

Univerzita Pardubice  
Fakulta ekonomicko-správní

Lokalizace softwaru pro české prostředí  
Bakalářská práce

Univerzita Pardubice  
Fakulta ekonomicko-správní  
Akademický rok: 2024/2025

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **David Šlachta**  
Osobní číslo: **E22726**  
Studijní program: **B0688A140011 Digitální podnikání**  
Téma práce: **Lokalizace softwaru pro české prostředí**  
Zadávající katedra: **Ústav systémového inženýrství a informatiky**

## Zásady pro vypracování

Cílem práce je vysvětlit proces a metody překladu a úpravu softwaru pro české prostředí a následně prakticky ukázat na vzorovém příkladu.

Osnova:

- Vysvětlení základních pojmů a metod
- Návrh postupu lokalizace softwaru
- Implementace ve vzorové aplikaci

Rozsah pracovní zprávy: **cca 35 stran**  
Rozsah grafických prací:  
Forma zpracování bakalářské práce: **tištěná/elektronická**

#### Seznam doporučené literatury:

EVJEN, Bill. *ASP.NET 2.0: programujeme profesionálně. Programujeme profesionálně*. Brno: Computer Press, 2006. ISBN ean:9788025114735.  
LOKALISE. *Glossary of translation and localization terms*. Online. 2023. Dostupné také z: <https://lokalise.com/blog/localization-and-translation-glossary-terms/>.  
MICROSOFT. *ASP.NET overview*. Online. 2022. Dostupné také z: <https://learn.microsoft.com/en-us/aspnet/overview>.  
MICROSOFT. *Globalizace a lokalizace aplikací .NET*. Online. 2024. Dostupné také z: <https://learn.microsoft.com/cs-cz/dotnet/core/extensions/globalization-and-localization>.

Vedoucí bakalářské práce: **RNDr. Ing. Oldřich Horák, Ph.D.**  
Ústav systémového inženýrství a informatiky

Datum zadání bakalářské práce: **1. září 2024**  
Termín odevzdání bakalářské práce: **30. dubna 2025**

L.S.

**prof. Ing. Jan Stejskal, Ph.D.** v.r.  
děkan

**doc. Ing. Hana Kopáčková, Ph.D.** v.r.  
garant studijního programu

V Pardubicích dne 1. září 2024

Prohlašuji:

Práci s názvem Lokalizace softwaru pro české prostředí jsem vypracoval(a) samostatně. Veškeré literární prameny a informace, které jsem v práci využil(a), jsou uvedeny v seznamu použité literatury. Byl(a) jsem seznámen(a) s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne

David Šlachta v. r.

## **Poděkování**

Tímto bych velice rád poděkoval RNDr. Ing. Oldřichu Horákovi, Ph.D., svému vedoucímu bakalářské práce, za odborné vedení a cenné rady, které mi pomohly tuto práci zpracovat. Dále bych pak chtěl poděkovat Ing. Martinu Šlachtovi za konzultace při psaní kódu aplikace v praktické části.

## **ANOTACE**

*Tato bakalářská práce se zaměřuje na problematiku lokalizace softwaru pro české prostředí. Práce poskytuje teoretický základ v oblasti lokalizace, internacionalizace a globalizace softwaru. Dále se zabývá praktickou ukázkou lokalizace softwaru demonstrovanou na vývoji aplikace pro HR fiktivní firmy s využitím technologií JavaScript/HTML a .NET (webová i desktopová verze). Práce zkoumá specifika lokalizace, jako jsou jazykové odlišnosti, formáty dat a kulturní aspekty, a poskytuje praktické poznatky pro nové vývojáře, kteří se zabývají lokalizací.*

## **KLÍČOVÁ SLOVA**

*Lokalizace, Internacionalizace, Překlad, Kulturní přizpůsobení, Formáty čísel a datumů*

## **TITLE**

*Software localization for the Czech Environment*

## **ANNOTATION**

*This bachelor thesis focuses on the issue of software localization for the Czech environment. The work provides a theoretical basis in the field of software localization, internationalization and globalization. It also includes a practical demonstration of software localization, shown on the development of an application for HR of fictional company using JavaScript/HTML and .NET (both web and desktop versions). The work explores the specifics of localization, such as linguistic differences, data formats and cultural aspects, and provides practical insights for new developers dealing with localization.*

## **KEYWORDS**

*Localization, Internationalization, Translation, Cultural adaptation, Number and Date formats*

## Obsah

Seznam obrázků .....	10
Seznam tabulek .....	10
Seznam zkratk .....	11
Úvod .....	12
1 Proč se k lokalizaci softwaru přistupuje .....	13
1.1 Porozumění jazyku.....	13
1.2 Rozdílná písma.....	15
1.3 Formáty čísel a datumů .....	17
1.4 Různý význam gest, znaků a symbolů .....	17
1.5 Barvy v různých kulturách .....	18
1.6 Co to všechno znamená.....	19
1.7 Shrnutí.....	20
2 Základy lokalizace .....	21
2.1 Představení lokalizace.....	21
2.2 Definice .....	21
2.2.1 Lokalizace.....	21
2.2.2 Internacionalizace .....	22
2.2.3 Globalizace .....	22
2.2.4 Překlad .....	23
2.3 Co všechno se v rámci lokalizace mění .....	23
2.3.1 Jazyk a písmo.....	24
2.3.2 Číselné, datumové a časové formáty .....	24
3 Proces Lokalizace .....	26
3.1 Analýza a plánování.....	26
3.2 Internacionalizace (i18n).....	26
3.3 Překlad a kulturní přizpůsobení .....	27

3.4	Integrace překladů.....	27
3.5	Kontrola a testování .....	27
3.6	Vydání lokalizované verze .....	27
3.7	Zpětná vazba a opravy .....	27
4	Metody, technologie a nástroje lokalizace .....	28
4.1	Metody lokalizace .....	28
4.1.1	Metody úpravy kódu pro zjednodušení lokalizace v budoucnu .....	28
4.1.2	Metody úpravy obsahu zobrazovaného uživateli .....	28
4.1.3	Metody pro kontrolu a ověření funkčnosti .....	29
4.2	Technologie a nástroje lokalizace .....	30
4.2.1	Strojový překlad .....	30
4.2.2	Systémy pro správu překladu .....	31
4.2.3	Lokalizační platformy.....	32
4.2.4	Nástroje pro testování lokalizace.....	33
5	Lokalizace v praxi.....	35
5.1	Úvod do praktické ukázky .....	35
5.2	Výchozí stav a kontext.....	35
5.3	Cílový stav .....	36
5.4	Lokalizace aplikací .....	37
5.4.1	Převod textů do MS Excel .....	37
5.4.2	Převod textů z MS Excel do vhodného formátu.....	38
5.4.3	Načtení textových řetězců ve webové aplikaci HTML/JS .....	39
5.4.4	Přepínání jazyků a zobrazení lokalizovaných dat ve HTML/JS.....	40
5.4.5	Načtení textových řetězců – .Net platforma .....	43
5.4.6	Přepínání jazyků a zobrazení lokalizovaných dat – .Net platforma .....	44
5.5	Závěr praktické ukázky .....	48
	Závěr.....	50

Použitá literatura .....	51
--------------------------	----

## Seznam obrázků

Obrázek 1 - Znalost cizích jazyků min. na základní úrovni v České populaci k roku 2022 ....	14
Obrázek 2 - Úroveň znalosti cizího jazyka v české populaci.....	15
Obrázek 3 - Porovnání latinky s azbukou .....	16
Obrázek 4 - Palec nahoru .....	18
Obrázek 5 - Start of Action (IEC 60204–1ref.no. 5104).....	18
Obrázek 6 - Význam červené barvy .....	19
Obrázek 7 - Význam černé barvy .....	19
Obrázek 8 - Proces lokalizace .....	26
Obrázek 9 - Expanze na nový trh .....	35
Obrázek 10 - Výchozí stav HR aplikace .....	36
Obrázek 11 - Cílový stav HR aplikace .....	36
Obrázek 12 - Překlad textu pomocí nástroje DeepL.....	38
Obrázek 13 - MS Excel texty .....	38
Obrázek 14 - Tlačítka pro změnu jazyka, Zdroj Vlastní .....	40
Obrázek 15 - Tlačítka pro změnu jazyka 2.....	44
Obrázek 16 - Výsledek lokalizace – .NET desktopová verze .....	48
Obrázek 17 - Výsledek lokalizace – ASP.NET webová verze .....	49
Obrázek 18 - Výsledek lokalizace – JavaScript a HTML .....	49

## Seznam tabulek

Tabulka 1 - Příklad formátovacích řetězců .NET.....	47
---	----

## **Seznam zkratk**

AES	Adult Education Survey
ČSÚ	Český statistický úřad
ECML	Evropské středisko pro moderní jazyky
HR	Human resources
ISO	Mezinárodní organizace pro normalizaci
LISA	Localization Industry Standards Association
UI	Uživatelské rozhraní

## Úvod

**Lokalizace softwaru** označuje proces přizpůsobení softwaru a digitálního uživatelského prostředí jazyku a kulturním normám koncového uživatele v libovolném zeměpisném regionu. Když jde o lokalizaci počítačového softwaru, jedná se o mnohem více, než jen o překlad z jednoho jazyka do druhého. (Kent State University, 2018)

Cílem práce je vysvětlit proces a metody překladu a úpravu softwaru pro (nejen) české prostředí a následně prakticky ukázat na vzorovém příkladu. Tato práce má být návodem pro všechny, kteří chtějí svůj software poskytnout uživatelům z různých zemí.

Jak již úvodní citace naznačuje, práce bude zaměřena na téma lokalizace softwaru. Proč jsem si zvolil pro svou bakalářskou práci právě toto téma? Zvolil jsem ho proto, že v dnešní době globalizace je stále častěji potřeba spolupráce v mezinárodním měřítku. Myslím si tedy, že lokalizace softwaru je dnes velice relevantní téma. Je to totiž právě lokalizace softwaru, jež umožňuje efektivní spolupráci mezi jednotlivými spolupracujícími subjekty, které nemluví stejným jazykem, nebo pocházejí z jiných kultur.

Dále je na místě v rychlosti shrnout, jak bude práce koncipovaná. Práce je rozdělena na dvě části. Část první je čistě teoretická a slouží k uvedení čtenáře do problematiky lokalizace. Druhá část pak předvádí některé metody popsané v první části na konkrétních praktických ukázkách a vysvětluje čtenáři, jak tyto metody použít v praxi. Ukazuje lokalizaci v prostředí .NET jak pro webové, tak desktopové aplikace a dále lokalizaci webových aplikací pomocí JavaScriptu.

# 1 Proč se k lokalizaci softwaru přistupuje

Jedním z důležitých prvků každé aplikace je uživatelské rozhraní. Můžeme mít naprosto revoluční a propracovaný software, který svými funkcemi hravě předčí konkurenci, ale pokud nebude uživatelské prostředí této aplikace srozumitelné, snadno ovladatelné, přehledné a atraktivní, dá uživatel často přednost méně sofistikované a kvalitní aplikaci, která má ale narozdíl od té naší pro uživatele přívětivější rozhraní.

