

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Rozšíření GUI datového skladu pro data softwarových projektů

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Lukáš HARINGER**
Osobní číslo: **A13B0306P**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informatika**
Název tématu: **Rozšíření GUI datového skladu pro data softwarových projektů**
Zadávací katedra: **Katedra informatiky a výpočetní techniky**


Z á s a d y p r o v y p r a c o v á n í :

1. Seznamte se s možnostmi analýzy a prezentace dat softwarových projektů, konkrétně s projektem SPADe a implementací jeho nástrojů.
2. Analyzujte možnosti rozšíření a vylepšení nástroje na prezentaci dat SPADe databáze, identifikujte případné vady nebo nedostatky současné implementace.
3. Navrhněte odpovídající úpravy implementace s ohledem na budoucí rozvoj aplikace.
4. Návrhy úprav konzultujte a odsouhlasené implementujte.
5. Výsledné řešení otestujte regresně, funkčně i uživatelsky.

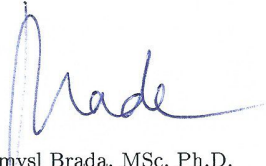
Rozsah grafických prací: **dle potřeby**
Rozsah kvalifikační práce: **doporuč. 30 s. původního textu**
Forma zpracování bakalářské práce: **tištěná**
Seznam odborné literatury:
dodá vedoucí bakalářské práce

Vedoucí bakalářské práce: **Ing. Petr Pícha**
Katedra informatiky a výpočetní techniky

Datum zadání bakalářské práce: **10. října 2017**
Termín odevzdání bakalářské práce: **3. května 2018**


Doc. Dr. Ing. Vlasta Radová
děkanka




Doc. Ing. Přemysl Brada, MSc. Ph.D.
vedoucí katedry

V Plzni dne 17. října 2017

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 1. května 2018

Lukáš Haringer

Abstract

The aim of this bachelor thesis is to extend the SPADe GUI tool, which is used to graphically display SPADe data warehouse data. For this purpose, the SPADe GUI will be analyzed to identify possible enhancements, flaws, and deficiencies of the current implementation. In the next part, this thesis will deal with implementation of given extensions and subsequent testing of the resulting tool. Part of the thesis is an introduction to the possibilities of analysis and presentation of data of software projects, namely SPADe and implementation of its parts.

Abstrakt

Cílem této bakalářské práce je rozšířit nástroj SPADe GUI, který slouží ke grafickému znázornění dat datového skladu nástroje SPADe. Za tímto účelem bude provedena analýza grafického uživatelského rozhraní nástroje SPADe GUI k identifikaci možných vylepšení, vad a nedostatků současné implementace. V další části se tato práce bude zabývat implementací daných rozšíření a následným testováním výsledného nástroje. Součástí práce je seznámení s možnostmi analýzy a prezentace dat softwarových projektů, konkrétně s projektem SPADe a implementací jeho částí.

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce Ing. Petru Píchovi za jeho vedení, trpělivost a cenné rady v průběhu vypracovávání celé této práce.

Obsah

1	Úvod	8
2	Kontext práce	9
2.1	Projektové řízení	9
2.1.1	Projekt	9
2.1.2	Proces vývoje SW	10
2.1.3	Metodiky	10
2.1.4	Anti-patterny	13
2.1.5	Grafy používané v projektovém řízení	13
2.2	SPADe	17
2.2.1	ALM nástroje	18
2.2.2	SPADe GUI	19
2.3	Požadavky a odsouhlasené návrhy na úpravu aplikace SPADe GUI	21
3	Analýza	24
3.1	Analýza podobných nástrojů	24
3.1.1	Kibana	24
3.1.2	Bitergia	24
3.1.3	Bugzilla	27
3.1.4	Redmine	27
3.2	Analýza změn aplikace	28
3.2.1	Anglická lokalizace	29
3.2.2	Drag and drop panel	29
3.2.3	Zvýšení přehlednosti	29
3.2.4	Uživatелеm definované grafy	30
3.2.5	Rozšíření možností filtrování	32
4	Implementace	33
4.1	Balík data.polozky	33
4.2	Balík data	33
4.3	Balík data.ciselnik	34
4.3.1	Třída Ciselnik	34
4.4	Balík data.custom	34
4.4.1	Třída CustomGraf	34
4.4.2	Třída PrepravkaUkladaniCustom	35

4.4.3	Třída <code>NastaveniCustomSloupec</code>	35
4.4.4	Třída <code>SablonaCustomGrafu</code>	35
4.5	Balík <code>database</code>	36
4.6	Balík <code>gui</code>	36
4.6.1	Třídy pro vykreslení oken	37
4.6.2	Třídy pro vykreslení panelů s filtry	38
4.6.3	Třídy pro vykreslení panelů s grafy	39
4.6.4	Třídy pro vykreslení uživatelem definovaných grafů	40
4.6.5	Třídy pro drag and drop	41
4.6.6	Ostatní třídy balíku <code>gui</code>	42
4.7	Balík <code>ostatni</code>	42
4.7.1	Třída <code>Ukladani</code>	42
4.7.2	Třída <code>Konstanty</code>	43
5	Zhodnocení	44
5.1	Testování	44
5.2	Rozsah změn implementace	45
5.3	Dosažené výsledky	45
5.4	Nedostatky současné implementace	46
5.5	Budoucí rozšíření	47
6	Závěr	50
	Literatura	53
	Přílohy	56
	A Obsah přiloženého DVD	57
	B Uživatelská dokumentace	58
B.1	Okno přihlašování	58
B.2	Hlavní okno	58
B.3	Okno uživatelských grafů	59
	C Ukázky testovacích scénářů	63
	D Výsledky uživatelských testů	65

1 Úvod

Při vývoji software dochází k často se opakujícím chybám projektového řízení i chování vývojových týmů. Tyto chyby jsou označovány jako „bad practices“ nebo také anti-patterny. Existuje velké množství literatury, školení a nástrojů (např. ALM – Application Lifecycle Management), které mohou napomoci při jejich odhalení. Doposud však nebyl vytvořen ucelený automatizovaný nástroj, který by byl schopný anti-patterny detekovat a pomoci při jejich odstranění. Z tohoto důvodu vzniká na Katedře informatiky a výpočetní techniky (KIV) Fakulty aplikovaných věd (FAV) Západočeské univerzity v Plzni (ZČU) nástroj SPADe (Software Process Anti-pattern Detector). Ve své finální podobě bude SPADe sloužit ke shromažďování dat z ALM nástrojů v datovém skladu a následné analýze dat za účelem identifikace anti-patternů a nasměrování vedoucích pracovníků vývojových týmů k jejich možnému vyřešení.

Cílem této bakalářské práce je rozšířit aplikaci SPADe GUI, který slouží ke grafickému znázornění dat v datového skladu nástroje SPADe. Za tímto účelem bude provedena analýza grafického uživatelského rozhraní (GUI – Graphical User Interface) nástroje SPADe (dále SPADe GUI) k identifikaci možných vylepšení, vad a nedostatků současné implementace. V další části se tato práce bude zabývat implementací daných rozšíření a následným testováním výsledného nástroje.

Součástí práce je seznámení s možnostmi analýzy a prezentace dat softwarových projektů, konkrétně s projektem SPADe a implementací jeho částí.

2 Kontext práce

Tato kapitola se věnuje seznámení se základy projektového řízení, nástrojem SPADe a požadavky na úpravu současné implementace aplikace SPADe GUI.

2.1 Projektové řízení

Ve své finální podobě bude SPADe sloužit jako nástroj pro projektové řízení. Projektové řízení [23] je aplikace znalostí, dovedností, nástrojů a technik do projektových aktivit tak, aby efektivně napomáhaly ke splnění požadavků projektu. Projektoví manažeři musí usilovat nejen o splnění specifických cílů, termínů a rozpočtů projektů, ale musí také usnadňovat celý proces, aby uspokojili potřeby a očekávání lidí zapojených do projektu.

2.1.1 Projekt

Projekt [23] je dočasnou snahou o vytvoření jedinečného produktu, služby nebo výsledku. Oproti tomu operační aktivita (anglicky "operations") je práce kontinuálně vykonávaná v organizacích pro udržení podnikání. Projekty se liší od operací tím, že končí, když byly dosaženy jejich cíle nebo byl projekt ukončen.

Atributy projektu

- **Projekt má jedinečný účel.** Každý projekt by měl mít jasně stanovený cíl.
- **Projekt je dočasný.** Projekt má určitý počátek a definitivní konec.
- **Projekt vyžaduje zdroje,** často z různých oblastí. Zdroje zahrnují lidi, hardware, software a další prostředky
- **Projekt je vyvíjen pomocí postupného zpracování.** Projekty jsou často na počátku definovány široce a postupem času se vyjasňují detaily.

- **Projekt zahrnuje nejistotu.** Vzhledem k tomu, že každý projekt je jedinečný, je někdy obtížné jasně definovat jeho cíle, odhadnout, jak dlouho bude trvat, nebo určit, kolik bude stát. Nejistota je také způsobována externími faktory, jako například dodavatelem, který ukončí podnikání, nebo členem projektu, který potřebuje neplánované volno. Tato nejistota je jedním z hlavních důvodů, proč je řízení projektů tak náročné, zejména pokud jde o projekty zahrnující nové technologie. [23]

2.1.2 Proces vývoje SW

Každý projekt má více či méně definovaný proces [12], kterým se řídí. Tyto procesy jsou často odvozeny od ověřených metodik (viz níže) nebo volnou kolekcí praktik vybraných z více metodik.

S ohledem na životní cyklus projektu popisujeme proces řízení projektu jako fázový model. Existuje pět hlavních fází řízení projektů:

- **Zahájení** – popis produktu projektu, počáteční dokumentace cílů projektu a přiřazení projektového manažera.
- **Plánování** – dokumentace projektového plánu a její aktualizace v průběhu projektu.
- **Vykonávání** – ověřitelné dokončení projektových výstupů.
- **Kontrola** – periodické měření pokroku oproti plánu projektu a v případě potřeby vykonání nápravných opatření.
- **Uzavření** – dokumentované přijetí výsledků projektu. [12]

2.1.3 Metodiky

Metodika [9] řízení projektů je soubor zásad a procesů řízení projektu. Výběr metodiky určuje způsob práce a vzájemné komunikace v rámci vývojového týmu.

Sekvenční

Sekvenční metodiky jsou silně orientované na požadavky. Je nutné znát všechny požadavky projektu předem, protože po zahájení projektu není žádný prostor pro opravu. Pokud se požadavky změni je nutné začít znovu.

Jejich hlavní výhodou je snadná pochopitelnost a dokumentovatelnost, což umožňuje v případě potřeby snadné začlenění dalších lidí do projektu. Hlavní nevýhodou je riziko, že v případě objevení chyby návrhu nebo potřeby změny je nutno začít znovu. Z těchto důvodů se tato metodika hodí zejména pro krátké a jasně definované projekty. [9]

Zástupci sekvenčních metodik jsou:

- **Vodopád** [16]
- **V-model** [15]
- **Structured systems analysis and design method**

Iterativní

Vývoj iterativní metodikou nevyžaduje na začátku detailní zpracování projektu. Místo toho se na začátku vytvoří základní verze, která tvoří základ pro budoucí požadavky. Tento proces se opakuje. Tyto verze nemusí být ideální, hlavní je, aby fungovaly. Každá další verze by měla být blíže konečné podobě projektu. [9]

Zástupci iterativních metodik jsou:

- **Rational Unified Process** [18]
- **Microsoft Solution Framework**

Agilní

Agilní [7] metodiky vznikly jako reakce na selhání sekvenčních metodik pro správu komplexních projektů. V přístupu a ideologii jsou agilní metodiky opakem sekvenčních. Jak název naznačuje, tyto metodiky se zaměřují na

rychlý a flexibilní vývoj. Nedochozí k žádnému náročnému shromažďování požadavků. Agilní metodiky jsou iterativní s malými přírůstkovými změnami, které reagují na měnící se požadavky.

Hlavními výhodami těchto metodik je schopnost reagovat na změny požadavků a pravidelná zpětná vazba od zadavatelů projektu. Nevýhodou těchto metodik je neexistence pevného, dlouhodobého plánu, což značně ztěžuje přiřazování zdrojů a vzájemnou komunikaci. Agilní metodiky se nejvíce hodí pro kreativní projekty, kde není od začátku jasná představa o výsledném produktu. [9]

Zástupci agilních metodik jsou:

- **Scrum** [22]
- **Extrémní programování**

Hybridní

Hybridní přístup, jak naznačuje název, je nejčastěji kombinací sekvenčních a agilních metodik. Vychází z nejlepších částí sekvenčních a agilních metodik a kombinuje je ve flexibilním, ale strukturovaném přístupu. Soustředí se na počáteční sbírání a analýzu požadavků a následně na iterační vývoj. Nejvíce se hodí pro středně velké projekty, kde máme základní představu o výsledném produktu, ale je možné, že v průběhu vývoje dojde ke změně. [9]

Zástupci hybridních metodik jsou:

- **Disciplined Agile Delivery** [8]

Lean

Lean [14] koncept řízení výroby se vyvinul a čerpá z japonského stylu řízení, který byl uplatňován v japonské automobilce Toyota v období po druhé světové válce.

Lean (česky "štíhlý") se zaměřuje na odstranění plýtvání skrze efektivní využívání zdrojů, ve snaze o minimalizaci nákladů a o co nejrychlejší a nejflexibilnější uspokojení zákaznickových potřeb.

Lean je někdy charakterizován jako agilní metodika vývoje softwaru, avšak od ostatních agilních metodik se poměrně odlišuje. Proto bývá někdy definován spíše jako filozofie či soubor principů pro vývoj softwaru, než jako metodika.

Zástupci lean metodik jsou:

- **Kanban**

2.1.4 Anti-patterny

Antipatterny [25] jsou často používaná řešení často se vyskytujících problémů, která jsou neúčinná a mohou mít za následek nežádoucí následky. Na rozdíl od návrhových vzorů, které jsou ověřeným řešením častých problémů a obecně se považují za dobrou vývojovou praxi, jsou anti-patterny opakem a jsou nežádoucí.

Známým anti-patternem je například Fire Drill [24]. Tento anti-pattern popisuje situaci, kdy je projekt iniciován, ale zaměstnanci zpožďují projektové a vývojové činnosti po dobu několika měsíců, zatímco je řešena řada technopolitických problémů na úrovni managementu. Následkem tohoto zdržení nezůstává dost času na vývoj software a jsou dělány kompromisy ohledně kvality produktu a jeho testování.

Mezi časté symptomy anti-patternů (které se projekt SPADe snaží identifikovat) patří například nadhodnocení nebo podhodnocení odhadů, nedodržení termínu iterace a nezaznamenávání podstatných informací.

