

Ročníkový projekt

Jaroslav Žáček

jaroslav.zacek@osu.cz

<http://www1.osu.cz/~zacek/infos1/>

Cíle předmětů

- Vytvoření fungující aplikace, která splňuje definované požadavky
- Vyzkoušet si celý životní cyklus projektu - specifikace zadání, formování týmu, konfigurace prostředí, vytvoření prototypu architektury, kódování, integrace a beta release
- Studenti si rozdělí jednotlivé týmové role, důraz je ovšem kladem na cross-functional týmy (každý v týmu je zastupitelný)
- Vývoj se bude řídit agilními metodikami - iterativně-inkrementálním vývojem, využívat se bude procesní framework OpenUP
- Pomoc při návrhu architektury
 - Java, Hibernate, Spring, Angular, React - Žáček
 - .NET, Entity Framework, ASP, .NET Core - Vajgl

ROPR1

- V ROPR1 realizace fáze Inception a Elaboration podle OpenUP
- Kritické je pochopení iterativního vývoje. Nezbytný předpoklad pro splnění je paralelní absolvování teorie - INFS1.
- Kritická je kooperace všech členů týmu.
- Příprava prostředí (IDE, repository) a nastavení nutných podpůrných procesů (testovací proces, modelovací nástroje)

Výstupy fáze Inception

- Definice a sestavení týmu - vytvoření přístupů pro GIT
- Definice rysů aplikace a identifikace klíčových požadavků
- Identifikování aktoři, nastíněné Use Case
- Prostředí a nástroje jsou vybrány, nakonfigurovány a připraveny k použití

Výstupy fáze Inception

- Dokumenty
 - Vize projektu
 - Risk list
 - Projektový plán
 - Návrh architektury
 - Detailní iterační plán pro 1. iteraci
- Jednoduchý prototyp - “Hello world” (pokud neznáte danou technologii)

Osobní časovač: Vize

Problém (co řešíme, co nefunguje, co nás trápí)

Gary není schopný sbírat konsistentní časové údaje od vývojářů reprezentující čas strávený na různých projektech. Není tedy možné monitorovat a porovnat postup oproti plánům, fakturovat řádné časy, platit externí spolupracovníky a samozřejmě také na základě těchto dat dělat věrné odhady dalších iterací.

Řešení (jak tento problém popsany v předchozím kroku budeme řešit)

Osobní časovač (OČ) měří čas strávený na projektech, shromažďuje a ukládá tato data pro pozdější zobrazení (stylem Post-it poznámek), aby mohl Gary systematicky organizovat a hodnotit projekty, sledovat aktuální postup prací a ty porovnávat s plánovanými odhady pro jednotlivé projekty

Zainteresované strany (anglicky *stakeholders*)

- jednotlivý vývojáři
- pracovníci administrativy
- projektový manažer

Use Cases (základní obecné funkčnosti)

- Změř čas aktivity
- Sesbírej týdenní data
- Sluč / konsoliduj data pro každý projekt
- Nastav nástroj a databázi pro projekt

Vize - příklad

Rozbor zadání

Zadání: Zadáním je plánovací kalendář do firmy, který bude sloužit pro sledování a spravování zdrojů ve firmě.

Problém: Nedostačující správa zdrojů a úkolů ve firmě.

Řešení: Implementace plánovacího kalendáře pro zefektivnění práce se zdroji.

Dopad: Zefektivnění zprávy zdrojů a lidské práce. Přehled o využití jednotlivých zdrojů a vytížení pracovníků.

Výhody: Přehled o využívání zdrojů, úkolech, výkonnosti pracovníků.

Cílová skupina

Cílový zákazník: mistr/předák

Plánovací kalendář: je plánovací software

Důvody pro koupi: lepší využitelnost zdrojů a lidské práce a tím zefektivnění podnikových procesů.

Alternativy: MS Outlook, Google Calendar, Plánovací kalendář 2012, My life Organized

Náš produkt: Produkt je vytvořen přímo pro potřeby firmy, neobsahuje zbytečně mnoho nepoužívaných funkcí, které rozptylují při práci.

Stakeholder Summary

Běžní uživatelé: řadoví zaměstnanci firmy

Nadřízení: vedení firmy

Vývojový tým: [blurred text]

Uživatelské prostředí

Na programu bude pracovat 5 vývojářů. Vývojový cyklus je jeden rok a výsledkem bude webová aplikace splňující požadavky zadavatele.

Vize - příklad

Informační systém: Vize

Problém:

Stávající stravovací systém je neefektivní a složitý. Neelektronický způsob objednávky stravy je nekomfortní a vyžaduje zbytečný personál. Není možnost dále rozvíjet současný postup, jak si objednat stravu: například, možnost změny objednávky přes internet. Dále není šance zjednodušit tvorbu jídelníčku z pohledu jídelny.

Řešení:

Vytvoření informačního systému, v němž budou shromažďovány informace o jídelníčku, včetně přihlašování strážníků a objednávání stravy. V systému bude také zakomponován systém burzy odhlášených objednávek.

Zainteresované strany:

- personál jídelny
- strážníci

Use Cases:

- UC0 -- správa jídelníčku
- UC1 -- evidence strážníků a plateb
- UC2 -- správa objednávek stravy

Risk list

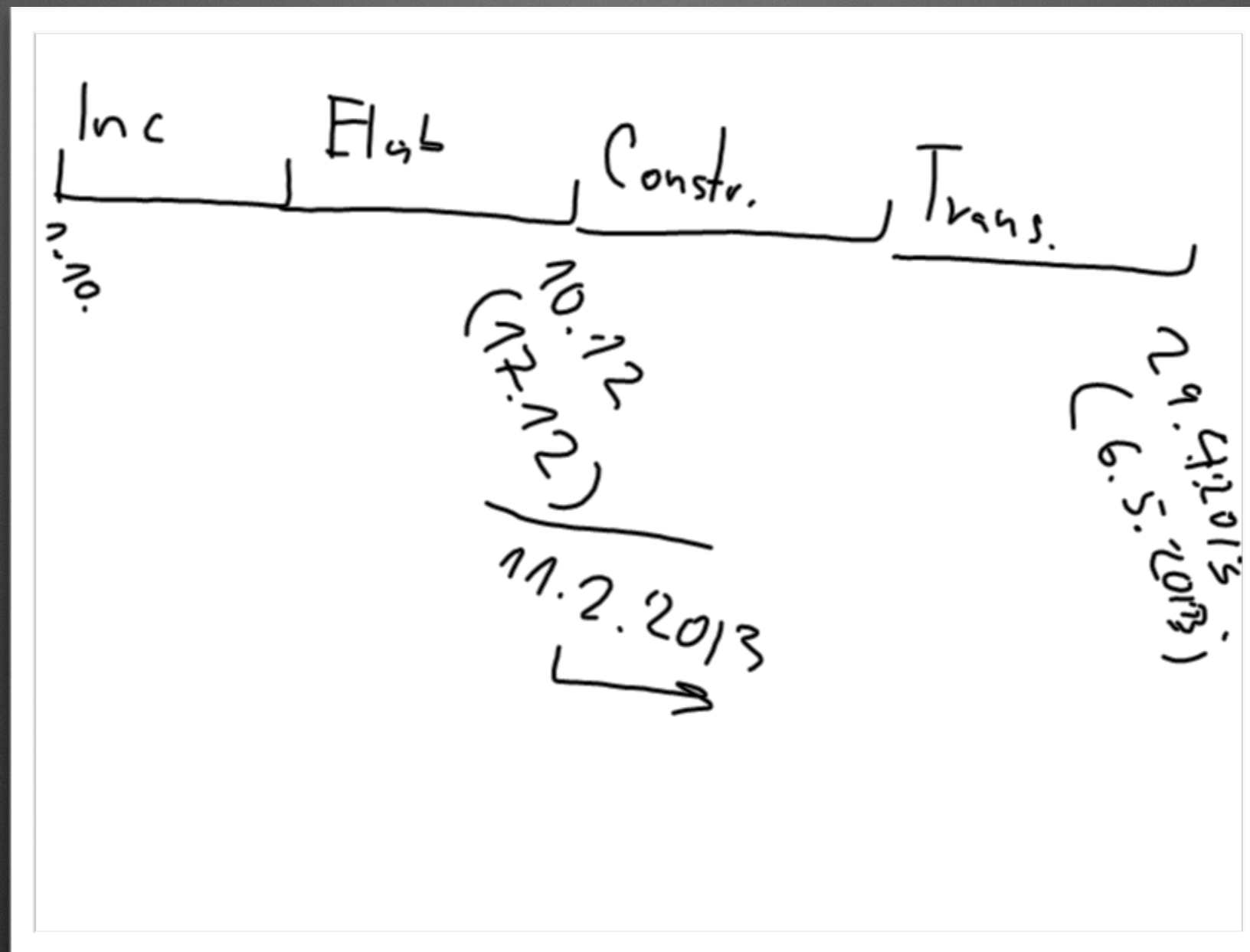
Název rizika	Popis	Strategie
Integrace se systémem X	Není zřejmé, jak integrovat naši aplikaci s historickým systémem X.	Strategie A: Snížení rizika Vytvoříme „tiger team“ sestávající z několika málo zkušených vývojářů, jejichž cílem je ověřit na integraci prototypu naší aplikace se systémem X. Integrace a její způsob může být velmi primitivní, cílem je však ověřit schopnost propojení se systémem X. Navrhujte, implementujte a testujte dané use casey v průběhu projektu, aby bylo toto propojení neustále validováno. Strategie B: Předejití rizika Změna náplně projektu tak, že daná integrace nebude obsahem, předmětem.
Nedostatečná zkušenost s JEE	Je riziko, že vytvoříme druhořadé, méně technicky kvalitní řešení v důvodu nezkušenosti s platformou JEE.	Strategie A: Snížení rizika Poslat několik vývojářů na školení Microsoft .NET a nalezení rezervy v rozpočtu na mentora 2 dny v týdnu po dobu prvních dvou měsíců trvání projektu. Nábor nového člena týmu se znalostí .NET platformy. Vytvoření prototypu pro ověření použití .NET platformy.

