

ТАЛЛИНСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет информационных технологий
Институт информатики

IDK40LT

Максим Андреев 123886 IABV

АНАЛИЗ И ПРИМЕНЕНИЕ ИНСТРУМЕНТОВ ДЛЯ ЛОКАЛИЗАЦИИ И ТЕСТИРОВАНИЯ

Бакалаврская работа

Руководитель: Инна Шварцман
Магистр

Таллинн 2017

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Informaatikainstituut

IDK40LT

Maksim Andrejev 123886 IABB

LOKALISEERIMISE JA TESTIMISE VAHENDITE ANALÜÜS JA KASUTAMINE

Bakalaurusetöö

Juhendaja: Inna Švartsman
Magister

Tallinn 2017

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Maksim Andrejev

03.01.2017

Аннотация

Целью данной дипломной работы является рассказ в общих чертах о локализации, её тестировании, возможных встречающихся трудностях и проблемах, а также анализ инструментов, способных упростить процесс локализации и тестирования.

Во второй главе рассказывается о локализации в целом, о её необходимости. Рассматриваются этапы локализации, а также всевозможные сопутствующие проблемы и трудности. Вдобавок повествуется о различных нюансах и особенностях, которые нужно учитывать при локализации.

В третьей главе речь идет непосредственно о тестировании - каким образом оно производится и что для этого необходимо.

В четвертой главе мы проводим анализ инструментов для локализации и тестирования. Сравниваем, какая из двух предложенных систем имеет больше плюсов и, соответственно, лучше.

Данная работа написана на русском языке и содержит 42 страницы, 5 глав, 19 иллюстраций, 10 таблиц.

Annotatsioon

Lokaliseerimise ja testimise vahendite analüüs ja kasutamine

Antud bakalarausetöö eesmärgiks on kirjeldada lokaliseerimise protsessi üldjoontes, testida lokaliseerimise protsessi, uurida, millised probleemid ja raskused võivad tekkida selle protsessi rakendamisel, analüüsida vahendeid lokaliseerimise protsessi lihtsustamiseks ja testimiseks.

Teises peatükis vaadeldakse lokaliseerimist ja selle vajadust. Kirjeldatakse kõik lokaliseerimise etapid ja igasugused selle protsessiga kaasnevad probleemid ja raskused. Lisaks räägitakse erinevatest nüanssidest ja eriomadusest, mis tuleb ka arvestada.

Kolmas osa kirjeldab testimist - kuidas testitatakse ja milleks see protsess nii vajalik.

Neljandas peatükis viiakse läbi lokaliseerimise ja testimise vahendite analüüs. Võrreldatakse kaks süsteemi ning saadud tulemuste põhjal valitakse parimat.

Lõputöö on kirjutatud vene keeles ning sisaldab teksti 42 leheküljel, 5 peatükki, 19 joonist, 10 tabelit.

Abstract

Analysis and usage of localization and testing tools

The main goal of this thesis is to tell in general about localization, its testing, possible occurring problems and troubles, also to make an analysis of tools, which can simplify the process of localization and testing.

In the second chapter is generally told about localization and its necessity. We look into stages of localization and all sorts of associated problems and difficulties. Moreover, is told about different nuances and features, which should consider during localization.

In the third chapter speech is directly about testing - how it is made and what is necessary for it.

In the fourth chapter is made analysis of localization and testing tools. We compare, which of two proposed systems has more advantages and consequently is better.

The thesis is in Russian and contains 42 pages of text, 5 chapters, 19 figures, 10 tables.

Словарь сокращений и понятий

ad hoc testing	интуитивное тестирование
i18n (англ. internationalization)	интернационализация
l10n (англ. localization)	локализация
баг (англ. bug)	ошибка в программе или системе
локаль (англ. locale)	региональные настройки. Зачастую служит синонимом слова "язык"
ОС	операционная система
ПО	программное обеспечение
тест кейс (англ. test case)	документация для тестировщика, где описана последовательность действий и ожидаемый результат

Содержание

1 Введение	12
2 Локализация	14
2.1 Для чего нужна локализация	14
2.2 Понятие интернационализации	14
2.3 Понятие локализации	15
2.3.1 Формат даты и времени	16
2.3.2 Формат бумаги	16
2.3.3 Символы валюты	17
2.3.4 Система мер	17
2.3.5 Формат телефонных номеров	17
2.3.6 Особенности человеческих имен	18
2.3.7 Символы на клавиатуре	19
2.4 Этапы локализации	19
2.4.1 Начальный шаг (обсуждение проекта, подготовка, анализ)	20
2.4.2 Перевод	21
2.4.3 Разработка (внедрение)	22
2.4.4 Тестирование	22
2.5 Трудности	23
2.6 Пример неудавшейся локализации	24
3 Тест кейсы	25
3.1 Тест кейсы в теории	25
3.2 Тест кейсы на практике	26
3.3 Примеры	29
4 Инструменты для локализации и её тестирования	31
4.1 Инструменты для визуализации	31
4.1.1 ALF	31
4.1.2 Slate	32
4.2 Системы отслеживания ошибок	34
4.2.1 Watson	35

4.2.2 Jira	38
4.3 Сравнение обеих систем	40
5 Заключение.....	41
Использованная литература	42

Перечень рисунков

Рисунок 1. Каскадная и итеративная модели процесса разработки программного обеспечения [11].	12
Рисунок 2. Зависимость суммы, требуемой на исправление ошибки, от времени нахождения ошибки [10].	13
Рисунок 3. Этапы локализации [4].	20
Рисунок 4. Интерфейс программы MemoQ.	21
Рисунок 5. Пример тест кейса в системе wiki.corp.adobe.com	29
Рисунок 6. Пример тест кейса в системе Jira.	30
Рисунок 7. Интерфейс программы ALF.	31
Рисунок 8. Интерфейс программы Slate.	32
Рисунок 9. Программа Slate. Функция проверки на согласованность.	33
Рисунок 10. Программа Slate. Статистика	33
Рисунок 11. Программа Slate. Статистика. Фильтрация по языку.	33
Рисунок 12. Программа Slate. Пример автобаги.	34
Рисунок 13. Интерфейс инструмента Watson.	35
Рисунок 14. Watson. Протоколирование ошибки.	36
Рисунок 15. Watson. Созданный, исправленный и уже закрытый протокол ошибки.	37
Рисунок 16. Jira. Возможность регулирования полей при вводе баги.	38
Рисунок 17. Jira. Список предназначенных для пользователя баг.	39
Рисунок 18. Jira. График созданных и разрешенных проблем.	39
Рисунок 19. Jira. Создание фильтра.	40

Перечень таблиц

Таблица 1. Пример различных форматов даты и времени.	16
Таблица 2. Пример различных размеров и форматов бумаги по стандарту ISO 216.	16
Таблица 3. Сравнение стандарта ANSI с ISO 216.	17
Таблица 4. Пример различных телефонных номеров.	18
Таблица 5. Список локалей на некоторых проектах Adobe Systems.	27
Таблица 6. Пример тест кейса для приложения Acrobat Scan.	27
Таблица 7. Пример тест кейса для приложения Adobe Sign.	27
Таблица 8. Пример тест кейса для приложения Adobe Sign.	28
Таблица 9. Пример тест кейса для приложения Adobe Sign.	28
Таблица 10. Пример тест кейса для приложения Acrobat Scan.	29

1 Введение

Мы живем в век стремительного развития инфотехнологий. Каждый день в свет выходят новые программы, игры, приложения и так далее. Малосведущему пользователю кажется, что выпуск нового продукта (а тем более "всего лишь" выпуск новой версии) не так-то сложен и времязатрачен. На самом деле, этому предшествует длительный этап разработки, который порой может длиться годами.

Существует множество различных моделей процесса разработки программного обеспечения. У каждой есть как свои плюсы, так и минусы. Рассмотрим две из них: итеративную ("iteration") и каскадную ("waterfall"). Основные различия мы можем рассмотреть на нижеприведенной картинке (Рисунок 1). Отталкиваясь от увиденного, можно сказать, что основное различие заключается в том, что у каскадной модели следующая фаза начинается только после полного завершения предыдущей. Данная особенность многими оценивается как главный минус этой модели. Особенность другой модели, исходя из названия ("iteration" - повторение), как раз в том, что все фазы повторяются. Наглядный пример "превосходства" итеративной модели над каскадной можно увидеть на следующем изображении (Рисунок 2). Чем раньше будет найдена ошибка/сбой/неисправность - тем лучше. Вовремя найденная ошибка может сэкономить много времени и средств.

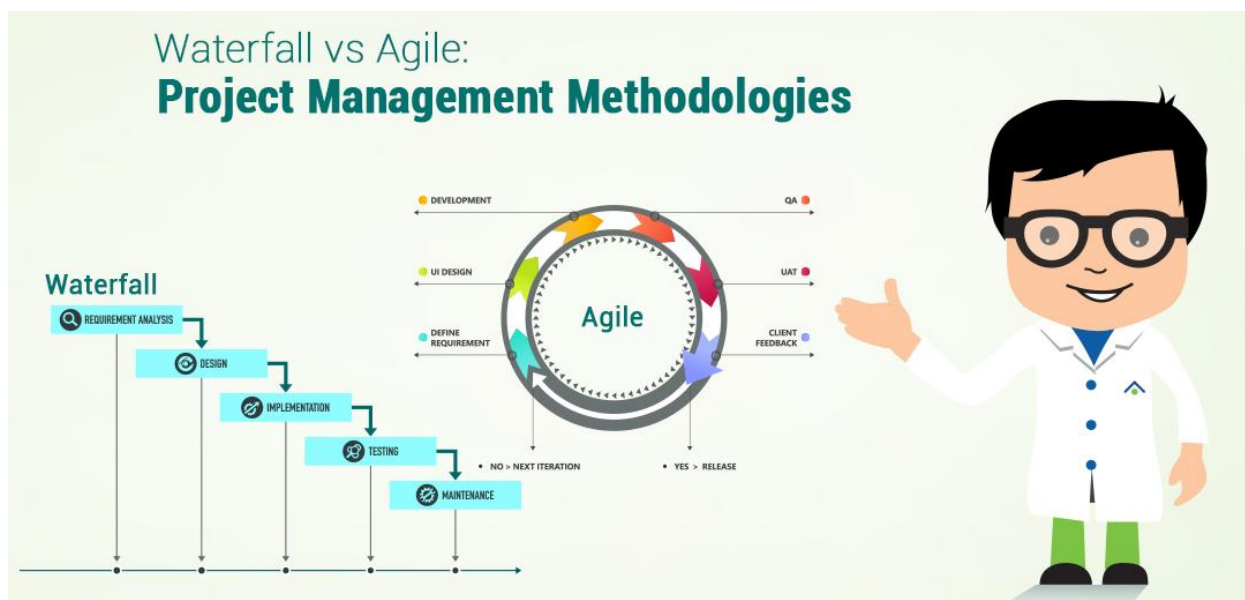


Рисунок 1. Каскадная и итеративная модели процесса разработки программного обеспечения [11].