2.1.5 Grafy používané v projektovém řízení

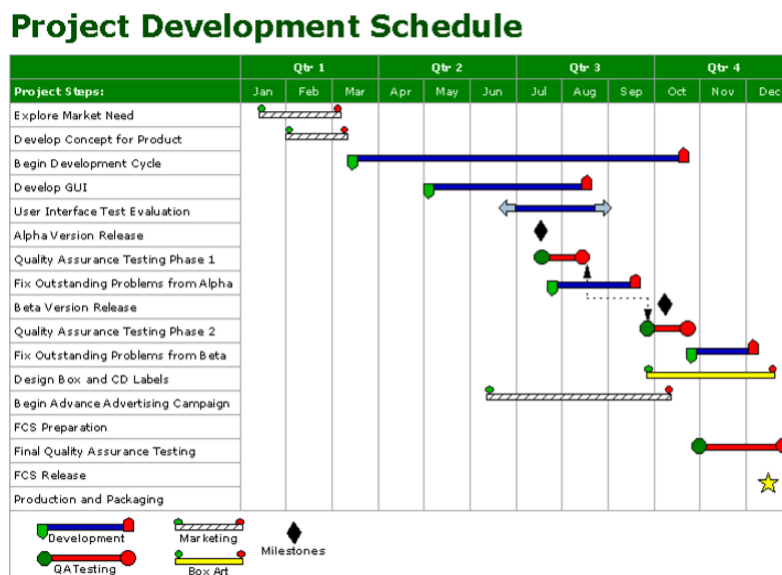
Každý projekt nezávisle na procesu vykazuje určité měřitelné veličiny pokroku KPI (Key Performance Indicators). Ačkoli měření úspěchu pouze těmito kvantitativními aspekty je považováno za špatnou praxi, jejich význam a hodnota informací, které poskytují, je přesto nezanedbatelná. Nejběžnější formou vizualizace takových metrik jsou dashboardové GUI se sadou různých grafů.

Grafy [11] jsou často používány v IT projektech, aby vizuálně pomohly

při řízení projektů, definovaly specifikace systému a zlepšily komunikaci mezi účastníky projektu. Strukturní diagramy, Ganttovy grafy a PERT grafy jsou určeny pro řízení projektů, zatímco výšečové a sloupcové grafy pro efektivní vizuální prezentaci informací.

Ganttův graf

Ganttův [11] diagram představuje znázornění plánu, který pomáhá naplánovat, koordinovat a sledovat konkrétní úkoly v projektu. Jedná se o horizontální sloupcový graf vyvinutý jako nástroj řízení výroby v roce 1917 Henrym L. Ganttem, americkým inženýrem a společenským vědcem. Obvykle se skládá ze dvou částí: úkoly projektu vlevo a přidělení času pro každý úkol vpravo. Milníky jsou speciálně označeny jako důležité termíny pro výstupy. Příklad Ganttova grafu je vidět na obrázku 2.1.



Obrázek 2.1: Ganttův graf. Převzato z [11]

Výšečové grafy

Výšečový graf [11] je specializovaný graf používaný ve statistice. Nezávislá proměnná je vynesena kolem kružnice ve směru nebo proti směru hodinových ručiček. Závislá proměnná (obvykle procenta) je vykreslena jako oblouk, jehož míra je úměrná velikosti veličiny. Každý oblouk je znázorněn konstrukcí



Obrázek 2.2: Výsečový graf

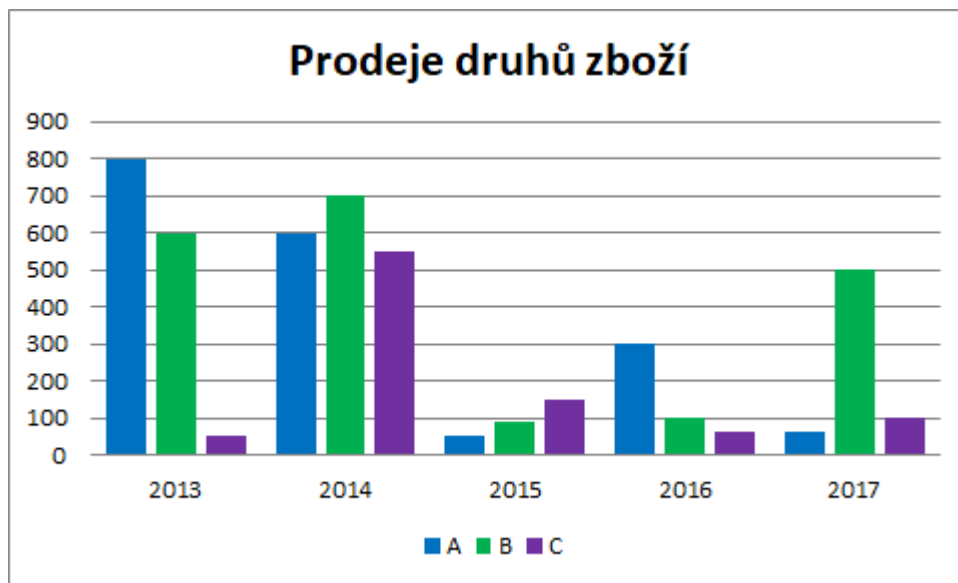
radiálních čar od jeho konců ke středu kruhu, čímž vzniká klínovitý "řez". Nezávislá proměnná může dosáhnout konečného počtu diskretních hodnot (například pět). Závislá proměnná může dosáhnout libovolné hodnoty od nuly do 100 procent. Příklad výsečového grafu je zobrazen na obrázku 2.3.

Sloupcové grafy

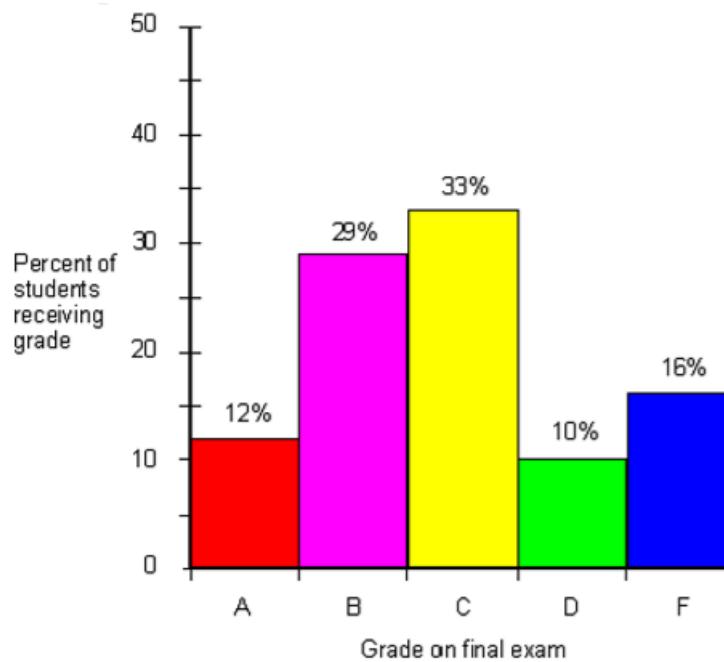
Sloupcové grafy [11] se používají pro porovnání řady nespojitých hodnot proti stejnému měřítku. Seskupování sloupcových grafů může také sloužit k zobrazení vztahů mezi datovými sadami. Příklad sloupcového grafu je vidět na obrázku 2.4.

Histogram

Histogram [11] je specializovaný graf používaný ve statistice. V nejběžnější podobě je nezávislá proměnná vynesena podél horizontální osy a závislá proměnná (obvykle procenta) je vynesena podél svislé osy. Nezávislá proměnná může dosáhnout pouze konečného počtu diskretních hodnot (například pěti). Závislá proměnná může mít spojitý rozsah. Následující vzorový histogram znázorňuje výsledky závěrečné zkoušky, která byla udělena hypotetické skupině studentů. Každá známka je označena svislou čarou určité barvy. Celkový součet procent se musí rovnat 100. Příklad histogramu je vidět na obrázku 2.5.



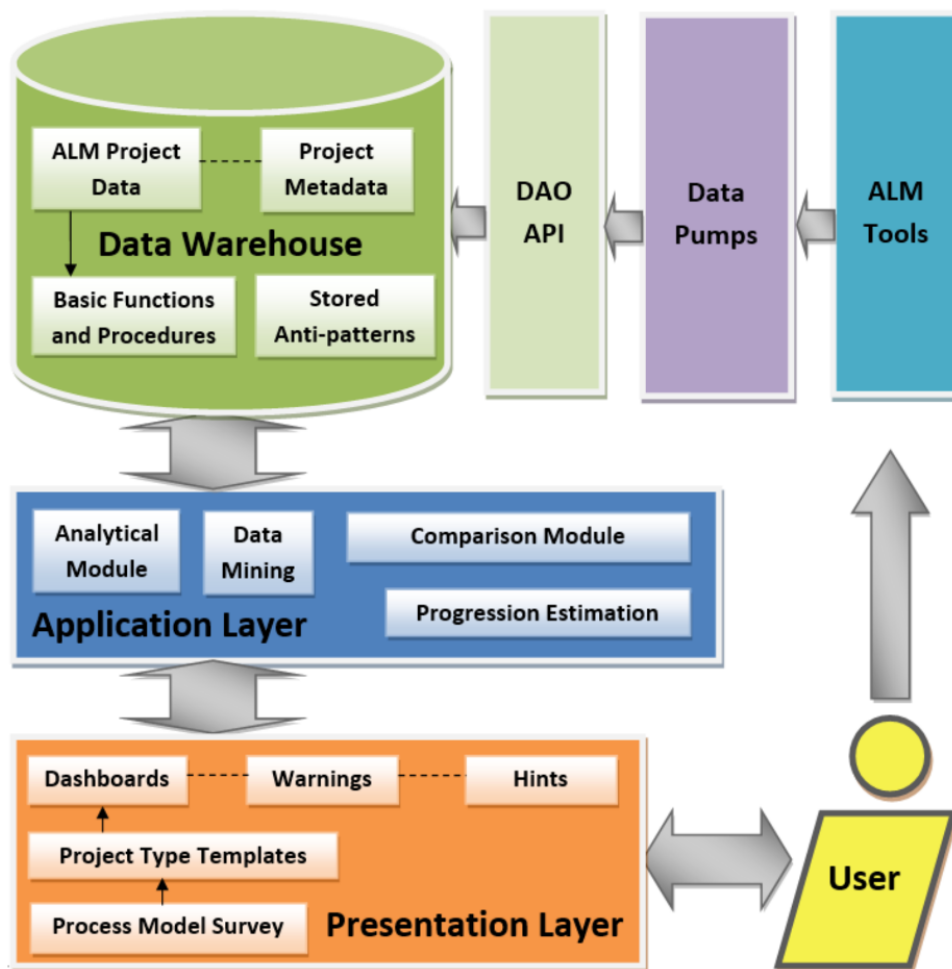
Obrázek 2.3: Sloupcový graf



Obrázek 2.4: Histogram. Převzato z [11]

2.2 SPADe

SPADe [21] [19] [20] je softwarový nástroj vyvíjený na KIV. Ve své finální podobě bude tento program za pomoci datových pump získávat data z řady tzv. Application Lifecycle Management nástrojů (ALM). Takto získaná data bude posléze ukládat do jednotného datového skladu a následně bude v těchto datech identifikovat „bad practices” neboli anti-patterny. Po úspěšné identifikaci anti-patternu by SPADe mohl podávat hlášení příslušnému uživateli o jeho výskytu a navrhnout jak anti-pattern odstranit. Obrázek 3.1 zobrazuje schéma nástroje SPADe.



Obrázek 2.5: Schéma nástroje SPADe. Převzato z [21]

2.2.1 ALM nástroje

ALM nástroje [21] [19] [20] [13] jsou podpůrné softwarové nástroje pro řízení životního cyklu aplikace. ALM nástroje mají za cíl sledovat a dokumentovat všechny změny provedené v průběhu celého procesu vývoje aplikace až do jejího vyřazení. Pojem ALM zahrnuje nástroje pro změnové řízení (bugtrackery / issuetrackery), konfigurační řízení, správu požadavků, řízení kvality, atd. V dnešní době v podstatě každý softwarový projekt, na kterém pracuje více než jeden člověk, používá nějaký ALM nástroj. Je možné používat i více ALM nástrojů současně.

ALM nástroje používané SPADe

Nástroj SPADe pracuje s následujícími ALM nástroji:

- Apache Subversion
- Assembla
- Atlassian Jira
- Bugzilla
- Git
- GitHub
- Redmine

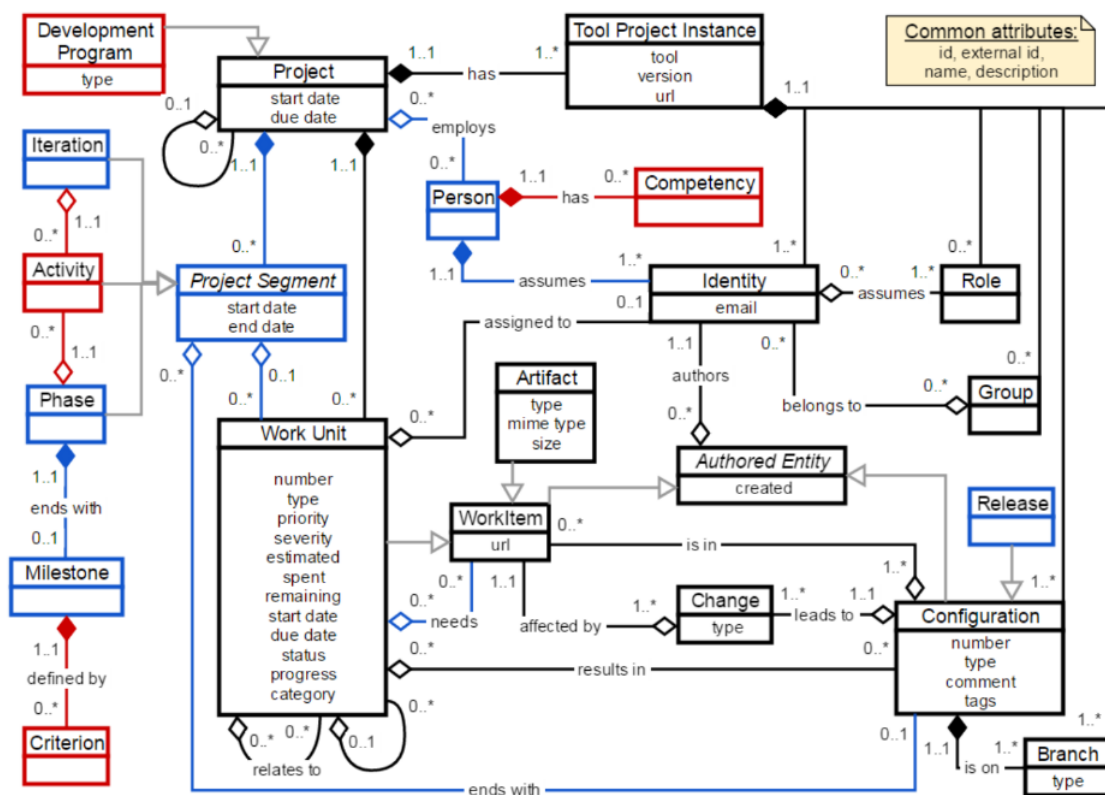
Datový sklad

Pojmem datový sklad (anglicky Data Warehouse) [21] [19] [20] [10] označujeme zvláštní typ databáze, určený primárně pro analýzy dat v rámci Business Intelligence (oblast analýzy dat sloužící jako podklady pro manažerské rozhodování).