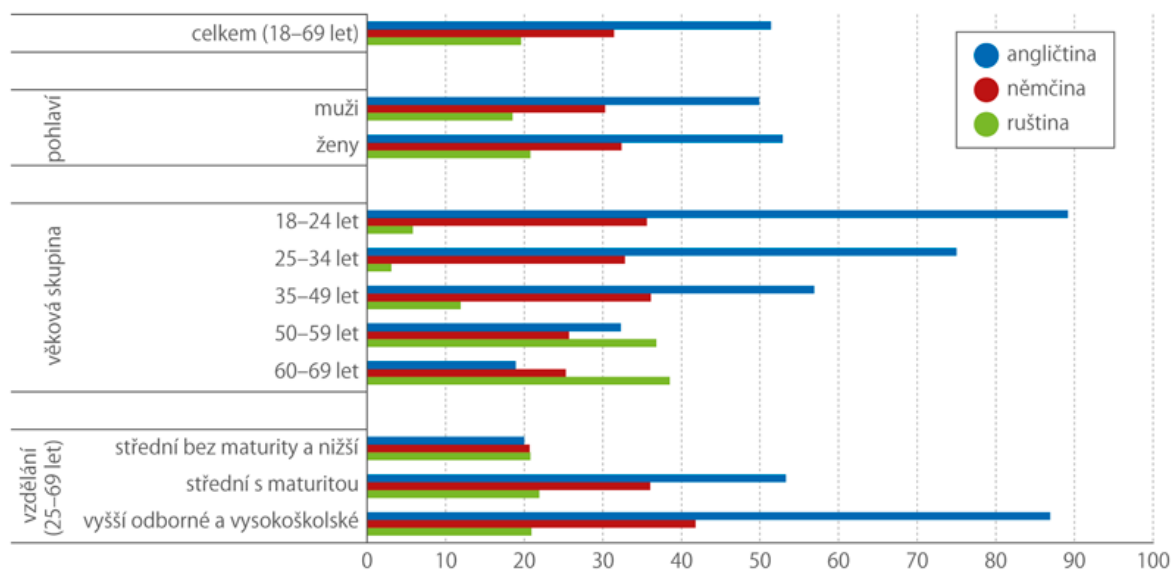
Co to tedy uživatelské rozhraní (UI) je: Jedná se o bod interakce mezi člověkem (uživatel) a softwarem v zařízení. (Hashemi-Pour, 2024) Pro správné fungování a splnění zamýšlené funkce softwaru je tedy třeba, aby si uživatel se softwarem rozuměl. Jenže tady nastává problém.

## 1.1 Porozumění jazyku

Podle Evropského střediska pro moderní jazyky (ECML) se ve světě aktivně používá mezi šesti a sedmi tisíci různých jazyků. Z toho jenom v Evropě (kde je nejmenší jazyková rozmanitost) existuje 225 původních jazyků.

Je sice pravda, že uživatelské rozhraní můžeme navrhnout v jazyce, kterým mluví velké procento lidí, jako je například angličtina. Bohužel angličtinou mluví “pouze“ jeden a půl miliardy z celkových osmi miliard a dvou set milionů lidí na Zemi. I přes to, že je angličtina, co se počtu mluvčích týče, nejrozšířenější jazyk na světě, stále ji ovládá pouze necelá pětina světové populace.

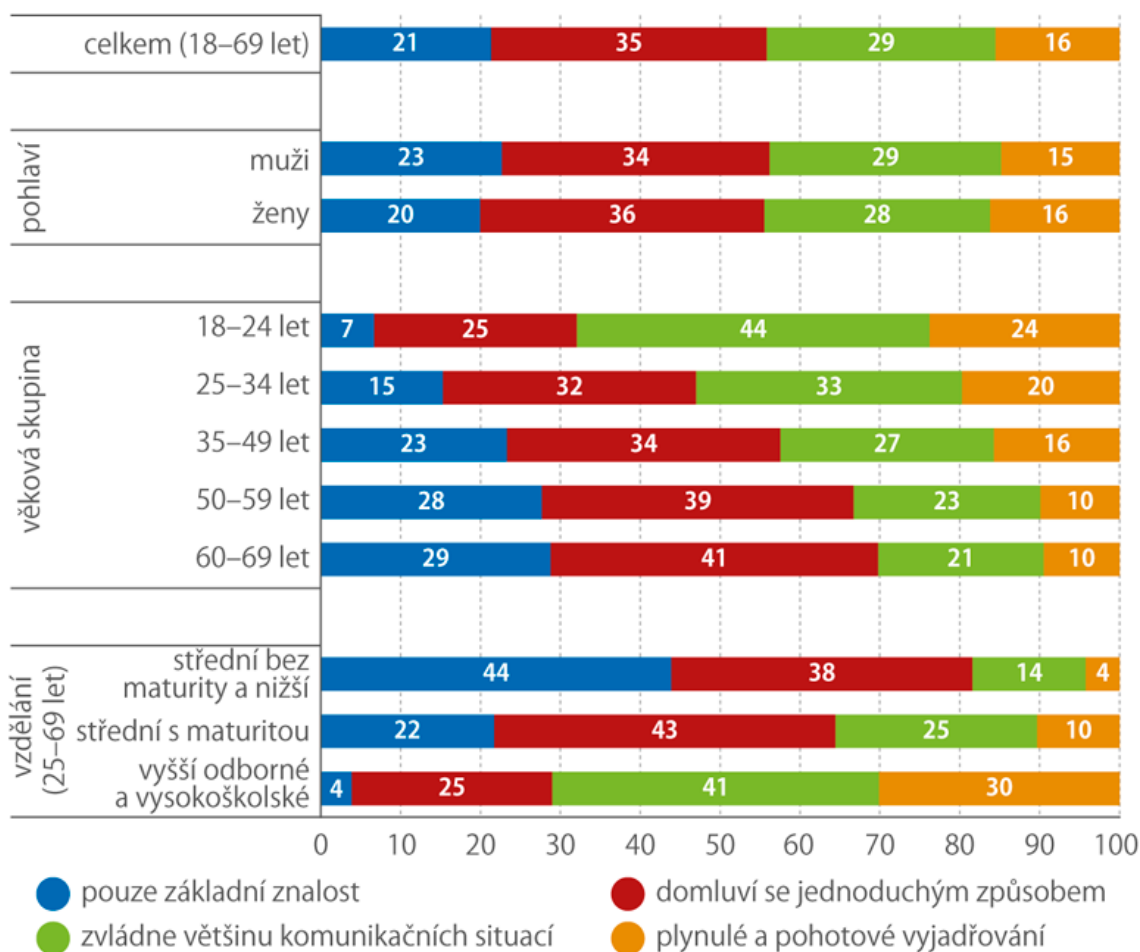
Pokud se pak budeme soustředit na český trh, nebude tato situace o moc lepší. Podle dat Českého statistického úřadu k roku 2022 (v roce se 2022 opět konalo celoevropské šetření o vzdělávání dospělých (Adult Education Survey – AES)) se anglicky domluví každý druhý obyvatel Česka. Rozvržení mezi populací je následující: Mezi mladými ve věku 18–24 let angličtinu ovládá 89 %, zatímco z osob ve věku 50–59 let je to pouhých 32 %, tedy necelá třetina. Ještě nižší podíl vykazali obyvatelé ve věku 60–69 let. Angličtinu alespoň na základní úrovni ovládá pouze necelá pětina (19 %) z nich. (Hykyšová, 2024). Pro ilustraci těchto čísel přidávám obrázek grafu od ČSÚ.



Obrázek 1 - Znalost cizích jazyků min. na základní úrovni v České populaci k roku 2022

Zdroj: ČSÚ

Dále je pak třeba počítat s tím, že relativně velká část z těch, co angličtinu ovládají, ji ovládá pouze na základní úrovni. Podle dat českého statistického úřadu 16 % z anglicky mluvících Čechů ovládá angličtinu pouze na úrovni A1. To znamená, že dokážou komunikovat pouze velice jednoduchým způsobem. Nemají širokou slovní zásobu, a ta slovní zásoba, kterou ovládají je zaměřena spíše na obecná témata. V případě zaměření na konkrétní odborná témata (např. ekonomika, IT, politika atd.) tak mohou mít problém porozumět danému tématu. Na základě grafu níže můžeme dokonce předpokládat, že z Čechů, kteří ovládají angličtinu, ji na úrovni, kdy se zvládne domluvit ve většině situací, ovládá pouze zhruba polovina.



Obrázek 2 - Úroveň znalosti cizího jazyka v české populaci

Zdroj: ČSÚ

Na základě tohoto je třeba se při vývoji softwaru tedy zaměřit na to, aby byl vyvíjen v různých jazycích podle toho, na jaký trh aplikace cílí. Ale pozor! Nestačí pouze přeložit texty v softwaru. Je třeba se zamyslet i nad tím, že uživatelé se liší i v jiných věcech než jazyk, kterým mluví.

## 1.2 Rozdílná písma

Mimo to, že relativně velká část populace neovládá cizí jazyky v mluvené formě, tak ještě větší část pak bude mít potíže s formou psanou. Hlavním důvodem je pak fakt, že různé světové jazyky používají i různou abecedu, různá písma. Jednotlivá písma se mohou lišit jak jednotlivými znaky, tak i tím, jak se zapisují. V Evropě používaná latinka se například zapisuje z levé strany směrem doprava. Oproti tomu například arabské písmo se zapisuje opačným směrem. Čínské, japonské a mongolské znaky se pak často zapisují shora dolů. V některých

historických případech se pak dokonce mohl směr psaní měnit. Příkladem je písmo starých Etrusků, které využívalo řádky střídavě psané zleva doprava a zprava doleva. Může proto nastat situace, kdy uživatel nejen že nerozumí jazyku, ale dokonce ani nedokáže přečíst písmo, které jazyk používá. Nebo naopak rozumí základním výrazům v mluvené formě, ale nedokáže se orientovat ve formě psané. (Coulmas, 1999)

Aa	Bb	Cc	Dd	Ee	Aa	Бб	Вв	Гг	Дд
Ff	Gg	Hh	Ii	Jj	Ee	Ёё	Жж	Зз	Ии
Kk	Ll	Mm	Nn	Oo	Йй	Кк	Лл	Мм	Нн
Pp	Qq	Rr	Ss	Tt	Оо	Пп	Рр	Сс	Тт
Uu	Vv	Ww	Xx	Yy	Уу	Фф	Хх	Цц	Чч
Zz					Шш	Щщ	Ъъ	Ыы	Ьь
					Ээ	Юю	Яя		

Obrázek 3 - Porovnání latinky s azbukou

*Zdroj: Vlastní*

Pro české prostředí blízkým příkladem může být třeba ukrajinština. I přes to, že některá ukrajinská slova jsou v mluvené formě téměř totožná s jejich českým protějškem, je pro průměrného Čecha velice složité těm samým slovům porozumět ve formě psané. Je tomu tak proto, že i přes příslušnost ke stejné jazykové skupině používají oba jazyky jinou abecedu. Jak čeština, tak ukrajinština patří do skupiny slovanských jazyků. Ukrajinština pak používá pro zápis azbuku (také označovaná jako cyrilice) zatímco čeština používá latinskou abecedu (také známá jako latinka).

Můžeme si to ukázat na následujících příkladech:

Новий (novyj) = nový

Старий (staryj) = starý

Малий (malyj) = malý

Великий (velykyj) = velký

Холодний (cholodnyj) = chladný (studený)

Теплий (teplyj) = teplý

### 1.3 Formáty čísel a datumů

Dalším, na první pohled sice drobným rozdílem, který ale může zásadně ovlivnit obsah sdělení, jsou formáty čísel a datumů. Ty se mohou podle zemí lišit.

Rozdíly ve formátu data mohou způsobit mnoho nepříjemností. V Evropě používáme formát DD.MM.YYYY (např. 12.02.2025). V USA je pak typický formát MM/DD/YYYY (např. 02/12/2025). Dále se také můžeme setkat s formátem YYYY-MM-DD (např. 2025-02-12) podle mezinárodního standardu ISO 8601, který se používá v Japonsku a dále technických systémech po celém světě. Problém tedy nastává, pokud dva spolu komunikující lidé nevědomky používají jiný formát datumu. Tento fakt můžeme krásně reprezentovat na příkladu.

John z New Yorku a jeho kolega Peter z pobočky ve Frankfurtu si v e-mailové komunikaci domluví online schůzku na 12:25 02/03/2025. Každý z nich má ale ve svém kalendáři jiné datum. Je tomu tak proto, že John se řídil formátem typickým pro USA, zatímco Peter pro Evropu. John tedy počítá se schůzkou třetího února a Peter druhého března. Kolegové se tak nesejdou, a to dokonce o celý měsíc. Vše kvůli rozlišnému formátu data.

To samé, co platí o datumu, bude platit i o zápisu čísel. Zde může být rozdílné jak oddělení tisíců, tak oddělení desetinných míst. V Evropě je typické oddělování desetinných míst pomocí čárky (např. 10,2). Pro anglicky mluvící země je ale běžně používaným oddělovačem v tomto případě tečka (např. 10.2). Co se týče oddělování tisíců, tak ve většině Evropy se setkáme s oddělením pomocí mezery (např. 1 000 000), v USA pak pomocí čárky (např. 1,000,000) a v některých zemích jako například Brazílie je k oddělení tisíců používána tečka (např. 1.000.000). I tady tak může nastat nedorozumění podobné příkladu s datumem.