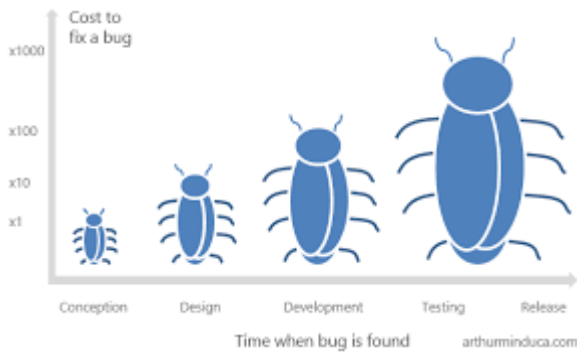


Рисунок 2. Зависимость суммы, требуемой на исправление ошибки, от времени нахождения ошибки [10].

Вот мы и подошли к главному: на Рисунке 1 мы видим, что фаза “Тестирование” присутствует в обеих моделях. Также она встречается во всех остальных моделях, которые не были упомянуты, но которые также используются при разработке программного обеспечения. Исходя из Рисунка 2, можно сказать, что данной фазой никак нельзя пренебрегать (впрочем, как и остальными). Далее мы более подробно рассмотрим именно эту фазу.

Тестирование - процесс, целью которого является обнаружение всех ошибок и недочетов программы и дальнейшее их устранение. Тестирование проходит многократно и конечная цель - дойти до той стадии, когда ошибки практически отсутствуют.

Существует множество разных типов классификации тестирования:

По знанию системы [1,2]:

- Тестирование черного ящика
- Тестирование белого ящика

По степени автоматизации:

- Мануальное (ручное) тестирование
- Автоматизированное тестирование

По времени проведения тестирования:

- Альфа-тестирование
- Бета-тестирование

По объекту тестирования:

- Функциональное тестирование
- Тестирование безопасности
- Тестирование производительности
- Тестирование локализации
- и прочие

Данный список можно продолжать и продолжать, так как классифицировать можно по разным критериям и стандартам. Мы же в дальнейшем затронем только малую составляющую тестирования - тестирование локализации.

Целью работы будет попытаться как можно более подробно рассказать о локализации, её тестировании, встречающихся трудностях и проблемах, а также о вспомогательных инструментах, которые сильно облегчают процесс локализации и тестирования.

2 Локализация

2.1 Для чего нужна локализация

Начнем с того, что же такое локализация и для чего вообще она нужна? С каждым годом, месяцем, днем наш мир становится всё более глобализованным. Всё становится более взаимосвязанным друг с другом, границы между государствами постепенно исчезают, происходит сближение и слияние различных культур. Данный процесс необратим.

Все эти процессы происходят и в сфере инфотехнологий. Данные явления означают то, что всевозможные компьютерные продукты (игры, приложения, браузеры и так далее; одним словом - ПО) нужно производить не только для одной страны и на одном языке (в которой производится), но и для всего остального мира. А для этого продукт должен быть переведен на другие языки и ориентирован на реалии других стран.

Все крупнейшие корпорации, фирмы, а также производители чего бы то ни было заинтересованы в как можно большем количестве потребителей. Для этого производимый ими товар должен быть на доступном для этих самых потребителей языке. Поэтому вся мало-мальская продукция переводится, то есть локализуется, на другие языки. Локализация способствует расширению возможностей, выходу на новые рынки, а также увеличению прибыли, к чему, собственно, и стремятся все компании.

Более того, распространение нелокализованного ПО в некоторых странах запрещено законодательно, что подразумевает о необходимости локализации.

Итак - локализация. Локализация – это лишь часть процесса адаптации, которой всегда предшествует глобализация или интернационализация. Эти термины используются для обозначения способа проектирования ПО, при котором возможность многоязыковой поддержки закладывается с самого начала.

2.2 Понятие интернационализации

Слова "интернационализация" и "глобализация", по сути, являются синонимами, поэтому объясняя значение одного, мы одновременно объясняем и другое.

Интернационализация - это создание и развитие содержания продукта, программы или документации так, чтобы позволить лёгкую локализацию для рынков, различающихся по культуре, региону или языку.

Интернационализация обозначается как "*i18n*", где 18 - это число букв между '*i*' and '*n*' в английском варианте слова "*internationalization*".

Интернационализация обычно включает в себя:

1. Разработку таким образом, чтобы не было преград при дальнейшей локализации и международном применении. Например, создание возможности применения Unicode или обеспечение соответствующего подхода к кодировке символов.
2. Обеспечение основы для элементов, которые невозможно использовать до процесса локализации.
3. Добавление в CSS основы для вертикального текста или нелатинских типографических свойств.
4. Добавление разметки в DTD основы для двустороннего текста.

То есть мы заранее подготавливаем продукт к возможным изменениям.

Интернационализация значительно облегчает локализацию продукта. Настройка лингвистически- и культурно-направленного объекта для глобального рынка безусловно является более сложным и продолжительным процессом, чем создание объекта с целью его глобального представления.[3]

Поэтому теоретически интернационализация является основным шагом в процессе создания и развития, нежели дополнением, которое может часто стать причиной неловких и дорогих технологического изменений.[3]

2.3 Понятие локализации

Локализация – это процесс адаптации ПО под конкретные национальные требования.

Локализация пишется как "*l10n*", где 10 - это число букв между буквами '*l*' and '*n*' в английском варианте слова "*localization*".

Локализация не ограничивается обычным переводом отдельных составляющих программного продукта – экранных форм, текстов сообщений, интерактивной справки и печатной документации. Собственно перевод – это трудоемкий, но далеко не единственный и не самый ответственный этап локализации. Помимо традиционного перевода всех исходных материалов требуется обеспечить корректность работы всей системы с новым интерфейсом и в новой языковой среде. Ниже я приведу многие, но далеко не все аспекты, на которые нужно обращать внимание при процессе локализации:

2.3.1 Формат даты и времени

Данный аспект, мне думается, является "самым вспоминающимся", то есть про него забывают реже всего и его тяжелее всего упустить из виду при локализации. В разных регионах и странах принято по-разному фиксировать формат даты и времени.

Страна/язык	Формат даты	Формат времени	Пример даты	Пример времени
США	MM-DD-YYYY	hh:mm:ss	01-24-2011	16:35:00
Россия	DD.MM.YYYY	hh:mm:ss	24.01.2011	16:35:00
Швейцария	DD.MM.YYYY	hh,mm,ss	24.01.2011	16,35,00
Испания	DD/MM/YYYY	hh:mm:ss	24/01/2011	16:35:00
Швеция	YYYY-MM-DD	hh.mm.ss	2011-01-24	16.35.00

Таблица 1. Пример различных форматов даты и времени.

Как мы видим из приведенной таблицы, данные хоть и немного, но различаются. И именно про эти различия нельзя забывать при подготовке выпуска продукта в другой стране.

2.3.2 Формат бумаги

Существует международный стандарт на бумажные форматы (ISO 216), который принят всеми странами, за исключением Соединённых Штатов, Канады, Японии. В Мексике и на Филиппинах, несмотря на принятие международного стандарта, по-прежнему широко используется американский формат «Letter».

По данному стандарту бумажные форматы классифицируются на 3 серии - А, В и С. И каждый из них делится на 11 размеров - от 0 до 10.

Например, самый распространенный формат А4 имеет следующие размеры: 210 x 297 мм (8,27 x 11,69 дюймов).

Формат	Серия А	Серия В	Серия С
Размер			
0	841 × 1189 мм	1000 × 1414 мм	917 × 1297 мм
1	594 × 841 мм	707 × 1000 мм	648 × 917 мм
...
10	26 × 37 мм	31 × 44 мм	28 × 40 мм

Таблица 2. Пример различных размеров и форматов бумаги по стандарту ISO 216.

В Северной Америке имеется собственный стандарт (ANSI), форматы в котором очень близки по значениям, а также по соотношению сторон к ISO 216, но всё же различия имеются:

Популярное название	Классификация ANSI	в мм	Соотношение сторон	Похожий формат ISO
Letter	ANSI A	216 × 280	1 : 1,3002	A4(210 × 297)

Legal		216 × 356	1 : 1,6471	
Ledger или Tabloid	ANSI B	432 × 279	1 : 1,5455	A3(297 × 420)

Таблица 3. Сравнение стандарта ANSI с ISO 216.

У формата A4 соотношение сторон соответствует 1 : 1,414, что весьма близко к ANSI A, но даже такие маленькие погрешности существенны.

В Японии также имеется свой стандарт JIS. Форматы JIS серии A соответствуют серии A по ISO 216, однако форматы JIS серии B отличаются от B по ISO; площадь японских B в 1,5 раза больше соответствующих A (а не в 1,414 раза, как по ISO).

2.3.3 Символы валюты

Всем известно, что у многих стран есть собственная валюта. Есть, конечно, "единая" валюта, которая используется сразу в ряде государств (например, евро и доллар), но список валют включает в себя более чем 150 стран.

Поэтому крайне важно не забывать об этом при локализации, особенно если локализуемый продукт может соприкасаться с банковскими операциями.

2.3.4 Система мер

Также дела обстоят и с разнообразием систем мер. Всем известно про основные единицы СИ (метр, килограмм, секунда, ампер, кельвин, моль и кандела). Но во многих странах и регионах имеются отличные от СИ единицы измерения. Этих систем множество и большая часть неизвестна широкому кругу, но про некоторые слышали все:

- 1 миля ~ 1,6 км
- 1 дюйм = 2,54 см
- 1 гектар = 10 000 м²
- 1 фунт ~ 409,5 гр
- и так далее

Чтобы избежать возможных непониманий, нужно учитывать особенности единиц измерения для региона локализуемого продукта.