Datové sklady jsou integrované – data, která jsou ukládána v datovém skladu, pochází z několika zdrojů. Data jsou na základě určitých pravidel spojována tak, aby poskytla koncovému uživateli celistvý pohled na oblast

jeho zájmu. Oproti běžným relačním databázím není kladen důraz na snížení redundance dat, ale na jejich srozumitelnost pro uživatele (dosážno standardní strukturou – faktová tabulka).

SPADe doluje data pro datový sklad za pomoci datových pump. Každá pumpa je kouskem softwaru na principu Extract – Transform – Load (ETL) specificky implementovaným pro extrakci dat z daného ALM (každý ALM má vlastní pumpu), převod dat do jednotného formátu a uložení do datového skladu skrze příslušné rozhraní. Obrázek 3.2 znázorňuje doménový model datového skladu nástroje SPADe.



Obrázek 2.6: Doménový model datového skladu SPADe. Převzato z [21]

2.2.2 SPADe GUI

Aplikace SPADe GUI [26] je samostatný program, který vznikl v roce 2017 na Katedře informatiky a výpočetní techniky Fakulty aplikovaných věd ZÁ-

padočeské univerzity v Plzni jako bakalářská práce Michala Všelka. Jedná se o desktopovou aplikaci, která slouží jako grafické rozhraní pro zobrazování statistik a grafů dat z datového skladu nástroje SPADe. Aplikace SPADe GUI nevychází z implementace "dolovací" (ETL) části nástroje SPADe a sdílí s ním pouze databázi. Cílem této bakalářské práce je vylepšení tohoto nástroje a oprava jeho případných nedostatků.

Dosavadní implementace nástroje SPADe GUI

Aplikace [26] je vytvořena stejně jako nástroj SPADe v programovacím jazyce Java. Pro vykreslování grafů používá knihovnu JFreeChart [17] a z toho důvodu používá pro uživatelské rozhraní knihovnu Swing, pro kterou je JFreeChart primárně určena. Pro připojení k MySQL [5] databázi je použito API (Application Programming Interface) JDBC (Java Database Connectivity) [3]. Na obrázku 3.3 je vidět architektura aplikace SPADe GUI.

Třídy aplikace jsou rozděleny do balíků:

- `data` – Balík `data` obsahuje datové třídy `Projekt`, `Ukol`, `Segment`, `Artefakt`, `Konfigurace` a `Osoba`. V těchto datových třídách jsou uloženy nejdůležitější informace z datového skladu nástroje SPADe pro grafické rozhraní. [26]
- `data.ciselnik` – Základem tohoto balíku je abstraktní třída `Ciselnik`, ze které ostatní třídy dědí. Dalšími třídami v tomto balíku jsou třídy `Priority`, `Severity`, `Resoluce`, `Status`, `Typ` a `Osoby`. [26]
- `data.polozky` – `data.polozky` obsahuje datové třídy pro uložení nejzákladnějších údajů datového modelu. Tyto třídy se využívají ve všech ostatních datových třídách. [26]
- `databaze` – Balík `databaze` obsahuje DAO (Data Access Object) třídy a jejich rozhraní, zajišťující výběr dat z databáze pomocí připojení realizovaného technologií JDBC. V balíku jsou třídy `ArtefaktDAO`, `CiselnikyDAO`, `KonfiguraceDAO`, `ProjektDAO` a `UkolDAO`. K těmto třídám jsou zde i rozhraní se stejným názvem doplněným o prefix „I-“ na začátku (`IArtefaktDAO`, `ICiselnikyDAO`, `IKonfiguraceDAO`, `IProjektDAO` a `IUkolDAO`). [26]
- `gui` – V tomto balíku jsou třídy grafického rozhraní. Balík obsahuje čtyři základní typy tříd: třídy pro vykreslení oken (`OknoHlavni`, `Okno-`

Prihlasovani, OknoProgresNacitani), třídy pro vykreslení panelů filtrů na hlavním okně (PanelFiltr, PanelFiltrCiselnik, PanelFiltrKonfigurace, PanelFiltrPolozkaPocatek, PanelFiltrPolozkaVytvoreni), třídy pro vykreslení panelů grafů na hlavním okně (PanelGrafu, PanelGrafuArtefakt, PanelGrafuKonfigurace, PanelGrafuRodic, PanelGrafuSegment, PanelGrafuUkol) a třídu ComboBoxDynamicka. [26]

- ostatni – Mezi ostatní třídy se řadí třída Konstanty a třída Aplikace, která slouží pro spuštění aplikace a načtení popisků lokalizace. [26]

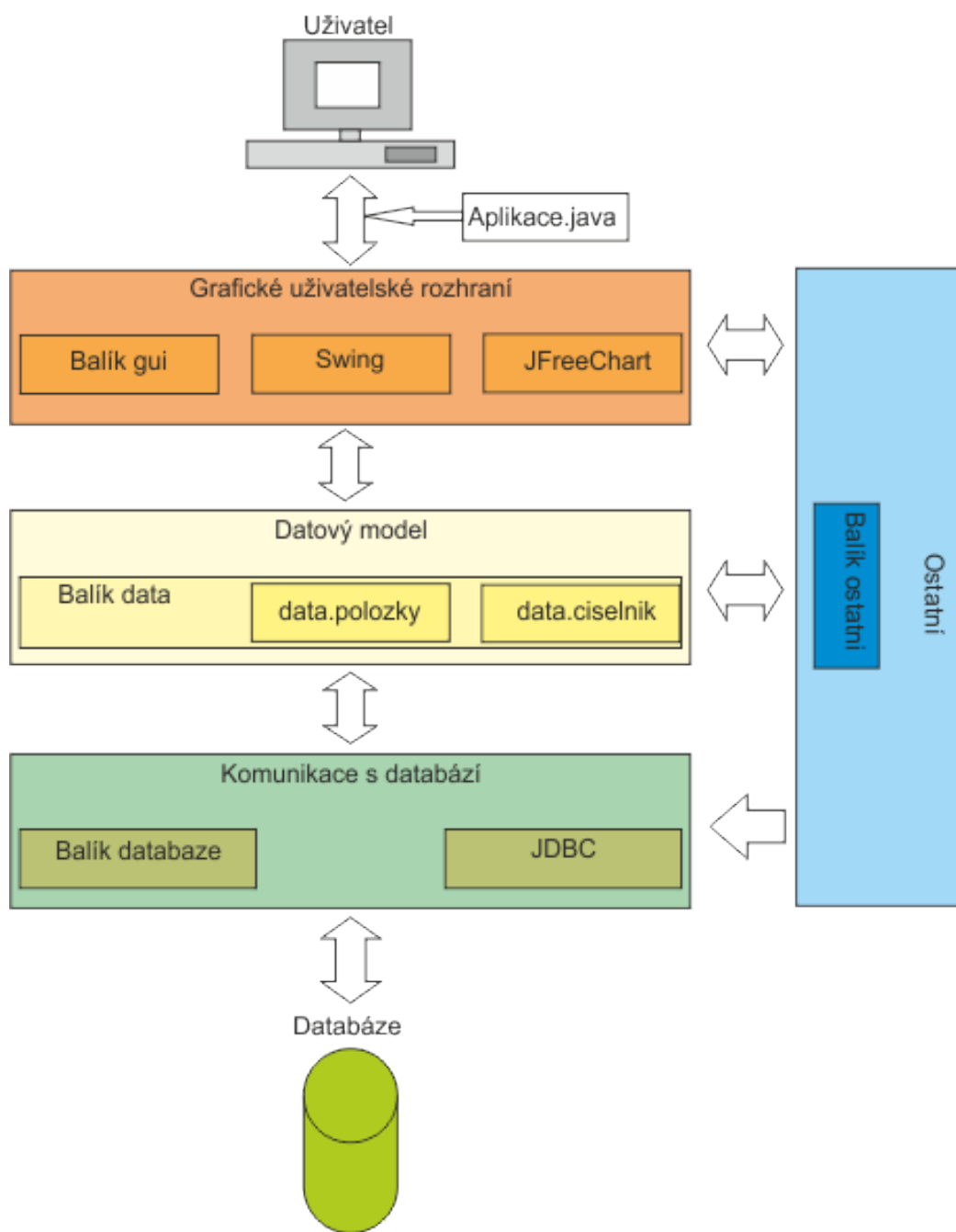
2.3 Požadavky a odsouhlasené návrhy na úpravu aplikace SPADe GUI

Tato kapitola popisuje funkční požadavky vzešlé z analýzy dosavadní implementace aplikace a návrhy na úpravu autora práce odsouhlasené vedoucím práce. Hlavními aspekty brány v potaz při analýze možných rozšíření bylo zvýšení použitelnosti a prohloubení schopností aplikace s ohledem na plnění jejího původního účelu (prezentace dat v datovém skladu SPADe a identifikovaných anti-patternů). Velkým mimofunkčním požadavkem byl také důraz na zachování konvencí kódu a další aspekty usnadňující budoucí úpravy aplikace.

- **Anglická lokalizace.** V současné době nabízí aplikace pouze českou lokalizaci, což značně omezuje využitelnost aplikace a to hlavně v případě prezentování výstupů SPADe zahraničnímu publiku. Proto bude přidána anglická lokalizace, čímž se značně zvýší počet potenciálních uživatelů a pozorovatelů aplikace.
- **Drag and drop panel.** Do aplikace bude přidán panel s miniaturami grafů, který bude podporovat funkci drag and drop, díky které bude možné grafy přesunout do slotu, kde se zvětší do původní velikosti. Tím dojde ke zpřehlednění práce s grafy a větší efektivitě při používání aplikace.
- **Zvýšení přehlednosti.** Layout statistik a filtrů bude změněn tak, aby byl uživatelsky příjemnější a přehlednější. Panel s filtry a panel statistik

bude možné schovat. Uvolněné místo bude využito pro vykreslení grafů, čímž se zlepší jejich čitelnost.

- **Uživatелеm definované grafy.** Bude přidána možnost vytváření grafů dle potřeb uživatele. Vytvořené uživatelské grafy bude možné ukládat a uložené grafy bude možné editovat. Tím dojde k ušetření času při nutnosti změny již vytvořeného grafu. Uložené grafy bude možné přenášet mezi počítači využívající aplikaci SPADe GUI, což výrazně pomůže při spolupráci více lidí. Bude možné ukládat šablony obsahující nastavení pro vykreslení uživatelského grafu. Tímto se ušetří čas při potřebě vytvořit více stejných grafů pro různá data.
- **Rozšíření možnosti filtrů.** Filtry budou rozšířeny o nové filtrovací možnosti, čímž dojde k rozšíření funkcionality aplikace. Filtr bude přidán hned po jeho výběru v menu bez nutnosti stisknutí tlačítka Použít filtr. Při přepínání projektů nebude docházet k odstranění filtrů. Díky tomuto opatření nebude nutné filtry nastavovat opakovaně, což urychlí a usnadní práci s aplikací. Po zapnutí filtru se objeví tlačítko pro jeho odstranění, díky čemuž bude možné filtr rychleji odstranit, což povede ke zrychlení práce s aplikací a její přehlednosti.



Obrázek 2.7: Architektura aplikace. Převzato z [26]

3 Analýza

3.1 Analýza podobných nástrojů

Tato kapitola se věnuje analýze podobných softwarových nástrojů, které inspirovali změny provedené na SPADe GUI. Nástroje Kibana a Bitergia sloužily jako inspirace pro zobrazování dat. Nástroje Bugzilla a Redmine byly použity jako inspirace pro vzhled a logické uspořádání filtrů.

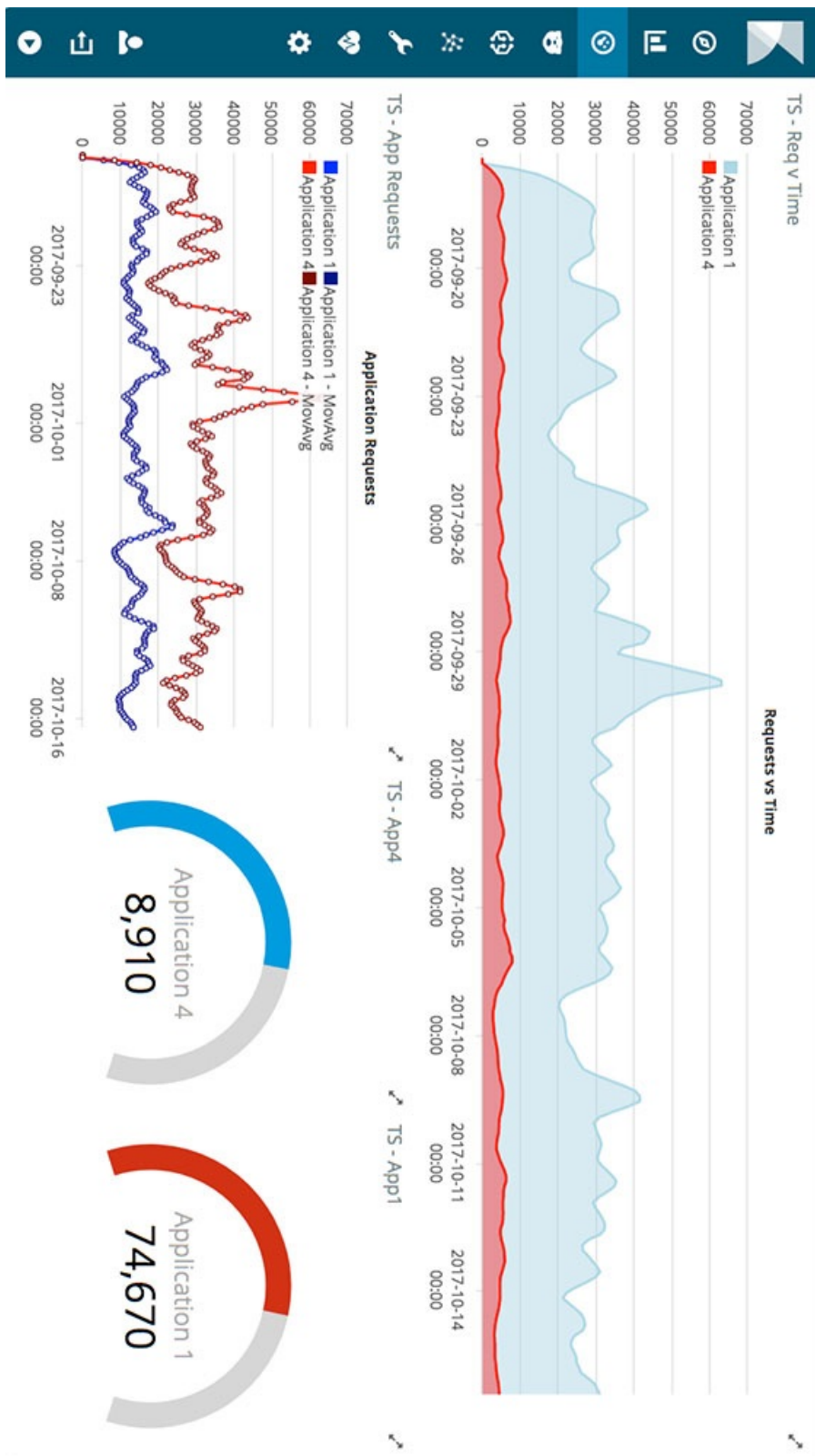
3.1.1 Kibana

Kibana [4] je open source plugin umožňující vizualizaci dat uložených v Elasticsearch ve webovém prohlížeči. Uživatelé mohou vytvářet sloupcové, spojnicové, rozptýlené a koláčové grafy a mapy na velkém objemu dat. Vytvořené grafy je možné vložit jako iframe do jiné aplikace, nebo jen použít dotaz, který Kibana vygenerovala. Kibana je vytvořená v jazyce Javascript a je rozšiřitelná pomocí řady pluginů. Vzhled aplikace Kibana je k vidění na obrázku 5.1.