Risk list - příklad týmu z ROPR

ID	Fakt	Popis rizika	Priorita	Příznaky rizika	Plán na zmírnění rizik	Vlastník	Míra odstranění rizika
R1	Žádný člen teamu nemá zkušenosti s prací ve skupině	Neshoda v komunikaci	3	Rozdělení dílčích rolí v teamu. Špatná synchronizace práce a komunikace mezi členy.	Schůze každé úterý od 12:30 - 18:00 a řešení dané problematiky. Párové programování.	Program	
R2	Nedostatečná znalost použitých technologií	Kompatibilita technologií	4	Neschopnost pokračovat ve vývoji.	Testování na funkčním prototypu.	Program	
R3	Časová náročnost projektu	Nedodržení termínu odevzdání projektu	2	Nedodržování termínů iterací.	Využití plánovacího softwaru.	Program	
R4	Žádná garance permanentního uložení dat	Ztráta dat	1	Ztráta dat	Týdenní záloha dat na externím uložišti.	Program	
R5	Nedostatek financí	Nenalezení volně dostupných technologií	5	Neschopnost pokračovat ve vývoji.	Využití studentských a zkušebních verzí.	Program	

Risk List 1: Priorita 1 až 5 (5 nejvyšší priorita)

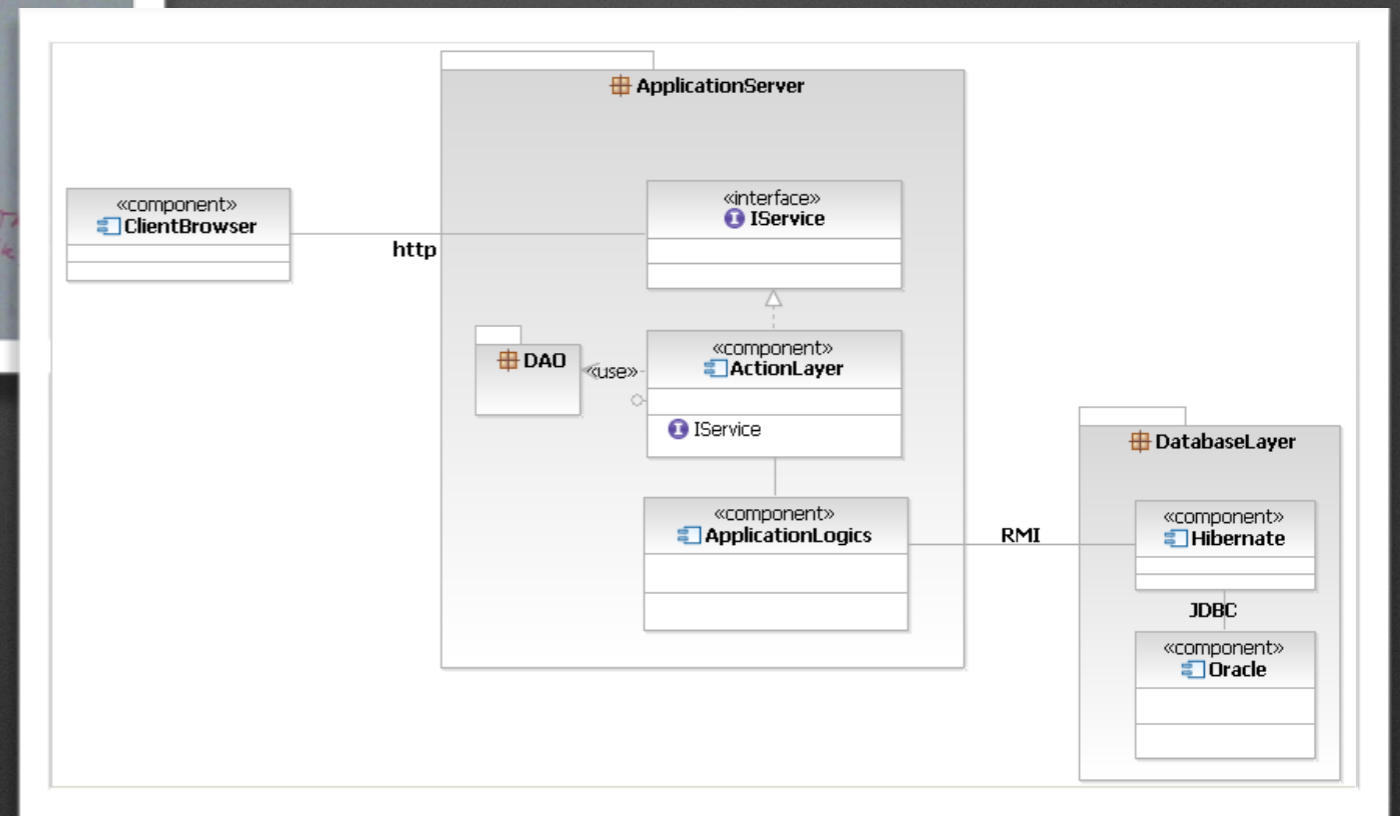
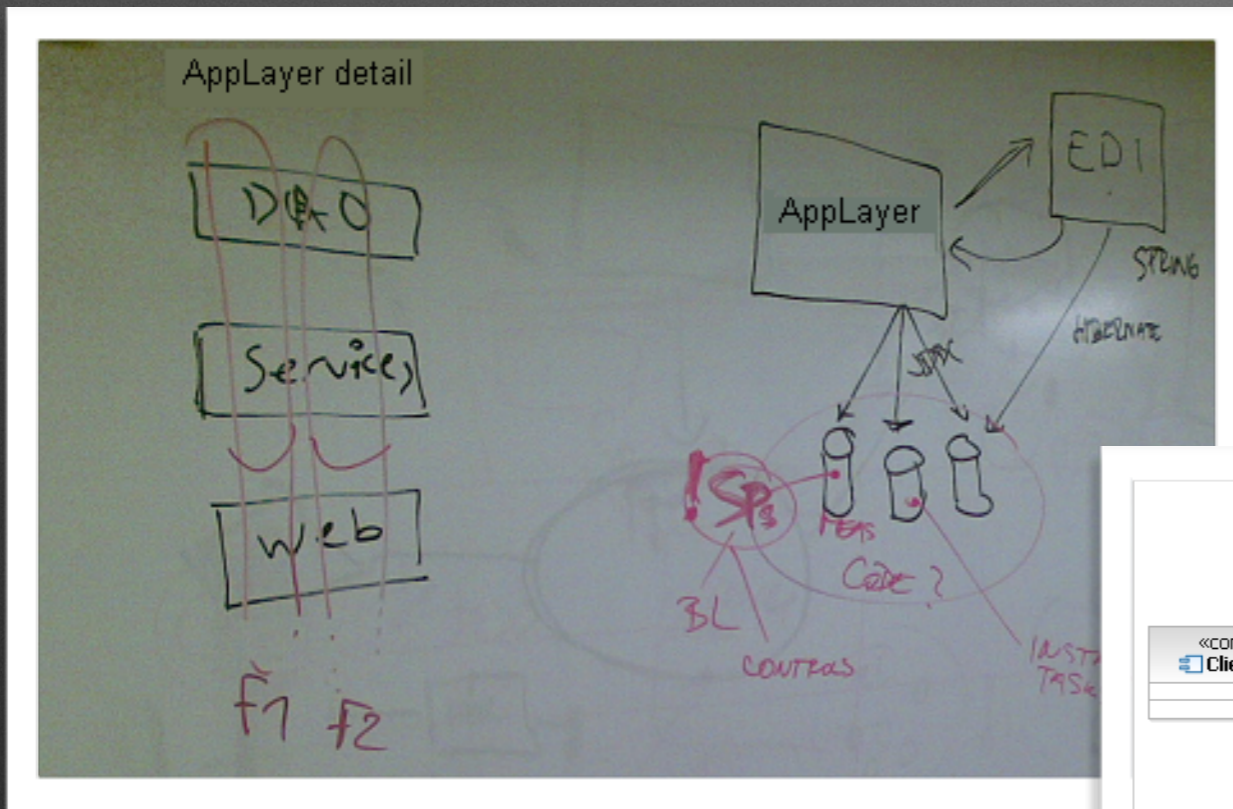
Projektový plán

