při instalaci serveru a není nutné ji nijak nastavovat.

Po dokončení instalace je nutné server spustit. V prostředí Windows to lze přes tlačítko *Start*, kde vybereme složku *Jazz Team Server* a spustíme pomocí volby *Spustit server Jazz Team Server a aplikace CLM*. Tím dojde ke spuštění aplikačního serveru Tomcat včetně potřebných balíčků Jazzu. Tento proces může trvat i několik minut v závislosti na hardwarovém výkonu. To, že se server

spustil správně lze poznat tak, že v příkazové řádce Tomcatu se zobrazí informace o tom, že server byl spuštěn.

### 3.2 Nastavení serveru

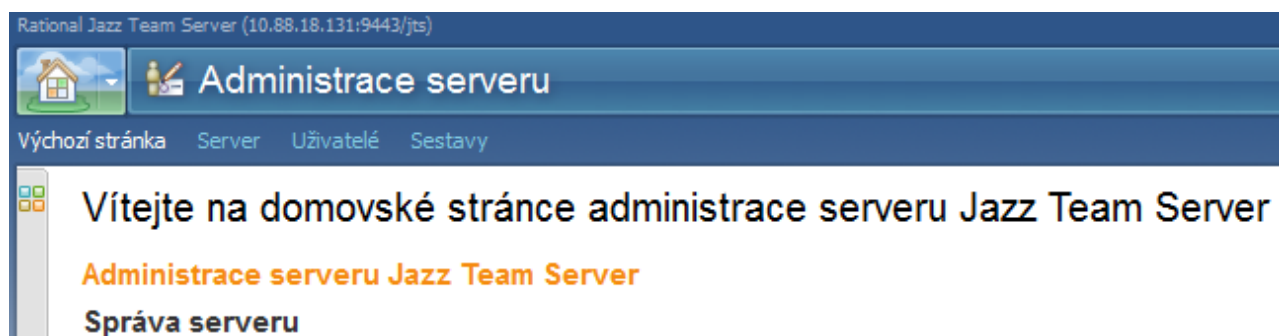
Jazz Team Server (dále jen JTS) se standardně spouští na portu 9443. Je proto vhodné tento port povolit v nastavení firewallu. K administraci serveru se přistupuje pomocí webového rozhraní. URL nastavení serveru je vidět z Ilustrace 3. Samozřejmě, že je vždy nutné nahradit *localhost* za správnou IP adresu nebo doménové jméno serveru v případě přístupu z jiného počítače.



*Ilustrace 3: URL adresa pro nastavení serveru JTS*

JTS podporuje prohlížeče Firefox od verze 3.6 a Internet Explorer od verze 7. Standardně je použito šifrování pomocí HTTPS a certifikátu IBM. Je proto možné, že při prvním spuštění budete muset přidat tento certifikát mezi povolené.

Přístupové údaje pro administraci jsou při instalaci nastaveny na jméno ADMIN a heslo ADMIN. Průvodce prvním nastavením zde opět nebudeme popisovat kompletně. Pouze bych zdůraznil, že v průběhu průvodce je nutné nastavit jedinečné URI pro všechny zdroje. Ideálně by takové URI mělo být v podobě *https://adresa.serveru:9443/jts*. Jinak v průběhu průvodce lze nechat všechny hodnoty na výchozích hodnotách. Po úspěšném dokončení celého procesu budete přesměrováni do administrace serveru, kterou můžete vidět na Ilustraci 4 níže.



*Ilustrace 4: Úvodní stránka Administrátorské části serveru*



### **3.3 Správa licenčních klíčů**

Od verze 3.0 existují v rámci Rational Team Concertu čtyři základní licence pro klienty. Nicméně je možné vytvářet i uživatele, kteří nemají žádnou licenci. Takový uživatelé mají vždy přístup pouze pro čtení.