A to třeba při nákupu akcií v určité hodnotě, kdy jedna strana má v úmyslu nakoupit akcie za 145,500, slovy sto čtyřicet pět tisíc pět set, ale druhá strana používající jiný formát zápisu to pochopí jako sto čtyřicet pět celých pět desetin.

### 1.4 Různý význam gest, znaků a symbolů

Mnoho lidí si myslí, že gesta, znaky a symboly jsou stejné po celém světě a jsou tedy univerzální. Je sice pravda, že v mnoha zemích mají základní gesta a symboly, jako je třeba palec nahoru, nebo použití počtu na prstech ruky, stejný nebo velmi podobný význam. I tady ale existují výjimky, které v některých případech mohou zcela změnit význam a pochopení toho, co se snažíme jejich pomocí sdělit. Pro lepší představu uveďme pár příkladů.

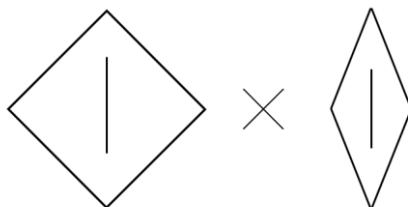
Pro západní svět univerzální gesto palce zdviženého nahoru, které v česku znamená. V Thajsku je ale toto gesto používáno místo vyplazování jazyka a v některých regionech Itálie a Řecka se toto gesto využívá ve vulgárním významu. (Labohá, 2017)



Obrázek 4 - Palec nahoru

*Zdroj: Ikony Microsoft Office Word*

Dalším příkladem může být třeba piktogram s názvem Start of Action (IEC 60204-1 symbol reference no. 5104). Jedná se o čtverec otočený o 45 stupňů s vertikální linií v jeho středu. Tento znak se podle ISO (Mezinárodní organizace pro standartizaci) používá pro identifikaci tlačítka start. (International Organization for Standardization; International Organization for Standardization) Nejčastěji můžeme tento symbol vidět na tiskárnách pro firemní a kancelářské účely a scannerech, kde slouží jako označení tlačítka pro zahájení úlohy (tiskové nebo skenovací). Jenomže velice podobný symbol, kosočtverec s vertikální linkou uvnitř, se v Česku používá i k vyjádření velice vulgárního výrazu. Přitom obdobný symbol je logem jedné francouzské automobilky.



Obrázek 5 - Start of Action (IEC 60204–1ref.no. 5104)

*Zdroj: Vlastní*

## 1.5 Barvy v různých kulturách

To samé, co platí o gestech znacích a symbolech platí také o barvách. Jednotlivé barvy mohou mít v různých kulturách velmi rozdílný význam. V případě barev nebývají rozdíly mezi kulturami tak časté a výrazné, ale i přesto se občas setkáme s jinou barevnou symbolikou. Proto se může jednoduše stát, že použití barvy pro podtrhnutí nějakého sdělení v aplikaci (například barevné ovládací prvky – tlačítka atd.), uživatelům z jednoho kulturního prostředí pomůže

význam správně pochopit, ale lidem, kteří pochází z jiného kulturního prostředí naopak zamotá hlavu.

Jako příklad můžeme použít červenou barvu. Zatímco v evropské a africké kultuře je červená barva od pradávna spojována s barvou krve a slouží tedy často jako barva pro výstrahu a nebezpečí, tak v Číně je červená barva symbolem štěstí a v Indii symbolizuje panenství a čistotu. (Kenner, 2006)



Obrázek 6 - Význam červené barvy

*Zdroj: Vlastní*

Další barvou, která má v různých kulturách různý význam je černá. V západní a evropské kultuře je černá barva spjatá s morovými ranami. Proto je používána jako symbol pro truchlení a smrt. V některých částech Asie je ale černá barva považována za symbol stability a prosperity. (Kenner, 2006)



Obrázek 7 - Význam černé barvy

*Zdroj: Vlastní*

## 1.6 Co to všechno znamená

V dnešní době globalizace většina softwarových vývojářů necílí pouze na lokální trhy, ale snaží se své softwarové řešení prosadit na trhu globálním, a to ať už se jedná o nezávislá softwarová řešení nebo softwarová řešení v rámci nadnárodních korporací. Na základě výše uvedených příkladů je ale jasně vidět, že není možné použít jednu identickou verzi softwaru pro všechny uživatele. Vždy je třeba přizpůsobit produkt trhu na který cílí, a proto je potřeba

jednotlivým trhům přizpůsobit i všechny prvky softwaru. Pokud by nebyl software pro konkrétní trh správně přizpůsoben, citelně by to ovlivnilo jeho funkčnost pro uživatele a tím snížilo jeho konkurenceschopnost.

Pokud by však měl softwarový vývojář pro každý trh (myšleno národní trhy – pro Česko, pro Německo atd.) vytvářet novou verzi softwaru bylo by to jak časově, tak finančně velmi nákladné. Proto vývojáři už od začátku při tvorbě zohledňují tuto potřebu diferenciaci některých prvků v uživatelském rozhraní a zpracování uživatelských vstupů. Software je budován tak, aby v budoucnu při potřebě uvedení softwaru na cizí trh tuto úpravu co nejvíce usnadnil. A přesně na to, jak to vývojáři dělají se podíváme v další kapitole.

## **1.7 Shrnutí**

Na světě je nespočet různých národů a kultur, které se mohou lišit jak svým jazykem, tak svými zvyklostmi, vnímáním symbolů nebo dokonce barev. Proto je třeba software určený pro použití v mezinárodním multikulturním prostředí vždy uzpůsobit dle potřeb uživatelů, na které cílí. Aby si to vývojáři softwaru usnadnili a nemuseli vytvářet pro každý trh jinou verzi aplikace, tak už při vývoji uzpůsobují software tomu, aby podporoval více jazyků. Zároveň je softwarové řešení tvořeno tak, aby do budoucna co nejvíce usnadnilo rozšíření o další podporované jazyky.

## 2 Základy lokalizace

### 2.1 Představení lokalizace

Během posledních desetiletí se lokalizace proměnila z okrajové aktivity několika softwarových vydavatelů na významné průmyslové odvětví v hodnotě miliard dolarů. Spolu s ní se do popředí zájmu firem dostaly i globalizace, jazykové inženýrství a internacionalizace softwaru. Tyto oblasti jsou dnes klíčové pro společnosti, které chtějí vstoupit na mezinárodní trhy a úspěšně tam prodávat své produkty. Lokalizace se navíc často ukazuje jako zásadní faktor pro přijetí a úspěch produktů u zahraničních zákazníků.

Slovo "lokalizace" vychází z termínu "locale", který původně označoval menší oblast nebo místo. V současnosti se však v technickém prostředí používá k označení konkrétního nastavení, které zahrnuje kombinaci jazyka, geografického regionu a znakové sady. Lokalizace v zásadě znamená překlad a úpravu softwarového produktu či webových stránek tak, aby odpovídal požadavkům daného trhu. Zahrnuje nejen samotnou softwarovou aplikaci, ale také veškeré související materiály, jako je dokumentace nebo uživatelské příručky. (Esselink, 2000)

### 2.2 Definice

#### 2.2.1 Lokalizace

LISA (Localization Industry Standards Association = Asociace standardů lokalizačního průmyslu), byla známá svým úsilím o zlepšení kvality a efektivity procesů lokalizace softwaru a překladů. Působila mezi roky 1990 a 2014. Vytvářela standardy a doporučení pro best practices v oblasti, které jsou využívány dodnes. LISA definuje Lokalizaci takto:

„Lokalizace zahrnuje přizpůsobení produktu jazykově a kulturně tak, aby byl vhodný pro cílovou lokalitu (zemi/region a jazyk), kde bude používán a prodáván.“

Ve většině případů (cca 80 %) se lokalizuje z angličtiny do ostatních jazyků. Je tomu tak proto, že relativně velká část softwaru je vyvíjena v USA. Dalším faktorem je to, že i vývojáři z jiných zemí často svůj produkt vytvářejí v angličtině pro jednodušší použití na mezinárodním trhu. V neposlední řadě je pak angličtina brána jako jazyk IT.

Pokud je software lokalizován dobře, koncový uživatel ho může plně využívat, a to za použití svého rodného jazyka a bez jakýchkoliv omezení. To znamená, že může číst obsah a rozumí všem prvkům softwaru od vyskakovacích oken (potvrzovací nebo chybové hlášky),

může využívat veškeré vstupy (vyplňovací pole, výběrové pole atd.) a to vše ve svém rodném jazyce a za použití lokální znakové sady a použití místních formátů.

Lokalizace bývá také někdy označována pomocí zkratky „L10n“, kde L a N značí první a poslední písmeno, zatímco číslovka 10 nahrazuje 10 písmen mezi těmito dvěma písmeny (platí pro anglickou verzi slova Localization) (Esselink, 2000)

### **2.2.2 Internacionalizace**

Pojem internacionalizace bývá často zaměňován s lokalizací, jedná se ale pouze o jednu z metod lokalizace, která má za cíl už při psaní kódu myslet na budoucí lokalizaci. Definice pojmu podle LISA (Localization Industry Standards Association = Asociace standardů lokalizačního průmyslu) zní takto:

„Internacionalizace je proces zobecnění produktu tak, aby mohl podporovat více jazyků a kulturních zvyklostí, aniž by bylo nutné jej znovu navrhovat. Internacionalizace probíhá na úrovni návrhu programu a tvorby dokumentace.“

Obecně tedy platí, že pokud má vývojář v plánu použití jeho produktu na globálním trhu, nebo na trhu s jiným kulturním pozadím a používaným jazykem, je pro něj výhodné produkt pro lokalizaci připravit již během vývoje. Právě tento proces označujeme jako internacionalizaci.

Internacionalizace není pouze záležitostí softwaru, ale zahrnuje také další aspekty softwarového produktu: online nápovědu, technickou dokumentaci a webové stránky, které je nutné přizpůsobit pro globální publikum. Tento proces se často označuje jako „psaní pro překlad“ nebo „psaní pro mezinárodní publikum“. To znamená, že je obsah strukturován tak, aby byl jednoduše lokalizovatelný do různých jazyků a kulturních kontextů. U webových stránek se internacionalizace kombinuje s překladem a přizpůsobením obsahu specifickým trhům. To se označuje jako „globalizace webových stránek“. Cílem je zajistit, aby informace byly relevantní a přístupné uživatelům po celém světě, a to bez ohledu na jejich jazyk nebo region. (Esselink, 2000)

### **2.2.3 Globalizace**

LISA (Localization Industry Standards Association = Asociace standardů lokalizačního průmyslu) definuje globalizaci následovně:

„Globalizace se zabývá širšími obchodními aspekty spojenými s uvedením produktu na globální trh. V případě high-tech produktů globalizace zahrnuje integraci lokalizace napříč

celou společností, včetně internacionalizace, návrhu produktu, marketingu, prodeje a zákaznické podpory na světovém trhu.“

S pojmem globalizace se setkáme v mnoha různých kontextech. Na nejvyšší obecné úrovni mluvíme o globalizaci z pohledu geopolitického. V tomto případě je globalizace vnímána jako jistá forma evoluce, vedoucí k propojení trhů v globálním měřítku. Na nižší úrovni pak mluvíme o globalizaci podniků, kde společnosti pomocí zakládání místních poboček dosahují své mezinárodní přítomnosti, případně dosahují mezinárodní přítomnosti na základě spolupráce s dalšími podniky.

V kontextu této práce se pak budeme soustředit na globalizaci v kontextu softwarového trhu, kde globalizace zahrnuje jak internacionalizaci a lokalizaci, tak marketing a distribuci. Zjednodušeně řečeno je globalizace v tomto kontextu souhrnem všech částí procesu, při kterém se softwarový vývojář snaží dostat na mezinárodní trh. (Esselink, 2000)

#### **2.2.4 Překlad**

Překlad je nedílnou součástí lokalizace softwaru. Je to samotný proces převodu psaného textu nebo mluveného slova z jednoho jazyka do jazyka jiného. U překladu je důležité zachování původního významu textu a kontextu. Proto často nelze text přeložit doslovně, protože některá rčení a ustálené výrazy po doslovném překladu nemusí dávat smysl. Nesprávný překlad může mít za následek ztrátu původního významu a tím vést k nedorozumění.