3.1.2 Bitergia

Bitergia Grimoire Development Dashboard [1] slouží pro automatické a inkrementální shromažďování dat z téměř všech nástrojů souvisejících s open source vývojem (nástroje pro správu zdrojového kódu, issue tracking systémy, fóra atd.)

Bitergia nabízí dashboards pro vizualizaci dat umožňující filtrování podle časového rozmezí, projektu, repozitáře, přispěvatele atd. Dále nabízí data dumpy (datové výpisy) a API přístup pro sestavování reportů a integraci s nástroji třetích stran. Vzhled aplikace Bitergia je kobrazen na obrázku 5.2.



Obrázek 3.1: Vzhled aplikace Kibana. Převzato z [4].



Obrázek 3.2: Vzhled Bitergia Grimoire Development Dashboard. Převzato z [1].

3.1.3 Bugzilla

Bugzilla [2] je webová open source aplikace pro bugtracking (proces zaznamenávání, hlášení a správy informací o chybách, ke kterým dochází v softwaru) původně vyvinutá a používaná organizací Mozilla. Bugzilla byla původně napsána Terryem Weissmanem v programovacím jazyce Tcl, ale ještě před zveřejněním byla přepsána do Perlu. První dostupná verze byla Bugzilla 2.0.

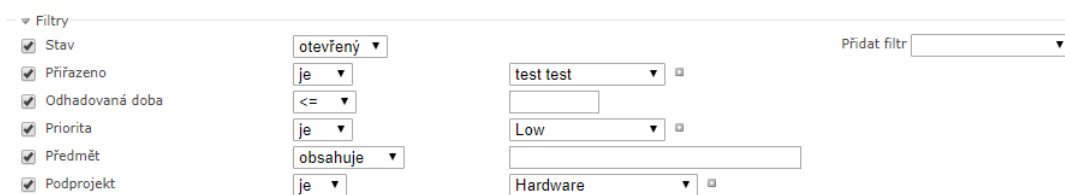
Bugzilla nabízí široké možnosti filtrování, ale díky tomu jsou filtry velmi komplikované. Všechny možnosti filtrování jsou vidět ihned po otevření panelu filtrů. Díky tomu jsou filtry sice přehledné, ale vyžadují velké množství místa. Z tohoto důvodu tento layout filtrů není vhodný pro SPADe.

3.1.4 Redmine

Redmine [6] je bezplatný a opensource webový nástroj pro správu projektů a issuetracking. Issuetracking systém je softwarová aplikace, která umožňuje podniku zaznamenávat a sledovat průběh každého problému neboli "issue", který uživatel počítačového systému identifikuje, dokud nebude problém vyřešen. Redmine umožňuje uživatelům spravovat více projektů a přidružených dílčích podprojektů. Pro každý projekt nabízí wiki, fórum, sledování času a nastavení přístupu založeného na rolích. Obsahuje kalendář a Ganttovy grafy, které pomáhají vizuálnímu znázornění projektů a jejich termínů. Redmine podporuje mnoho různých verzovacích systémů a obsahuje prohlížeč repozitářů a jejich změn. Redmine je napsaný s využitím frameworku Ruby on Rails. Je to multiplatformní a multidatabázový systém.

V nástroji Redmine se filtry přidávají postupným výběrem v rozbalovacím menu. Filtr se přidá okamžitě po výběru. Při přidání více filtrů se filtry skládají do řádků pod sebe. Jednotlivé filtry dynamicky mění svůj vzhled v závislosti na vyplněných datech. Každý filtr obsahuje zaškrtnutí pro jeho deaktivaci. Vzhled filtrů nástroje Redmine je vidět na obrázku 5.3.

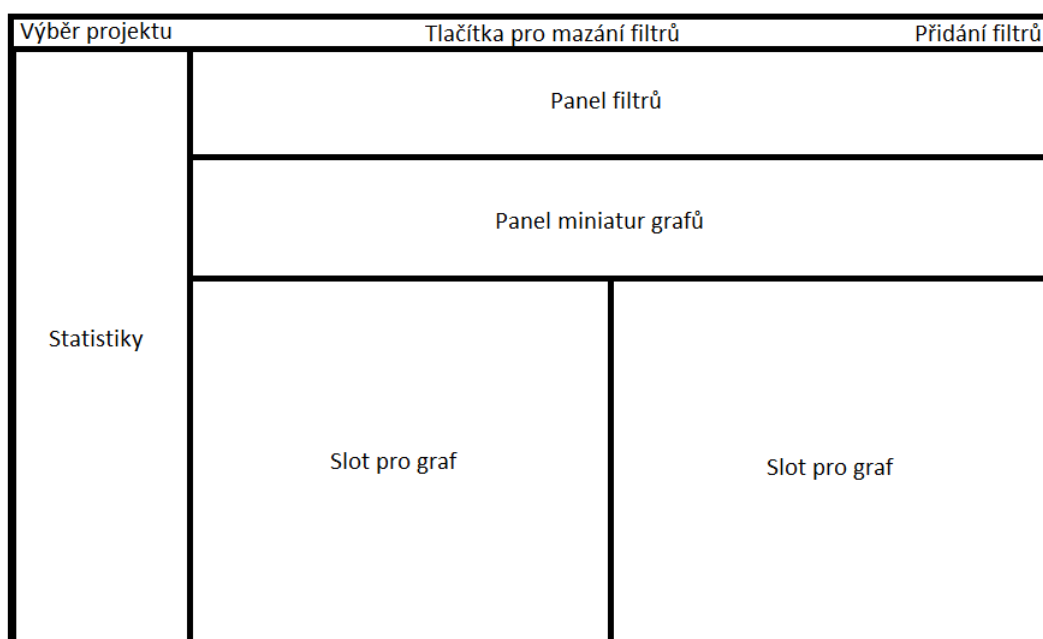
Nástroj Redmine byl využit jako hlavní inspirace pro nový vzhled filtrů aplikace SPADe a jejich logické uspořádání, přestože oproti SPADe jsou filtry v Redmine uzpůsobeny jen na jeden typ objektů (tickety).



Obrázek 3.3: Panel filtrů nástroje Redmine.

3.2 Analýza změn aplikace

Tato kapitola se zabývá analýzou požadavků na aplikaci z kapitoly 2.3 a způsobem jejich vyřešení. Na obrázku 6.1 je zobrazen návrh vzhledu a rozložení komponent hlavního okna nové verze SPADe GUI.



Obrázek 3.4: Návrh hlavního okna aplikace

3.2.1 Anglická lokalizace

V současné době nabízí SPADe GUI pouze české uživatelské rozhraní, což velmi limituje jeho případné využití. Proto bude přidána anglická lokalizace a program bude připraven pro přidávání dalších lokalizací, které půjdou přepínat za běhu programu. Výchozí jazyk aplikace bude změněn na angličtinu.

Za tímto účelem budou ze zdrojového kódu programu odstraněny textové řetězce, které budou přesunuty do externího souboru, ze kterého se budou načítat. Následně bude možno vytvořit více souborů pro různé jazyky a poté pouze určovat, ze kterého souboru má program řetězce načítat, čímž dojde k přepnutí jazyku programu. V rámci této práce bude přidán soubor s anglickými řetězci. Pro účely přepínání jazyků bude přidána horní ovládací lišta s nastavením, ve které bude možno jazyk programu přepnout.

3.2.2 Drag and drop panel

V současné implementaci nástroje SPADe GUI jsou všechny grafy vykreslovány do jednoho velkého panelu. I přes relativně malou velikost grafů, která ztěžuje čtení informací z grafů, je celkový panel velký a vyžaduje použití vertikálního i horizontálního scrollování, což nadále ztěžuje orientaci a práci v programu.

Z těchto důvodů bude tento panel zrušen a nahrazen panelem s miniaturami těchto grafů, ze kterých budou odstraněny popisy, osové čáry, kontextové menu a ponechán pouze název a náhled grafu. Tyto miniatury bude možno přetáhnout do dvou připravených slotů, které se budou nacházet pod panelem s miniaturami grafů. Tímto přetažením bude odstraněna miniatura grafu z panelu miniatur a objeví se jeho velká verze s popisky, osovými čarami a kontextovým menu. Pokud bude do obsazeného slotu přetažen jiný graf, dojde k nahrazení původního grafu novým grafem a původní graf se vrátí zpět do panelu miniatur.

3.2.3 Zvýšení přehlednosti

V současné implementaci mají statistiky podobu čtvercového panelu. Tímto dochází k plýtvání místem na obrazovce, protože prostor vedle statistik zůstával nevyužitý. Toto plýtvání se stalo ještě výraznějším po odstranění pů-

vodního panelu s grafy a jeho nahrazením panelem s miniaturami.

Proto bude panel statistik převeden do podoby postranního sloupce a umístěn do levé části hlavního okna.

Filtry budou přemístěny z levé části hlavního okna, kde se budou nacházet statistiky, do horní části okna nad panel s miniaturami. Layout filtrů bude změněn do podoby obdélníkového panelu s horizontálním scrolováním. Do panelu filtrů budou přidány popisky ke kontrolním tlačítkům a výběrovému menu. Dále bude přidána stromová struktura filtrů, tak aby filtry spadající pod úkoly byly přidávány z panelu filtru úkolů. Dále bude z panelu filtrů odstraněno menu pro výběr projektu a bude přesunuto na samostatný panel, který se bude nacházet v horní části obrazovky.

Nejdůležitějším prvkem hlavního okna budou dva sloty, které zobrazují aktuálně vybrané grafy. Aby tyto sloty mohly mít co největší velikost, bude možno schovávat panel filtrů a statistik. Uvolněné místo na obrazovce bude využito ke zvětšení těchto slotů, čímž dojde ke zlepšení jejich čitelnosti.

Pro schovávání těchto panelů budou vedle nich umístěna tenká tlačítka s popiskem šipky a možnost schování těchto panelů bude přidána i do menu nastavení na horní liště programu.

3.2.4 Uživatelem definované grafy

Pro ovládání uživatelem definovaných grafů bude přidáno menu Uživatelské grafy na horní ovládací lištu. V tomto menu bude tlačítko pro vytvoření uživatelského grafu. Pro účel vytvoření uživatelského grafu se otevře nové okno. V okně bude možné zvolit SQL dotaz a jeho parametry, který následně načte odpovídající data pro aktuálně zvolený projekt. Okno bude nabízet náhled tabulky vybraných dat. Nad každým sloupcem tabulky se budou nacházet ovládací prvky pro vykreslení grafu (použití dat v grafu, typ grafu a barva pro vykreslení v grafu). Ve spodní části okna bude náhled grafu, který se bude dynamicky měnit dle nastavení. Na obrázku 6.2 je zobrazen návrh vzhledu a rozložení komponent okna pro vytváření uživatelem definovaných grafů.

Pro ukládání uživatelských grafů bude sloužit tlačítko v okně pro vytvoření uživatelského grafu. Po jeho stisknutí se graf uloží jako miniatura do panelu miniatur v záložce uživatelské grafy. S takto uloženými grafy půjde

Výběr SQL					Název grafu
<input checked="" type="checkbox"/> použít	<input checked="" type="checkbox"/> použít	<input checked="" type="checkbox"/> použít	<input checked="" type="checkbox"/> použít	<input checked="" type="checkbox"/> použít	<input checked="" type="checkbox"/> použít
Barva	Barva	Barva	Barva	Barva	Barva
Typ grafu	Typ grafu	Typ grafu	Typ grafu	Typ grafu	Typ grafu
Název sloupce	Název sloupce	Název sloupce	Název sloupce	Název sloupce	Název sloupce
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data
Náhled grafu					
Uložení grafu			Uložení šablony		

Obrázek 3.5: Návrh okna pro vytváření uživatelských grafů.

pracovat stejně jako s ostatními grafy. Grafy půjdou přetáhnout do slotů pro zvětšení a budou mít kontextové menu.

Pro editaci uživatelských grafů bude přidáno tlačítko editace do jejich kontextového menu. Kontextové menu grafu se objeví, pokud je graf přetažen do slotu a je na něj kliknuto pravým tlačítkem myši. Po stisknutí tlačítka editace se otevře okno se stejným nastavením jako při uložení grafu. Poté bude možné graf uložit pod stejným jménem a tím přepsat původní verzi grafu nebo název změnit a zachovat obě verze.

Pro ukládání šablon uživatelských grafů bude sloužit tlačítko v okně pro vytvoření uživatelského grafu. Po jeho stisknutí se šablona přidá do horního menu uživatelských grafů. V tomto menu budou tlačítka pro vytvoření a smazání dané šablony. V případě stisknutí tlačítka pro vytvoření grafu ze šablony se otevře okno pro vytvoření uživatelského grafu se stejným nastavením, jako když byla šablona uložena.