#### **3.3.1 Contributor**

Tato licence Contributor je určena profesionálním členům týmů, kteří se aktivně účastní projektu, ale nejsou vývojáři. Uživatel, kterému byla přiřazena licence Contributor Client Access License má úplný přístup pro čtení a pro zápis k řízení změn, přizpůsobení sestavy a plánování. Tato licence také zajišťuje přístup pro čtení k funkcím Software Configuration Management, Automatizace (Sestavovací systém), aplikace Requirements Management a Správa testů, pokud to dále neomezují oprávnění rolí v rámci procesu.

#### **3.3.2 Developer**

Tato licence Developer je určena profesionálním vývojářům, kteří se aktivně účastní projektu. Uživatel, kterému byla přiřazena licence Developer Client Access License má úplný přístup pro čtení a pro zápis k aplikaci Change Management, přizpůsobení sestavy, plánování, k nástroji Software Configuration Management a automatizaci (sestavovací systém). Tato licence také zajišťuje přístup pro čtení k funkcím aplikace Requirements Management a Správa testů, pokud to dále neomezují oprávnění rolí v rámci procesu.

#### **3.3.3 Developer for IBM Enterprise Platforms**

Tato licence Developer-IEP je určena profesionálním vývojářům, kteří se aktivně účastní projektu. Uživatel, kterému byla přiřazena licence Developer-IEP Client Access License má úplný přístup pro čtení a pro zápis k aplikaci Change Management, přizpůsobení sestavy, plánování, k nástroji Software Configuration Management, automatizaci (sestavovací systém) a rozšíření schopnosti specifických podnikovým platformám IBM. Tato licence také zajišťuje přístup pro čtení k funkcím aplikace Requirements Management a Správa testů, pokud to dále neomezují oprávnění rolí v rámci procesu

### 3.3.4 Stakeholder

Tato licence Stakeholder je určena pro podporu rozšířených uživatelů, jako např. externích zákazníků, podpůrného personálu nebo těch, kteří chtějí upravovat pracovní položky a sledovat průběh projektu. Uživatel, kterému byla přidružena licence Stakeholder Client Access License, má oprávnění pro čtení a pro zápis k řízení změn a oprávnění ke čtení sestav, plánování, pokud tento přístup není dále omezen prostřednictvím oprávnění k procesům na základě rolí.

### 3.3.5 Licence zdarma

Z webu Jazz.net lze získat licenci pro 10 vývojářů zdarma. Licenci je nutné nahrát na server. To lze provést přes administraci JTS, kde v záložce *Server* vpravo vybereme z menu Licence položku *Správa licenčních klíčů*. Dále vybereme *Přidat...* v *Typech licencí pro klientský přístup*. Po úspěšném nahrání souboru s licencemi staženými z webu Jazz.net můžeme vytvořit až 10 uživatelů s licenci Developer.