### **2.3 Co všechno se v rámci lokalizace mění**

Jak již bylo naznačeno v kapitole „Proč se vlastně k lokalizaci softwaru přistupuje“, tak nestačí pouze přeložit texty. Co všechno je ale potřeba v rámci lokalizace zohlednit?

1. *Jazyk*
2. *Písmo*
3. *Číselné, datumové a časové formáty*
4. *Jednotkový systém*
5. *Použití měny*
6. *Použitá klávesnice*
7. *Třídění a řazení*
8. *Symboly, ikony a barvy*
9. *Text a grafika obsahující odkazy na objekty, akce nebo myšlenky, které mohou být v určité kultuře nesprávně interpretovány nebo vnímány jako necitlivé*

## 10. Různé právní požadavky

## 11. Další specifické požadavky

V některých případech je dokonce třeba kompletně přehodnotit logiku softwaru, jeho vizuální prezentaci a design, nebo dokonce i funkce lokalizovaného softwaru. (Příkladem může být účetnictví, kde se pro algoritmy pro různý trh budou lišit na základě právních stanov daného trhu/státu) (Richard Ishida, 2005)

### 2.3.1 Jazyk a písmo

Základním prvkem, který se během lokalizace mění jsou texty použité v softwaru. Tyto texty je třeba přeložit/převést do jazyka trhu, pro který je software lokalizován (například pro Německo do němčiny). Zároveň je třeba použít i písmo, které se k tomuto jazyku váže (Azбуka pro ruštinu atd.)

### 2.3.2 Číselné, datumové a časové formáty

#### 2.3.2.1 Číselné formáty

U číselných formátů se v rámci lokalizace mění použité oddělovače používané při psaní desetinných čísel a tisíců.

Při psaní desetinných čísel se používají dva různé formáty:

1. Desetinná část oddělená čárkou => 10,25 (slovy: deset celých dvacet pět setin)
2. Desetinná část oddělená tečkou => 10.25 (slovy: deset celých dvacet pět setin)

Při psaní čísel v řádech tisíců a výš se pak využívá třech různých formátů:

1. Řády odděleny pomocí mezery => 2 110 000 (slovy: dva miliony sto deset tisíc)
2. Řády odděleny pomocí čárky => 2,110,000 (slovy: dva miliony sto deset tisíc)
3. Řády odděleny pomocí tečky => 2.110.000 (slovy: dva miliony sto deset tisíc)

#### 2.3.2.2 Datumové formáty

V případě datumových formátů se rozlišuje krátký a dlouhý zápis data, s tím že jak dlouhý, tak krátký zápis datumu může nabývat mnoha různých formátů. Pro znázornění použijeme české formáty, formáty používané v USA a formáty používané mezinárodně (ISO 8601).

### Česká republika (ČSN 01 6910)

Krátký formát:

1. DD.MM.YYYY => 01.03.2025

2. D. M. YYYY => 1. 3. 2025

Dlouhý formát:

1. D. měsíc YYYY => 1. března 2025
2. Den v týdnu, D. měsíc YYYY => Sobota, 1. března 2025

Úřední zápis (ISO 8601 kompatibilní):

1. YYYY-MM-DD => 2025-03-01

#### **USA (MM/DD/YYYY, americký formát)**

Krátký formát:

1. M/D/YY => 3/1/25
2. MM/DD/YYYY => 03/01/2025

Dlouhý formát:

1. Month D, YYYY => March 1, 2025
2. Day, Month D, YYYY => Saturday, March 1, 2025

#### **ISO 8601 (Mezinárodní standard, používaný v databázích a administrativě)**

Krátký formát:

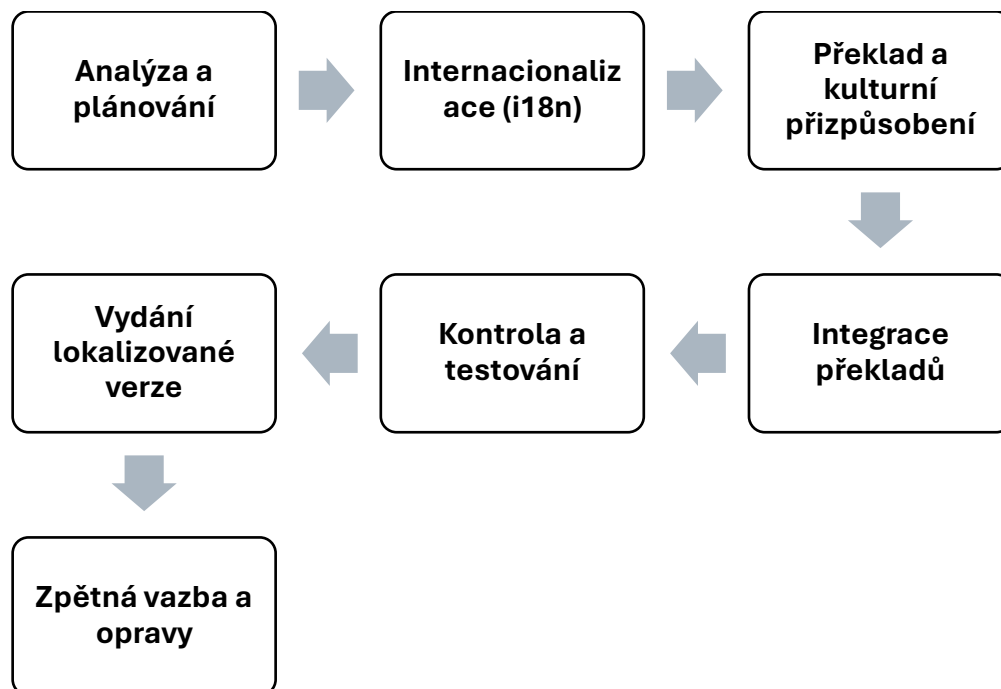
1. YYYY-MM-DD => 2025-03-01

Dlouhý formát:

1. Den v týdnu, YYYY-MM-DD => Saturday, 2025-03-01

### 3 Proces Lokalizace

Tato kapitola slouží pro pochopení procesu lokalizace, aby pak čtenář správně chápal následující kapitolu v jeho kontextu. Proces lokalizace můžeme rozdělit následovně.



Obrázek 8 - Proces lokalizace

Zdroj: Vlastní

#### 3.1 Analýza a plánování

Jedná se o často přehlíženou fázi lokalizačního procesu, a to i přes to, že se jedná o naprostý základ každého úspěšného projektu (ať už lokalizačního nebo jakéhokoliv jiného). V této fázi je třeba analyzovat výchozí stav (stav trhu, stav produktu atd.) a určit si stav cílový (tj. nastavit si cíle). Také je třeba si správně nastavit omezení projektu (časový plán, budget atd.). V případech, kdy je tato fáze zanedbána nebo opomíjena se velmi často stává, že projekt dopadne neúspěšně.

#### 3.2 Internacionalizace (i18n)

V této fázi dochází k oddělení zdrojového kódu a lokalizovaných prvků, zavádí se použití Unicode (zajišťuje zobrazení různých znakových sad) a zajišťuje se flexibilita uživatelského rozhraní. Ve zkratce řečeno se software upravuje pro lokalizaci. Správně provedená fáze internacionalizace umožňuje bezproblémovou budoucí lokalizaci.

### **3.3 Překlad a kulturní přizpůsobení**

Po fázi internacionalizace již dochází k samotným překladům a kulturním přizpůsobením. Je třeba vyextrahovat všechny textové řetězce a následně je přeložit do cílových jazyků. Dále se v této fázi přizpůsobují další prvky softwaru (jako formáty datumů a čísel, měny, symboly atd.) tak, aby byly vhodné pro cílový trh.

### **3.4 Integrace překladů**

Zde se přistupuje k začlenění přeložených textových řetězců a dalších upravených prvků zpět do softwaru.

### **3.5 Kontrola a testování**

Následně je potřeba lokalizovaný produkt (software) otestovat a zkontrolovat. Zkoumá se, zda nebyla při lokalizaci narušena funkčnost softwaru (zda se po lokalizaci neobjevily tzv. bugy atd.) a také jestli jsou všechny lokalizované prvky přeloženy a upraveny správně (tj. dávají smysl, nemění význam sdělení, zachovávají si kontext atd.) pro cílový trh. Pokud testování odhalí nějaký problém, tak je třeba se vrátit k předchozím fázím a problém vyřešit.

### **3.6 Vydání lokalizované verze**

Pokud všechny předchozí fáze proběhly v pořádku, je možné přistoupit k uvedení lokalizované verze softwaru na trh. Je důležité, aby tato fáze proběhla hladce. Při špatném postupu můžeme ohrozit funkcionalitu softwaru v jeho předchozí verzi (př. Nedostupnost serverů po dobu aktualizace – zastavení výroby – finanční ztráta).

### **3.7 Zpětná vazba a opravy**

Tato fáze probíhá už po uvedení produktu (lokalizované verze softwaru) na trh. Spočívá v shromažďování zpětné vazby od zákazníků a v případné opravě, pokud zákazník odhalí problém (který nebyl odhalen ve fázi testování).

(Esselink, 2000)

## 4 Metody, technologie a nástroje lokalizace

V rámci lokalizace se používá množství různých metod nástrojů a technologií, které proces lokalizace co nejvíce usnadňují a některé aspekty lokalizace dokonce dokážou automatizovat. Na místě je tedy zmínit a popsat ty nejpoužívanější napříč všemi platformami.

### 4.1 Metody lokalizace

Metody lokalizace jsou ve zkratce základní pilíře lokalizace a jejího zjednodušení. Metody lze rozdělit na ty, které jsou použity už během tvorby kódu. Dále pak rozeznáváme metody, které upravují různý obsah, který se bude dostávat k uživateli. Neméně důležité jsou pak metody, které slouží k otestování lokalizace a ověření, že nový lokalizovaný obsah je v pořádku a uživatelsky přívětivý.

#### 4.1.1 Metody úpravy kódu pro zjednodušení lokalizace v budoucnu

Jedná se o metody, které jsou aplikovány už při prvotním vývoji softwaru. Ve zkratce slouží k předpřípravě softwaru pro jeho následnou lokalizaci. Do této kategorie se řadí internacionalizace a správa překladových pamětí.

##### 4.1.1.1 Internacionalizace v kontextu metod lokalizace

Je to základní krok, který připravuje aplikaci pro podporu více jazyků a regionálních nastavení. Zahrnuje oddělení textových řetězců od kódu a použití lokalizačních knihoven.

##### 4.1.1.2 Správa překladových pamětí

Ukládání a znovupoužití už dříve přeložených textů pro zvýšení konzistence a efektivity překladu v budoucnu

(Esselink, 2000)

#### 4.1.2 Metody úpravy obsahu zobrazovaného uživateli

Jedná se o nejobsáhlejší skupinu metod lokalizace, protože v rámci lokalizace obsahu je třeba měnit téměř vše. Radíme sem následující metody:

- **Překlad textu** – Zabývá se převedení textových prvků do cílového jazyka s ohledem na jeho jazykové a kulturní specifika.
- **Formátování datumů a čísel** – Přizpůsobování formátu datumů, času, měn a dalších číselných prvků podle místních konvencí.