2.3.5 Формат телефонных номеров

Достаточно распространенная ошибка - неучитывание особенностей телефонных форматов для различных государств. Достаточно часто при регистрации аккаунта или других действиях, где нужно указывать телефонный номер, поля для ввода данных лимитированы, то есть можно написать не более определенного количества цифр. И бывает, что поле для заполнения предусматривает меньшее количество цифр, чем есть на самом деле. Или же наоборот - введен верный

номер, а система просит ввести еще какое-то количество цифр. Чтобы избежать таких ошибок - опять же, при локализации нужно учитывать регион.

Вот насколько могут различаться номера разных стран:

Страна	Код страны	Номер
США	1	707-402-9563
Австрия	43	0688 517 94 79
Гренландия	299	34 84 22
Германия	49	03496 47 74 77
Дания	45	40-74-99-32
Швеция	46	0498-7015135

Таблица 4. Пример различных телефонных номеров.

Как мы видим из этой неполной таблицы, количество цифр варьируется от 6 до 11 и это еще не беря во внимание различие между мобильными и стационарными телефонами, телефоны спецслужб, присутствие внутри номера цифр, привязанных к определенному городу/области/региону.

2.3.6 Особенности человеческих имен

В большинстве стран имена людей выглядят практически одинаково - имя и фамилия. Но как и везде, здесь также имеются свои исключения. Данные для регистрации, обращения, да вообще любое использование человеческого имени - везде нужно учитывать свои особенности. Рассмотрим некоторые исключения:

- *Страна:* Голландия
Особенность имени: многие фамилии имеют префикс: "van", "de", "het"
Пример: Laurens van Toor
- *Страна:* Бельгия
Особенность имени: во многих фамилиях, как и в голландских, используются те же самые префиксы, но при этом они пишутся с заглавной буквы
Пример: Maarten Van Der Elst
- *Страна:* Ирландия
Особенность имени: отличительной особенностью является префикс " O' " или " Mac" перед фамилией
Пример: Donagh O'Casey
- *Страна:* Шотландия
Особенность имени: префикс " Mac" перед фамилией
Пример: Simidh MacCoinnigh
- *Страна:* Россия
Особенность имени: после имени идет имя отца (отчество)
Пример: Никитин Тимур Викторович

- *Страна:* Англия
- *Особенность имени:* дети в Англии получают 2 имени - личное и среднее
- *Пример:* Jamie Richard Vardy

2.3.7 Символы на клавиатуре

Во многих программах доступно использование "горячих" клавиш. При определенных сочетаниях выполняются определенные действия. И если все буквенные клавиши на локализованном продукте скорее всего учтены, то клавиши управления (такие как "Esc", "Ctrl", "Alt", "Shift" и другие) часто остаются без перевода. А в некоторых раскладках на нескольких языках имеются уникальные обозначения:

Немецкая клавиатура:

Ctrl = Strg

Shift = Umschalt

Финская клавиатура:

Shift = Vaihto

Французская клавиатура:

Shift = Maj

Esc = Echap

Итальянская клавиатура:

Shift = Maiusc

2.4 Этапы локализации

Если сильно упростить, то процесс локализации можно разбить на 4 основных этапа, которые запечатлены на рисунке 3.

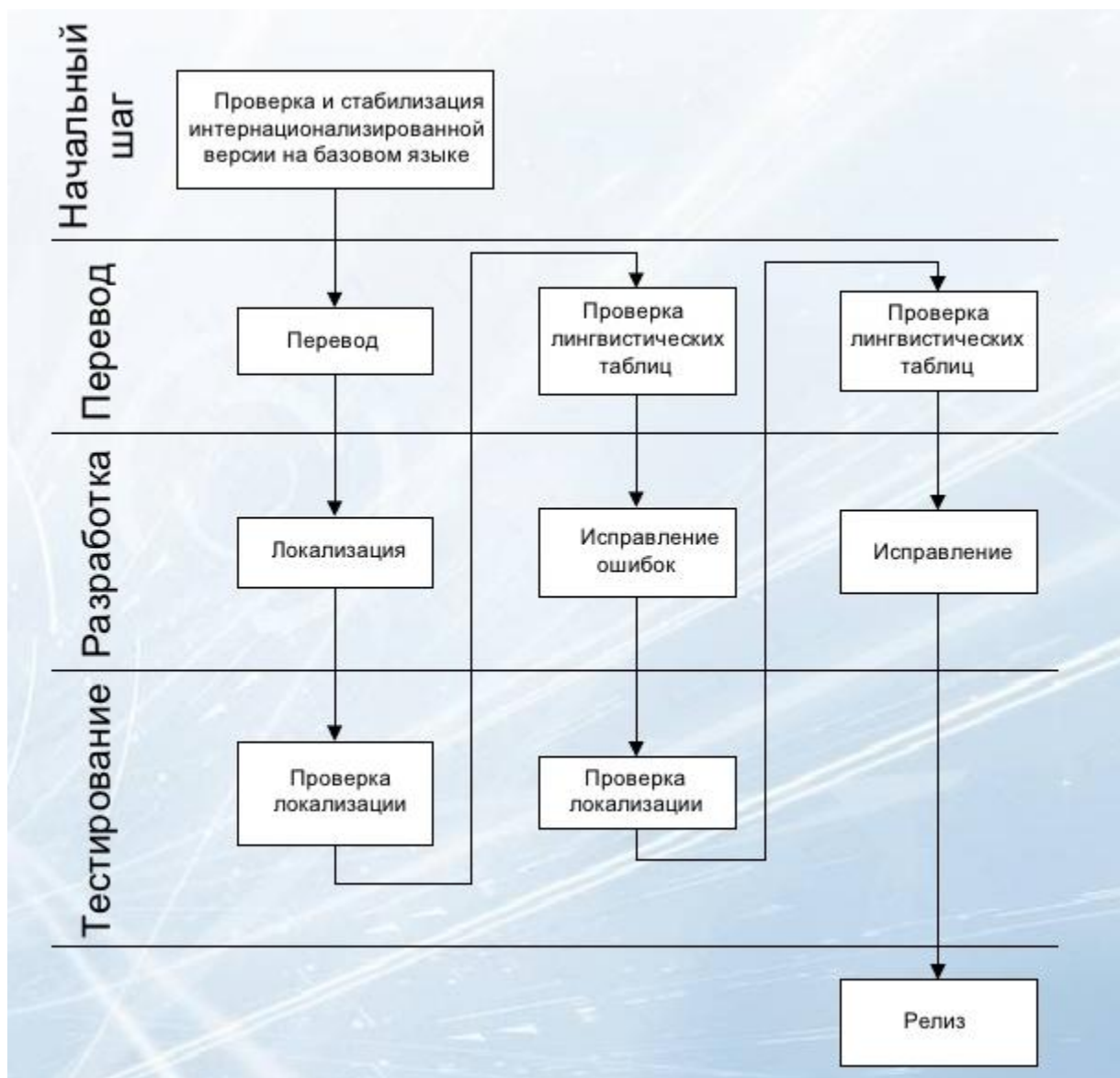


Рисунок 3. Этапы локализации [4].

Разберем каждый из них более подробно:

2.4.1 Начальный шаг (обсуждение проекта, подготовка, анализ)

Продукт может быть:

- адаптированным к локализации
- неадаптированным

Заранее подготовленный к локализации продукт намного проще и дешевле локализовать, чем неподготовленный. Поэтому при разработке перспективных, с большими дальнейшими планами, программ нужно заранее закладывать возможность многоязыковой поддержки, чтобы в будущем избежать огромных затрат, как денежных, так и временных.

Во время анализа обсуждаются всевозможные аспекты - временные рамки, стоимость услуг, составление документации и так далее.

2.4.2 Перевод

Случается, что переводчикам требуются дополнительные материалы для лучшего понимания контекста переводимого текста (рисунки, схемы, руководства данного проекта).

В наше время вряд ли можно встретить переводчика/переводчиков, не использующих при переводе никаких программ. В последние годы стремительный рост набирает программа МемоQ. Она, как и множественные её аналоги, сильно облегчает задачу переводчикам, а именно:

- В ней хранятся все предыдущие переводы ("*Translation Memory*") различных проектов. При соответствии слов/предложений/абзацев, можно заимствовать старые переводы, что обеспечит как идентичный перевод, так и значительную экономию времени.
- Все данные синхронизируются между собой, поддерживается много разных форматов для перевода (xlz, xml, html, sgml, ...).
- Готовые переводы после поступления мгновенно архивируются, сразу же обновляя базу данных.
- Очень простой и доступный интерфейс (Рисунок 4)

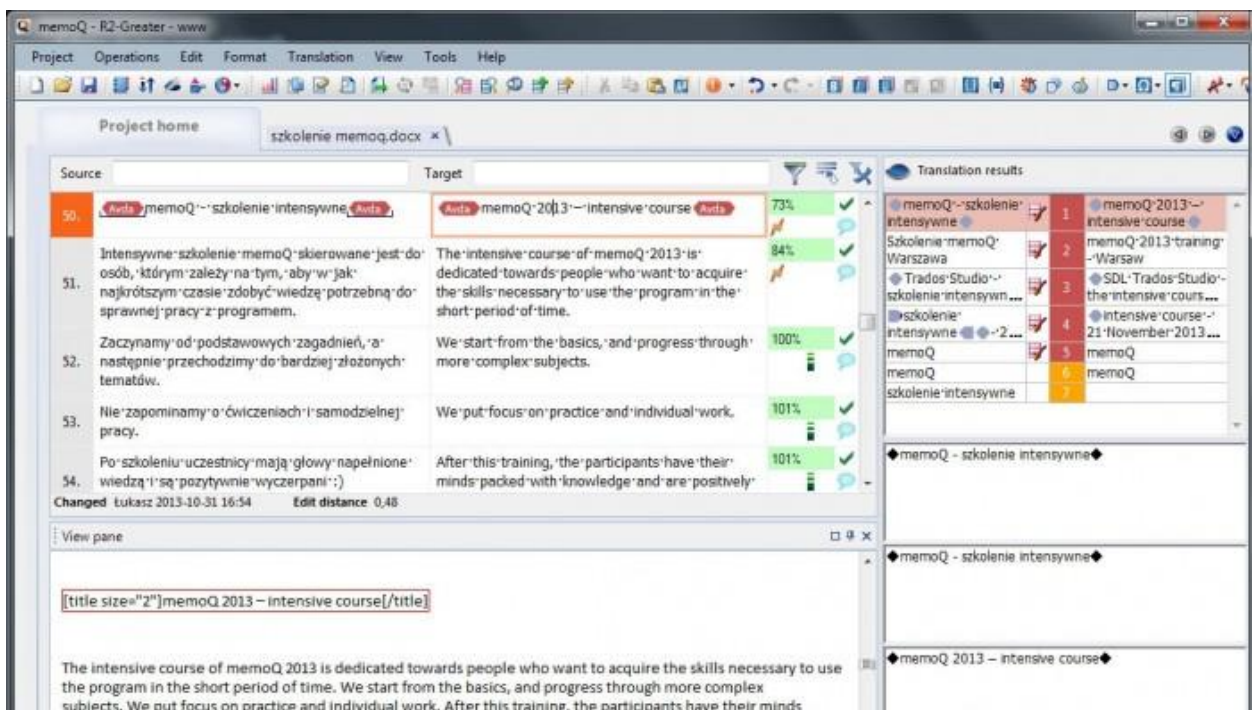


Рисунок 4. Интерфейс программы МемоQ.