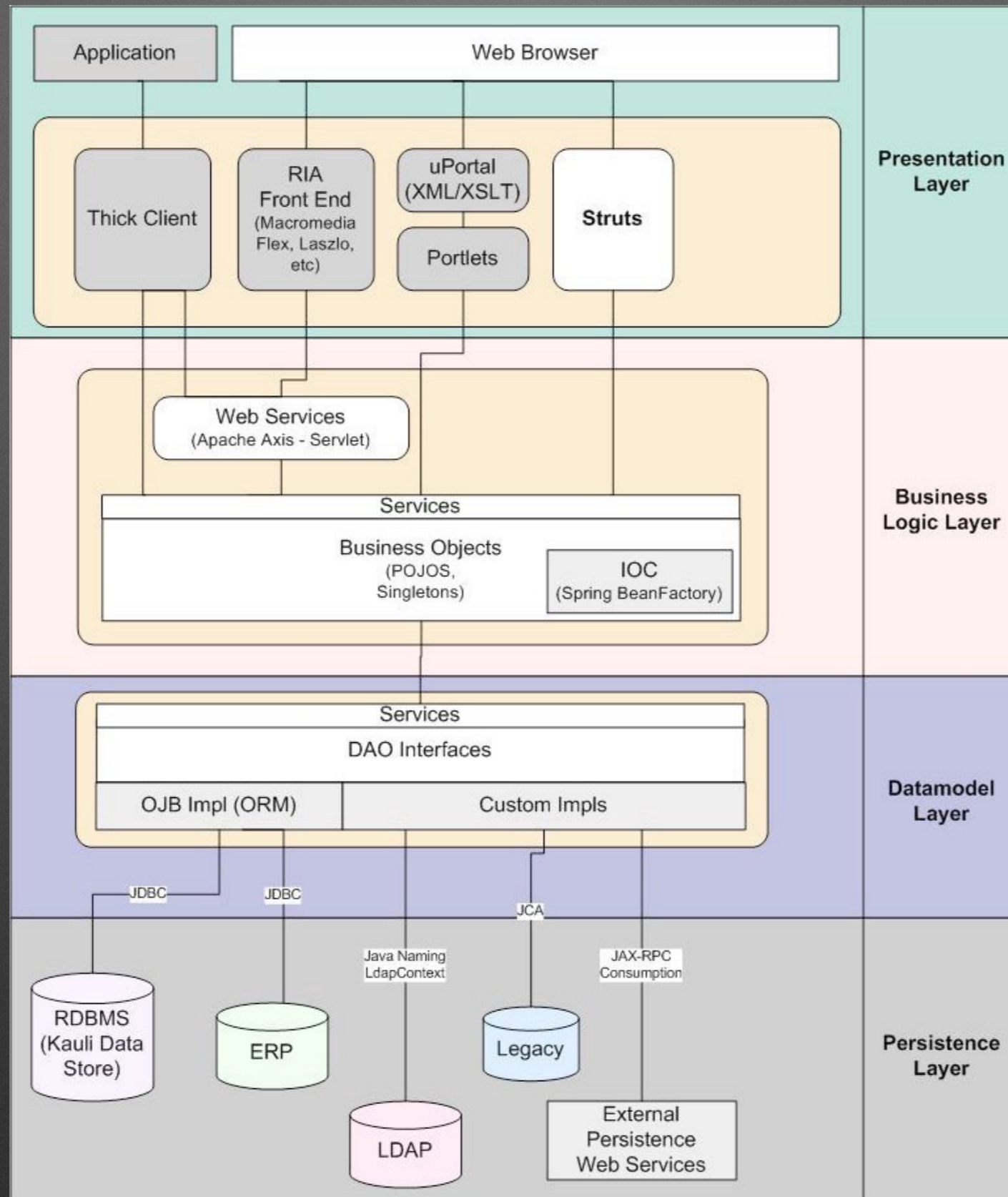


Projektový plán - příklad

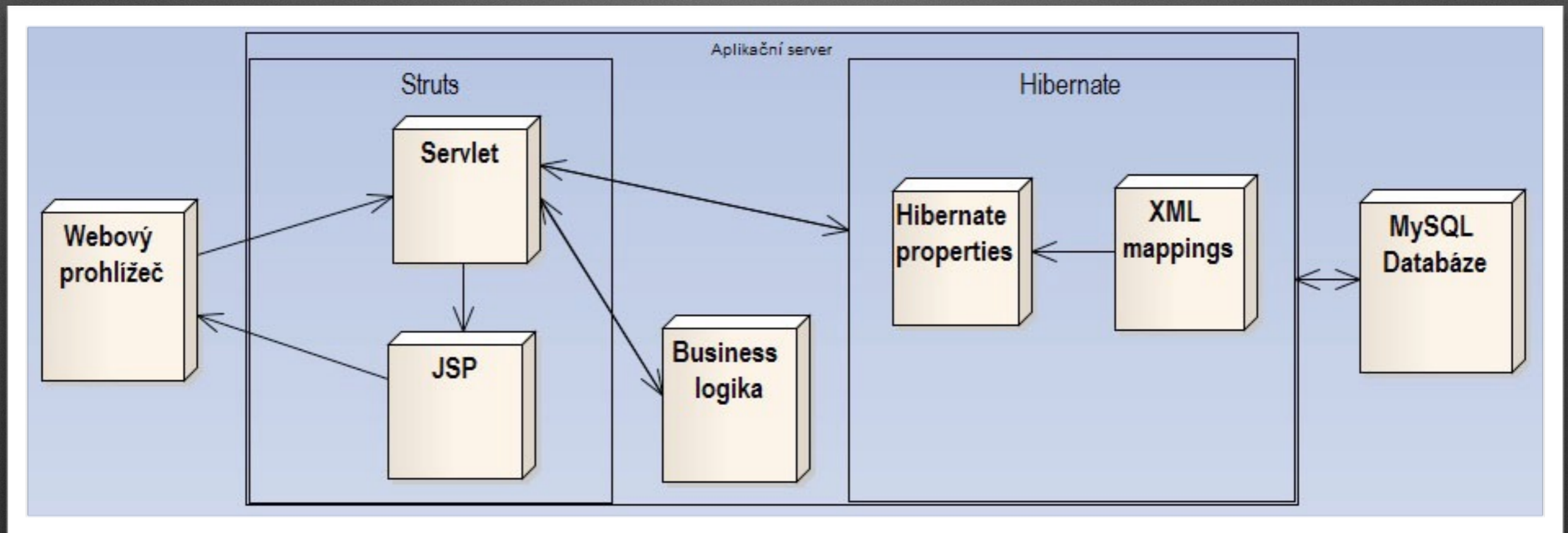
Fáze	Iterace	Primární úkoly	Datum od - do
Inception	I0	Vize, definice rozsahu systému, jeho hranic. Identifikace nejkritičtějších Use Casů. Návrh architektury. Srozumnění s plánem a riziky Definice/úprava procesu, výběr a nastavení nástrojů. Interační plán	26.09.2011 - 10.10.2011
	I1	Vytvoření prototypu	10.10.2011 - 17.10.2011
		LOM	17.10.2011
Elaboration	E1	UC1: [BF1] Prohlížení úkolů (zahrnuje Ganttův graf)	
		UC2: [BF] Prohlížení zdrojů	
	E2	UC1: [BF2] Vytvoření úkolu	
		UC3: [AF1] Identifikace uživatele (přihlášení, odhlášení)	
	E3	Rezerva	
		LCA	5.12.2011
Construction	C1	odstranění rizik	Letní semestr
	C2		
		IOP	
Transition	T1		
		PRM	4.30.12

Architektura



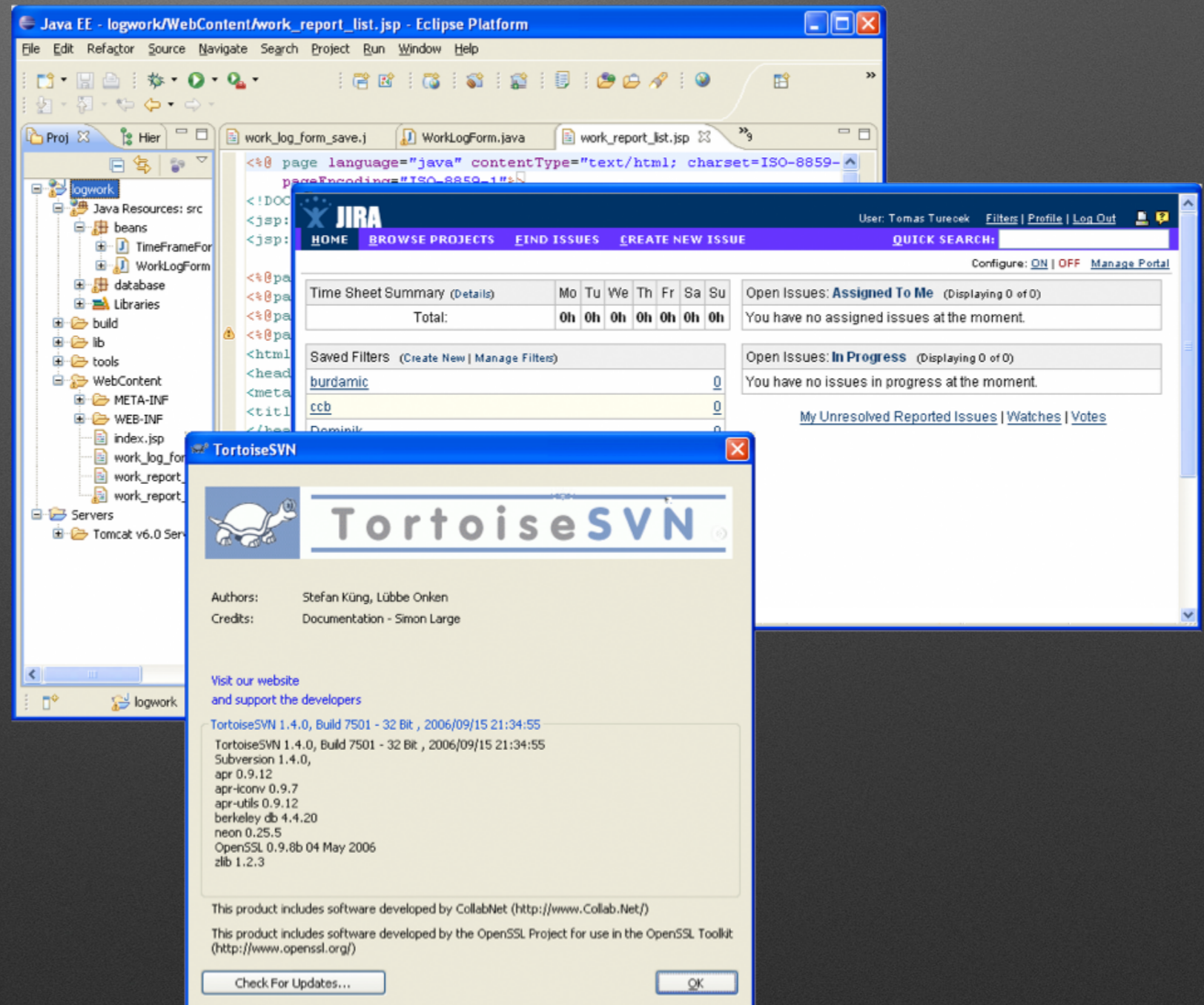


Architektura - jiný příklad



Prostředí

- IDE (Idea, VS)
- GitLab
- Týmová Wiki?
- xUnit



Prostředí - příklad

- Repository - používáte školní GitLab, v praxi jsou alternativy SourceForge, Bitbucket, GitHub Enterprise...
- IDE - IntelliJ/Eclipse/NetBeans a Visual Studio + napojení na repository
- Testy řízený vývoj - nejprve test, pak kód
- Automatický proces sestavení a spouštění testů - využití Ant/Maven/Gradle/Gulp a xUnit

Iterační plán

- Plán iterace E1
 - Začátek iterace (plánovací meeting dané iterace): 3.10.2017
 - Konec iterace (demo, assessment): 10.10.2017

- Cíle iterace:

- Implementace UC1 [BF].
- Odstranění rizika R1 a R2.

- Evaluační kritéria:

- 60 % kódu pokryto unit testy.
- 100 % unit testů prošlo.
- 70 % implementovaných funkčních testů prošlo.
- Sníženo riziko R1.
- Bylo předvedeno demo zákazníkovi.
- Zákazník demo akceptoval.