- **Kulturní adaptace obsahu** – Úprava obsahu pro maximální relevanci a soulad s místními kulturními normami a hodnotami.
- **Přizpůsobení uživatelského rozhraní** – Přizpůsobení designu/návrhu aplikace nebo webu na základě lokálních zvyklostí a požadavků.
- **Transkrece** – Kombinace překladu a tvorby nového obsahu s cílem lépe oslovit cílové publikum/uživatele. Používá se převážně v marketingu (například jiná reklama pro softwarový produkt na různých trzích)
- **Úprava obrázků a další grafiky** – Adaptace vizuálních/grafických prvků kulturním a estetickým preferencím cílového trhu.
- **Úprava zvukových a video souborů** – Přidání dabingu, titulek nebo náhrada zvukových stop s cílem plně zpřístupnit tento obsah uživatelům na dalších trzích.
- **Optimalizace vyhledávání** – Přizpůsobení vyhledávání pro cílový trh – tzn. překlad a změna klíčových slov, tak aby byly výsledky vyhledávání relevantní pro místní potřeby.
- **Právně-regulační přizpůsobení** – Zajištění souladu softwarového produktu a jeho obsahu se všemi místními právními předpisy a normami.
- **Lokalizace platby a fakturace** – Přizpůsobení platebních metod, možností a fakturačních procesů pro cílový trh. (protíná se s právně – regulačními přizpůsobeními)

### 4.1.3 Metody pro kontrolu a ověření funkčnosti

Tento segment bývá často opomíjen i přesto že se jedná o velmi důležitou součást procesu lokalizace. Tyto metody zajišťují to že výsledný produkt, který se dostane k uživateli neobsahuje žádné chyby a nedokonalosti, které by mohli pokazit jeho zkušenost s produktem. Těmito metodami jsou:

- **Kulturní konzultace** – Spolupracování s kulturními experty pro cílový trh, která má za cíl odhalení skrytých kulturních nuancí.
- **Testování lokalizace** – Testování lokalizovaného obsahu místními uživateli pro kontrolu jeho správnosti a srozumitelnosti.
- **Testování uživatelského rozhraní** – Testování uživatelského prostředí pro cílový trh pro ověření jeho plné funkčnosti a použitelnosti.

## 4.2 Technologie a nástroje lokalizace

Díky rychlému vývoji technologií v poslední době je dnes k dispozici velké množství nástrojů, které výrazně ulehčují a zjednodušují proces lokalizace.

Patří sem:

- technologie strojového překladu,
- systémy pro správu překladů,
- lokalizační platformy,
- nástroje testování lokalizace,
- frameworky a knihovny pro internacionalizaci softwaru.

### 4.2.1 Strojový překlad

Strojový překlad (také machine translation, nebo zkráceně MT) je technologie strojového učení k automatizovanému překladu velkého množství textu. Umožňuje překládat texty, webové stránky a dokumenty mezi různými jazyky. Díky technologiím umělé inteligence a hlubokých neuronových sítí je strojový překlad v dnešní době velice přesný a neposkytuje pouze doslovný překlad textu, ale bere v potaz i kontext jednotlivých částí textu s ohledem na celek. Proto je dnes technologie MT hojně využívána právě v oblasti lokalizace. Samozřejmě i tato technologie má stále své nedostatky, a proto je potřeba i touto metodou přeložený text zkontrolovat. (Microsoft, 2025)

#### 4.2.1.1 Nástroje strojového překladu

Existuje nespočet různých nástrojů pro strojový překlad, zde jsou nejčastěji používané:

##### **Google Translator**

Také známý jako Google překladač - je to bezplatný nástroj pro strojový překlad, využívá umělou inteligenci k analýze a překladu textu. Aktuálně podporuje více než 100 jazyků. Jedná se o nejpoužívanější nástroj strojového překladu na světě.

##### **DeepL**

DeepL je vysoce kvalitní nástroj pro strojový překlad, který využívá neuronové sítě k dosažení vysoké kvality překladu. Je známý svou schopností zachovat kontext a produkovat přirozeně znějící překlady. Aktuálně podporuje více než 30 jazyků a je ceněn zejména pro svou přesnost v překladech mezi evropskými jazyky.

## Microsoft Translator

Tento nástroj využívá neuronové sítě. Nabízí překladové služby, které lze integrovat do aplikací a webů. Podporuje více než 100 jazyků a nabízí funkce jako překlad textu v reálném čase, překlad konverzací a překlad obrázků. Jeho použití můžeme vidět v mnoha produktech společnosti Microsoft.

(Schoening, 2025)

### 4.2.2 Systémy pro správu překladu

Systémy pro správu překladu (také označovány jako TMS) jsou software, který automatizuje opakující se nebo manuální úkoly, aby zjednodušil proces překladu a učinil jej co nejefektivnější. Tento systém je navržen pro pomoc firmám při globalizaci tím, že usnadňuje překlad a správu obsahu pro mezinárodní trhy.

TMS také podporuje proces lokalizace, což znamená přizpůsobení překladu kulturním a jazykovým specifikům cílového trhu. To zahrnuje nejen samotný překlad textu, ale také úpravy, jako jsou měnové symboly, formáty telefonních čísel nebo adres.

(Schoening, 2024)

#### 4.2.2.1 Funkce systému pro správu překladu

- **Automatizace pracovních postupů** – Snižuje riziko chyb a zrychluje proces překladu.
- **Správa týmu a projektů** – Umožňuje efektivní spolupráci mezi všemi zúčastněnými stranami, včetně překladatelů, vývojářů a specialistů na trh.
- **Správa překladových pamětí** – Ukládá již přeložený obsah, který lze znovu použít, což zvyšuje konzistenci.
- **Slovníky a styly** – Zajišťují, aby se používala správná terminologie a styl překladu.
- **Zabudované kontroly kvality** – Pomáhají odstraňovat chyby a zajišťovat přesnost překladů.
- **Integrace s ostatními systémy** – Umožňuje snadné propojení s content management systémy (CMS), e-shopy a dalšími platformami.

(Schoening, 2024)

#### 4.2.2.2 Příklady systémů pro správu překladu

V rámci celosvětového trhu existuje nespočet systémů pro správu překladu. Zde jsou relevantní příklady:

## **Lokalise**

Lokalise je pokročilý systém pro správu překladu, který automatizuje a řídí překladatelské procesy pro více než 700 jazykových kombinací. Využívá moderní technologie pro efektivní zvládnání obsahu na globálních trzích, nabízí integraci s populárními platformami, paměť překladů, automatizované kontroly kvality a podporuje spolupráci v reálném čase.

## **Smartling**

Smartling je moderní systém pro správu překladu, který zjednodušuje a automatizuje překladatelské procesy pro více než 150 jazyků. Využívá pokročilé technologie pro efektivní správu obsahu na globálních trzích, nabízí integraci s populárními platformami, paměť překladů, automatizované kontroly kvality a podporuje strojový překlad s možností post-editing.

## **Phrase**

Phrase je pokročilý systém pro správu překladu, který automatizuje a řídí překladatelské procesy pro více než 150 jazyků. Využívá moderní technologie pro efektivní správu obsahu na globálních trzích, nabízí integraci s populárními platformami, paměť překladů, automatizované kontroly kvality a podporuje vizuální editory s předzpracováním v reálném čase.

(K, 2025)

### **4.2.3 Lokalizační platformy**

Lokalizační platforma je komplexní nástroj, který pomáhá firmám přizpůsobit své produkty a služby kulturním a jazykovým specifikám cílových trhů. Zahrnuje překlad, ale také úpravy, které zohledňují místní zvyklosti, právní požadavky a uživatelské preference. Lokalizační platformy často nabízejí funkce pro správu obsahu, integraci s ostatními systémy a automatizaci pracovních postupů.

Přestože se může zdát, že TMS a Lokalizační platformy jsou to samé, není tomu tak. Lokalizační platformy se zaměřují na širší spektrum úprav než jen překlad, včetně kulturních adaptací a technických úprav, zatímco TMS se primárně soustředí na automatizaci a řízení překladatelských procesů, což je jen jedna část lokalizace.

(ALPHA, 2025)

#### 4.2.3.1 Funkce lokalizačních platforem

- **Komplexní přizpůsobení obsahu** – Zahrnuje nejen překlad, ale také kulturní a technické úpravy pro přizpůsobení produktu nebo služby cílovému trhu.
- **Správa lokalizačních projektů** – Umožňuje koordinaci mezi překladateli, lokalizačními specialisty, vývojáři a dalšími zúčastněnými stranami pro zajištění souladu s místními požadavky.
- **Podpora místních formátů** – Zahrnuje úpravy dat, měn, adres a dalších specifických prvků pro daný region.
- **Testování a kontrola kvality** – Nabízí nástroje pro testování lokalizovaného obsahu v reálném prostředí, včetně funkčního testování a kontroly uživatelského zážitku.
- **Integrace s více systémy** – Kromě CMS a e-shopů se lokalizační platformy často integrují s marketingovými automatizacemi a systémy pro správu produktů.
- **Analýza a optimalizace uživatelského zážitku** – Poskytuje nástroje pro analýzu chování uživatelů a optimalizaci lokalizovaného obsahu na základě získaných dat.
- **Podpora pro více typů obsahu** – Zahrnuje nejen text, ale také multimédia, software a další typy obsahu, které vyžadují lokalizaci.

(ALPHA, 2025)

#### 4.2.4 Nástroje pro testování lokalizace

Existuje několik nástrojů a technik pro testování lokalizace softwaru. Tyto nástroje pomáhají zajistit, aby testování lokalizovaného softwaru probíhalo hladce a konzistentně a zároveň automatizují některé aspekty testování. Tyto nástroje lze rozdělit do 3 kategorií:

1. *Nástroje pro správu testovacích případů*
2. *Nástroje pro správu a sledování problémů*
3. *Nástroje pro automatizaci testování lokalizace*

##### 4.2.4.1 Nástroje pro správu testovacích případů

Používají se pro udržení konzistence při testování. Slouží pro definování testovacích scénářů. To pomůže manažerům testování efektivně řídit a sledovat případy a postupně zlepšovat pokrytí testování. Zde je pár příkladů nástrojů pro správu testovacích případů:

## **TestRail**

TestRail je robustní nástroj pro správu testovacích případů, který vyniká integrací s různými systémy pro sledování chyb, jako je Redmine. Jeho silnou stránkou je možnost nasazení na vlastních serverech, což poskytuje větší kontrolu nad daty a bezpečností.

## **TestLodge**

TestLodge je moderní a uživatelsky přívětivý nástroj pro správu testovacích případů, který se vyznačuje snadným používáním a rychlým nastavením. Je dostupný pouze jako hostované řešení, což znamená, že nevyžaduje žádnou údržbu na straně uživatele.

### **4.2.4.2 Nástroje pro správu a sledování problémů**

Tyto nástroje slouží k udržení konzistence při sledování a řešení problémů během testování. Pomáhají definovat a sledovat pracovní postupy, což usnadňuje manažerům testování efektivně řídit a monitorovat stav jednotlivých chyb. Díky tomu lze postupně zlepšovat efektivitu a kvalitu testovacího procesu.

## **Remine**

Jedná se o flexibilní a mnohostranný nástroj pro sledování a správu problémů, který se vyznačuje širokou škálou možností přizpůsobení. Vyniká svou integrací se systémy pro správu testovacích případů (např. TestRail), což umožňuje plynulé sledování a řešení chyb v rámci testovacího procesu. Také umožňuje nasazení na vlastních serverech, což poskytuje týmům plnou kontrolu nad daty a bezpečností.

## **Lighthouse**

Lighthouse je jednoduchý a moderní nástroj pro sledování chyb. Nabízí jednoduché a intuitivní uživatelské rozhraní. Jako plně hostované řešení nevyžaduje žádnou instalaci ani údržbu, což ho činí ideálním pro menší týmy nebo projekty potřebující rychlé nasazení. Jeho jednoduchost pomáhá efektivně řídit a sledovat stav chyb během testování.

### **4.2.4.3 Nástroje pro automatizaci testování lokalizace**

Tyto nástroje slouží k zefektivnění testování lokalizace. I přestože většina procesů při testování lokalizace musí být provedena manuálně, některé procesy lze automatizovat. Tady pak přicházejí na scénu nástroje pro automatizaci testování lokalizace. Ty pomáhají definovat a provádět opakované úkoly. Automatizace umožňuje snížit časové nároky a zvýšit přesnost, čímž se postupně zlepšuje kvalita a efektivita celého testovacího procesu.