## 3.4 Zhodnocení

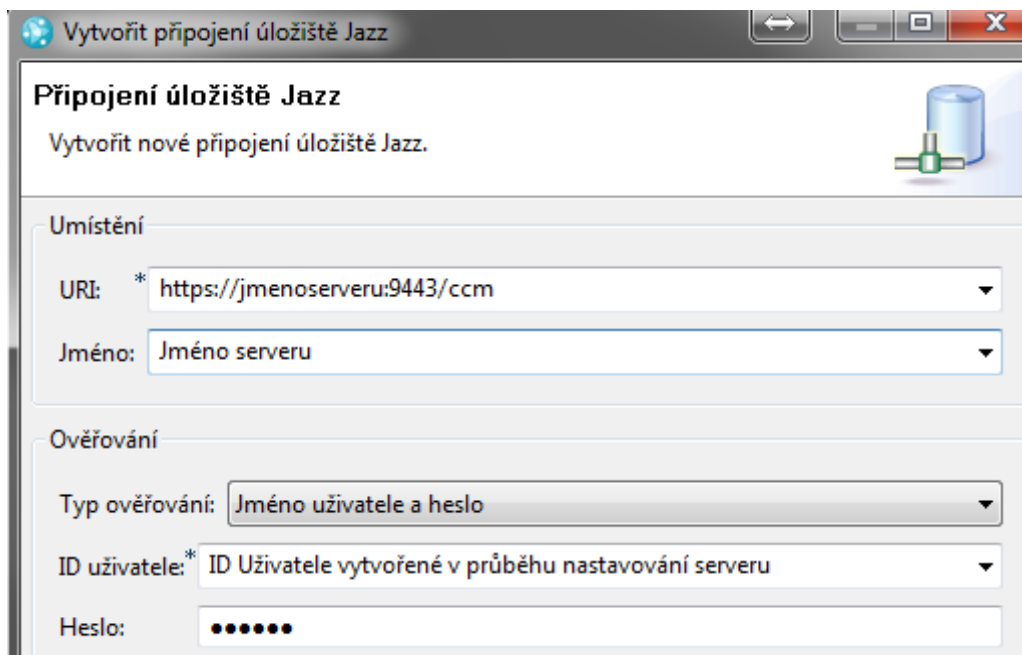
Z pohledu uživatele není instalace a nastavení serveru JTS složité. Díky tomu, že je téměř všechny položky možné nachat v defaultním nastavení, tak si s procesem instalace není problém poradit.

Horší hodnocení si však zaslouží systém licenčních klíčů, který není příliš přehledný a není zcela jasné, jakým způsobem jsou tyto licence využity. Navíc, jak si ukážeme v další kapitole není vazba mezi rolí uživatele v projektu a jeho typem licence. Může tak docházet k tomu, že uživatel dle své role v týmu by měl mít oprávnění k určitému úkonu, ale protože jeho licence to neumožňuje, tak mu to nebude dovoleno a naopak.

## 4 Rational Team Concert Eclipse klient

Tato kapitola se zabývá prací s klientem RTC pro Eclipse. Předpokladem je základní znalost tohoto IDE a úspěšně nainstalovaný RTC klient pro Eclipse.

Před zahájením práce je nutné klienta připojit k úložišti v rámci JTS. To provedeme tak, že po spuštění klienta se přepneme do perspektivy Administrace Jazz a zde vybereme *Vytvořit připojení úložiště*. Připojení k úložišti potom nastavíme dle Ilustrace 5.