3.2.5 Rozšíření možností filtrování

Možnost filtrování bude rozšířena o další atributy pro filtrování jako je například strávený čas, velikost artefaktu, nebo datum konce iterace. Dále bude přidána možnost spojovat filtry úkolů za pomoci logických funkcí AND, OR, NOT a XOR.

Bude odebráno tlačítko přidej filtr, které sloužilo pro přidání filtru, poté co byl vybrán v menu filtrů. Filtry se budou automaticky přidávat ve chvíli, kdy dojde k jejich výběru v menu.

Po aktivování filtru bude do volného prostoru mezi menu pro výběr projektu a menu pro výběr filtru na horní liště aplikace přidáno tlačítko, které bude sloužit k odstranění daného filtru. Po jeho stisknutí dojde k vymazání daného filtru a tlačítko zmizí. Pokud bude aktivní více než jeden filtr, objeví se také tlačítko pro odstranění všech filtrů.

V aktuální implementaci dochází při změně projektu k vymazání nastavených filtrů. Z důvodu zvýšení uživatelské přívětivosti a zrychlení práce bude nastavení filtrů udržováno a bude se odstraňovat pouze to nastavení filtrů, které je specifické pro data projektu (např. názvy iterací nebo osob).

4 Implementace

Tato kapitola se věnuje implementaci aplikace. Aplikace vychází z původní implementace aplikace SPADe GUI a to především z původního datového modelu a jeho načítání.

Aplikace je napsaná v jazyce Java, pro vykreslení grafů používá knihovnu JFreechart a data načítá z MySQL databáze za použití rozhraní JDBC.

Třídy aplikace jsou rozděleny do balíků `data`, `data.ciselnik`, `data.custom`, `data.polozky`, `databaze`, `gui` a `ostatni`.

4.1 Balík `data.polozky`

Balík `data.polozky` obsahuje datové třídy pro uložení základních dat z databáze. Tento balík obsahuje třídy `Polozka`, `PolozkaCiselnik`, `PolozkaVytvoreni` a `PolozkaPocatek`. Z těchto tříd dědí všechny třídy z balíků `data` a `data.ciselnik`.

4.2 Balík `data`

Třídy v tomto balíku slouží pro uložení nejdůležitějších dat z databáze. Balík `data` obsahuje třídy `Projekt`, `Ukol`, `Segment`, `Artefakt`, `Konfigurace` a `Osoba`.

Všechny třídy v balíku `data` obsahují metodu `nactiData()` a její přetíženou verzi `nactiData(podminky pro načtení)`, tyto metody slouží pro načtení dat do seznamů dané třídy. Metoda `nactiData()` načte z databáze všechna data týkající se dané třídy, zatímco metoda `nactiData(podminky pro načtení)` načte pouze data odpovídající zadaným podmínkám. Jako dané podmínky zde slouží seznamy vyhovujících id v podobě `ArrayList<Integer>`. Jedinou třídou, která tyto metody neobsahuje je třída `Ukol`, protože neobsahuje žádné seznamy k načtení.

4.3 Balík `data.ciselnik`

Tento balík obsahuje třídy `Ciselnik`, `Priority`, `Severity`, `Resoluce`, `Status`, `Typ` a `Osoby`. Nejdůležitější třídou tohoto balíku je abstraktní třída `Ciselnik`, ze které všechny ostatní třídy tohoto balíku dědí.

4.3.1 Třída `Ciselnik`

Abstraktní třída `Ciselnik` obsahuje dva atributy. Atribut `seznam`, který je typu `arraylist` a obsahuje seznam položek daného číselníku. Druhým atributem této třídy je `typ`, který udává typ daného číselníku. Nejdůležitější metodou této třídy je abstraktní metoda `nactiPolozky(int idProjekt)`, kterou musí implementovat všechny třídy zděděné z této třídy. V této metodě musí jednotlivé třídy načíst příslušná data daných číselníků.

4.4 Balík `data.custom`

Třídy v tomto balíku slouží pro uložení dat uživatelských grafů a šablon pro vytvoření uživatelských grafů. Všechny třídy v tomto balíku jsou nově vytvořené. Balík obsahuje třídy `CustomGraf`, `nastaveniCustomSloupec`, `prepravkaUkladaniCustom` a `sablonaCustomGrafu`.

4.4.1 Třída `CustomGraf`

Třída `CustomGraf` slouží pro uložení dat potřebných pro vykreslení uživatelského grafu. Třída má tři atributy. Atribut `ArrayList<String> nazvySloupcu` slouží pro uložení názvů sloupců tabulky dat. Atribut `ArrayList<String> datумы` slouží pro uložení sloupce tabulky s časovými údaji. Poslední atribut `ArrayList<ArrayList<Double>> data` slouží pro uložení hodnot z tabulky dat. Všechny atributy této třídy jsou soukromé, ale třída obsahuje metody pro získání příslušných dat.

4.4.2 Třída `PrepravkaUkladaniCustom`

Třída `PrepravkaUkladaniCustom` slouží pro uložení vytvořeného uživatelského grafu na disk. Třída obsahuje následujících pět atributů `String` `Nazev` (obsahuje název grafu), `int` `projectID` (ID projektu, ke kterému se graf vztahuje), `int` `typGrafu` (určuje typ grafu) a `JFreeChart` `panel` (panel s vykresleným grafem). Všechny atributy této třídy jsou soukromé, ale třída obsahuje getry pro všechny atributy.

4.4.3 Třída `NastaveniCustomSloupec`

Třída `NastaveniCustomSloupec` slouží pro uložení informací pro vykreslení jednoho sloupce z tabulky dat uživatelského grafu. Třída obsahuje následující tři atributy `boolean` `pouzit` (udává, zda se mají data z vybraného sloupce vykreslit do grafu), `int` `typGrafu` (udává typ grafu, v jakém se mají data z vybraného sloupce vykreslit do grafu) a `Color` `barva` (udává barvu, v jaké se mají data z vybraného sloupce vykreslit do grafu). Všechny atributy této třídy jsou soukromé, ale třída obsahuje getry a setry pro všechny atributy.

4.4.4 Třída `SablonaCustomGrafu`

Třída `SablonaCustomGrafu` slouží pro uložení vytvořené šablony uživatelského grafu na disk a editaci již vytvořených grafů. Třída obsahuje atributy `String` `Nazev` (obsahuje název šablony), `Projekt` `projekt` (Odkaz na projekt. Používá se při editaci existujícího grafu, při vytváření nového není použit.), `int` `sql` (určuje SQL pro vykreslení grafu), `int` `osoby` (určuje zvolené osoby pro vykreslení grafu), `int` `iterace` (určuje zvolené iterace pro vykreslení grafu) a `ArrayList<nastaveniCustomSloupec>` `sloupce` (arraylist s nastavením pro vykreslení jednotlivých sloupců tabulky dat). Třída má soukromý konstruktor, ale obsahuje veřejné metody pro vytvoření instance třídy v závislosti na zvoleném SQL a jeho parametrech.

4.5 Balík databaze

Tato kapitola popisuje balík databaze, který slouží k výběru dat z databáze. Balík `databaze` obsahuje DAO (Data Access Object) třídy a jejich rozhraní, zajišťující výběr dat z databáze pomocí API JDBC. Balík obsahuje třídy `ArtefaktDAO`, `CiselnikyDAO`, `KonfiguraceDAO`, `ProjektDAO` a `UkolDAO`, `IArtefaktDAO`, `ICiselnikyDAO`, `IKonfiguraceDAO`, `IProjektDAO` a `IUkolDAO`. V konstruktoru každé třídy je přiřazeno připojení k databázi, které se navazuje po přihlášení uživatele.

4.6 Balík gui

Balík `gui` obsahuje třídy grafického rozhraní aplikace. Tento balík byl nejvíce pozměněn oproti původní verzi implementace. Tento balík obsahuje několik typů tříd:

- třídy pro vykreslení oken – `OknoPrihlasovani`, `OknoProgresNacitani`, `OknoHlavni`, `PanelProjektu`,
- třídy pro vykreslení panelů s filtry – `PanelFiltr`, `PanelFiltrCiselnik`, `PanelFiltrPolozkaPocatek`, `PanelFiltrPolozkaVytvoreni`, `PanelFiltrCas`, `PanelFiltrPocatekSegment`, `PanelFiltrPolozkaVytvoreniArtefakt`,
- třídy pro vykreslení grafů – `PanelGrafuRodic`, `PanelGrafuArtefakt`, `PanelGrafuKonfigurace`, `PanelGrafuSegment`, `PanelGrafuUkol`, `PanelGrafuCustom`,
- třídy pro vykreslení uživatelem definovaných grafů – `OknoCustomGraf`, `PanelCustomNahled`, `PanelSloupceCustomGrafu`,
- třídy pro drag and drop – `DragGestureHandler`, `DropHandler`, `DropChartPanel`, `PanelDataFlavor`, `PanelTransferable`,
- ostatní třídy – `ColorChooserButton`, `CheckBoxbezNul`, `TlacitkoMazaniGrafu`, `TlacitkoEditaceGrafu`.

4.6.1 Třídy pro vykreslení oken

Tato skupina tříd se stará o vykreslení oken aplikace.

Třída `OknoPrihlasovani`

Instance této třídy je vytvořena ihned po zapnutí aplikace. Třída se stará o vykreslení přihlašovacího okna. Pokud jsou vyplněny správné přihlašovací údaje, třída naváže za pomoci JDBC připojení k databázi a vytvoří instanci hlavního okna aplikace.

Třída `OknoProgresNacitani`

Z důvodu velkého objemu dat v databázi zabere načítání určitý čas, je proto potřeba uživateli sdělit co se děje. Okno `OknoProgresNacitani` se spouští při načítání dat a ukazuje průběh načítání a jaká data se právě načítají.

Třída `OknoHlavni`

Tato třída slouží pro zobrazení hlavního okna aplikace. Přihlašovací okno předává této třídě v konstruktoru připojení k databázi. Třída následně v konstruktoru připojení nastaví do proměnné `PRIPOJENI` ve třídě `Konstanty`.

Poté je spuštěna metoda `nactiProjekty()`, která načte z databáze seznam projektů pro jejich zobrazení v rozbalovacím menu s výběrem projektu. Data samotného projektu se načítají při jeho výběru v menu. Při prvním spuštění okna se načtou data prvního projektu v seznamu.

Po dokončení načítání dat je spuštěna metoda `nastavZobrazeni()`, která nastaví okno a přidá do něj příslušné komponenty. Následně se spustí metoda `nastavAkce()`, která přiřadí příslušné akce ovládacím prvkům okna.

Třída `PanelProjektu`

Jedná se o hlavní panel hlavního okna. Stará se o rozložení panelu statistik, filtrů, drop slotů a miniatur grafů. Třída ve svém konstruktoru načte

data aktuálně vybraného projektu. Poté spustí metodu `nastavZobrazeni()`, která přidá do panelu příslušné komponenty. V této metodě se také vytvoří panel statistik za pomoci metody `getPopisProjektu()`.

4.6.2 Třídy pro vykreslení panelů s filtry

Tyto třídy slouží pro zobrazení panelů s filtry. Hlavní třídou balíku je `PanelFiltr`, ze kterého všechny ostatní třídy tohoto balíku dědí.

Třída `PanelFiltr`

`PanelFiltr` je abstraktní třída, ze které dědí všechny ostatní panely filtrů. V konstruktoru se nastavuje seznam hodnot, které filtr obsahuje. Následně se spustí metoda `nastavPanel()`, která nastaví titulek panelu, přidá checkbox pro použití filtru a menu se seznamem hodnot pro výběr. Tyto tři komponenty obsahují všechny panely filtrů. Dále třída obsahuje abstraktní metodu `nastavAkce()`, která slouží pro přiřazení akcí komponentám a ostatní třídy ji musí implementovat. Další abstraktní metodou je `getSeznamId()`, která slouží k vrácení seznamu id položek, které odpovídají nastavení filtru.

Třída `PanelFiltrCiselnik`

Třída se používá pro filtrování datových tříd zděděných z třídy `Ciselnik`. Těmito třídami jsou `Priority`, `Severity`, `Status`, `Typ`, `Resoluce` a `Osoby`. Panel ke zděděné funkcionalitě z `PanelFiltr` přidává možnost filtrování podle třídy a supertřídy.

Třída `PanelFiltrPolozkaPocatek`

Tato třída je zděděná ze třídy `PanelFiltr`. Ke zděděné funkcionalitě přidává možnost filtrování položek podle jejich data počátku. Datum počátku obsahují datové třídy `Ukoly` a `Segmenty` (Iterace, Fáze, Aktivity).

Třída PanelFiltrPolozkaPocatekSegment

Tato třída je zděděná ze třídy `PanelFiltrPolozkaPocatek`. Ke zděděné funkcionalitě přidává možnost filtrování položek podle jejich data konce. Dále je přidáno menu pro výběr logické vazby mezi filtrováním datem počátku a konce. Datum počátku a konce obsahuje datová třída `Segment` (Iterace, Fáze, Aktivita).

Třída PanelFiltrPolozkaVytvoreni

Tato třída je zděděná ze třídy `PanelFiltr`. Ke zděděné funkcionalitě přidává možnost filtrování položek dle data vytvoření. Datum vytvoření obsahují datové třídy `Konfigurace` a `Artefakt`.

Třída PanelFiltrPolozkaVytvoreniArtefakt

Tato třída je zděděná ze třídy `PanelFiltrPolozkaVytvoreni`. Ke zděděné funkcionalitě přidává možnost filtrování položek dle velikosti souboru. Velikost souboru obsahuje datová třída `Artefakt`.

4.6.3 Třídy pro vykreslení panelů s grafy

Třídy pro vykreslování grafů jsou `PanelGrafuRodici`, `PanelGrafuSegment`, `PanelGrafuUkol`, `PanelGrafuKonfigurace`, `PanelGrafuArtefakt` a `PanelGrafuCustom`. Tyto třídy slouží pro vykreslení a zobrazení miniatur grafů v hlavním okně. Všechny tyto třídy jsou zděděné z třídy `PanelGrafuRodici`.

Třída PanelGrafuRodici

`PanelGrafuRodici` je abstraktní třída zděděná ze třídy `JPanel`. Třída obsahuje abstraktní metodu `vlozGrafy()`, kterou musí ostatní třídy implementovat, a která slouží pro vytvoření a vložení příslušných grafů do panelu. Dále třída obsahuje metody `nastavZobrazeni()`, `nastavGraf()` a `vratMiniaturu()`. Metoda `nastavZobrazeni()` slouží k nastavení velikosti všech grafů v panelu, metoda `nastavGraf()` slouží k nastavení zobrazení grafu dle

jeho typu a metoda `vratMiniaturu()` slouží pro vrácení miniatury grafu do panelu, po tom co byl nahrazen ve slotu pro graf jiným grafem.