Úkoly:

Název / Popis	Priorita	Odhad (body)	Přiřazeno	Odhad (hodin)
Instalace build mechanismu (Ant)	2		Johan	70
Analýza scénáře UC1 (BF)	2	8	Lisa	
			Lisa, Ann,	
Návrh scénáře a implementace		8	Johan	12
Implementace a testy server části			Ann	14
Implementace a testy klient části			Johan	28
Sestavení dema pro assessment	3	5	Johan	18
Vytvoření uživatelské dokumentace	2	5	Lisa	65
Vytvoření install manuálu	2	1	Lisa	5
Vytvoření release notes	2	1	Johan	4
Vytvoření online help stránek	3	2	Ann	22

Iterační plán - příklad

Iteration #10 plan

- Start of iteration: 26. 09. 2011
 - End of iteration: 10. 10. 2011
-

Main Goal

- Vytvoření vize
 - Vytvoření risk-listu
 - Vytvoření projektového plánu
 - Vytvoření UC diagramu
 - Výběr a nastavení nástrojů
-

Evaluation criteria

- Vize schválena všemi členy týmu
- Vize schválena zákazníkem
- Risk-list chválen všemi členy týmu
- UC diagram schválen všemi členy týmu
- Projektový plán schválen všemi členy týmu
- Projektový plán schválen zákazníkem
- Použité nástroje schváleny všemi členy týmu

Iterační plán - příklad

Iteration backlog (work item list)

Název / Popis	Priorita	Přiřazeno	Odhad (hodin)
Vytvoření projektového plánu	2	Karel	8
Vytvoření risk-listu	2	Tomáš	8
Vytvoření vize	2	Libor	8
Vytvoření UC diagramu	2	Jirka	8
Výběr a nastavení nástrojů	2	Martin	8

1 - nejvyšší priorita, 5 - nejmenší priorita

Iteration #10 assessment

Demo result

Kritérium	Splněno / Nesplněno
Vize schválena všemi členy týmu	Splněno
Vize schválena zákazníkem	Splněno
Risk-list chválen všemi členy týmu	Splněno
UC diagram schválen všemi členy týmu	Splněno
Projektový plán schválen všemi členy týmu	Splněno
Projektový plán schválen zákazníkem	Splněno
Použité nástroje schváleny všemi členy týmu	Splněno

Nástroje pro podporu vývoje

- Vše bude zaznamenáno v Jira a GitLab
- Podmínka pro udělení zápočtu je dokumentovaná aktivita
 - Počet commitů a vlastní obsah commitů
 - Přidělování práce
- Nástroje jsou dostupné pouze ze sítě OU:
 - Jira <http://172.16.101.101:8080/>
 - GitLab <http://195.113.102.176:8085>

Organizace práce

JIRA Dashboards ▾ Projects ▾ Issues ▾ Boards ▾ **Create** Search 🔍 🔔 ⓘ ⚙️ 👤

TP board Board ▾ ⌆

Backlog

🔍 QUICK FILTERS: Only My Issues Recently Updated

VERSIONS EPICS

Iterace 2 2 issues Start Sprint ⋮

- TP-39 Navrh
- TP-49 Dokumentacia

+ Create issue

2 issues Estimate 0

Backlog 22 issues Create Sprint

- TP-5 As a team, I'd like to commit to a set of stories to be completed in a sprint (or iteration) >> Click "Create Sprint" then drag the foot 1
- TP-34 analýza požadavků
- TP-40 Implementace
- TP-4 As a team, I'd like to estimate the effort of a story in Story Points so we can understand the work remaining >> Try Version 3.0 5
- TP-44 Udělat vše.
- TP-31 otestovat aplikaci na 10 probandech
- TP-6 As a scrum master, I'd like to break stories down into tasks we can track during the sprint >> Try creating a task by clicking the Su 1
- TP-30 Otestovat aplikaci

Testing project / TP-43 ⋮ ×

My new awesome super user story

Estimate: Unestimated

Details

Status: **TO DO** (View Workflow)

Priority: ↑ Medium

Component/s: None

Labels: None

Affects Version/s: None

Fix Version/s: None

Epic Link: None

People

Reporter: Jaroslav Žáček

Assignee: Unassigned [Assign to me](#)

Dates

Created: Monday 11:36 AM

Updated: Tuesday 5:01 PM

Description

Kde najdu co mám dělat?

JIRA Dashboards ▾ Projects ▾ Issues ▾ Boards ▾ **Create** Search 🔍 🔔 ? ⚙️ 👤

Testing project / TP-44
Udělat vše.

Edit Comment Assign More ▾ To Do In Progress Workflow ▾ Admin ▾ Export ▾

Details

Type:	Story	Status:	IN PROGRESS (View Workflow)
Priority:	Medium	Resolution:	Unresolved
Affects Version/s:	None	Fix Version/s:	None
Labels:	None		
Sprint:	My testing sprint		

Description

Jako uživatel chci editovat svůj profil, abych mohl změnit následující údaje:

- Jméno a příjmení
- Věk
- Kontaktní telefon

AC:

- 100% unit testů prošlo
- Kód prošel core review
- Aplikace při každé změně profilu odesílá testovací e-mail.
- REST API otestováno skriptem z Postmanu

Attachments

Drop files to attach, or [browse](#).

Activity

People

Assignee: Jaroslav Žáček
Reporter: Tomáš David
Votes: 0 [Vote for this issue](#)
Watchers: 1 [Start watching this issue](#)

Dates

Created: 6 days ago
Updated: Just now

Agile

Completed Sprint: [My testing sprint ended 24/Sep/18](#)
[View on Board](#)

HipChat discussions

Do you want to discuss this issue? Connect to HipChat.

[Connect](#) [Dismiss](#)

Koncept iterací/sprintů

JIRA Dashboards ▾ Projects ▾ Issues ▾ Boards ▾ Create

Search 🔍 🔔 ? ⚙️ 👤

TP board 🕒 5 days remaining Complete Sprint Board ▾ ⌆

Iterace

QUICK FILTERS: Only My Issues Recently Updated

To Do	In Progress	Ready for QA	In Testing	Verify by Marek	Done
<p>✓ TP-46 DONE 1 sub-task Udělat celý projekt</p>					<p>TP-47 ↑ analyza požadavok1</p>
<p>✓ TP-37 DONE 1 sub-task Analyza požadavku</p>					<p>TP-48 ↑ zaver analyzy</p>
<p>✓ TP-34 ↑ analýza požadavků</p>	<p>TP-40 ↑ Implementace</p>		<p>TP-5 ↑ As a team, I'd like to commit to a set of stories to be</p>		<p>TP-45 ↑ Vymyslet co dělat.</p>
<p>TP-4 ↑ As a team, I'd like to estimate the effort of a story in Story</p>					

Jak dodat výsledek?

Releases

Manage Versions

QUICK FILTERS: Released Unreleased Archived

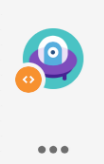
Merge versions

Version name Start date (optional) Release date (optional) Description (optional) Add

Version	Status	Progress	Start date	Release date	Description	Actions
Version 3.0	UNRELEASED	<div style="width: 100%;"></div>				⋮
Version 2.0	UNRELEASED	<div style="width: 50%;"></div>		28/Feb/18		⋮



Kde najdu minulé sprinty?

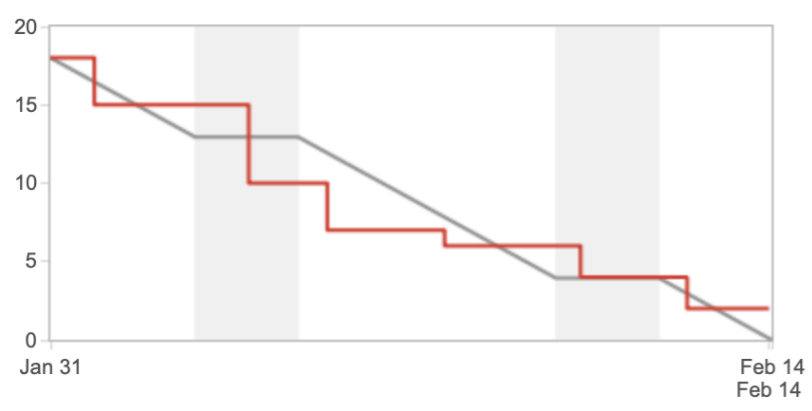


TP board
Sprint Report [Switch report ▾](#)

Board ▾ ⏴



Sample Sprint 1 ▾
Closed Sprint 31/Jan/18 4:32 AM - 14/Feb/18 3:12 AM [Linked pages](#)