2.4.3 Разработка (внедрение)

После того, как весь необходимый перевод был получен, остается загрузить его в продукт. Из базы данных с переводами (мы рассматривали программу МемоQ) все локализованные строчки "заливаются" на сервер, где содержится продукт. Если же это уже повторный цикл (были найдены лингвистические ошибки при тестировании, и текст был заново отправлен на перевод), то старые строчки удаляются, а вместо них загружаются новые.

2.4.4 Тестирование

Как видно из Рисунка 3, тестирование проходит в несколько этапов, с постепенным исправлением ошибок. Количество циклов может быть очень велико - оно зависит от масштаба продукта, а также от временных рамок.

Тестирование локализации, также как и перевод, может быть качественно выполнено специалистом, в совершенстве владеющим языком тестируемой локали, так как выявленные ошибки могут быть разного характера:

1. нелокализованный (непереведенный) текст
2. обрезанный текст - самая популярная ошибка при переводе на другие языки
3. перекрывающиеся элементы управления

И если для вышеперечисленных ошибок знание языка необязательно, то для следующих баг нужен именно носитель языка проверяемой локали:

4. проверка грамматики
5. проверка орфографии
6. проверка особенностей, описанных в пунктах 2.3.1 - 2.3.7

Ошибки №2 и №3 возникают в связи с тем, что у всех языков разная "длина" - на одних языках текст получается длинее, на других короче. Если брать за исходный язык английский, то опять же, можно выделить 2 категории:

- Длиннее, чем по-английски

Сюда входит множество языков, в основном европейские, такие как: французский, испанский, немецкий, польский и так далее.

- Короче, чем по-английски

Здесь же преобладает иероглифическая письменность, то есть азиатские и арабские языки.

По данным исследования живого онлайн-переводчика Nitro, рассмотрим несколько интересных фактов:

- Локализация с английского на итальянский или немецкий увеличивает объём текста в среднем на 17,3%. [5]

- При необходимости сделать перевод с английского на португальский язык следует учесть, что локализация на «португальский португальский» увеличит объём текста на 14,3%, а на «бразильский португальский» — чуть меньше, чем на 13%. [5]

- Перевод с английского на традиционный китайский уменьшит объём текста на 63,8%, а на упрощённый китайский — почти на 62%. [5]

Понятно, что все эти данные очень округлены и каждый случай уникален, но можно понять представление, почему же после перевода возникают ошибки. Сообщения-строки в разных языках имеют существенно разные размеры и поэтому происходит наложение элементов друг на друга, обрезание и другие проблемы с пользовательским интерфейсом.

2.5 Трудности

Большинство трудностей возникают, как уже говорилось, из-за особенностей языка. Невозможно сразу предусмотреть все из них, некоторые будут проявляться со временем и фикситься в последующих версиях.

Нижеприведенный список можно было бы включить в пункт 1.3, однако указанные здесь тонкости и нюансы менее распространены и чаще забываются при локализации, поэтому отнесем их к трудностям.

- Для представления иероглифов в азиатских языках (японский, китайский, корейский) используются двухбайтные символы. Символы для всех остальных языков можно представить однобайтными символами. Данное различие порой может очень сильно повлиять на пользовательский интерфейс программы (для того, чтобы иероглифы четко отображались, шрифт нужно ставить значительно больше, нежели у однобитных символов), а порой и на её функциональную часть.
- Те же самые последствия могут возникнуть у языков с нестандартным направлением написания: справа налево (арабский язык и иврит); сверху вниз (японский язык - в Японии распространены 2 типа написания: слева направо и сверху вниз).
- Локализовать нужно не только текст в самой программе, но и на изображениях, которые встречаются в этой программе. Если программа содержит аудио или видео файлы - они также нуждаются в переводе и записи новых файлов. Все эти действия, как можно понять, гораздо более тяжелы в исполнении, нежели перевод простого текста. Зачастую клиенты даже не задумываются, что такие вещи также нуждаются в локализации, изначально держа в уме только простые строчки-сообщения.

- Помимо текста на изображениях, случается, что необходимо переделывать и сами изображения. Например, если в продукте присутствует карта мира или глобус, то в локализованной версии, его нужно развернуть той частью света, на которую продукт рассчитан.

При локализации игр, очень важно, чтобы переводчик понял контекст игры и всего происходящего в ней. Поэтому существует такая практика при локализации игр - перед началом перевода дать переводчикам поиграть в эту самую игру, требующую перевода. И тогда переводчики, ориентируясь в игре гораздо лучше, предоставят более качественный перевод, не такой "сухой", каким мог бы быть без знания азов игры.

2.6 Пример неудавшейся локализации

К чему может привести выход на рынок не до конца продуманного продукта (а точнее - всего лишь с непроверенным на разных языках названием), можно рассмотреть на примере выпуска автомобилей итальянской фирмой "Fiat" - "Fiat Punto".

В 1993 году, в год выпуска вышеупомянутой модели, по всему миру данная модель была самой кассовой и била все рекорды по продажам. Везде, кроме испаноговорящих стран.

Дело в том, что в переводе с испанского, слово "Punto" имеет весьма нелицеприятное значение. На остальных же языках данное слово никакого негатива не вызывает.

И вот так, всего лишь из-за недостаточно продуманного и локализованного названия модели, концерн "Fiat" недосчитался прибыли.

3 Тест кейсы

3.1 Тест кейсы в теории

В пункте 1 мы рассматривали различные типы классификации тестирования. Можно рассмотреть еще один - по степени подготовленности к тестированию:

- ad hoc testing
Это тестирование без каких-либо четких шагов и указаний - интуитивное тестирование. Большим плюсом служит то, что оно занимает гораздо меньше времени (в основном за счет того, что не нужно предварительно описывать все сценарии - тест кейсы и в дальнейшем неукословно их соблюдать), при этом качество тестирования не всегда страдает, а порой и более эффективно, так как не ограничено ни в каких рамках.
- тестирование по документации
Тестирование подразумевает четкое следование заранее написанным тест кейсам. Поэтому наверняка вся важная функциональная часть будет протестирована. Более того, при наличии документации гораздо проще планировать тестирование: тест кейсы можно поделить между разными тестировщиками. Вдобавок, значительно проще внедрять новых людей, тк у них есть четкое описание всех их действий.

Что же такое тест кейс? Это документация для тестировщика, где описана последовательность действий и ожидаемый результат. Зачастую тест кейс содержит еще предварительное условие и послеусловие.

Предварительное условие - Список действий, которые приводят систему к состоянию пригодному для проведения основной проверки. Либо список условий, выполнение которых говорит о том, что система находится в пригодном для проведения основного теста состоянии. [8]

Послеусловие - Список действий, переводящих систему в первоначальное состояние. [8]

И все же, соблюдение данных правил необязательно - все описывают тест кейсы так, как им удобно. Иногда специфика тестируемого продукта предполагает использование других критериев при написании.

Помимо тест кейсов перед началом тестирования должны быть обозначены и другие детали - список тестируемых локалей и операционных систем, а также сроки, к которым тестирование должно быть выполнено.

Локали представлены в виде двух субтегов - основной субтег для идентификации языка и дополнительный субтег для уточнения региона использования, например:

- en_GB (британский английский)
- en_US (американский английский)
- en_AU (австралийский английский)

Операционные системы: в первую очередь указание того, на чем мы тестируем - компьютеры, смартфоны, планшеты и так далее. С дальнейшим указанием конкретных версий и систем.

3.2 Тест кейсы на практике

Рассмотрим на примере мобильных приложений выпускаемых компанией Adobe Systems.

Локализация совершается на 18 языках, следовательно, тестируются они же:

Язык	Языковой тег
французский	fr_FR
немецкий	de_DE
японский	ja_JP
итальянский	it_IT
испанский	es_ES
нидерландский	nl_NL
португальский	pt_BR
шведский	sv_SE
датский	da_DK
финский	fi_FI
норвежский	nb_NO
китайский (упрощенный)	zh_CN
китайский (традиционный)	zh_TW
корейский	ko_KR
чешский	cs_CZ
польский	pl_PL
русский	ru_RU

турецкий	tr_TR
----------	-------

Таблица 5. Список локалей на некоторых проектах Adobe Systems.

Что касемо систем, то у мобильных девайсов идет разделение на iOS и Android системы. И при тестировании указывается, какие версии должны быть затроны. Например, все iOS системы начиная с 9.3.5 и выше.

Примеры тест кейсов:

Title:	Review Screen: Page Index with rotation
Description:	Verify that the page index is properly shown on Review screen.
Test step:	Expected Result:
1. Launch Scan	Capture screen should display
2. Capture 1 image	Thumbnail stack should fade in once the first image is captured.
3. Rotate the device 90 degree.	
4. Capture another image	
5. While still holding the device in same orientation as step 3, tap the thumbnail stack	Review screen should display. The page index should appear below the image (between the image and the bottom bar). The page index should say "Page 1 of 2".

Таблица 6. Пример тест кейса для приложения Acrobat Scan.