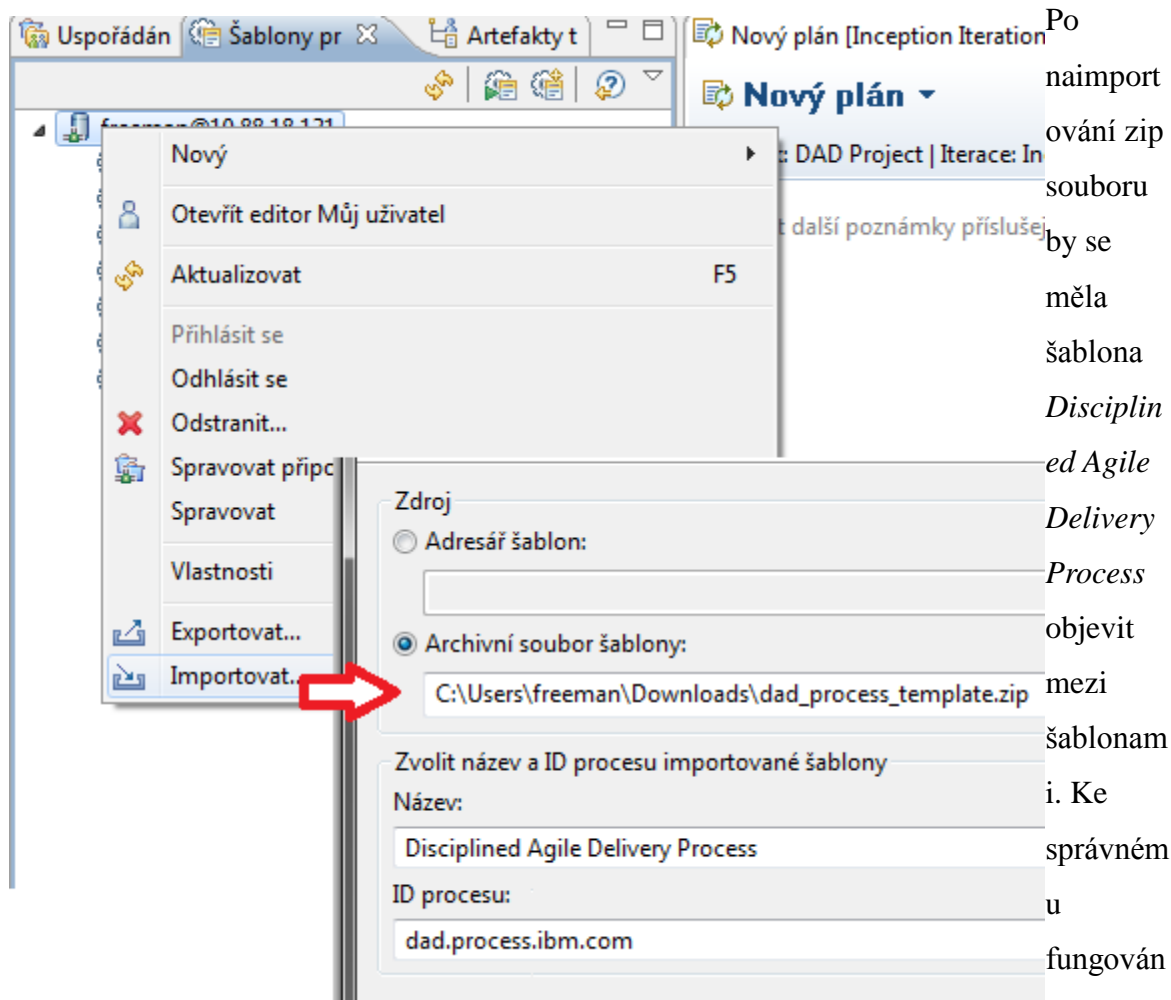
Ilustrace 5: Nastavení připojení k úložišti Jazz

### 4.1 Šablony procesů

Jak je uvedeno v odstavci na straně 4, tak jednou ze základních komponent JTS je proces, který je chápán jako základní pilíř celého projektu. Respektive projekt lze chápat jako proces. Pro usnadnění práce při vytváření projektů a zakládání jednotlivých částí projektů jsou v rámci JTS již obsaženy základní šablony procesů. Nicméně šablona pro DAD framework mezi nimi není.

### 4.2 Přidání šablony pro DAD

Ze stránek <https://www-304.ibm.com/support/docview.wss?uid=swg24028581> je možné stáhnout šablony procesů pro DAD a také pro Agility@Scale. Šablona se vždy skládá ze dvou částí. ZIP soubor se semotnou šablonou a WAR soubor. Nejdříve musíme nainstalovat šablonu ze ZIP souboru. To provedem opět v rámci perspektivy Administrace dle Ilustrace 6 níže.



Ilustrace 6: Importování šablony procesů

je však ještě potřeba na Tomcat server nahrát balíček *DAD.war*. To provedeme tak, že tento balíček nakopírujeme do složky `..\JazzTeamServer\server\tomcat\webapps`. Poté musíme znovu spustit JTS. Dojde tak k nahrání balíčku na server. Po dokončení těchto kroků je platforma Jazz připravena k použití společně s frameworkem DAD.