Status Report

Completed Issues

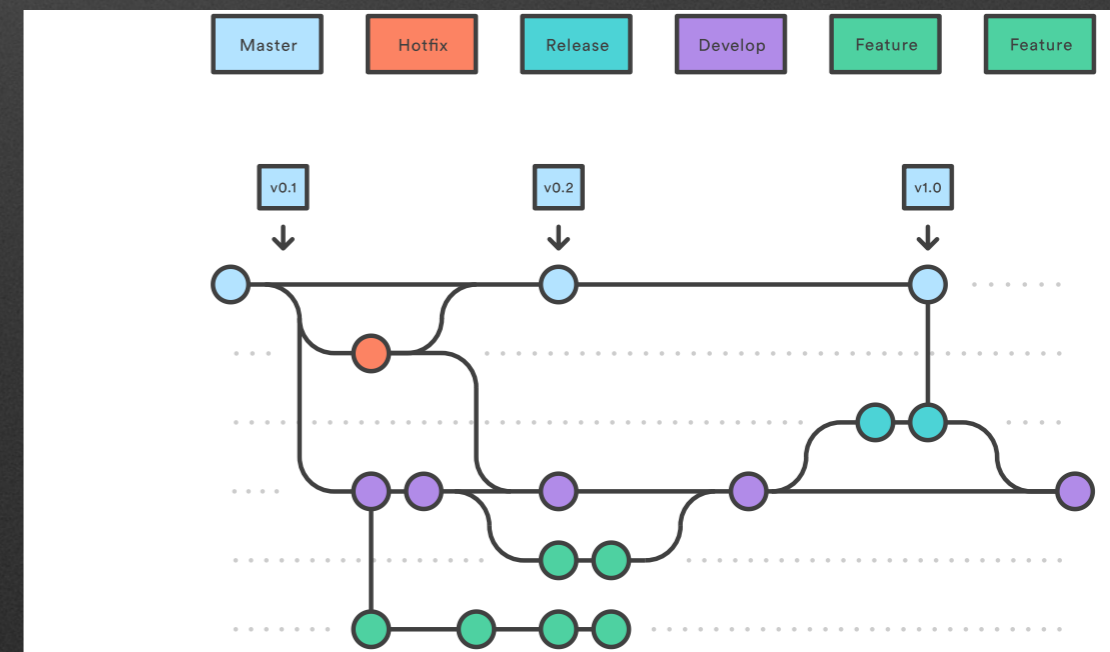
[View in Issue navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (16)
TP-18	As a user, I'd like a historical story to show in reports	Story	↑ Medium	DONE	3
TP-19	As a user, I'd like a historical story to show in reports	Story	↑ Medium	DONE	5
TP-20	As a user, I'd like a historical story to show in reports	Story	↑ Medium	DONE	3
TP-21	As a user, I'd like a historical story to show in reports	Story	↑ Medium	DONE	1
TP-22	As a user, I'd like a historical story to show in reports	Story	↑ Medium	DONE	2



Verzovací systém

- Pro nastartování můžete využít toto video: <https://youtu.be/47uih9Tp6H8>
- Prakticky si to můžete zkusit zde: <https://learngitbranching.js.org>
- Během vývoje musíte používat vlastní větve (branches)
- Postupně do procesu zapojíte pull requesty
- ...a budete používat GitFlow



Projekt v GitHubu

Engin1980 / ropr-snadNeRUPnem Private

Watch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights

No description, website, or topics provided.

501 commits 16 branches 0 releases 4 contributors

Branch: master New pull request Create new file Upload files Find File Clone or download

Yahkem Remove overlay tab Latest commit 23bba2b on 8 Aug

snadNeRUPnem.Tests	change link validity to 6 hours	6 months ago
snadNeRUPnem	Remove overlay tab	2 months ago
.gitattributes	Add .gitignore and .gitattributes.	last year
.gitignore	implement basic project sharing	5 months ago
README.md	Update README.md - Verze+Hosting info	2 months ago
snadNeRUPnem.sln	add Moq and password test	10 months ago

README.md

ropr-snadNeRUPnem

Projekt skupiny snadNeRUPnem pro předmět RPRO1/RPRO2. Členové skupiny:

- Richard Kubný
- Jan Mojžíš
- Martin Rucký

Změna API klíče pro Google maps

API klíč je v souboru `snadNeRUPnem/ClientApp/src/index.html` na 26. řádku. Klíč se dosadí do `src` atributu tagu `<script>` ve tvaru:

```
https://maps.googleapis.com/maps/api/js?v=3.35&key=<API_KEY>&libraries=geometry
```

Historie v GitHubu

Engin1980 / ropr-snadNeRUPnem Private

Watch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights

Branch: master

Commits on Aug 8, 2019

- Remove overlay tab
Yahkem committed on 8 Aug
23bba2b

Commits on Jul 30, 2019

- Update README.md - Verze+Hosting info
Yahkem committed on 30 Jul
Verified 95afa9b

Commits on Jul 15, 2019

- Create README.md
Yahkem committed on 15 Jul
Verified 3cdf05

Commits on Jun 3, 2019

- Merge pull request #21 from Engin1980/feat/overlays
mojzisy committed on 3 Jun
Verified faa7c8c
- updated overlays
mojzisy committed on 3 Jun
2aec5e4
- Merge pull request #20 from Engin1980/feat/nerupnem-113
mojzisy committed on 3 Jun
Verified 26a1435

Commits on May 7, 2019

- remove old TODO comments
Yahkem committed on 7 May
87e54bd

Commits on May 6, 2019

- příprava na práci s overlays
MRucky committed on 6 May
8446034
- aktualizace overlay creator komponenty
MRucky committed on 6 May
f6a88d1

Výstupy fáze Elaboration

- Ustálený vývojový proces a prostředí (funguje Repository, umíte ovládat IDE, funguje Wiki a všichni s ní umí pracovat, fungují testy)
- Jsou popsány kritické scénáře
- Dokumenty jsou umístěny ve společném repozitáři či Wiki
- Architektura je stabilní, otestovaná a spustitelná - lze na ní projekt realizovat celý
- Všechny významné rizika jsou odstraněny (minimalizovány)

Ověření

- V závěru semestru bude práce ohodnocena vyučujícími a ostatními týmy, zaměříme se hlavně na:
 - správně pochopené potřeby zákazníka a vybrané klíčové požadavky
 - architektura je stabilní oproti testům i klíčovým požadavkům zákazníka
 - budete demonstrovat základní UC v aplikaci
 - ukážete výsledky testů
 - prezentujete model architektury
- kontrola, jestli je risk list pravidelně aktualizován, doplňován a jestli jsou podle něj plánovány iterace