Třídy `PanelGrafuSegment`, `PanelGrafuUkol`, `PanelGrafuKonfigurace`, `PanelGrafuArtefakt`, `PanelGrafuCustom`

Tyto třídy jsou zděděné ze třídy `PanelGrafuRodic`. Každý z těchto panelů obsahuje metody pro vytvoření svých příslušných grafů a jejich následné zobrazení.

4.6.4 Třídy pro vykreslení uživatelem definovaných grafů

Tyto třídy slouží pro vytvoření uživatelsky definovaných grafů a šablon pro uživatelsky definované grafy.

Třída `OknoCustomGraf`

Tato třída slouží pro zobrazení okna, které vytváří uživatelsky definované grafy a šablony. Při vytváření okna se nejprve volá metoda `nastavZobrazeni()`, která vytvoří příslušné komponenty okna. Poté se volá metoda `nastavMenu()`, která přidá ovládací prvky dle parametrů zvoleného SQL. Následně se metodou `nactiData()` načtou a zobrazí data odpovídající zvolenému SQL. Nakonec metoda `nastavAkce()` nastaví akce daným komponentám. Jelikož okno obsahuje živý náhled výsledného grafu, všechny ovládací prvky kromě svých primárních akcí spouštějí ještě akci pro překreslení náhledu.

O vykreslení grafu se stará metoda `nakresliGraf()`, která vytvoří datasety ze zvolených dat a ty následně předá odpovídajícímu konstruktoru třídy `PanelCustomNahled`.

Třída `PanelCustomNahled`

Třída `PanelCustomNahled` je zděděná ze třídy `JPanel`. Tato třída slouží pro vytvoření a zobrazení náhledu uživatelsky definovaného grafu. `PanelCus-`

`tomNahled` má dva konstruktory. Jeden konstruktor slouží pro vytvoření kombinovaných grafů a druhý pro vytvoření koláčových grafů.

4.6.5 Třídy pro drag and drop

Tyto třídy slouží pro přetahování grafů mezi panelem miniatur a sloty pro grafy.

Třída `DragGestureHandler`

Třída `DragGestureHandler` slouží jako handler pro gesto přetažení. Tato třída implementuje třídy `DragGestureListener` a `DragSourceListener`. Nejdůležitějšími metodami třídy jsou `dragGestureRecognized()` a `dragDropEnd()`. Metoda `dragGestureRecognized()` slouží pro zachycení gesta přetažení. Odebere položku z containeru, vytvoří transferable wrapper a zahájí proces přetažení. Metoda `dragDropEnd()` slouží pro ukončení přetažení, v případě neúspěšného přetažení vrátí panel předchozímu rodiči.

Třída `DropHandler`

Tato třída slouží jako handler pro zpracování dropu drag and drop gesta. Implementuje třídu `DropTargetListener`. Kromě přetažení grafu se stará o povolení kontextového menu přetaženého grafu, zvětšení velikosti přetaženého grafu a vrácení nahrazeného grafu zpět do panelu miniatur.

Třída `DropChartPanel`

Tato třída je zděděna ze třídy knihovny JFreechart `ChartPanel`. Ke zděděné funkcionalitě přidává podporu pro drag and drop. Proměnná třídy `boolean dragable` slouží pro zakázání přetahování, poté co je panel přetažen z panelu miniatur do slotu.

4.6.6 Ostatní třídy balíku gui

Třída ColorChooserButton

Tlačítko, které po stisku otevře okno pro výběr barvy. Třída je zděděná ze třídy JButton. ColorChooserButton vykresluje ikonu obdélníku ve zvolené barvě místo popisku tlačítka. Třída obsahuje vlastní metodu pro přidávání action listenerů addColorChangeListener(), aby docházelo k upozornění listenerů při zvolení barvy a ne při prvním kliku na tlačítko.

Třída CheckBoxbezNul

Třída je zděděna ze třídy JCheckBox a implementuje třídy Serializable a ActionListener. Slouží pro přidání checkboxu do kontextových menu sloupcových grafů, který slouží pro schování nulových hodnot v daných grafech.

Třída TlacitkoEditaceGrafu

TlacitkoEditaceGrafu je zděděno ze třídy JMenuItem a implementuje třídu ActionListener. Slouží pro přidání možnosti editace grafu do kontextového menu uloženého uživatelského grafu.

4.7 Balík ostatni

V balíku ostatni jsou třídy Ukladani a Konstanty.

4.7.1 Třída Ukladani

Třída Ukladani slouží pro ukládání a načítání uživatelských grafů a šablon pro uživatelské grafy. Třída ukládá grafy do souboru CustomCharts.dat a šablony pro grafy do souboru Templates.dat.

Hlavními metodami třídy jsou:

- `static void save()` – uloží všechny grafy do souboru CustomCharts.dat,

- `static void save(File file)` – uloží všechny grafy do souboru daným parametrem `file`,
- `static void save(File file, int id)` – uloží grafy projektu specifikovaného parametrem `id` do souboru daným parametrem `file`,
- `static void load()` – načte grafy ze souboru `CustomCharts.dat`,
- `static void saveSablony()` – uloží všechny šablony do souboru `Templates.dat`,
- `static void loadSablony()` – načte všechny šablony ze souboru `Templates.dat`.

4.7.2 Třída Konstanty

Třída `Konstanty` slouží pro ukládání hodnot využívaných v celé aplikaci. Například velikost komponent, cesty k souborům, `properties` s popisky a další.

5 Zhodnocení

Tato kapitola popisuje průběh testování výsledné aplikace, rozdíl v implementaci mezi původní a novou verzí aplikace SPADe GUI, dosažené výsledky práce, nedostatky současné implementace a možná budoucí rozšíření aplikace.

5.1 Testování

K testování aplikace byly použity metody bílé i černé skříňky. Metodu bílé skříňky používal autor práce při psaní kódu aplikace. K tomuto účelu byly vytvořeny JUnit testy, kterých je celkem 43 a pokrývají 70,4 % kódu. Pro správné fungování testů je nutné dočasně změnit některé modifikátory přístupu na `public`.

Pro testování metodou černé skříňky bylo využito pět testerů. Testerů nejprve podrobili aplikaci monkey testům (náhodnému klikání na ovládací prvky aplikace). Následně testerů provedli testování dle předpřipravených scénářů. Po dokončení testování dle scénářů byly testerů seznámeni se starou verzí aplikace, aby mohli provést srovnání nové verze oproti původní. K tomuto účelu byl použit předpřipravený dotazník.

Během testování aplikace se neobjevily žádné závažné chyby. Většina objevených chyb byla pouze kosmetických. Nalezené chyby byly opraveny a následně byly provedeny regresní testy (testy, zda nová úprava aplikace nezhoršila původní funkčnost).

Ve všech otázkách dotazníku, kromě rychlosti běhu aplikace, hodnotili testerů aplikaci kladně nebo neutrálně. Pět z pěti testerů ohodnotilo aplikaci jako uživatelsky příjemnější, s přehlednějšími grafy a s větším množstvím funkcionality. Jeden z pěti testerů označil původní verzi jako rychlejší, což bylo nejspíše zapříčiněno rychlostí připojení. Zbývající testerů hodnotili rychlost obou verzí jako srovnatelnou. Detailní výsledky dotazníku jsou k vidění v příloze D.

Akceptační testy byly prováděny v průběhu celého vývoje aplikace během

konzultačních schůzek vedoucím bakalářské práce.

5.2 Rozsah změn implementace

K porovnání implementací byla použita funkce compare nástroje GitHub, která porovnávala první inicializační commit s posledním commitem. Původní implementace měla 46 souborů a 7,269 řádků. Nová implementace má 73 souborů a 14,297 řádků. Z porovnání vyplývá, že celkem bylo přidáno 28 souborů, 1 soubor byl odstraněn a 44 bylo změněno. Celkem bylo přidáno 8,255 (57,7 %) řádků a 1,227 (8,6 %) bylo odstraněno.

5.3 Dosažené výsledky

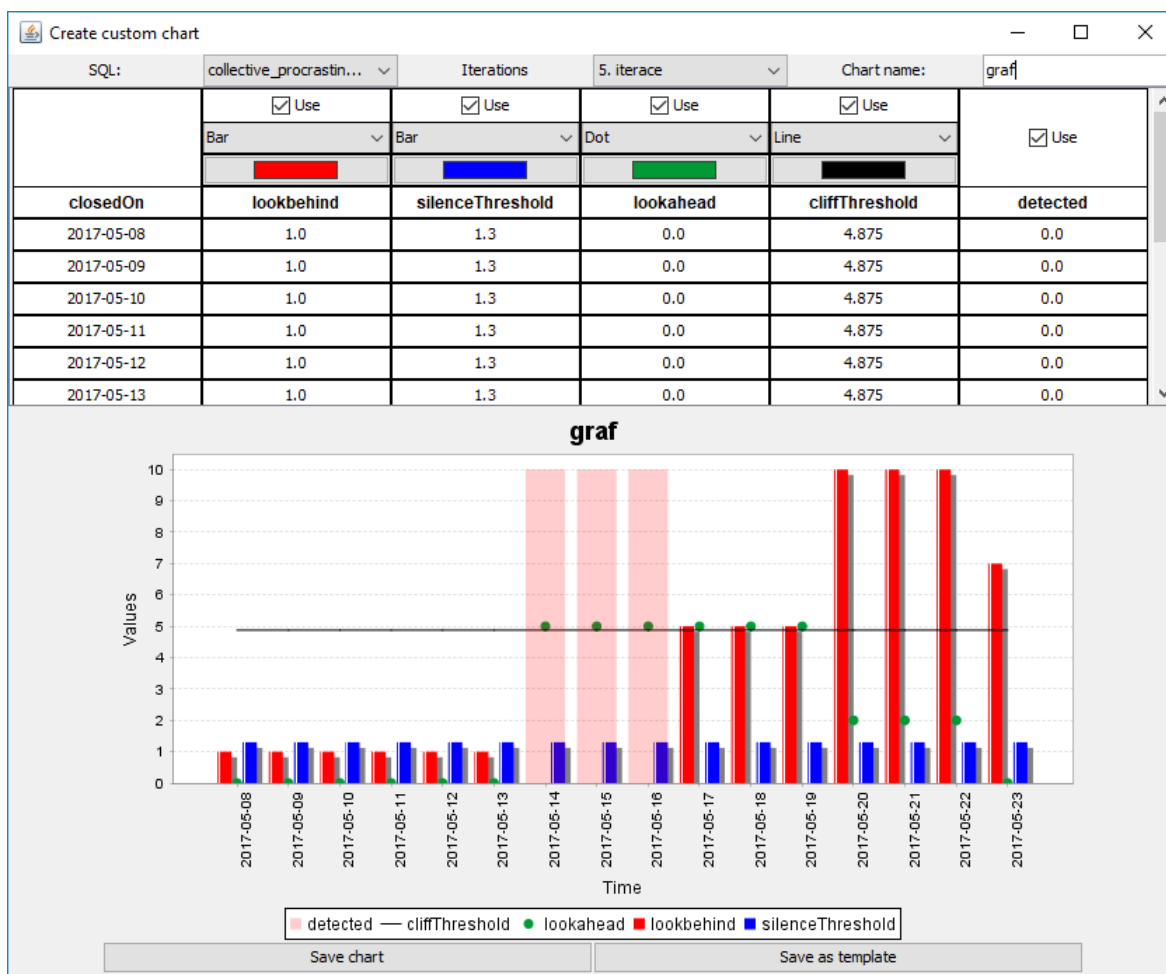
Do aplikace SPADe GUI byla přidána anglická lokalizace a aplikace byla připravena pro přidávání dalších lokalizací. Výchozí jazyk aplikace byl změněn na angličtinu. Lokalizace je možné přepínat za běhu programu.

Dále byl změněn layout celého okna aplikace za účelem větší přehlednosti. Původní panel s grafy byl nahrazen záložkovým panelem s miniaturami, které lze přetáhnout do slotů a tím je zvětšit. Byl změněn layout panelu filtrů a panelu statistik a přidána možnost schovat oba panely za účelem uvolnění místa pro grafy. Vzhled nové a původní verze aplikace je možné porovnat na obrázcích 9.2 a 9.3.

Filtry byly rozšířeny o nové filtrovací možnosti a filtry úkolů je možné spojovat logickými funkcemi AND, OR, XOR a NOT. Filtry se přidávají hned po jejich zvolení v menu a byla přidána tlačítka pro smazání aktivních filtrů. Při přepínání projektů se zachovává nastavení filtrů, které není specifické pro daný projekt.

Největším přínosem práce je přidání možnosti vytvářet uživatelsky definované grafy a jejich šablony. Díky tomuto rozšíření je možné vykreslovat libovolné grafy a detekovat v nich anti-patterny, což je primární účel nástroje SPADe. Vytvořené grafy je možné importovat, exportovat a editovat. Vzhled okna pro vytváření uživatelských grafů je zobrazen na obrázku 9.1.

V dotazníku uživatelského testování většina testerů hodnotila novou verzi



Obrázek 5.1: Vzhled okna pro vytváření uživatelských grafů a jejich šablon.

aplikace jako uživatelsky přívětivější, s přehlednějšími grafy a filtry a větší funkcionalitou.

5.4 Nedostatky současné implementace

Nedostatkem současné implementace je nemožnost změny lokalizace kontextových menu grafů a v okně pro výběr barvy za běhu programu. Jelikož jako výchozí jazyk aplikace byla zvolena angličtina, byla obě tato okna nastavena do angličtiny.

Dalším nedostatkem je pouze "pásový" zoom u uživatelských grafů, což je zapříčiněno vlastnostmi knihovny JFreechart.

Z důvodu načítání dat pro uživatelsky definované grafy separátním SQL dotazem nepodléhají tyto grafy nastavení filtrů, to však nebylo požadavkem práce.

5.5 Budoucí rozšíření

U aplikace se počítá s budoucím rozšířením. Možným budoucím rozšířením aplikace by mohlo být přidání dalších lokalizací aplikace, na což je aplikace připravena. Přidáním dalších lokalizací by se opět zvýšil okruh potenciálních uživatelů aplikace.

Dalším rozšířením by mohlo být přidání dalších SQL pro načítání dat uživatelsky definovaných grafů. Případně přidání možnosti vytvářet tyto SQL dotazy přímo v okně pro vytváření uživatelsky definovaných grafů.

Uživatelské grafy by se mohly dále rozšířit o možnost vytváření víceúrovňových koláčových grafů a možnost skládání sloupcových grafů na sebe.

Dále by bylo možné ještě více rozšířit filtry o další filtrovací možnosti a přidat možnost filtrování uživatelsky definovaných grafů.

Parametry programu

Rozšíření nástroje Bubble Navigator (R...)