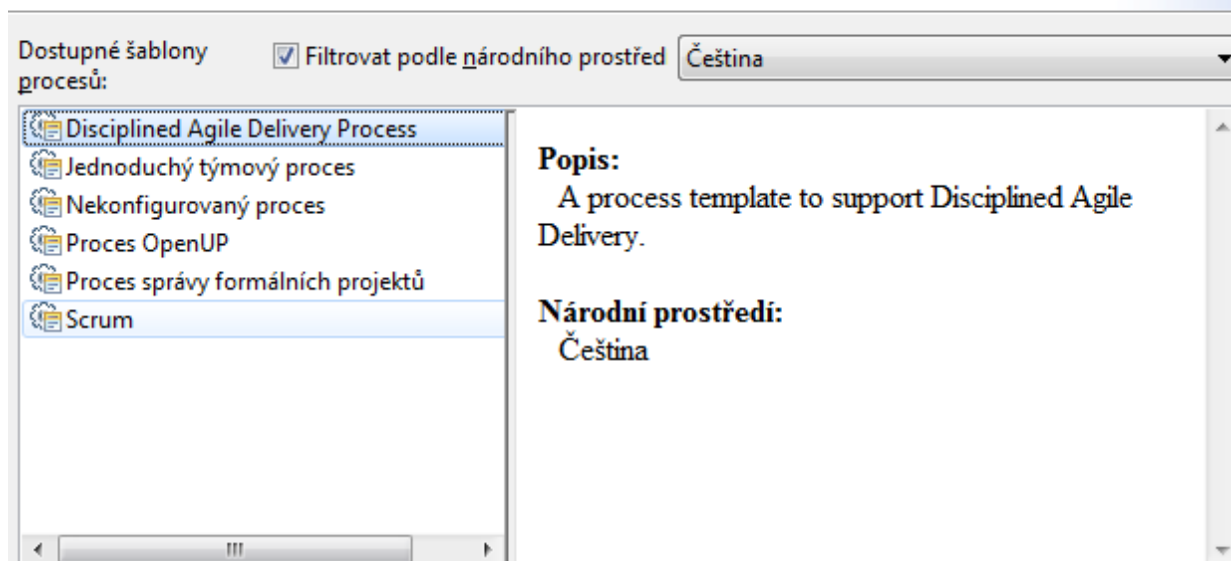
### 4.3 Využití šablon při vytváření projektu

Každý projekt se skládá z řady procesů. Aby při zakládání nového projektu nebylo nutné všechny nové procesy znovu definovat, tak lze využít šablon procesů. V předchozí kapitole jsme si importovali šablonu pro podporu DAD frameworku. Nicméně z Ilustrace 7 můžeme vidět, že kromě námi naimportované šablony pro DAD framework jsou již standardně podporovány procesy pro další typy projektů dle různých metodik. **Jednoduchý týmový proces** vytváří jednoduchou strukturu projektu v které je možné se rychle zorientovat. Členové týmu jsou oprávněni provádět úpravy všeho druhu. **Nekonfigurovaný proces** vytvoří prázdný projekt bez konfigurace. Je tedy

nutné všechny procesy vytvořit ručně. Je to vhodný typ šablony pouze v případě, že si chceme vytvořit vlastní strukturu projektu bez použití metodiky. **Proces OpenUP** je jedním z procesů vytvořených projektem Eclipse Process Framework (EPF)<sup>1</sup>. **Proces správy formálních projektů** slouží k podpoře tradičního procesu správy projektů, kde vývoj probíhá v po sobě následujících fázích: Požadavky, Návrh, Implementace a Testování.

### Zvolit šablonu procesu

Vyberte šablonu pro proces, který bude řídit oblast projektu.



Ilustrace 7: Výběr šablony procesů podle které bude oblast projektu vytvořena

K vytvoření nového projektu musíme na námi vytvořeném připojení k úložišti kliknout pravým tlačítkem a dát *Nový* → *Oblast projektu*. Budeme vyzváni k vyplnění názvu a popisu projektu a v dalším kroku se dostaneme k výběru šablony (viz Ilustrace 7), kde zvolíme *Disciplined Agile Delivery Process* a dáme *Vytvořit*.