Title:	[iOS] GPS Location. Application dialog - "Adobe Sign Would Like to Record Your Current Location"
Description:	[iOS] GPS Location. Application dialog - "Adobe Sign Would Like to Record Your Current Location"
Test step:	Expected Result:
1. Install Adobe Sign app.	
2. Select "Send for Signature"	
3. Add Document and one recipient	
4. Press Send button	
5. Open the agreement from "Waiting for Other" tab	
6. Tap on Signature icon. Sign Agreement and tap Finish Signing	Verify that: 1. Following Dialog appears - Adobe Sign Would Like to Record Your Current Location" 2. Verify that dialog was translated to all languages

Таблица 7. Пример тест кейса для приложения Adobe Sign.

Title:	Adobe Sign Mobile FTE appears when user sign-out from Adobe Sign app.
Description:	Adobe Sign Mobile FTE appears when user sign-out from Adobe Sign app.
Test step:	Expected Result:
1. Install Adobe Sign app	
2. Tap on "Sign In" button on FTE screen	oAuth login page appears
3. Login into Adobe Sign App	Adobe Sign Dashboard appears
4. Sign out from Adobe Sign app	Verify that FTE page appears. See att. video Verify that FTE redirect to Login Page automatically without taping on "Sign In" button

Таблица 8. Пример тест кейса для приложения Adobe Sign.

Title:	[Widget]Update Waiting for You count in the Widget
Description:	[Widget]Update Waiting for You count in the Widget
Test step:	Expected Result:
1. Device 1:Login into Adobe Sign app with my Adobe ID	
2. Device 1: Send a agreement to esignauto@gmail.com to sign	
3. Device 2: Login to esignauto@gmail.com/esignauto	
4. Device 2: Bring up Adobe Sign Widget from home screen, look at Waiting for you	The count of agreements should increase one. See screenshot in the attachment.
5. Device 1: Delete a agreement	
6. Device 2: Bring up Adobe Sign Widget from home screen, look at Waiting for you	The count of agreements should decrease one. See screenshot in the attachment.
7. We need to test in iPhone and iPad	

Таблица 9. Пример тест кейса для приложения Adobe Sign.

Title:	Review Screen: Delete confirmation message is displayed for each page deletion
Description:	Verify that the message dialog displays when user deletes each page.
Test step:	Expected Result:
1. Launch Scan	Capture screen should display
2. Capture an image	Thumbnail stack should fade in once the image is captured.
3. Tap the photo library icon, and select a photo to add.	Review screen should display

4. View page 1 and tap the Delete icon.	"Do you want to delete this page?" with options "NO" and "YES" should display.
5. Choose YES	The message dialog should go away. The proper page (page 1) is deleted. Review screen should remain open.
6. Tap the Delete icon again.	"Do you want to delete this page?" with options "NO" and "YES" should display.
7. Choose "YES"	The message dialog should go away. The page is deleted. Review screen should remain open.

Таблица 10. Пример тест кейса для приложения Acrobat Scan.

3.3 Примеры

Вот так выглядит пример реального тест кейса на wiki.corp.adobe.com:

S.no	Test Case	Steps	Expected Result	Sanity cases	Automation status
FTE/Coach mark - A coach mark will only once after installing application when any agreement list will be opened.					
1	FTE string	-	"Tap to reveal quick actions" in blue color coach mark	P0	FTEInComplete
2	FTE will dismiss in 7seconds automatically.	-	FTE will start dismissing after 5 sec and takes 2seconds to get fully dismissed.	P0	FTEInComplete
3	FTE will get dismissed tapping anywhere on screen.	-	FTE will get dismissed tapping anywhere on screen	P0	
4	Open waiting for you list having agreements first time after installing.	1. Open waiting for you list having agreements first time after installing.	FTE will come under more option button of first agreement	P0	
5	Open waiting for other list having	1. Open waiting for other list having agreements first time after installing.	FTE will come under	P0	

Рисунок 5. Пример тест кейса в системе wiki.corp.adobe.com

Тест кейс в системе Jira:

https://jira.corp.adobe.com/browse/DCME-5893

Adobe JIRA Data Center Dashboards - Projects - Issues - Capture - Agile - Tests - Portfolio - easyBI - Create

DCM Sign iOS / DCME-5893
Adobe Sign Mobile FTE appears when user sign-out from Adobe Sign app.

Edit Comment In Progress Ready for Review Ready for Execution Inactive Clone More Actions Execute Export

Details

Type: Test
 Priority: Normal
 Affects Version/s: None
 Component/s: UI
 Labels: Functional Sanity
 Team: eSign iOS

Status: **TODO** (View Workflow)
 Resolution: Unresolved
 Fix Version/s: None

People

Assignee: Unassigned
 Assign to me
 Reporter: Andrey Etremov
 Votes: Vote for this issue
 Watchers: Start watching this issue

Description

Adobe Sign Mobile FTE appears when user sign-out from Adobe Sign app.

Test Details

Test Step	Test Data	Expected Result
1	Install Adobe Sign app	
2	Tap on "Sign In" button on FTE screen	oAuth login page appears
3	Login into Adobe Sign App	Adobe Sign Dashboard appears
4	Sign out from Adobe Sign app	Verify that FTE page appears. See att. video Verify that FTE redirect to Login Page automatically without tapping on "Sign In" button

Add

Dates

Created: 31/Oct/16 04:25
 Updated: 31/Oct/16 04:29

Agile

Рисунок 6. Пример тест кейса в системе Jira.

4 Инструменты для локализации и её тестирования

Существует огромное множество разнообразных программ, инструментов, платформ, которые могут облегчить и значительно ускорить процесс локализации. Критерии по выбору этих самых инструментов самые разнообразные, начиная от цены за использование программы, заканчивая простотой использования.

Мы рассмотрим некоторые из них, которые используются в компании Adobe Systems на многих проектах.

4.1 Инструменты для визуализации

Как уже говорилось в пункте 2.4.3 все переводы хранятся на сервере. Чтобы визуализировать эти данные, можно использовать как устанавливаемые программы, так и браузерные приложения.

4.1.1 ALF

Для начала ознакомимся с программой под названием ALF (Adobe Localization Framework). Все данные, а также сделанные в программе изменения синхронизируются с сервером. На Рисунке 7 показан интерфейс после входа в систему и выбора проекта и языка.

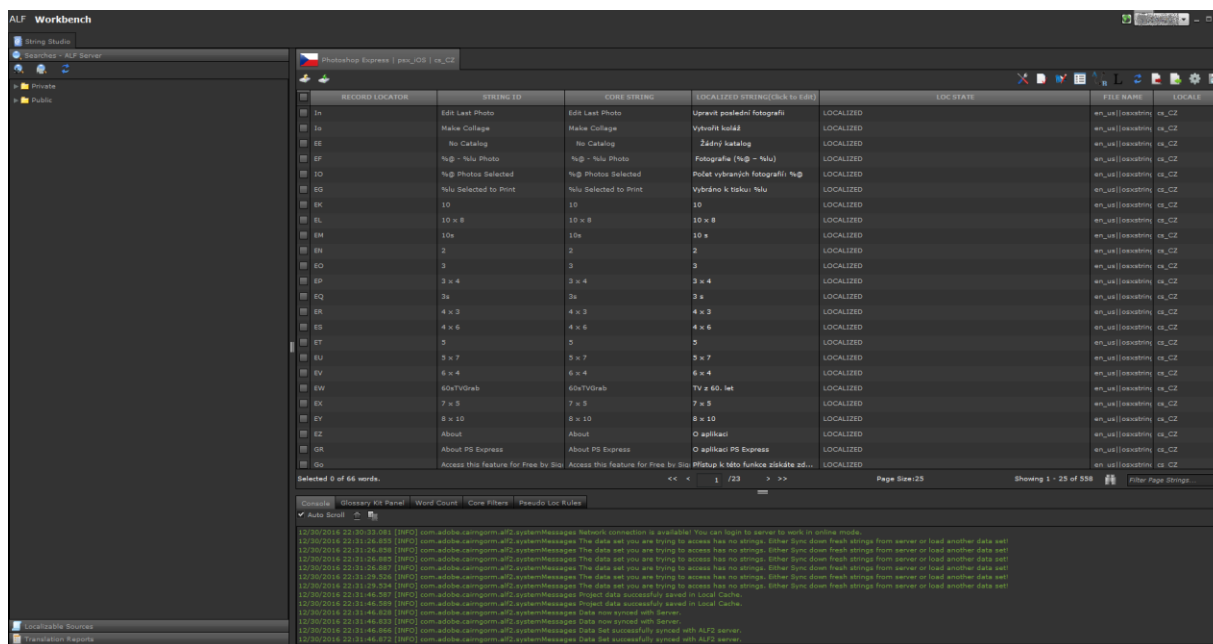


Рисунок 7. Интерфейс программы ALF.

Через ALF мы можем:

- Вызвать все строки определенного проекта
- Сделать сортировку строк по языкам/дате изменения/дате внесения в базу
- Найти любую строку по нижеперечисленным параметрам:
 - "Core string" (исходная строка)
 - "Loc string" (переведенная строка)
 - String/mock ID
 - И другим параметрам
- Мануально поменять перевод

4.1.2 Slate

Все те же функции имеются и у браузерного-приложения Slate (Рисунок 8) . Slate более прост в использовании и не требует установки, что позволяет использовать его в любом месте. Поэтому мы рассмотрим его более подробно.

Record Locator	Core String	String ID	Localized String	Locale	Localized State
H+	Install Photoshop Mix and Retry	Install Photoshop Mix and Retry	Nainstalovat aplikaci Photosh...	cs_CZ	LOCALIZED
H/	Update	Update	Aktualizovat	cs_CZ	AVAILABLE FOR REVIEW
H0	Edit Photo	Edit Photo	Upravit fotografii	cs_CZ	LOCALIZED
H1	Inner	Inner	Vnitřní	cs_CZ	LOCALIZED
H2	MAKE COLLAGE	MAKE COLLAGE	VYTVOŘIT KOLÁŽ	cs_CZ	LOCALIZED
H3	Next	Next	Další	cs_CZ	AVAILABLE FOR REVIEW
H4	Outer	Outer	Vnější	cs_CZ	LOCALIZED
H5	Recents	Recents	Poslední	cs_CZ	LOCALIZED
H6	Replace	Replace	Nahradit	cs_CZ	LOCALIZED
H7	Do you wish to disconnect you...	Do you wish to disconnect you...	Chcete svůj účet Google od al...	cs_CZ	LOCALIZED
H8	You have removed your accou...	You have removed your accou...	Odebrali jste účet z aplikace /...	cs_CZ	LOCALIZED
H9	Install	Install	Nainstalovat	cs_CZ	LOCALIZED
Ha	Photoshop Express needs acc...	Photoshop Express needs acc...	Aplikace Photoshop Express v...	cs_CZ	LOCALIZED
Hb	Photoshop Express needs acc...	Photoshop Express needs acc...	Aplikace Photoshop Express v...	cs_CZ	LOCALIZED
Hc	Please Give Permission	Please Give Permission	Udělte oprávnění.	cs_CZ	LOCALIZED
He	Finish the photo you started e...	Finish the photo you started e...	Dokončete úpravy fotografie v...	cs_CZ	LOCALIZED

Рисунок 8. Интерфейс программы Slate.