Fit vzhledu Přidat fit

Ukázky Použít fit

Konfigurace s nastavením a zadáním
 Sestavení ze s AGM procesem
 Příprava prostředí pro vývoj
 Karta s modely pro vývoj

Datum podniku:

Konfigurace Použít fit

Datum vytvoření:

Iterace

Znovi fit

Konfigurace na:	0	14.4	69
Číslo:	0	127	119
Artikl na:	0	6.1	31
Číslo:	0	6.1	31

Rozřazení segmentů v čase - Fáze

Datum	01.01.1975	01.01.1990	01.01.1975	01.01.2000
1900				

Iterace

1. Iterace
2. Iterace
3. Iterace
4. Iterace
5. Iterace

Rozřazení segmentů v čase - Iterace

Datum	15.03.2017	30.03.2017	14.04.2017	29.04.2017	14.05.2017

Aktivity

1. Iterace
2. Iterace
3. Iterace
4. Iterace
5. Iterace

Rozřazení s

Datum	01.01.1900	01.01.1925

Priority

Priority - Tridy

Pr

Obrázek 5.2: Vzhled původní verze aplikace.

SPADL
File, Custom chart, Settings

Program parameters

Remove all **Tasks X** **Phase X** **Activities X** **Activities** **Apply/Filter**

Task selection: Rozšířená nástrojová biblioteka

Use filter: Tasks: Start date: between End date: Filter tasks by:

Phase selection: Use filter: Phase:

Project information:
Name: Rozšířená nástrojová biblioteka
Navigator (NLP) - ASWI Brothers
Start date: 08.03.2017
End date: Unlimited

Element count:
Phase: 5
Iterations: 5
Activities: 5
Tasks: 61
Configurations: 644
Branches: 374
Tags: 6
Artifacts: 61

Tasks: Segments, Configurations, Artifacts, Custom charts

Resolution - Classes

Priorities

Resolution - Superclasses

Types

Types - Classes

Assigned

Phase

Statistics: **Min Average Max**

Tasks per:	Min	Average	Max
Phase:	8	12.2	15
Iteration:	8	12.2	15
Activity:	8	12.2	15
Person:	0	6.1	19

Task:
Time estimated: 1.0 3.0 8.0
Time spent: 0.5 3.65 11.5

Configuration per:
Person: 0 14.4 69

Artifact per:
Configuration: 0 1.27 119
Person: 0 6.1 31

Cumulative artifacts cumulative count over time

Obrázek 5.3: Vzhled nové verze aplikace.

6 Závěr

Cílem této práce bylo rozšířit a opravit případné nedostatky aplikace SPADe GUI, která slouží ke znázornění dat v datovém skladu nástroje SPADe.

Pro splnění tohoto cíle bylo potřeba se seznámit s projektovým řízením vývoje software a možnostmi prezentace dat softwarových projektů (kapitola 2.1). Dále bylo nutné seznámit se s nástrojem SPADe a implementací aplikace SPADe GUI (kapitola 2.2). Následně bylo zapotřebí identifikovat nedostatky a možná rozšíření současné implementace aplikace SPADe GUI (kapitola 2.3). Z důvodu vyřešení požadavků z kapitoly 2.3 bylo třeba analyzovat nástroje podobné nástroji SPADe (kapitola 3).

Výstupem této práce je nová implementace aplikace SPADe GUI spolu s popisem její implementace (kapitola 4), dokumentací zdrojového kódu aplikace, uživatelské příručky v příloze B a otestováním této implementace (kapitola 5.1).

Bakalářská práce splňuje zadání ve všech bodech a je připravena k obhajobě.

Seznam zkratek

ALM Application Lifecycle Management

API Application Programming Interface

DAO Data Access Object

ETL Extract – Transform – Load

FAV Fakulta aplikovaných věd

GUI Graphical User Interface

JDBC Java Database Connectivity

KIV Katedra informatiky a výpočetní techniky

KPI Key Performance Indicators

SPADe Software Process Anti-pattern Detector

ZČU Západočeská univerzita v Plzni

Seznam obrázků

2.1	Ganttův graf. Převzato z [11]	14
2.2	Výšečový graf	15
2.3	Sloupcový graf	16
2.4	Histogram. Převzato z [11]	16
2.5	Schéma nástroje SPADe. Převzato z [21]	17
2.6	Doménový model datového skladu SPADe. Převzato z [21]	19
2.7	Architektura aplikace. Převzato z [26]	23
3.1	Vzhled aplikace Kibana. Převzato z [4].	25
3.2	Vzhled Bitergia Grimoire Development Dashboard. Převzato z [1].	26
3.3	Panel filtrů nástroje Redmine.	28
3.4	Návrh hlavního okna aplikace	28
3.5	Návrh okna pro vytváření uživatelských grafů.	31
5.1	Vzhled okna pro vytváření uživatelských grafů a jejich šablon.	46
5.2	Vzhled původní verze aplikace.	48
5.3	Vzhled nové verze aplikace.	49
B.1	Okno pro přihlášení	58
B.2	Hlavní okno aplikace.	60
B.3	Okno pro vytváření uživatelských grafů.	61
B.4	Okno pro vytváření uživatelských grafů.	62

Literatura

- [1] Bitergia Dashboards [on-line]. Dostupné z: <https://bitergia.com/products/dashboards/>.
- [2] Bugzilla [on-line]. Dostupné z: <http://www.bugzilla.org>.
- [3] JDBC Overview [on-line]. Dostupné z: <http://www.oracle.com/technetwork/java/overview-141217.html>.
- [4] Kibana [on-line]. Dostupné z: <https://www.elastic.co/products/kibana>.
- [5] MySQL [on-line]. Dostupné z: <https://www.mysql.com/>.
- [6] Redmine [on-line]. Dostupné z: <http://www.redmine.org>.
- [7] AMBLER, S. W. – HOLITZA, M. *Agile for Dummies*. Hoboken, NJ: John Willey & Sons, Inc., 2012. ISBN 9781118305065.
- [8] AMBLER, S. W. – LINES, M. *Disciplined Agile Delivery (DAD): A Practitioner's Guide to Agile Software Delivery in the Enterprise*. IBM Press, 2012. ISBN 0132810131.
- [9] COHEN, E. Project Management Methodologies: Everything You Need to Know [on-line]. Dostupné z: <https://www.workamajig.com/blog/project-management-methodologies>.
- [10] DANEL, R. *Datový sklad [on-line]*. Vysoká škola báňská. Dostupné z: http://homel.vsb.cz/~dan11/is_skripta/IS%202010%20-%20Danel%20-%20Datovy%20sklad.pdf.
- [11] DAWN, L. – MCQUOWN, G. Charting Your Course: Charts and Graphs for IT Projects [on-line]. Dostupné z: <http://analytics.ncsu.edu/sesug/2003/SE14-Li.pdf>.
- [12] DUNCAN, W. R. The process of project management [on-line]. Dostupné z: <https://www.pmi.org/learning/library/basic-process-project-management-2114>.

- [13] GRIFFIN, L. What is Application Lifecycle Management (ALM)? [on-line]. Dostupné z: <https://study.com/academy/lesson/what-is-application-lifecycle-management-alm-definition-tools.html>.
- [14] HEFNEROVÁ, L. *Lean software development, bakalářská práce*. Vysoká škola ekonomická v Praze, 2012.
- [15] ISTQB. What is V-model - advantages, disadvantages and when to use it? [on-line], . Dostupné z: <http://istqbexamcertification.com/what-is-v-model-advantages-disadvantages-and-when-to-use-it/>.
- [16] ISTQB. What is Waterfall model - advantages, disadvantages and when to use it? [on-line], . Dostupné z: <http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/>.
- [17] JFREE. JFreeChart [on-line]. Dostupné z: <http://www.jfree.org/jfreechart/>.
- [18] KROLL, P. – KRUCHTEN, P. *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. Addison-Wesley Longman Publishing Co., Inc., 2003. ISBN 0-321-16609-4.
- [19] PÍCHA, P. – BRADA, P. ALM Tool Data Usage in Software Process Metamodeling. In *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, s. 1–8, Aug 2016. doi: 10.1109/SEAA.2016.37.
- [20] PÍCHA, P. et al. Towards Architects Activity Detection through a Common Model for Project Pattern Analysis. In *2017 IEEE International Conference on Software Architecture Workshops (ICSAW)*, s. 175–178, April 2017. doi: 10.1109/ICSAW.2017.46.
- [21] PÍCHA, P. *Datový model nástroje SPADe a jeho mapování na projektová data z ALM nástrojů, referát*. Západočeská univerzita.
- [22] SCHWABER, K. – SUTHERLAND, J. The Scrum Guide [on-line]. 2014. Dostupné z: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>.
- [23] SCHWALBE, K. *Information Technology Project Management*. Cengage Learning, 2009. ISBN 9780324786927.

- [24] SOURCEMAKING.COM. Fire Drill [on-line]. Dostupné z: <https://sourcemaking.com/antipatterns/fire-drill>.
- [25] STAMELOS, I. Software Project Management Anti-patterns. *J. Syst. Softw.* January 2010, 83, 1, s. 52–59. ISSN 0164-1212. Dostupné z: <http://dx.doi.org/10.1016/j.jss.2009.09.016>.
- [26] VŠELKO, M. *Uživatelské rozhraní prezentující obsah datového skladu nástroje SPADe, bakalářská práce*. Západočeská univerzita, 2017.

Přílohy

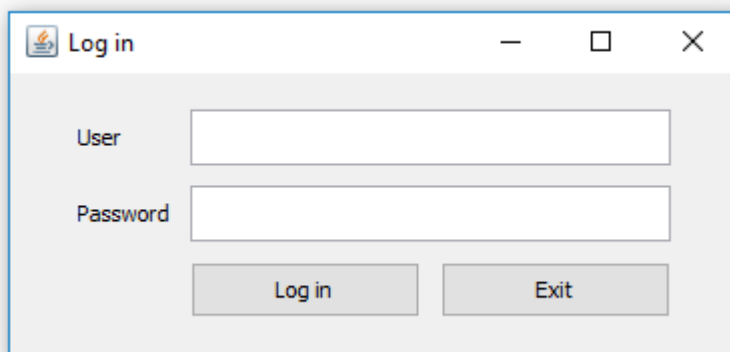
A Obsah přiloženého DVD

- **Dokumentace**
 - **text** – PDF soubor s textem bakalářské práce.
 - **zdroj** – zdrojové soubory textu bakalářské práce a obrázky.
 - **Javadoc** – dokumentace aplikace.
- **Projekt**
 - **.settings**
 - **lib** – externí knihovny aplikace.
 - **bin** – přeložené class soubory.
 - **src** – zdrojové kódy.
- **Původní verze programu**
 - **GUISpade.jar** – spustitelná aplikace.
 - **czech.properties** – soubor textů pro aplikaci v češtině.
- **Spustitelný program**
 - **GUISpade.jar** – spustitelná aplikace.
 - **czech.properties** – soubor textů pro aplikaci v češtině.
 - **english.properties** – soubor textů pro aplikaci v angličtině.
 - **sql.properties** – SQL dotazy pro uživatelsky definované grafy.
 - **sqlVar.properties** – parametry pro SQL dotazy.
 - **CustomCharts.dat** – soubor pro uložení uživatelsky definovaných grafů.
 - **Templates.dat** – soubor pro uložení šablon uživatelsky definovaných grafů.
- **Testování**
 - **JUnit testy** – zdrojové kódy použitých JUnit testů.
 - **Scénáře** – scénáře pro uživatelské testování ve formátu PDF.
- **CTLME.txt** – obsahuje tento rozpis obsahu CD nosiče

B Uživatelská dokumentace

B.1 Okno přihlašování

Pro spuštění aplikace je potřeba nainstalovat Java JRE 1.8. Po spuštění aplikace se zobrazí přihlašovací okno pro zadání uživatelského jména a hesla. Okno přihlášení je zobrazeno na obrázku B.1.



Obrázek B.1: Okno pro přihlášení

Tlačítkem „Exit“ se aplikace ukončí. Po zadání platného názvu uživatelského účtu a hesla se lze přihlásit tlačítkem „Log in“. Přihlašovací údaje je možné získat od Ing. Petra Píchy (případně od současného správce nástroje SPADe). Po ověření zadaných údajů se spustí načítání dat z databáze. Po dokončení načítání dat se přes celou obrazovku spustí hlavní okno aplikace.

B.2 Hlavní okno

Na obrázku B.2 je k vidění hlavní okno aplikace spolu s číselnými popisky jednotlivých komponent.

- **Číslo 1** – označuje rozbalovací menu pro výběr projektu. Po výběru projektu se spustí načítání dat zvoleného projektu.

- **Číslo 2** – označuje tlačítka, která se objeví po zapnutí filtrů a slouží pro jejich odstranění.
- **Číslo 3** – označuje rozbalovací menu pro výběr filtru. Po výběru filtru se objeví panel s daným filtrem.
- **Číslo 4** – označuje tlačítko, které slouží pro zapnutí navolených filtrů.
- **Číslo 5** – označuje panel, ve kterém lze nastavit parametry jednotlivých přidaných filtrů.
- **Číslo 6** – označuje panel se statistikami o projektu.
- **Číslo 7** – označuje záložkové menu, ve kterém lze zvolit jakou skupinu miniatur grafů chceme.
- **Číslo 8** – označuje panel s miniaturami grafů, které lze přetáhnout do slotů označených číslem 9 a tím je zvětšit.
- **Číslo 9** – označuje dva sloty pro přetažení miniatur grafů.
- **Číslo 10** – označuje horní ovládací menu. V tomto menu je možné měnit jazyk programu, vytvářet a exportovat uživatelské grafy nebo schovávat panely hlavního okna.
- **Číslo 11** – označuje tlačítko pro schování panelu filtrů.
- **Číslo 12** – označuje tlačítko pro schování panelu statistik.

B.3 Okno uživatelských grafů

Obrázek B.3 zobrazuje okno pro vytváření uživatelsky definovaných grafů a jejich šablon. Obrázek obsahuje číselné popisky jednotlivých komponent okna.

- **Číslo 1** – označuje rozbalovací menu pro výběr SQL dotazu pro načtení dat.
- **Číslo 2** – označuje rozbalovací menu pro výběr parametru pro zvolené SQL.