## 4.4 Uživatelé a jejich role

Ke každému projektu je možné přiřadit libovolný počet uživatelů. Ke každému uživateli je potom možné přiřadit jakoukoliv roli v rámci týmu. Je samozřejmě také možné jednomu uživateli přiřadit více rolí. Role uživatele v projektu není nijak závislá na typu licenčního klíče, který je danému uživateli přiřazen v rámci vytváření uživatele. Předdefinované role v rámci projektu DAD jsou:

- **Tester:** Tato role je zodpovědná za hlavní aktivity v rámci testování. Tyto aktivity zahrnují identifikování, definování, implementace a zhodnocení potřebných testů včetně hlubší analýzy testů.

<sup>1</sup> **Eclipse Process Framework (EPF)** si klade za cíl vytvořit upravovatelný framework vývoje software, který bude sloužit jako ukázkový postup, jak vytvářet širokou škálu software různými způsoby vývoje.

- **Test Manager:** Tato role je zodpovědná za kompletní fázi testování včetně dodržení kvality testování. Plánování a řízení zdrojů potřebných pro testování a řešení případných problémů.
- **Team Lead:** Tato role je zaměřená na kompletní plánování projektu a řízení vztahu se Stakeholdery. Cílem této role je zajistit, aby tým dosáhl požadovaných výsledků
- **Stakeholder:** Tato role představuje osobu jejíž zájmy jsou projektem dotčeny. Typicky se jedná o zákazníka pro kterého je projekt realizován.
- **Developer:** Role zodpovědná za vývoj části systému včetně jeho designu a začlenění do architektury. Může se také zabývat vytvářením prototypů uživatelského rozhraní a implementací unit-testů.
- **Architect:** Vytváří kompletní architekturu systému a rozhoduje o technologické platformě
- **Analyst:** Cílem této role je komunikovat se Stakeholdery a koncovými uživateli a sbírat jejich požadavky a seznamovat se s jejich problémy. Ty poté pochopit a navrhnout jejich řešení včetně prioritizace požadavků.

## 4.5 Plánování projektu

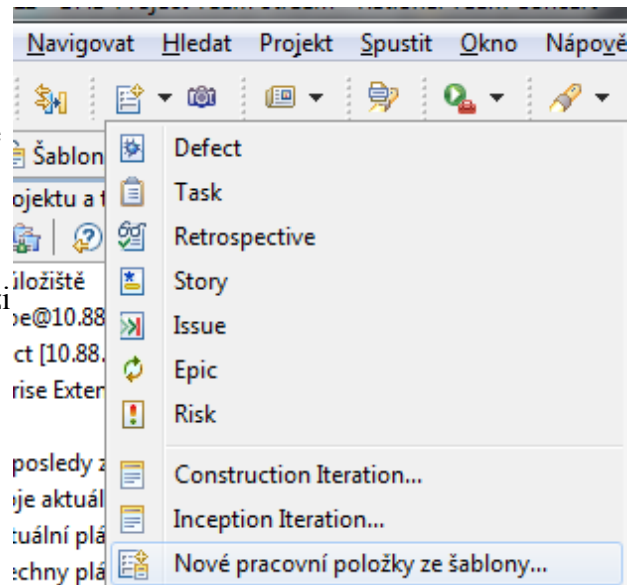
Mezi základní funkce, které poskytuje Rational Team Concert je plánování průběhu celého projektu. Pro snadnější plánování dle frameworku DAD je již při vytváření oblasti projektu vytvořena také základní struktura plánu. Protože je DAD framework zaměřen na celý životní cyklus software, tak je každý release rozdělen do tří fází:

- 1) **Inception:** Jedná se úvodní fázi, která přechází samotný vývoj software. V této fázi se typicky provádí sběr požadavků na software. Sestavování týmu a základní modelování a plánování organizace vývoje.
- 2) **Construction:** Fáze převzatá ze Scrumu. Jedná se tedy o cyklus samotného vývoje
- 3) **Transition:** Po tom, co je vývoj dokončen je v této fázi software připraven na nasezení do produkce.