Naše očekávání

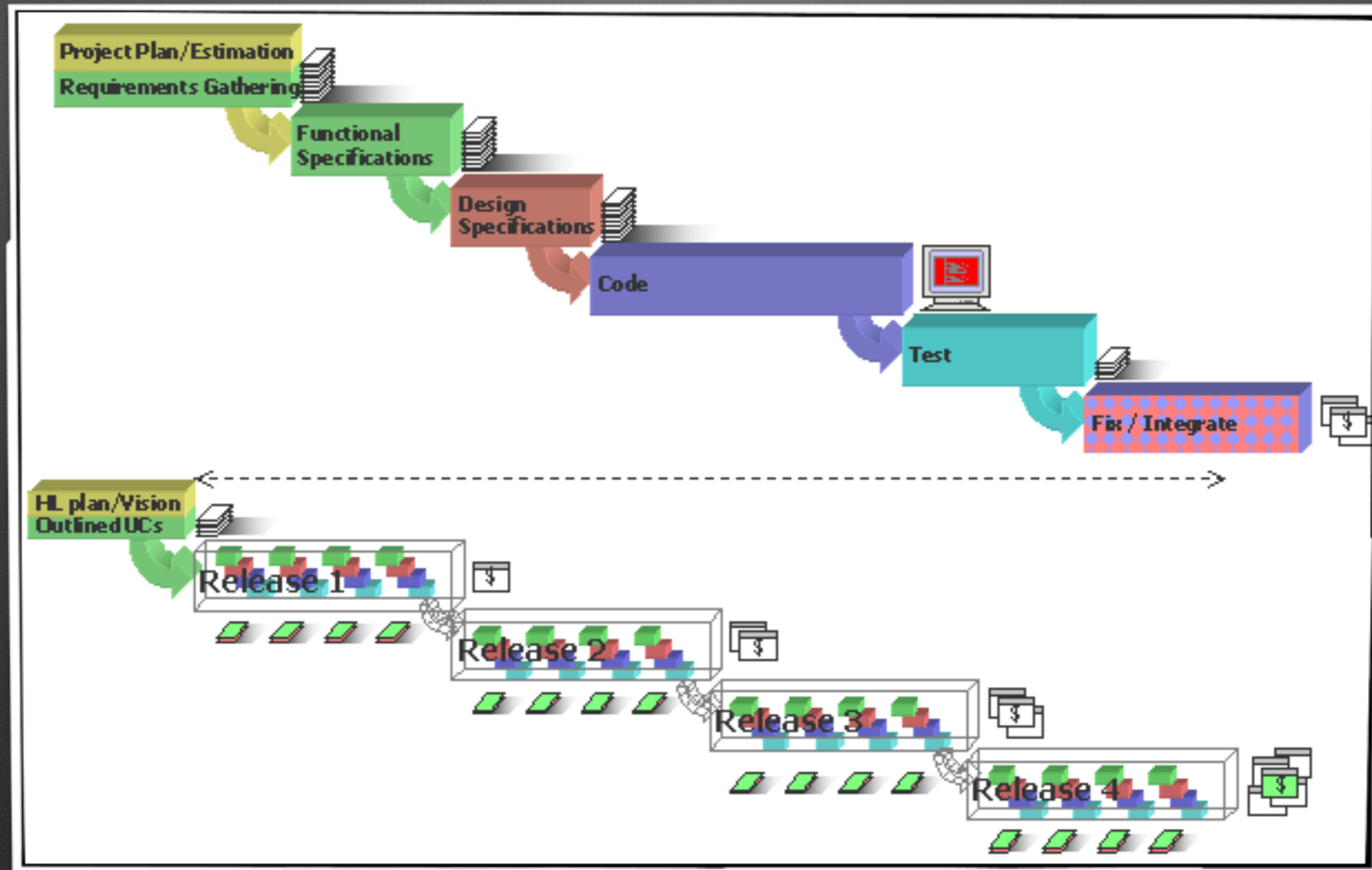
- NE 100% pokrytí testů (ani unit, ani systémové)
 - NE 100% možných a popsanych UC implementováno
 - NE vymakané GUI u všech scénářů
 - NE profesionální/robustní architektura
-
- ALE alespoň nějaké existující unit testy a funkční testy (tabulka v Excelu, Fitnesse, TPTP, ...)
 - ALE alespoň 2-3 scénáře implementovány
 - ALE nějaké frameworky použité a ověřené (JUnit, Struts, Spring ...)
 - ALE pracovat v duchu objektově-orientovaného paradigmatu
 - ALE alespoň základní prostředí nastavené (IDE, SVN, Wiki pro dokumentaci a návody, testovací nástroje, ...)

ROPR2

- Navazuje na ROPR1 dalšími fázemi:
 - Construction - hlavní fáze tvorby software, převážně kódovací, postaveno na ověřené architektuře z ROPR1, dopracování cca 80% procent požadavků
 - Transition - jen některé aktivity - funkční a nefunkční testování, demonstace zákazníkovi

Proces založený na RUP/OpenUP

Základní principy

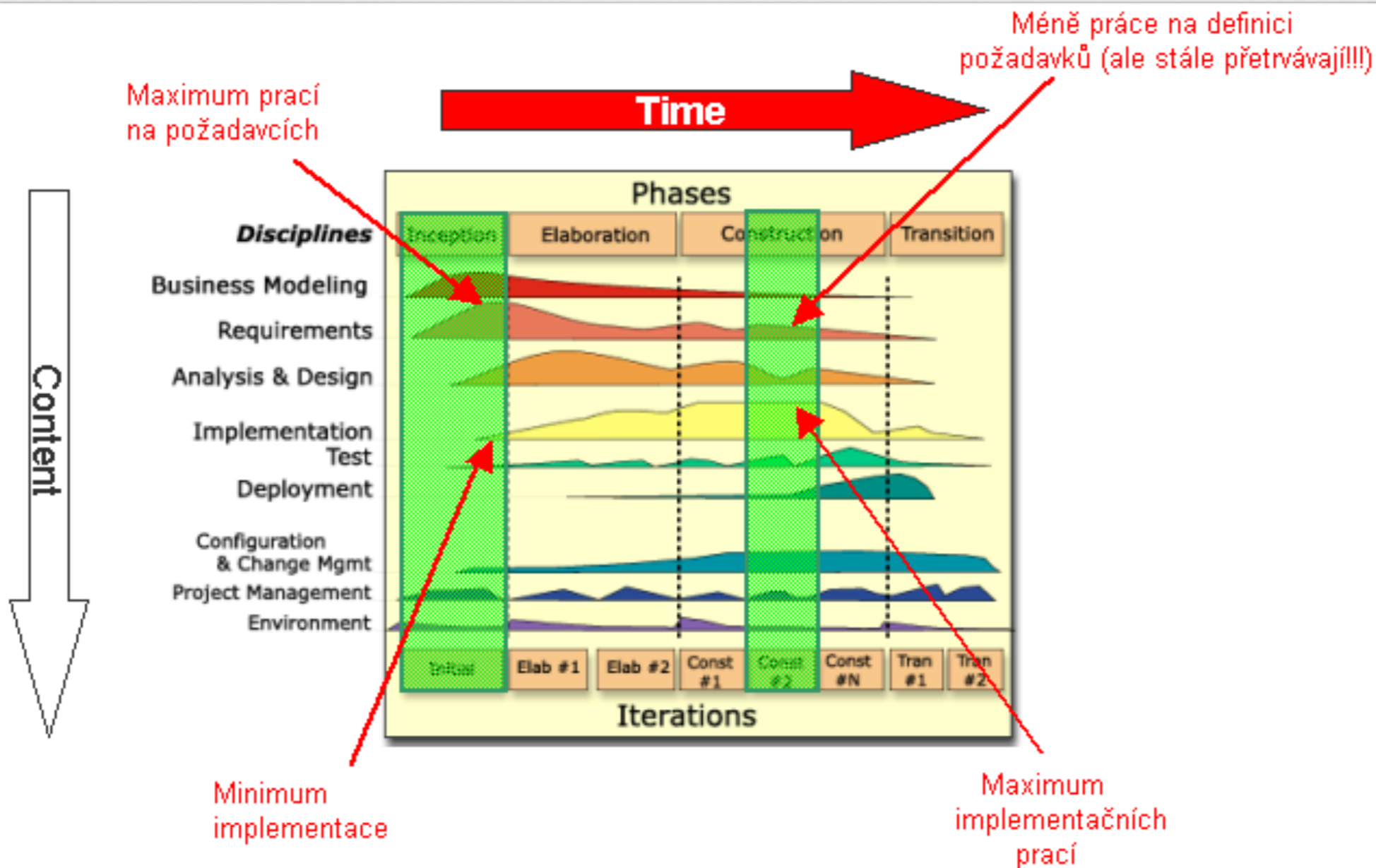


Rational Unified Process (RUP)

- UC driven (řízen pomocí UC)
- Zaměřen na rizika (nejrizikovější věci dělám nejdříve)
- Iterativní (každá iterace produkuje spustitelný a otestovaný build)
- Kooperace (analytik, designer, programátor, tester těsně spolupracují)
- Orientován na architekturu