(SUMMA LINGUAE, 2022)

## 5 Lokalizace v praxi

### 5.1 Úvod do praktické ukázky

Je vhodné výše popsané principy ukázat na praktickém příkladu. Bohužel není možné ukázat všechny nástroje, metody a technologie v práci tohoto rozsahu. Proto jsem vybral k lokalizaci menší aplikaci vlastní tvorby, na které si ukážeme principy popsané výše. Tato praktická ukázka je vhodná spíše pro „menší“ vývojáře, nebo menší společnosti. Aplikace je vytvořena pro webovou platformu (ve dvou variantách – JS a ASP.Net) a desktopovou pro Windows OS (varianta .Net), aby měl čtenář možnost si prohlédnout proces napříč několika platformami (Samozřejmě platform je mnohem více, ale udělat ukázky pro všechny je bohužel mimo rozsah této práce.).

### 5.2 Výchozí stav a kontext

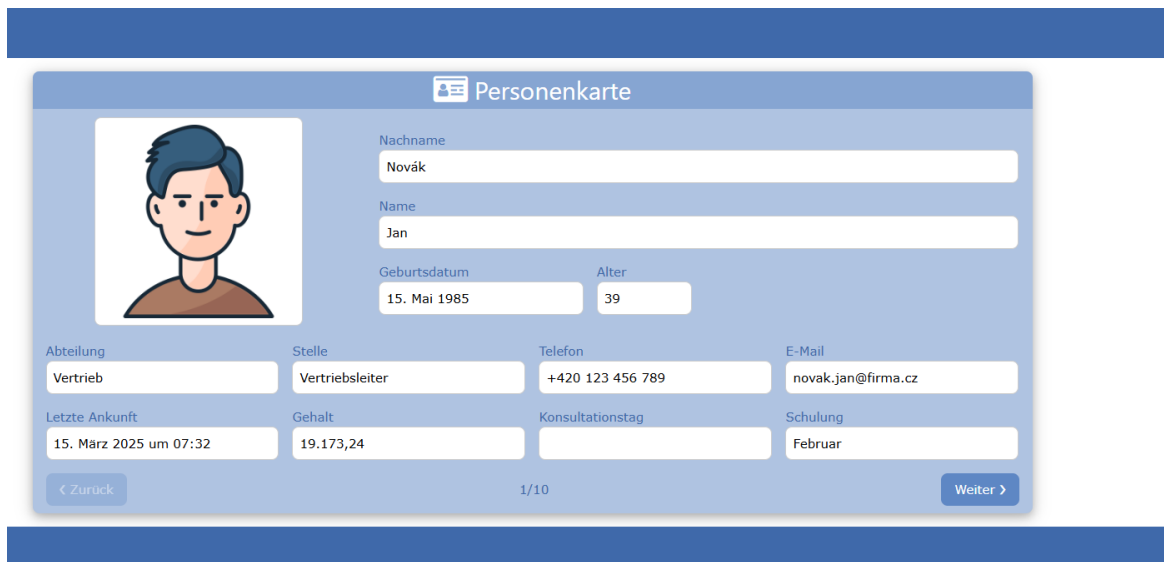
Pro lepší pochopení názorného příkladu nejdříve uvedu kontext. Menší německá fiktivní firma (dále označovaná pouze jako FF) se rozhodla expandovat směrem na východ. Pro tuto expanzi je potřeba otevřít několik nových poboček, mimo jiné i na území České republiky. Do budoucna pak firma plánuje další expanzi a tentokrát na ukrajinský trh.



Obrázek 9 - Expanze na nový trh

*Zdroj: Vlastní úprava map Google*

Pro zajištění hladkého chodu potřebuje FF, aby jejich stávající HR software byl použitelný i v nových pobočkách na českém trhu. Proto se vedení FF rozhodlo pro lokalizaci jejich dosavadní jednoduché HR aplikace v němčině.

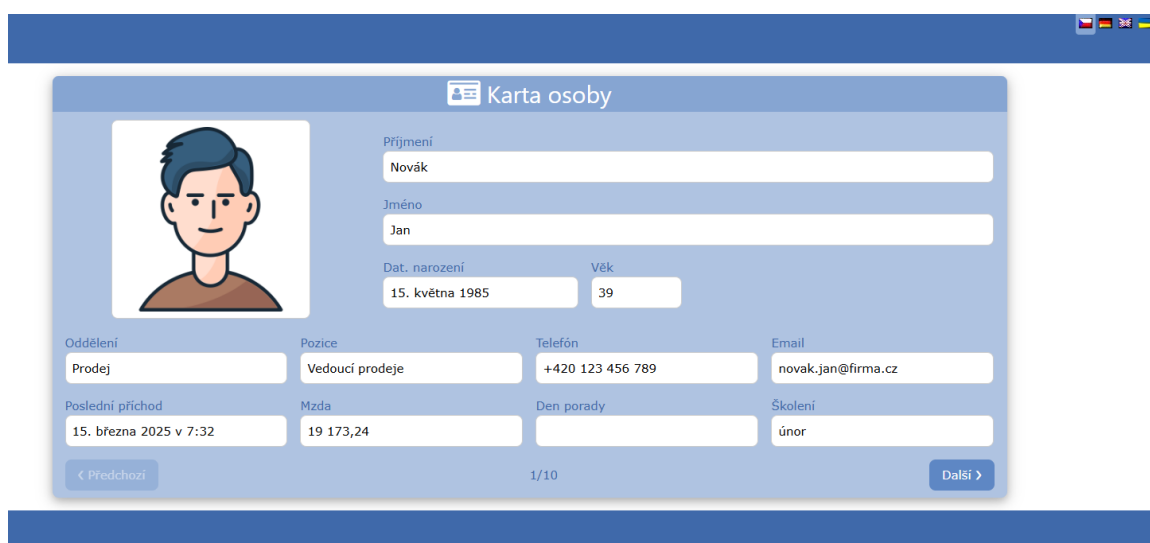


Obrázek 10 - Výchozí stav HR aplikace

Zdroj: Vlastní

### 5.3 Cílový stav

Jelikož v kontextu práce se tato část zaměřuje na technickou stranu lokalizace, tak vynechám výše zmíněné fáze jako je tvorba budgetu a analýza trhu a další. Cílový stav je plně lokalizovaný HR software. Software je lokalizovaný jak pro český, tak ukrajinský trh (usnadnění budoucí expanze). Zároveň je zde i možnost použití v anglické jazyce (region USA) pro případ budoucí spolupráce s jinou firmou. Aplikace je po lokalizaci funkční a otestovaná tak, aby veškerý její obsah dával smysl a veškeré prvky uživatelského rozhraní fungovali bez problémů.



Obrázek 11 - Cílový stav HR aplikace

Zdroj: Vlastní

## 5.4 Lokalizace aplikací

### 5.4.1 Převod textů do MS Excel

Už původní německá verze aplikace si brala textové řetězce a data z MS Excel (aplikaci jsem takto vytvářel, protože se mi v prostředí MS Excel s texty pracuje mnohem lépe než přímo ve zdrojovém kódu). V MS Excel jsem si vytvořil list s názvem „texty“, na tomto listu jsem si vytvořil tabulku, kde první sloupec a první řádek obsahují klíče (na jejich základě pak program vyhledává a doplňuje obsah, vysvětleno níže). V prvním řádku je jako klíč použita zkratka jazyka a v prvním sloupci jsou použity klíče, kterými se na jednotlivé texty budeme odkazovat.

Např. v JavaScriptu jsem u všech elementů, které se mění při lokalizaci, vytvořil atribut *locKey* a jeho hodnota je právě onen klíč.

```
<label ID="lblJmeno" locKey="1001">lblJmeno</label>
```

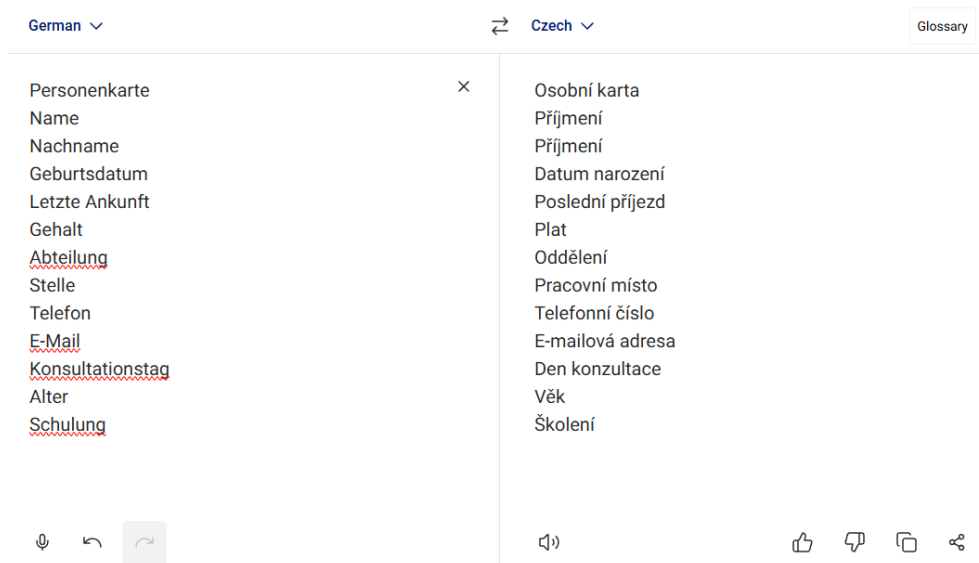
Obdobným způsobem jsou řešené tooltipové texty (text nápovědy po najetí myši), kde je za stejným účelem nastaven atribut *locTipKey*.

```
<input ID="txtPrijmeni" locTipKey="1002" Class="w3-input w3-border w3-round-large" readonly>
```

V .Net platformě pak tyto klíče používáme jako parametry pro získávání příslušných textů.

Výhodou použití Excelu pro práci s překládanými texty je, že takto při případné lokalizaci do dalšího jazyka stačí přidat další sloupec s novým jazykem.

Následně jsem do tabulky pouze přepíroval texty v němčině a samotný překlad textů jsem pak provedl pomocí nástroje strojového překladu DeepL. Proces překladu byl velice rychlý, protože jak MS Excel, tak DeepL umožňují velice jednoduché kopírování a vkládání textů.



Obrázek 12 - Překlad textu pomocí nástroje DeepL

Zdroj: Vlastní použití nástroje DeepL

Po překladu všech textů do požadovaných jazyků jsem jednotlivé sloupce ještě barevně oddělil, aby se mi v tabulce lépe orientovalo v případě, že bych potřeboval něco upravovat. Výsledkem byla tabulka na obrázku níže. (na obrázku je výřez části)

KEY	cs	de	en	uk
1000	Karta osoby	Personenkarte	Person card	Картка особи
1001	Jméno	Name	Name	Ім'я
1002	Příjmení	Nachname	Surname	Прізвище
1003	Dat. narození	Geburtsdatum	Date of birth	Дата народження
1004	Poslední příchod	Letzte Ankunft	Last Arrival	Останнє прибуття
1005	Mzda	Gehalt	Salary	Зарплата
1006	Oddělení	Abteilung	Department	Відділ
1007	Pozice	Stelle	Position	Посада
1008	Telefón	Telefon	Phone	Телефон
1009	Email	E-Mail	Email	Електронна пошта
1010	Den porady	Konsultationstag	Consultation day	День консультації
1011	Věk	Alter	Age	Вік
1012	Školení	Schulung	Training	Тренінг
1013				
1014				

Obrázek 13 - MS Excel texty

Zdroj: Vlastní

## 5.4.2 Převod textů z MS Excel do vhodného formátu

Bohužel ani JavaScript ani .Net nepracují nativně s daty přímo ze souboru ve formátu .xlsx. (formát dat v MS Excel). Data z excelu bylo proto potřeba vyexportovat do vhodného formátu. K tomu jsem použil již zabudovanou funkci MS Excel – Makra.

Vytvořil jsem si vlastní Makro „SaveResource2JSON“. Toto makro exportuje překlady z listu „Texty“ do JSON objektu a ukládá jej jako .js soubor. Formát JSON jsem zvolil, protože je podporován jak v JavaScriptu tak v .NET.

Makro nejdřív vytvoří .js soubor a do něj následně zapisuje data. V tabulce najde poslední řádek a sloupec obsahující data a tím si určí rozsah, který bude exportovat. Makro pak prochází každý řádek od druhého do posledního (první řádek se používá pro záhlaví). Z prvního sloupce se načte klíč (název překladu). Pro každý další sloupec se načte jazyková verze textu.