V rámci každé iterace je možné vytvořit jeden a více plánů, kterým lze poté přiřazovat jednotlivé pracovní položky.

## 4.6 Pracovní položky

RTC podporuje různé druhy pracovních položek (viz Ilustrace 8). Navíc také podporuje přidání pracovních položek ze šablony. Díky této funkci je možné přidávat předdefinované úlohy do projektu. Pro DAD jsou definovány tři šablony. Každá šablona popisuje úlohy typické pro jednotlivou fázi životního cyklu vývoje software dle DAD. Díky těmto šablonám, tak nemusíme tyto základní pracovní položky vytvářet ručně. Co je však nutné provést je přiřazení jednotlivých pracovních položek vybraným uživatelům a nastavení odhadu jejich pracnosti.



Ilustrace 8: Přidání pracovní položky

## 4.7 Další funkce

Kromě plánování práce na projektu, lze Rational Team Concert využít pro mnoho dalších operací, které v průběhu životního vývoje software potřebujeme řešit. Typicky to je správa kódu a automatizované sestavování verzí. Navíc RTC umožňuje vytvářet celou řadu sestav. Opět jsou některé z nich předdefinované pro urychlení práce.

## 4.8 Zhodnocení

Produkt Rational Team Concert je velmi robustním řešením. Díky šablonám procesů umožňuje rychlé vytváření projektu a nadefinování jeho základních parametrů.

Z hlediska týmové spolupráce bych ho spíše doporučil pro práci týmů, které jsou geograficky rozdělené nebo je počet členů větší. Paradoxně právě snaha IBM vytvořit nástroj, který kompletně postihne celý životní cyklus vývoje software ukazuje na určité problémy, které s tím mohou potenciální uživatelé mít. To je důvod, proč bych v menším týmu (do 10 osob) volil spíše jednodušší online nástroje, které neposkytují takovou funkcionalitu a nepokrývají celý životní cyklus vývoje, ale díky svému jednoduchému rozhraní se s nimi velmi rychle naučí pracovat každý a náklady na údržbu jsou téměř nulové. Nástroj RTC je přesným opakem a minimálně klient pro Eclipse nepatří svou přehledností k příliš propracovaným a minimálně z počátku se bez nahlížení do návodu neobejdete. Navíc rozdíl v grafické podobě mezi klientem pro Eclipse a webovým rozhraním je velký. Je tak problém se v obou rozhraních orientovat. Dokonce je funkcionalita obou

rozhraní lehce jiná, což nelze hodnotit pozitivně.

Podpora frameworku DAD je v nástroji po doinstalování šablon velmi dobrá. Navíc díky nainstalování balíku *dad.war* (viz kapitola 4.2 na straně 11) si můžeme pomocí webového prohlížeče zobrazit stránku <https://adresaserveru:9443/dad/>, kde je kompletní přehled frameworku DAD v návaznosti na RTC.

## 5 Závěr

Platforma Jazz se neomezuje na podporu určité činnosti nebo metodiky, ale snaží se nabídnout podporu pro celou řadu metodik a všechny fáze životní hocyklu vývoje software. To přináší výhody, ale bohužel i obtíže.

Výhodou může být, že lze celou platformu rozšiřovat dle vlastních potřeb a že existuje mnoho šablon pro různé metodiky a frameworky včetně frameworku DAD. Na druhou stranu právě možnost vlastního rozšiřování a úprav může být pro nové uživatele problém, protože pro správně fungování a orientaci v platformě Jazz musí strávit mnoho hodin až dní studiem celé platformy, aby jí mohli plnohodnotně využít. Je tedy na každém týmu, zda jsou ochotni investovat do nové platformy, která koncentruje všechny potřebné nástroje pro celý životní cyklus vývoje do jedné sady nástrojů, nebo zda je pro ně pohodlnější využívat více jednoduchých účelově zaměřených nástrojů jako je Pivotal Tracker a další.