Кроме схожих функций с ALF, в Slate также встроен ряд своих, весьма полезных:

- Позволяет сделать проверку на согласованность всех переводов между всеми продуктами Adobe. Согласованность переводов высоко ценится со стороны заказчиков, поэтому выполнение этого пункта особо важно.

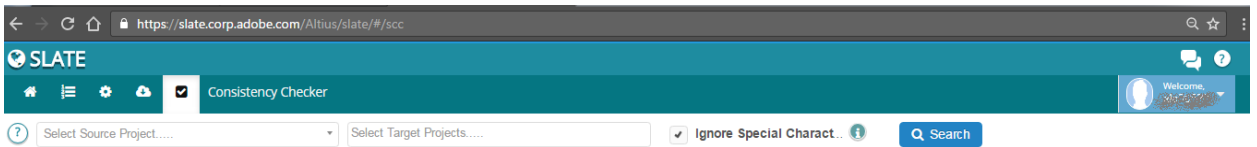


Рисунок 9. Программа Slate. Функция проверки на согласованность.

- Большой статистический функционал. Позволяет делать статистические выкладки по всем имеющимся проектам, где видны такие данные как: количество строк доступных для перевода, количество строк отправленных на перевод, количество уже локализованных строк и так далее.

Product	Release	Available for Translation	Sent For Translation	Localized	Available for Review	Sent For Review	Last updated
CC Apps Combined	CC Apps Combined	0	143	17606	0	0	2016-12-29:08:15:18
CC SDK Combined	CC SDK Combined	0	0	4318	0	0	2016-12-22:17:29:50
Adobe Capture Android	Project Gather	0	0	2797	26	0	2016-12-20:16:19:18
Photoshop Express	psx_iOS	18	0	2631	52	0	2016-12-12:23:24:42
PSMixApp	Develop	20	0	1688	2358	0	2016-12-06:12:30:36
Photoshop Mix Android	PSMixAndroid	0	0	648	640	0	2016-12-06:11:49:48
PSFixApp	Fix_Dev	29	0	3262	135	0	2016-12-06:09:15:52
Creative SDK Android	CSDK-Android	0	2	3415	38	0	2016-12-05:14:31:47
Photoshop Fix Android	PSFixAndroid	0	0	446	602	0	2016-12-05:11:16:05
Adobe Illustrator Draw	Draw	0	0	930	75	0	2016-12-05:09:49:25
PSMixApp	PSMixApp	0	0	1708	2358	0	2016-11-14:11:33:20

Рисунок 10. Программа Slate. Статистика

- Вдобавок к предыдущему пункту имеется возможность фильтровать данные как по определенному проекту, так и по языку.

Product	Release	Available for Translation	Sent For Translation	Localized	Available for Review	Sent For Review	Last updated
Project_Central	Project_Central_main	53	0	2729	0	0	2016-11-09:17:14:44
PSFixApp	PSFixApp	0	0	3282	135	0	2016-11-04:13:05:33
CC SDK Image Aviary Andr...	Aviary Android 03	0	0	4004	0	0	2016-11-03:17:24:03
CC Apps Combined	Release_Branches	0	0	904	0	0	2016-10-24:06:50:49
PhotoSDK	PhotoSDK	0	0	222	7	0	2016-10-21:13:32:50
CC SDK Combined	CSDK_IOS_Feature_Branch	0	0	2458	0	0	2016-10-18:09:35:47

Рисунок 11. Программа Slate. Статистика. Фильтрация по языку.

- Имеется возможность забивать авто-баги (которые автоматически исправляются в следующей сборке). Выделенная часть на Рисунке 12 генерируется автоматически, что значительно сокращает затрачиваемое время.

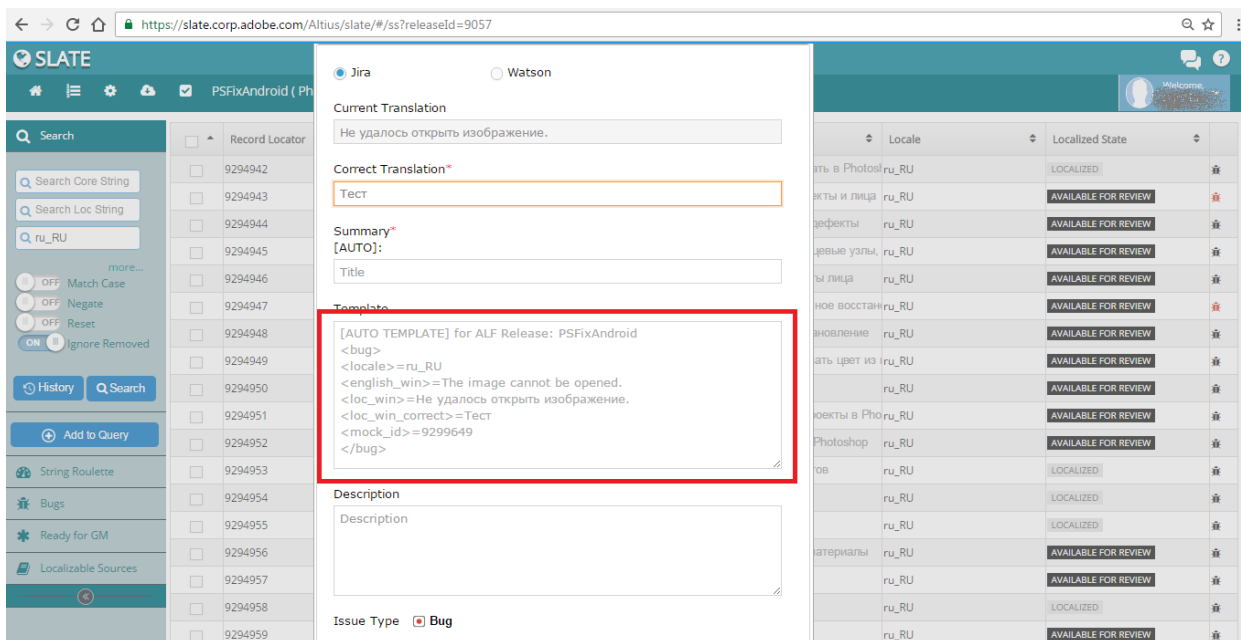


Рисунок 12. Программа Slate. Пример автобаги.

Несмотря на то, что Slate не требует установки, его функционал выглядит внушительнее и предпочтительнее. Но использование обеих платформ также возможно, ведь как и везде, у обеих из них есть какие-то плюсы и минусы. Они могут сосуществовать одновременно, все данные синхронизируются и параллельная работа над одним проектом несколькими людьми не противопоказана.

4.2 Системы отслеживания ошибок




В пункте 4.1 мы разобрали инструменты, которые лишь отображают сведения с сервера. Но также существуют инструменты, которые помогают работать с этими данными. По большей части они нужны для того, чтобы протоколировать ошибки, а затем отслеживать их состояние.

В недавнем времени Adobe Systems перешли с одной площадки на другую. С "Watson" на "Jira". Попробуем найти логическое объяснение этому:

4.2.1 Watson

Система более старая, поэтому, что логично, имеет меньше возможностей. За отсутствием большого количества функционала, имеет достаточно простой и доступный интерфейс, что можно отнести к плюсам. Но в то же время недостаток функций, по сравнению с конкурентами, является недостатком.

На рисунке 13 запечатлена стартовая страница. Как видно из панельного меню сверху, имеется всего лишь 3 иконки-вкладки:

-  - "*Home*"
Домашняя страница, на которой есть только лишь описание функционала, а также возможность перейти в другие места программы.
-  - "*New Bug*"
Переход на страницу ввода ошибок в систему.
-  - "*Search*"
Страница с расширенным поиском (можно задавать всевозможные параметры для поиска).

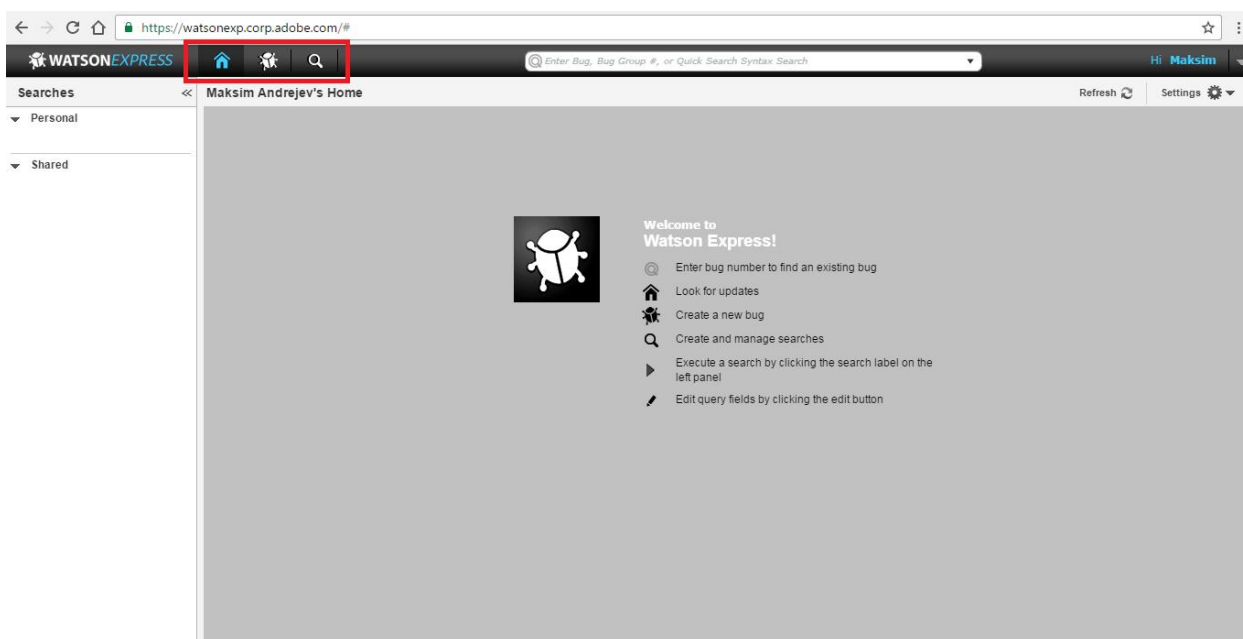


Рисунок 13. Интерфейс инструмента Watson.