SPADE
File Custom chart Settings 10

Program parameters Rozšířen nástroje Bubble Navigator (N... 1 Remove all Tasks X Phase X 2 3 Phase 4 Apply filter

Filter selection
Tasks
5
Phase
Use filter
Tasks
Start date: between ... Filter tasks by: Priorities
5 Iterace
4 Iterace
3 Iterace
2 Iterace
1 Iterace

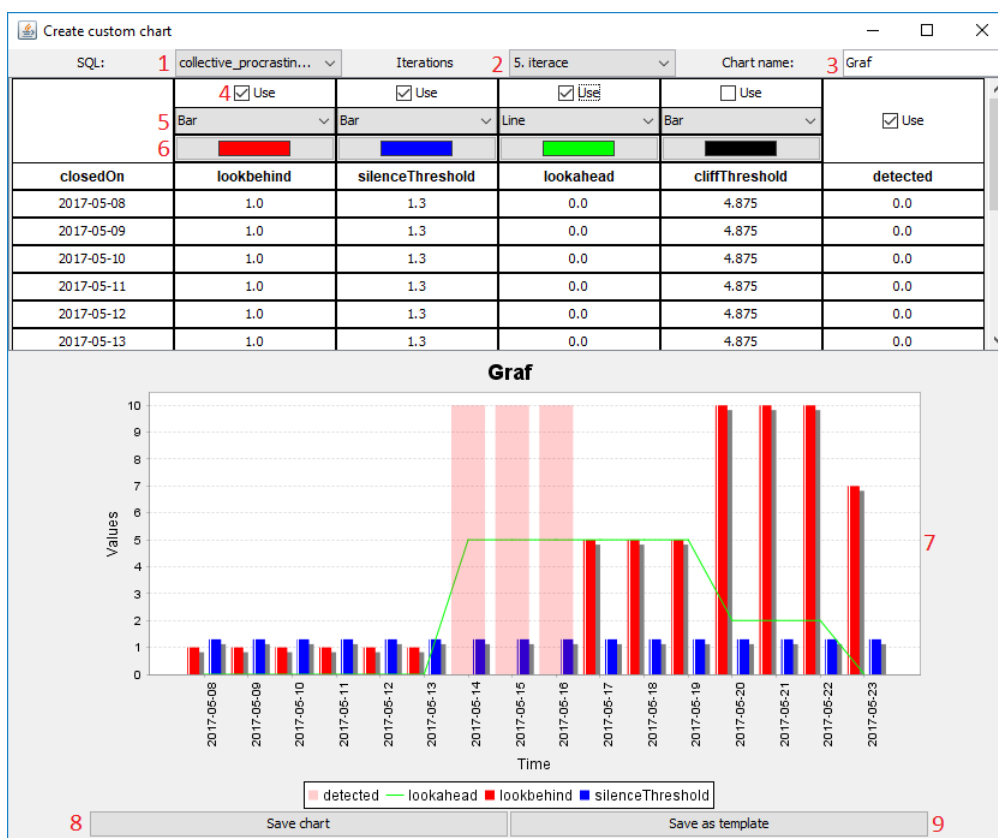
Project information
Name: Rozšířen nástroje Bubble Navigator (NLP) - ASWIT/brothers 6
Start date: 08.03.2017
End date: Unlimited
Elements count:
Phase: 5
Iterations: 5
Activities: 5
Tasks: 61
Configurations: 644
Branches: 374
Tags: 6
Artifacts: 61

Tasks Segments Configurations Artifacts Custom charts 7
8
Priorities
Priorities - Classes
Priorities - Superclasses
Severity
Severity - Classes
Severity

Statistics
Tasks per: Min Average Max
Phase: 8 122 15
Iteration: 8 122 15
Activity: 8 122 15
Person: 0 6.1 19
Task:
Time estimated: 1.0 3.0 8.0
Time spent: 0.5 3.65 11.5
Configuration per:
Person: 0 14.4 69
Artifact per:
Configuration: 0 1.27 119
Person: 0 6.1 31

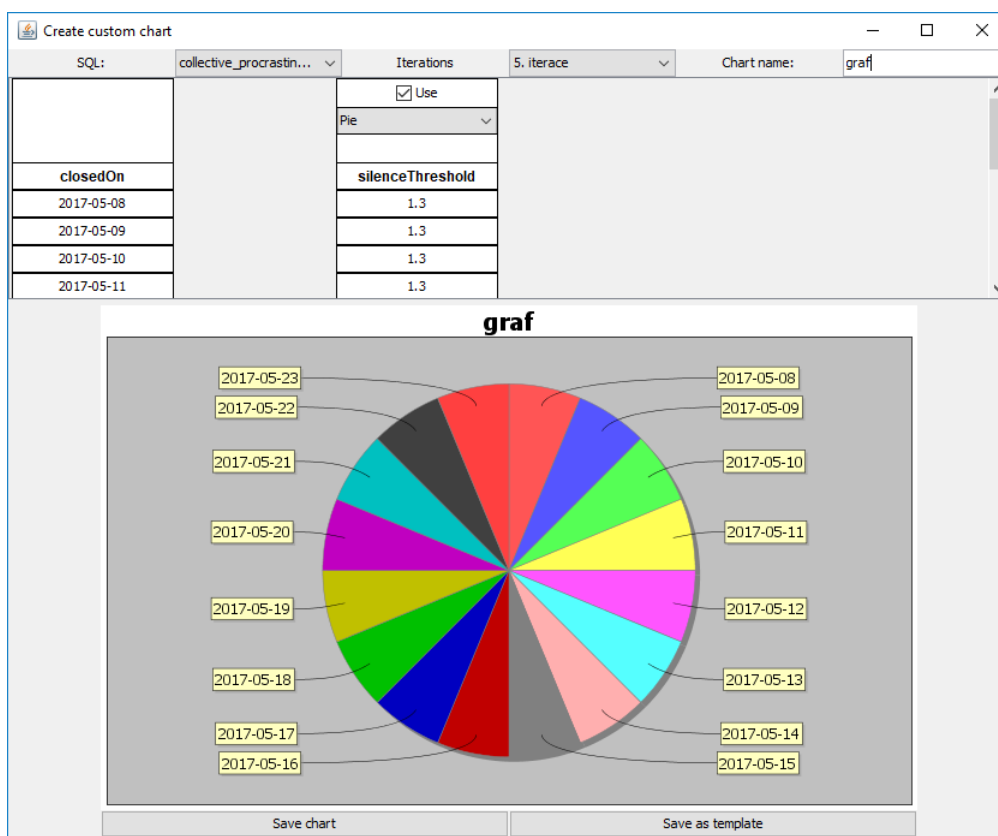
Drop chart here 9 Drop chart here 12

Obrázek B.2: Hlavní okno aplikace.



Obrázek B.3: Okno pro vytváření uživatelských grafů.

- Číslo 3 – označuje textové pole pro zadání názvu vytvářeného grafu nebo šablony.
- Číslo 4 – označuje zaškrtnutí, které určuje, zda se mají data z daného sloupce zobrazit v grafu.
- Číslo 5 – označuje rozbalovací menu pro výběr typu grafu pro data z daného sloupce. V případě zvolení koláčového grafu u některého ze sloupců, bude zakázán výběr ostatních sloupců. Viz obrázek B.4.
- Číslo 6 – označuje tlačítko, které slouží pro výběr barvy dat daného sloupce. Po stisku tlačítka se otevře nové okno s výběrem barvy.
- Číslo 7 – označuje náhled, jak bude vypadat výsledný graf.
- Číslo 8 – označuje tlačítko, které uloží vytvořený graf. Takto uložené grafy se objeví v záložce Custom Charts panelu miniatur.



Obrázek B.4: Okno pro vytváření uživatelských grafů.

- **Číslo 9** – označuje tlačítko, které uloží nastavení grafu jako šablonu. Z takto uložené šablony je posléze možné vygenerovat graf s tímto nastavením pro různá data.

C Ukázky testovacích scénářů

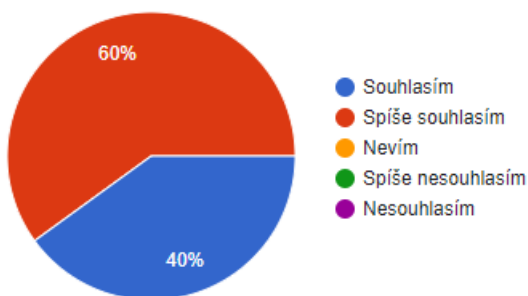
Print Test Case: TC.02.01 Přihlašovací okno

Test Case TC.02.01 Přihlašovací okno		
<u>Author:</u>	Lukáš Haringer	
<u>Summary:</u> Kontrola rozložení prvků v přihlašovacím okně		
<u>#:</u>	<u>Step actions:</u>	<u>Expected Results:</u>
1	Spustí se program	Otevře se okno Přihlášení
2	Okno má popisek Log in	
3	Existuje pole pro zapsání loginu	
4	Pole pro zapsání loginu má titulek User	
5	Existuje pole pro zapsání hesla	
6	Pole pro zapsání hesla má titulek Password	
7	Existuje tlačítko Log in	
8	Existuje tlačítko Exit	
<u>Execution type:</u>	Manual	
<u>Estimated exec. duration (min):</u>		
<u>Importance:</u>	Medium	
<u>Requirements</u>	None	
<u>Keywords:</u>	None	

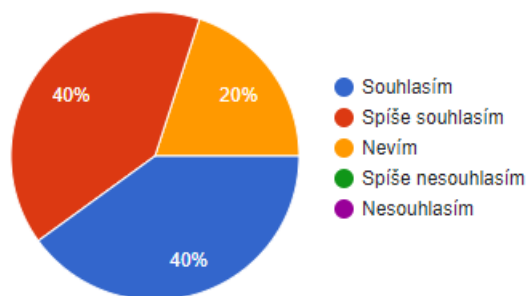
Test Case TC.05.03 Import/Export custom grafů		
<u>Author:</u>	Lukáš Haringer	
<u>Summary:</u>	Import/Export custom grafů	
<u>Preconditions:</u>	<p>Je spuštěna aplikace SPADe GUI.</p> <p>Je zvolen libovolný projekt.</p> <p>Je vytvořený alespoň jeden custom graf ve zvoleném projektu</p>	
<u>#:</u>	<u>Step actions:</u>	<u>Expected Results:</u>
1	Zvolení položky Custom charts v horním ovládacím menu	Rozbalí se ovládací menu
2	Zvolení položky Import/Export of charts v menu Custom charts	Rozbalí se ovládací menu
3	Zvolení položky Exports charts of this project v menu Import/Export of charts	Otevře se okno pro výběr souboru pro export
4	Potvrzení výběru souboru	Vytvoří se soubor s exportovanými grafy
5	Zvolení položky Custom charts v horním ovládacím menu	Rozbalí se ovládací menu
6	Zvolení položky Remove Charts v menu Custom charts	Rozbalí se ovládací menu
7	Zvolení položky Remove all charts from this project v menu Custom charts	Odstraní se všechny custom grafy z projektu
8	Zvolení položky Custom charts v horním ovládacím menu	Rozbalí se ovládací menu
9	Zvolení položky Import/Export of charts v menu Custom charts	Rozbalí se ovládací menu
10	Zvolení položky Import charts from a file v menu Import/Export of charts	Otevře se okno pro výběr souboru pro import
11	Potvrzení výběru souboru	Exportované a poté smazané grafy se vrátí zpět do projektu
<u>Execution type:</u>	Manual	
<u>Estimated exec. duration (min):</u>		
<u>Importance:</u>	Medium	
<u>Requirements</u>	None	
<u>Keywords:</u>	None	

D Výsledky uživatelských testů

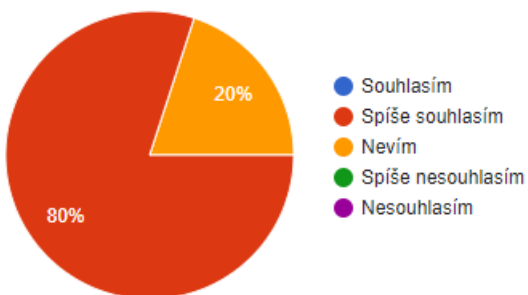
Ovládání aplikace je intuitivní.



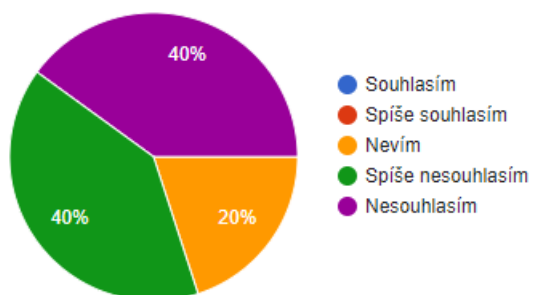
Myslím, že většina lidí by se naučila s touto aplikací pracovat rychle.



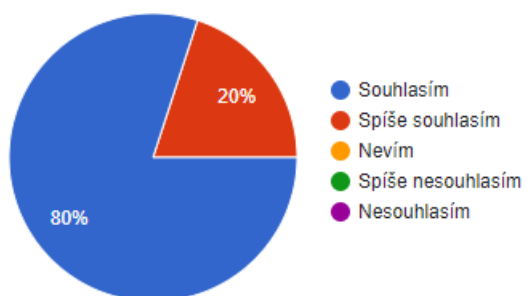
Aplikace pracovala rychle



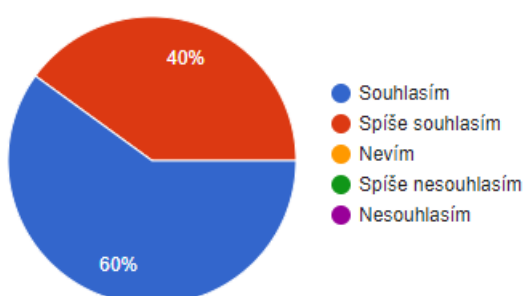
Ovládání aplikace je zbytečně složité



Tlačítka byla dobře organizovaná a snadno k nalezení.



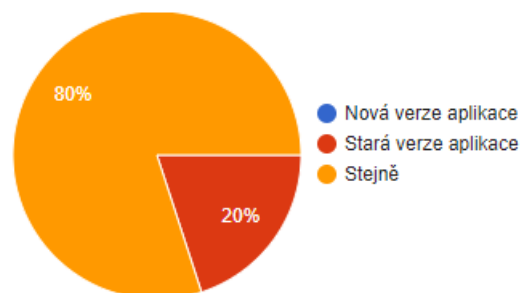
Okamžitě jsem rozuměl/a funkcím všech tlačítek.



Uživatelsky příjemnější byla



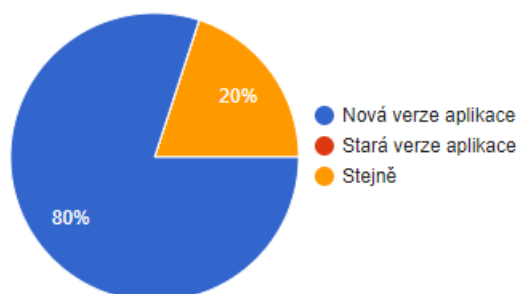
Rychleji pracovala



Grafy byly přehlednější v



Statistiky byly přehlednější v



Filtry byly přehlednější v



Více funkcionality nabízela



Více se mi líbil vzhled



Celkově lepší byla