Platforma Jazz je pro framework DAD dobře přizpůsobená. Pokud se tým rozhodně pracovat dle frameworku DAD, tak je vhodně uvažovat i o nasazení platformy Jazz. Nicméně, jak jsem již zmiňoval, tak dle mého názoru to platí pouze pro týmy nad 10 osob, nebo týmy, kde je problematická komunikace a je tedy nutné mít vyšší stupeň formálního popisu procesů.



## 6 Zdroje

- [1] *IBM.com* [online]. 2011 [cit. 2011-12-05]. Přehled a cíle návrhu technologické platformy Jazz. Dostupné z WWW:  
<[http://publib.boulder.ibm.com/infocenter/rtc/v2r0m0/index.jsp?topic=/com.ibm.jazz.platform.doc/topics/c\\_jazz-architecture.html](http://publib.boulder.ibm.com/infocenter/rtc/v2r0m0/index.jsp?topic=/com.ibm.jazz.platform.doc/topics/c_jazz-architecture.html)>.
- [2] *IBM.com* [online]. 2011 [cit. 2011-12-05]. Architektura technologické platformy Jazz. Dostupné z WWW:  
<[http://publib.boulder.ibm.com/infocenter/rtc/v2r0m0/index.jsp?topic=/com.ibm.jazz.platform.doc/topics/c\\_jazz-architecture.html](http://publib.boulder.ibm.com/infocenter/rtc/v2r0m0/index.jsp?topic=/com.ibm.jazz.platform.doc/topics/c_jazz-architecture.html)>
- [3] *IBM.com* [online]. 2011 [cit. 2011-12-05]. Komponenty jádra Jazz. Dostupné z WWW:  
<[http://publib.boulder.ibm.com/infocenter/rtc/v2r0m0/index.jsp?topic=/com.ibm.jazz.platform.doc/topics/c\\_jazz-architecture.html](http://publib.boulder.ibm.com/infocenter/rtc/v2r0m0/index.jsp?topic=/com.ibm.jazz.platform.doc/topics/c_jazz-architecture.html)>
- [4] AMBLER, Scott . Disciplined Agile Delivery (DAD) Lifecycle . *DeveloperWorks* [online]. 2010-07-15, [cit. 2011-11-03]. Dostupný z WWW:  
<[https://www.ibm.com/developerworks/mydeveloperworks/blogs/ambler/entry/disciplined\\_agile\\_delivery\\_dad\\_lifecycle14?lang=en\\_us](https://www.ibm.com/developerworks/mydeveloperworks/blogs/ambler/entry/disciplined_agile_delivery_dad_lifecycle14?lang=en_us)>
- [5] LINES, Mark; AMBLER, Scott. *Disciplined Agile Delivery : A Practitioner's Guide to Agile Software Delivery in the Enterprise*. [s.l.] : IBM Press, 2012. 400 s. ISBN 978-0-13-281013-5
- [6] LINES, Mark. Why do we need the DAD framework?. *Disciplinedagiledelivery.com* [online]. 2011-10-06, [cit. 2011-11-03]. Dostupný z WWW:  
<<http://www.disciplinedagiledelivery.com/>>
- [7] AMBLER, Scott; LINES, Mark. *Disciplined Agile Delivery : An introduction*. Somers, NY 10589 : IBM Corporation, 2011. 20 s. Dostupné z WWW:  
<<http://dev0.slideshare.com/ibmrational/raw14261-usen>>. RAW14261-USEN-00
- [8] AMBLER, Scott. *The Agile Scaling Model (ASM) : Adapting Agile Methods for Complex Environments*. Somers, NY 10589 : IBM Corporation, 2009. 35 s. Dostupné z WWW:  
<<ftp://ftp.software.ibm.com/common/ssi/sa/wh/n/raw14204usen/RAW14204USEN.PDF>>. RAW14204-USEN-00