Рассмотрим более подробно вкладку "*New Bug*" - Рисунок 14 :

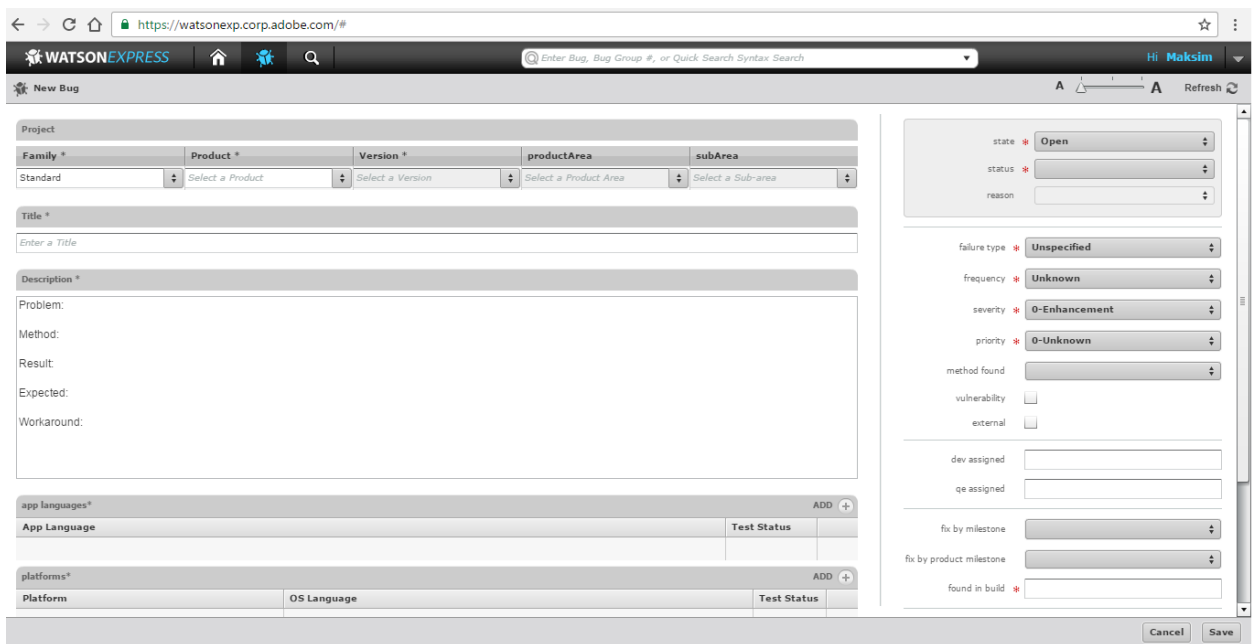


Рисунок 14. Watson. Протоколирование ошибки.

Страница имеет типичные поля для протоколирования ошибок.

Разберем более подробно обязательные (помечены символом " * ") из них:

- **"Family" - "Product" - "Version"**
Здесь мы выбираем все данные о проекте: данные о продукте, его конкретная версия и другие направляющие данные.
- **"Title"**
Соответственно даем название нашей баге. В названии должно содержаться краткое описание проблемы.
- **"Description"**
Пишем подробное описание проблемы, четкие шаги для воспроизведения ошибки, различия между ожидаемым результатом и настоящим и другие вспомогательные данные.
- **"app languages" и "platforms"**
Как можно понять из названия - в данных полях следует указать языки и ОС, на которых описываемая ошибка воспроизводится.
- **"state"**
Состояние может быть или открытым или закрытым. При создании баги, естественно статус будет *"open"*, а в дальнейшем он может меняться в соответствии с происходящими изменениями.
- **"status"**

Поле "статус" имеет различные значения, такие как "to test", "in development", "to defer", "to fix" и другие. Также как и состояние, в процессе изменений меняется и статус.

- "failure type"
Описываем тип проблемы, например, потеря данных, крах программы или косметическая ошибка.
- "frequency"
Некоторые ошибки можно воспроизвести со 100% вероятностью, некоторые же появляются периодически, что мы и указываем в данной графе.
- "severity" и "priority"
В зависимости от критичности баги, выставляем приоритет. Ошибки с самым высоким приоритетом будут рассмотрены разработчиками в первую очередь.
- "found in build"
Указываем номер билда (сборки), в котором была найдена ошибка.

Пример уже введенной, исправленной и закрытой ошибки (Рисунок 15):

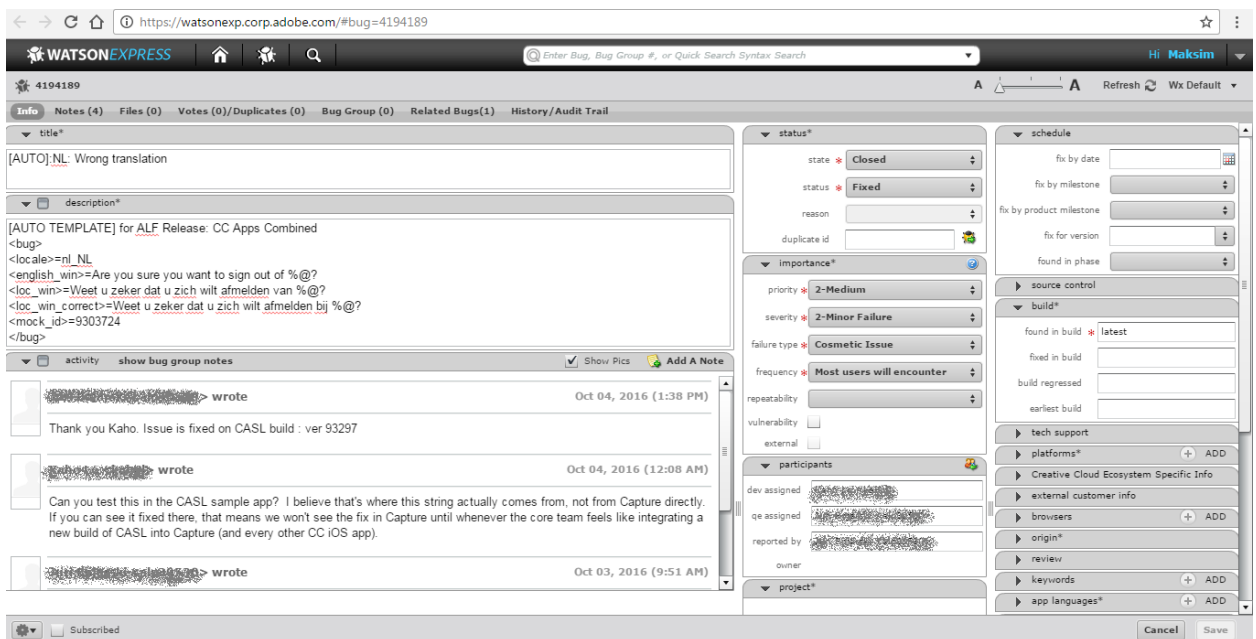


Рисунок 15. Watson. Созданный, исправленный и уже закрытый протокол ошибки.

По сути, на этом все возможности инструмента "Watson" заканчиваются. Система имеет узкий профиль, не предлагая ничего специфического.

4.2.2 Jira

Переход на новую платформу большинства проектов Adobe состоялся по одной, но существенной причине - помимо функций, имеющихся у "Watson", у "Jira" также имеется множество других "полезностей".

Далее мы ознакомимся со многими преимуществами этого инструмента.

- Индивидуальная конфигурация для каждого проекта, которая задается его руководителем. Перечень всевозможных данных при протоколировании баги регулируется и меняется в зависимости от потребностей проекта. Более того, все эти поля можно можно регулировать самому (Рисунок 16), оставляя лишь необходимые и часто используемые поля.

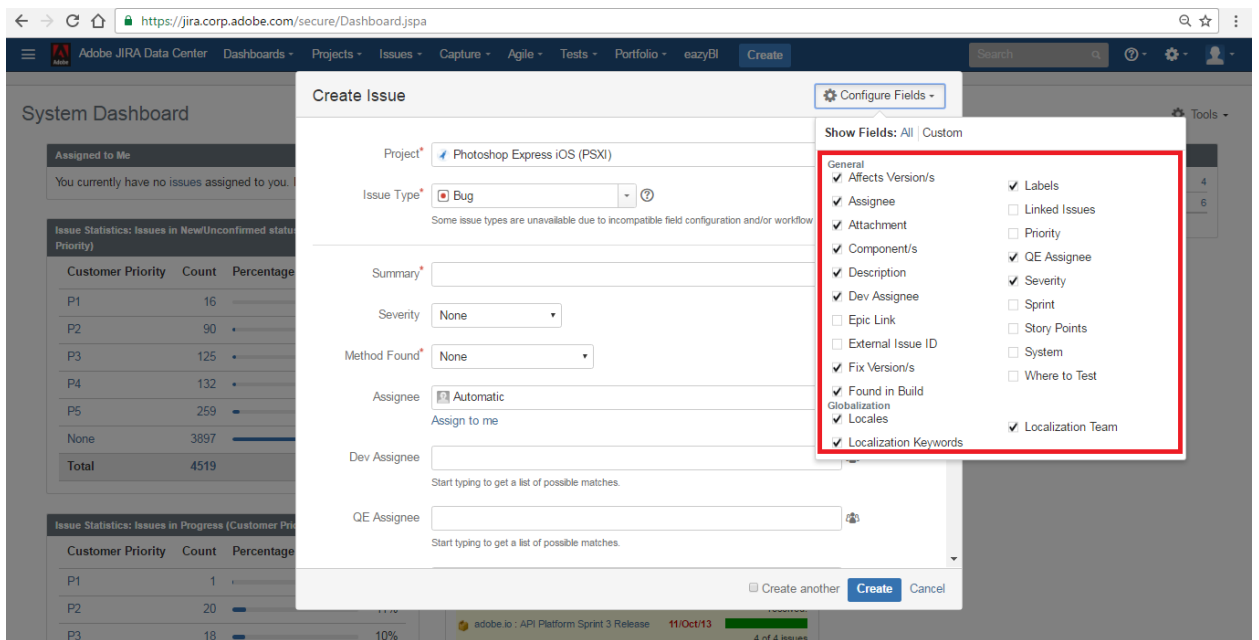


Рисунок 16. Jira. Возможность регулирования полей при вводе баги.