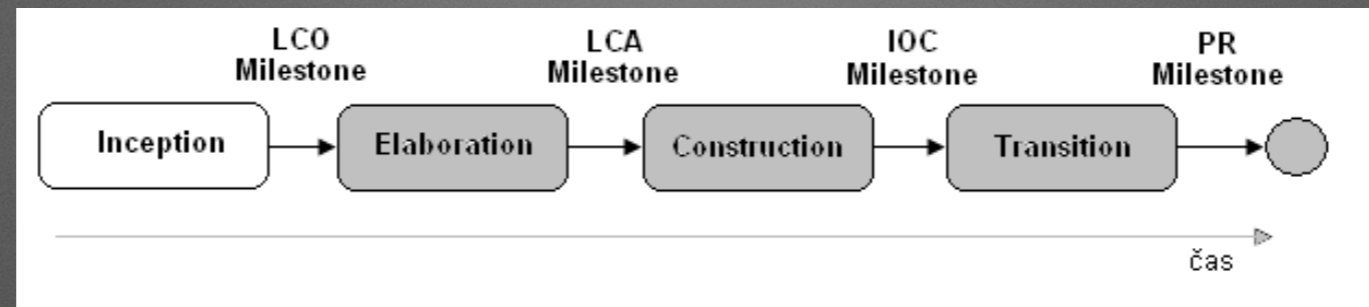
RUP fáze a disciplíny



Iterace (sprint)

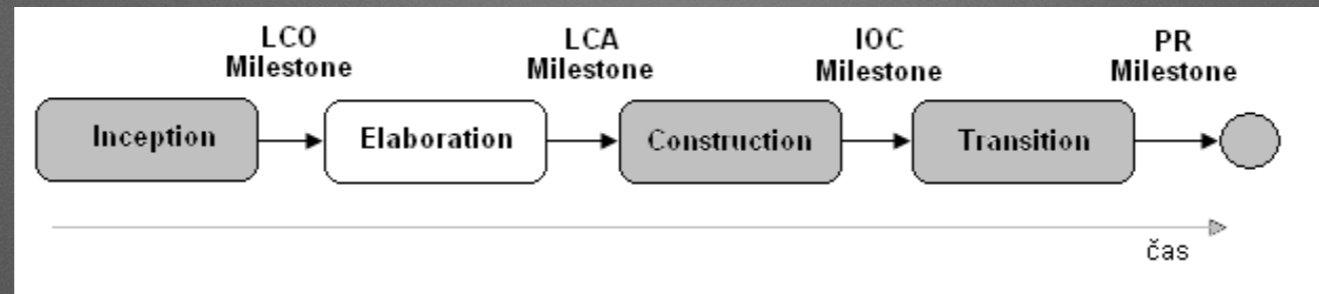
- Každá iterace produkuje spustitelný a otestovaný build obsahující nově implementované funkčnosti (scénáře) – z důvodu zpětné vazby od uživatele
- Každá iterace má definovaný přesný cíl, který se snažíme naplnit (paralelní analýza, návrh, implementace a testování vybrané nové funkčnosti – jejich scénářů).
- Iterace je miniprojekt, má svůj začátek, konec a pevně definovaný časový rozsah (2-6 týdnů) – což se již předvídá a detailně plánuje lépe, než například 2 roky.
- V průběhu 1 iterace provádíme všechny disciplíny! Tj. definice požadavků, analýza a návrh, implementace, integrace a testování!!!
- Zřetězením iterací nabalujeme jednotlivé funkčnosti až do výsledného produktu.

Fáze inception



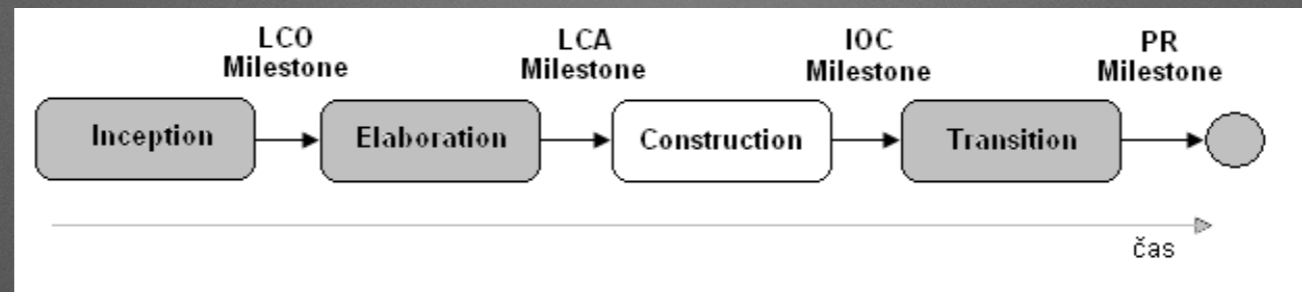
- Pochopení problematiky, nastínění funkcí systému (základní UC) – Vize
- Identifikace rizik + akce na jejich snížení / odstranění
- Návrh možného řešení (architektura)
- Složení týmu, financování
- Definice procesu, výběr a nastavení nástrojů

Fáze elaboration



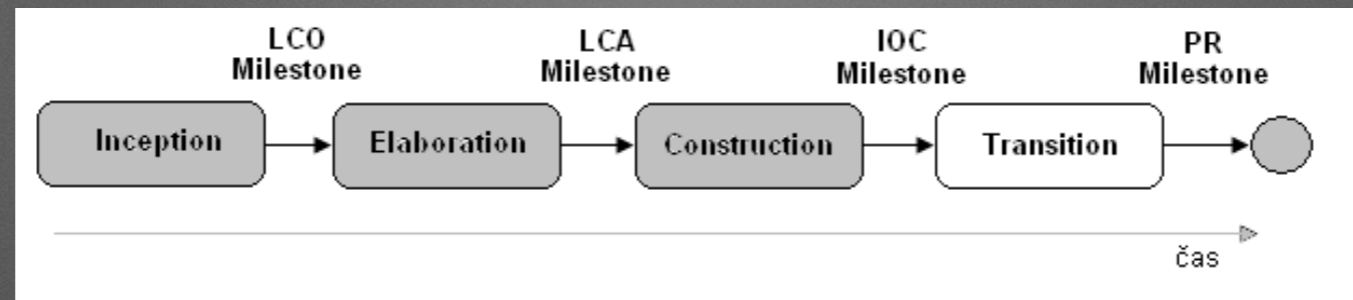
- Snížení technických rizik
- Návrh implementace a testování architektury – na základě klíčových potřeb (asi 20-30% UC)
- Výsledek fáze: stabilní, otestovaná architektura
- Architektura – rozhraní subsystémů, komunikační mechanismy, odchytávání výjimek, ukládání dat, ...

Fáze construction



- Detailní analýza, návrh, implementace a testování zbývajících 70-80% UC v několika iteracích
- Využíváme mechanismy definované architekturou
- Výsledek fáze: beta-release
- Všechny disciplíny (analýza, návrh, vývoj, testování) jsou v každé iteraci prováděny paralelně

Fáze Transition



- Beta testování
- Školení (podpory, uživatelů)
- Příprava dat a prostředí (souběžný provoz verzí, transformace dat), instalace SW na klientské stanice
- Další aktivity (marketing, distribuce, prodej, CASE study, white papers, zprávy pro tisk)
- Lessons learnt - pro zlepšení procesu

Jak to udělat špatně

- Částečně udělaná práce
- Extra rysy aplikace (zákazník nikdy nepoužije)
- Znovu se učení
- Předávání práce, dokumentů, znalosti
- Přepínání mezi jednotlivými úkoly (context switching)
- Zpoždění (čekání na schválení, fronty)
- Defekty

Jak to udělat dobře

- TDD
- Iterative development (feedback, možnost zahrnout v průběhu vývoje změny)
- UC driven (reuse TC, traceability)
- Continuous integration (automatické sestavení + automatický testing)
- Mále dokumentace + dokumentace návrhu kódem
- Implementace pouze potřebných požadavků (klíčové potřeby – 20% UC)
- Posunout rozhodnutí