Tyto hodnoty se zapíšou do souboru ve formátu JSON. Výsledná struktura v souboru .js obsahuje objekt translations, který má klíče (např. "1001") a hodnoty, které jsou dalšími objekty. Tyto vnořené objekty obsahují překlady pro různé jazyky ("cs", "de", "en", "uk"). Zde je pro ukázkou část tohoto .js Souboru:

```
"1001": {
  "cs": "Jméno",
  "de": "Name",
  "en": "Name",
  "uk": "Ім'я"
},
```

V tomto kroku byl postup stejný pro všechny platformy, protože jak jsem uváděl výše, tak všechny platformy podporují výsledný formát. V dalším kroku (tj. nahrávání textových řetězců zpět do softwaru) se již bude postup pro jednotlivé platformy lišit.

### 5.4.3 Načtení textových řetězců ve webové aplikaci HTML/JS

Data ve formátu uvedeném výše (tedy JSON objekt/konstanta v .js souboru) jsem pak potřeboval nahrát zpět do softwaru tak, aby se zobrazovala správná data (podle klíče) a ve správném jazyce (podle nastaveného jazyka).

Nejdřív je třeba naimportovat nový soubor texty.js pomocí HTML elementu `<script src="texty.js"></script>`. Tím se objekt translations stane součástí globálního prostředí JavaScriptu na stránce a díky tomu ho můžeme dál používat.

Pro získání konkrétních textových řetězců pro jednotlivé elementy v HTML pak slouží funkce `getText`. Ta má jako parametr klíč, který má element v atributu `locKey`.

```
<label ID="lblPrijmeni" locKey="1002">lblPrijmeni</label>
```

Následně vrací odpovídající textový řetězec pro daný klíč a aktuálně nastavený jazyk html dokumentu. Pokud překlad pro daný klíč a jazyk neexistuje, vrátí „???“.

```
//Vrátí text pro daný klíč a jazyk
function getText(key) {
```

```

// Nacteme akt. jazyk
var lng = document.documentElement.lang;

// Pokud existuje hodnota pro daný klíč a jazyk vrať
// tuto hodnotu, jinak "???"
if (translations[key] && translations[key][lng]) {
    return translations[key][lng];
}
else {
    return "???" ;
}
}

```

#### 5.4.4 Přepínání jazyků a zobrazení lokalizovaných dat ve HTML/JS

Aby bylo možné pomocí výše uvedené funkce *getText* zobrazovat v aplikaci texty v požadovaném jazyce, tak bylo třeba umožnit uživateli přepínat mezi jednotlivými jazyky. Proto jsem přidal do pravého horního rohu UI (uživatelské rozhraní) div (prvek html) obsahující tlačítka ve formě vlajček.



Obrázek 14 - Tlačítka pro změnu jazyka, Zdroj Vlastní

Každému tlačítku jsem pak přiřadil jazyk, atribut *lang*. Kliknutí na tlačítko pak volá funkci *changeLanguage*. Funkce má dva parametry a to *btn*, což je tlačítko, na které jsme kliknuli, a *lang*, což je jazyk, který má dané tlačítko přiřazeno. Uvnitř funkce pak nastaví atribut *lang* kořenového elementu (<html>) na nový jazyk. Tím se změní jazyk celého dokumentu.

```

//Přepne jazyk, změni lokalizaci
function changeLanguage(btn, lang) {
    // Nastavení jazyka HTML dokumentu na jazyk přiřazený tlačítku
    document.documentElement.lang = lang;
    . . .
}

```

##### 5.4.4.1 Texty

Dále je třeba, aby byla po přepnutí jazyka znovu volána funkce *getText* (vysvětlena výše) pro všechny elementy, které se lokalizují (tzn. obsahují atribut *locKey*, nebo *locTipKey*) K tomu se v rámci funkce *changeLanguage* provádí další operace.

První operace slouží k aktualizaci textů. Vybere všechny HTML prvky v dokumentu, které mají atribut *locKey*. Následně pro všechny tyto elementy získá hodnotu atributu *locKey*.

Nakonec pak pro všechny tyto elementy volá funkci *getText*, která vrátí textový řetězec pro každý element na základě hodnoty atributu *locKey* a nově zvoleného jazyka.

```
// Aktualizace textů
document.querySelectorAll('[locKey]').forEach(element => {
  const key = element.getAttribute('locKey');
  element.textContent = getText(key);
});
```

Druhá operace pak slouží k aktualizaci tooltip textů. První část je této operace je stejná jako v operaci předchozí pouze místo elementů s atributem *locKey* hledá elementy s atributem *locTipKey*.

Druhá část této operace se pak liší proto, že u tlačítek pro změnu jazyka chci, aby se tooltip text u každého tlačítka ukazoval v jazyce, který tlačítko reprezentuje. Zjišťuje, zda nemají elementy nadefinovaný atribut *locLng* (Tento atribut mají nadefinovaný pouze tlačítka pro změnu jazyka. Jeho hodnota je pak jazyk, který dané tlačítko reprezentuje). Pokud element tento atribut má, tak se pro něj hledá titulek pro daný klíč tohoto elementu a jazyk, který má tento element nadefinován v atributu *locLng*. V případě že element atribut *locLng* nemá, tak hledá titulek pro daný klíč a jazyk nastavený pro celý html dokument.

```
// Aktualizace textů tooltipu
document.querySelectorAll('[locTipKey]').forEach(element => {
  const key = element.getAttribute('locTipKey');

  // Nastavení jazyka na hodnotu locLng pokud ji element má
  var lng;
  if (element.hasAttribute('locLng')) {
    lng = element.getAttribute('locLng');
  } else {
    lng = lang;
  }

  // nastavení hodnoty titulku podle klíče
  // a jazyka daného elementu
  if (translations[key] && translations[key][lng]) {
    element.title = translations[key][lng];
  }
});
```

#### 5.4.4.2 Data

Poslední operace funkce *changeLanguage* je volání funkce *displayOsoba* s parametry *lang* (aktuálně zvolený jazyk html dokumentu) a *currentIndex* (index aktuálně zobrazované osoby). Tato funkce má na starosti zobrazení dat pro zobrazovanou osobu naformátované pro aktuálně zvolený jazyk.

Data osob jsou stejně jako textové řetězce vyexportovány z MS excelu do souboru osoby.js ve formátu JSON.

```
{
  "Jmeno": "Jan",
  "Prijmeni": "Novák",
  "DatumNarozeni": "1985-05-15",
  "PoslPrichod": "2025-03-15T07:32:15",
  "Mzda": 19173.24,
  "Fotka": "m1",
  "KodOddeleni": "01",
  "KodPozice": "P1",
  "Telefon": "+420 123 456 789",
  "Email": "novak.jan@firma.cz",
  "DenPorady": -1,
  "Skoleni": 1
},
```

Z tohoto souboru si funkce *displayOsoba* načítá data osoby z pole osob na základě indexu. Tyto data je třeba zformátovat pro daný jazyk/region. K tomu v jazyce JavaScript slouží třída **Intl**. Ta poskytuje funkce, které umožňují formátovat data, čísla a měny dle regionálních a jazykových zvyklostí. Hlavní funkce pro třídu *Intl* jsou:

- **Intl.DateTimeFormat** – formátování datumů a času
- **Intl.NumberFormat** – formátování číselných hodnot a měn

Tyto funkce mají různé kombinace parametrů, které ovlivňují výsledný formát. Základní parametr je *locale* – určuje jazyk a region, pro který se má použít formátování. Například „cs-CZ“ pro češtinu v České republice.

Pro zjednodušení a zpřehlednění volání funkcí jsem si pro konfiguraci parametrů pro jednotlivé jazyky vytvořil struktury určující parametry pro výsledný formát.

```
'cs': {
  locale: 'cs-CZ',
  currency: 'CZK',
  dateFormat: { dateStyle: 'long' },
  dateTimeFormat: { dateStyle: 'long', timeStyle: 'short' },
  weekdayFormat: { weekday: 'long' },
  monthFormat: { month: 'long' },
  decimalFormat: { minimumFractionDigits: 2, maximumFractionDigits: 2 }
},
```

Ze seznamu formátovacích parametrů pro jednotlivé jazyky si vybereme tu, která odpovídá aktuálně zvolenému jazyku html dokumentu.

```
const format = localeFormats[lang];
```

Textové údaje osoby (jméno, příjmení atd.) zobrazujeme přímo (tzn. bez formátování)

```
document.getElementById('txtJmeno').value = osoba.Jmeno;
```

Datumové a časové položky (datum narození, čas příchodu atd.) se formátují pomocí výše zmíněné funkce *Intl.DateTimeFormat*.

```
document.getElementById('txtDNarozeni').value = new
Intl.DateTimeFormat(format.locale, format.dateFormat).format(new
Date(osoba.DatumNarozeni));
```

Číselné hodnoty jsou pak formátovány pomocí funkce *Intl.NumberFormat*.

```
document.getElementById('txtMzda').value = new
Intl.NumberFormat(format.locale,
format.decimalFormat).format(parseFloat(osoba.Mzda));
```

Jako parametry používáme strukturu formátovacích parametrů daného jazyka.

V rámci zobrazení dat jsem si vytvořil i další pomocné funkce:

- *calculateAge* – vrací aktuální věk osoby
- *getDayName* – vrací jméno dne v týdnu v daném jazyce
- *getMonthName* – vrací jméno měsíce v daném jazyce

```
function getMonthName(format, MonthIndex) {
    if ((MonthIndex >= 1) && (MonthIndex <= 12)) {
        const date = new Date(2000, MonthIndex, 1);

        // Získání názvu dne

        return Intl.DateTimeFormat(format.locale,
format.monthFormat.format(date);
    }
    else {
        return "";
    }
}
```

## 5.4.5 Načtení textových řetězců – .Net platforma

Pro platformu .NET jsem vytvořil dvě varianty aplikace – jednu pro desktopové prostředí a druhou pro web v ASP.NET. Abych si práci usnadnil rozhodl jsem se veškeré funkcionality související s lokalizací zabalit do jedné třídy *CNation* v rámci knihovny, kterou budou využívat oba projekty na platformě .NET.

Stejně jako v případě JavaScriptu (viz. výše) bude třída *CNation* načítat textové řetězce ze souboru ve formátu .js obsahujícím data formátu JSON. Třída *CNation* obsahuje pro každý jazyk „dictionary“ - seznam dvojic klíč a hodnota, kde klíč odpovídá klíči daného textu a hodnotou je textový řetězec v daném jazyce.

```
Public Class CNation
    Private _culture As CultureInfo
    Private _Jazyk As tJazyky
    Private CZ As New Dictionary(Of String, String)
    Private EN As New Dictionary(Of String, String)
```

```

Private DE As New Dictionary(Of String, String)
Private UK As New Dictionary(Of String, String)

. . .

End Class

```

Pro získání konkrétních textových řetězců pro jednotlivé prvky aplikace pak slouží `CNation.Text` (obdoba `getText` v JS) s parametrem klíč.

```

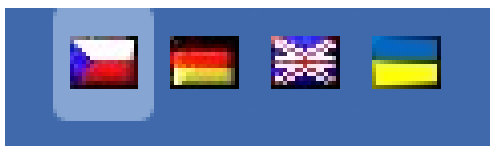
Public Function Text(ByVal pKod As String) As String
    Dim pom As String = ""
    Try
        'podle aktualniho jazyka vrat dle klice text
        Select Case Jazyk
            Case tJazyky.CZ
                pom = CZ(pKod)
            Case tJazyky.EN
                pom = EN(pKod)
            Case tJazyky.DE
                pom = DE(pKod)
            Case tJazyky.UK
                pom = UK(pKod)
        End Select
    Catch ex As Exception
        pom = "???"
    End Try
    Return pom
End Function

```

Pokud překlad pro daný klíč a jazyk neexistuje, vrátí „???“.