- Имеется поле, где видны все баги, которые предназначены для совершившего вход пользователя для дальнейших действий (Рисунок 17).

The screenshot shows the Jira System Dashboard. On the left, a table titled "Assigned to Me" lists issues with columns for "T", "Key", "Summary", and "P". The issues listed are:

T	Key	Summary	P
ISDK-4153773	[AUTO]:JP: Wrong translation	↑	
ISDK-4153774	[AUTO]:JP: Wrong translation	↑	
ISDK-4173600	[AUTO]:KO: Consistency	↑	
ISDK-4173601	[AUTO]:KO: Consistency	↑	
ISDK-4173602	[AUTO]:KO: Consistency	↑	
ISDK-4173603	[AUTO]:KO: Consistency	↑	
ISDK-4173604	[AUTO]:KO: Consistency	↑	
ISDK-4173605	[AUTO]:ES: Consistency	↑	
ISDK-4173614	[AUTO]:TW: Consistency	↑	
ISDK-4173616	[AUTO]:TW: Consistency	↑	

Below the table, it shows "1-10 of 94" and a pagination control. To the right, there is a "JIRA Road Map: Next 60 Days (Until 03/Mar/17)" section with various project updates and their completion percentages. Further right, there is a "Favorite Filters" section with a filter named "bagri PSF" and a count of 0.

Рисунок 17. Jira. Список предназначенных для пользователя баг.

- Большие возможности для ведения и использования продвинутой статистики. Всевозможные графики (например, рисунок 18), отслеживающие прогресс выполнения заданий.

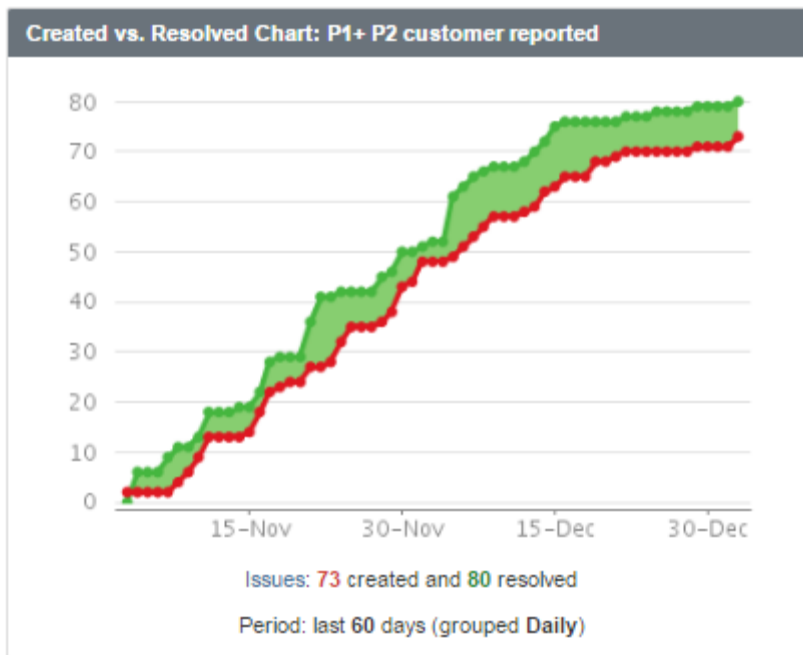


Рисунок 18. Jira. График созданных и разрешенных проблем.

- Создание настраиваемых фильтров, позволяющих сортировать баги по нужным критериям (проекту, типу, времени создания, статусу и многим другим параметрам). Возможность в дальнейшем редактировать эти

фильтры, отображать популярные, наиболее используемые и так далее (Рисунок 19)

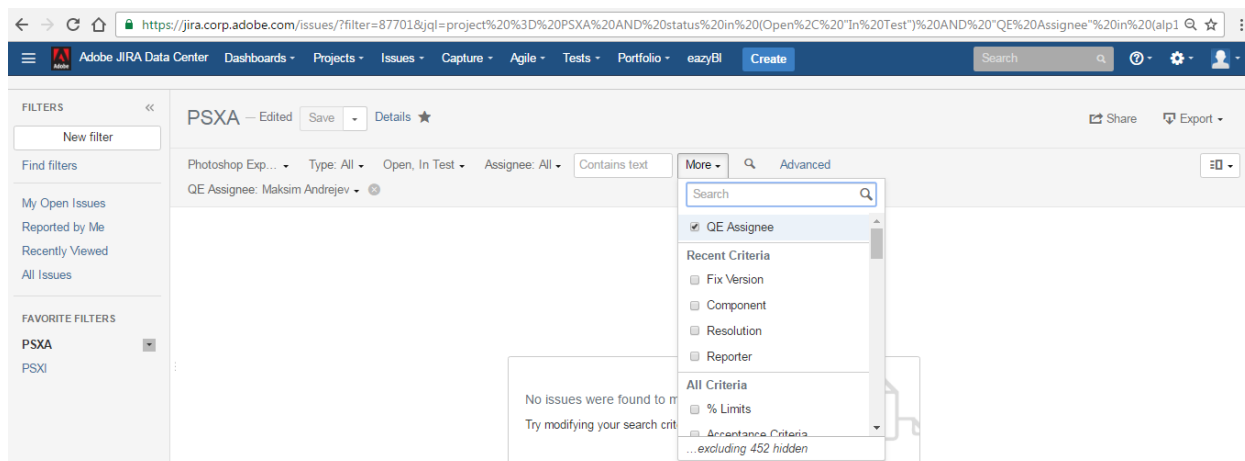


Рисунок 19. Jira. Создание фильтра.

- Можно приспособить "Jira" для непрофильных задач, таких как: планирование и описывание итераций, спринтов, стендапов; написание в ней тест кейсов (Рисунок 6), с последующим редактированием их состояния.

4.3 Сравнение обеих систем

Как можно убедиться из написанного выше, Jira имеет значительно больше плюсов, за исключением очевидных минусов (единственное, к такому можно отнести более сложную систему, для разбирания в тонкостях которой потребуется гораздо больше времени).

Использование более передовых технологий значительно повышает эффективность, позволяет экономить время и повышает работоспособность. Именно из-за всех этих достаточно важных нюансов большинство проектов Adobe Systems "переехали" в новую систему отслеживания ошибок.

Что же касается инструментов для визуализации - то тут предпочтение, как уже было оговорено, отдается платформе Slate, в связи с тем, что функционал у неё значительно больше. Разумеется, имеются и другие инструменты для локализации, у которых также будут какие-то иные существенные преимущества. Но в данной работе был анализ именно тех систем, которые используются в компании Adobe.

5 Заключение

Главной целью дипломной работы был анализ инструментов и систем, предназначенных для оптимизации локализации и дальнейшего тестирования, а также ознакомление с локализацией и многими её особенностями.

В начале работы мы попытались понять, что представляет из себя локализация. Рассмотрели такие важные для локализации аспекты, как: формат даты и времени, система мер, особенности человеческих имен и формат телефонных номеров и так далее. Были достаточно подробно разобраны этапы локализации, начиная от обсуждения проекта и заканчивая тестированием продукта и последующим релизом. Также мы рассмотрели пример из реальной жизни, к чему может привести недостаточно продуманная локализация. Я думаю, было успешно дано обобщенное представление о локализации.

Вдобавок были запланированы анализ и сравнение инструментов для локализации, которые и были успешно проведены. По результатам сравнения мы выявили более удобный и эффективный инструмент для визуализации, ровно как и более продуктивную систему для отслеживания ошибок. Анализ проводился на основе инструментов, используемых компанией Adobe Systems.

Использованная литература

- [1] Testing types. (2015). [WWW]
https://en.wikipedia.org/wiki/Software_testing#Testing_types (05.11.2016)
- [2] Тестирование программного обеспечения. (2016). [WWW]
https://ru.wikipedia.org/wiki/Тестирование_программного_обеспечения
(05.11.2016)
- [3] Локализация по сравнению с интернационализацией. (2010). [WWW]
<https://www.w3.org/International/questions/qa-i18n.ru.php> (19.11.2016)
- [4] Локализация программных продуктов и тестирование локализации. (2010).
[WWW]
<http://www.slideshare.net/VLDCORP/ss-4133794> (02.12.2016)
- [5] Язык в интерфейсе: на короткий-длинный рассчитайся! (2014). [WWW]
<http://app2top.ru/columns/язык-в-интерфейсе-на-короткий-длинный-41251.html> (03.12.2016)
- [6] Особенности локализации программного обеспечения на примере SCADA-системы WinCC. [WWW]
<http://www.sms-automation.ru/about/publications/Localization.pdf> (14.12.2016)
- [7] Исследовательское тестирование: когда его стоит применять и как это делать. (2016). [WWW]
<https://habrahabr.ru/company/redmadrobot/blog/280618/> (14.12.2016)
- [8] Тестовый случай (Test Case). [WWW]
<http://www.protesting.ru/testing/testcase.html> (15.12.2016)
- [9] Dean Leffingwell, "Agile Software Requirements", 2010, 47-61.
- [10] Don't let bugs get into production environments. (2015). [WWW]
<https://crmbusiness.wordpress.com/2015/06/19/dont-let-bugs-get-into-production-environments/> (15.10.2016)
- [11] Waterfall Vs Agile: Project Management Methodologies. (2016). [WWW]
<https://www.techwyse.com/blog/internet-marketing/waterfall-vs-agile-project-management-methodologies/> (15.10.2016)