## 5.4.6 Přepínání jazyků a zobrazení lokalizovaných dat – .Net platforma

Aby bylo možné pomocí výše uvedené funkce `Text` zobrazovat v aplikaci texty v požadovaném jazyce, tak bylo třeba umožnit uživateli přepínat mezi jednotlivými jazyky. Proto jsem přidal do pravého horního rohu UI (uživatelské rozhraní) tlačítka ve formě vlaječek.



Obrázek 15 - Tlačítka pro změnu jazyka 2

*Zdroj: Vlastní*

Každému tlačítku jsem pak nastavil atribut *jazyk*, resp. *Lang*. Kliknutí na tlačítko pak volá funkci *cmdLang\_Click*. Událost v prvním obsahuje odkaz na tlačítko, které ji vyvolalo. Uvnitř události je pak podle atributu *Lang* nastaven jazyk instance třídy *CNation* na jazyk zmáčknutého tlačítka. Následně se volají funkce pro zobrazení textů a dat osoby v nově zvoleném jazyce.

```
Public Sub cmdLang_Click(sender As Object, e As EventArgs)
    'zmenime def. jazyk v tride
    Select Case CType(sender, ImageButon).Attributes("Lang")
        Case "CZ"
            nation.Jazyk = CNation.tJazyky.CZ
        Case "DE"
            nation.Jazyk = CNation.tJazyky.DE
        Case "EN"
            nation.Jazyk = CNation.tJazyky.EN
        Case "UK"
            nation.Jazyk = CNation.tJazyky.UK
    End Select

    SetCssClass(cmdLangCZ, "w3-theme-l2", cmdLangCZ Is sender)
    SetCssClass(cmdLangDE, "w3-theme-l2", cmdLangDE Is sender)
    SetCssClass(cmdLangEN, "w3-theme-l2", cmdLangEN Is sender)
    SetCssClass(cmdLangUK, "w3-theme-l2", cmdLangUK Is sender)

    Call Texty()
    Call FillData()
End Sub
```

#### 5.4.6.1 Texty

Funkce *Texty* pak zajišťuje zobrazení textů v UI dle zvoleného jazyka a klíče odpovídajícího danému prvku voláním metody *Text* třídy *CNation* zmíněné výše. Stejným způsobem jsou pak nastaveny tooltipové texty jednotlivých prvků UI.

```
Private Sub Texty()
    lblNadpis.Text = nation.Text("1000")
    lblJmeno.Text = nation.Text("1001")
    . . .
End Sub
```

#### 5.4.6.2 Data

Poslední operace události *cmdLang\_Click* je volání funkce *FillData*. Tato funkce má na starosti zobrazení dat pro zobrazovanou osobu naformátované pro aktuálně zvolený jazyk.

Data osob jsou stejně jako textové řetězce vyexportovány z MS excelu do souboru *osoby.js* ve formátu JSON a načteny do seznamu osob – třída *COsoba*

```

Public Class COsoba
    Public Property Jmeno As String
    Public Property Prijmeni As String
    Public Property DatumNarozeni As Date
    Public Property PoslPrichod As Date
    Public Property Mzda As Double
    Public Property Fotka As String
    Public Property KodOddeleni As String
    Public Property KodPozice As String
    Public Property Telefon As String
    Public Property Email As String
    Public Property DenPorady As Long
    Public Property Skoleni As Long
End Class

```

Z tohoto seznamu si funkce *FillData* načítá data osoby na základě indexu. Tyto data je třeba zformátovat pro daný jazyk/region. K tomu v .NET slouží třída *CultureInfo*. Třída *CultureInfo* v .NET poskytuje informace o specifických regionálních zvyklostech pro formátování dat, čísel a měn, podobně jako *Intl* v JavaScriptu.

Při vytváření instance *CultureInfo* pro specifický jazyk a region je vždy třeba definovat parametr jazyk/region. Například „cs-CZ“ pro češtinu v České republice. V třídě *CNation* je nastavení regionu pro *CultureInfo* realizováno ve vlastnosti *Jazyk* při změně jazyka. Instanci *CultureInfo* pro aktuální jazyk má třída v privátní proměnné *\_culture*.

```

Public Property Jazyk As tJazyky
    Get
        Return _Jazyk
    End Get
    Set(value As tJazyky)
        _Jazyk = value
        Select Case value
            Case tJazyky.CZ
                _culture = New CultureInfo("cs-CZ")
            Case tJazyky.EN
                _culture = New CultureInfo("en-US")
            Case tJazyky.DE
                _culture = New CultureInfo("de-DE")
            Case tJazyky.UK
                _culture = New CultureInfo("uk-UA")
        End Select
    End Set
End Property

```

Na rozdíl od *Intl* má ale již přednastavené formáty, ze kterých si může uživatel zvolit pomocí zadání parametru. Tím odpadá krok s definicí parametrů formátů, protože již obdobné struktury obsahuje. *CultureInfo* ale neobsahuje formátovací funkce jako *Intl*, používá se jako parametr u metody *ToString()* jednotlivých datových typů.

- `Date.ToString(<formátovací řetězec>, <CultureInfo>)`
- `Int.ToString(<formátovací řetězec>, <CultureInfo>)`
- ...

Tabulka 1 - Příklad formátovacích řetězců .NET

Typ dat	Formátovací řetězec	Příklad výstupu (cs-CZ)	Příklad výstupu (en-US)
Čísla	"N" (standardní)	1 234,56	1,234.56
	"E" (vědecký)	1,23E+03	1.234560E+003
Data	"d" (krátký)	05.10.2023	10.05.2023
	"F" (dlouhý)	Úterý, 5. října 2023	Thursday, October 5, 2023
Měna	"C" (měna)	1 234,56 Kč	\$1,234.56

Pro zjednodušení a zpřehlednění volání funkcí jsem si pro formátování dat v třídě *CNation* vytvořil sadu formátovacích funkcí. Například pro formátování čísel:

```
Public Function FormatNumber(value As Double, Optional decimals As Integer = 2) As String
    Return value.ToString($"N{decimals}", _culture)
End Function
```

Funkce *FillData* tedy naformátuje pomocí těchto funkcí data *Osoby*. Textové údaje osoby (jméno, příjmení atd.) zobrazujeme přímo (tzn. bez formátování)

```
txtJmeno.Text = Osoba.Jmeno
txtPrijmeni.Text = Osoba.Prijmeni
```

Datové a časové položky (datum narození, čas příchodu atd.) se formátují pomocí funkce *CNation.FormatShortDate*

```
txtDNarozeni.Text = nation.FormatShortDate(Osoba.DatumNarozeni)
```

Číselné hodnoty jsou pak formátovány pomocí funkce *CNation.FormatNumber*

```
txtMzda.Text = nation.FormatNumber(Osoba.Mzda, 2)
```

Pro zobrazení dat jsem vytvořil i další pomocné funkce v třídě *CNation*:

- *GetDayName* – vrací jméno dne v týdnu v daném jazyce
- *GetMonthName* – vrací jméno měsíce v daném jazyce

```
Public Function GetDayName(dayOfWeek As DayOfWeek, Optional abbreviated As Boolean = False) As String
    If abbreviated Then
        Return _culture.DateTimeFormat.GetAbbreviatedDayName(dayOfWeek)
    Else
        Return _culture.DateTimeFormat.GetDayName(dayOfWeek)
    End If
```

## 5.5 Závěr praktické ukázky

V ukázce jsem předvedl lokalizaci softwaru do (nejen) českého prostředí v praxi na aplikaci pro HR fiktivní firmy. Aplikace byla lokalizována jak na platformě JavaScriptu v kombinaci s HTML, tak na platformě .NET ve webové i desktopové verzi. Jak jsem již zmiňoval v úvodu, postup lokalizace a platformy zvolené v ukázce jsou vhodné pro menší projekty (například aplikace do prostředí malých firem). Pro lokalizaci větších projektů jsou potom vhodné metody uvedené v kapitole 5 (lokalizační platformy atd.). Na obrázcích níže pak můžete vidět výsledek lokalizace.

The screenshot shows a desktop application window titled "Karta osoby" with a dark blue header and a light blue main area. The window contains a user profile card with a cartoon avatar of a man. The form fields are as follows:


Karta osoby	
Příjmení	Jan
Jméno	Novák
Dat. narození	15.05.1985
Věk	40
Oddělení	Prodej
Pozice	Vedoucí prodeje
Telefón	+420 123 456 789
Email	novak.jan@firma.cz
Poslední příchod	sobota 15. března 2025 7:32:15
Mzda	19 173,24
Den porady	
Školení	leden

At the bottom of the card, there are navigation buttons: "< Předchozí", "1/10", and "Další >".

Obrázek 16 - Výsledek lokalizace – .NET desktopová verze

Zdroj: Vlastní

Karta osoby



Příjmení: Novák

Jméno: Jan

Dat. narození: 15.05.1985      Věk: 40

Oddělení: Prodej      Pozice: Vedoucí prodeje      Telefón: +420 123 456 789      Email: novak.jan@firma.cz

Poslední příchod: sobota 15. března 2025 7:32:1      Mzda: 19 173,24      Den porady:      Školení: leden

< Předchozí      1/10      Další >

Obrázek 17 - Výsledek lokalizace – ASP.NET webová verze

Zdroj: Vlastní

Karta osoby



Příjmení: Novák

Jméno: Jan

Dat. narození: 15. května 1985      Věk: 39

Oddělení: Prodej      Pozice: Vedoucí prodeje      Telefón: +420 123 456 789      Email: novak.jan@firma.cz

Poslední příchod: 15. března 2025 v 7:32      Mzda: 19 173,24      Den porady:      Školení: únor

< Předchozí      1/10      Další >

Obrázek 18 - Výsledek lokalizace – JavaScript a HTML

Zdroj: Vlastní

## **Závěr**

Lokalizace softwaru je v dnešním globalizovaném světě nepostradatelnou součástí vývoje softwaru. Tato práce poskytuje základní přehled o lokalizaci softwaru a nastiňuje její klíčové aspekty.

Práce definuje lokalizaci a zasazuje ji do kontextu globalizace a českého prostředí. Zdůrazňuje, že lokalizace není pouze o překladu textu, ale zahrnuje i přizpůsobení softwaru kulturním a technickým požadavkům cílového trhu.

V práci jsou popsány různé metody, technologie a nástroje, které se při lokalizaci softwaru používají. Dále jsou na praktických ukázkách předvedeny postupy lokalizace softwaru pro menší webové i desktopové projekty.

Cílem této práce bylo poskytnout ucelený úvod do problematiky lokalizace softwaru. A na lokalizaci do českého prostředí si ukázat základy. Práce může sloužit jako vodítko pro začínající softwarové vývojáře, kteří chtějí své produkty zpřístupnit mezinárodnímu publiku, nebo chtějí upravit nějaký open source program tak, aby byl vhodný pro české prostředí.

## Použitá literatura

- ALPHA. 2025.** ALPHA: Localization platforms in 2025: everything you need to know. *ALPHA*. [Online] ALPHA, 12. březen 2025. [Citace: 25. březen 2025.] <https://alphacrc.com/localization/localization-platform-guide/>.
- Coulmas, Florian. 1999.** *The Blackwell Encyclopedia of Writing Systems*. místo neznámé : Wiley-Blackwell, 1999. 978-0631214816.
- Esselink, Bert. 2000.** *A Practical Guide to Localization*. s.l. : John Benjamins Publishing, 2000. 9789027298188, 9027298181.
- Hashemi-Pour, Cameron. 2024.** user interface (UI). *TechTarget*. [Online] 2024. <https://www.techtarget.com/searcharchitecture/definition/user-interface-UI>.
- Hykyšová, Alena. 2024.** Anglicky umí polovina Čechů - STATISTIKA&MY. *STATISTIKA&MY časopis českého statistického úřadu*. [Online] 17. Duben 2024. <https://statistikaamy.csu.gov.cz/anglicky-umi-polovina-cechu>.
- International Organization for Standardization. ISO IEC60204-1. ISO.** [Online] <https://www.iso.org/obp/ui#iec:grs:60417:5104>.
- K, Karol. 2025.** TranslatePress: 4 of the Best Translation Management Software in 2025. *TranslatePress*. [Online] TranslatePress, 16. Leden 2025. [Citace: 25. březen 2025.] <https://translatepress.com/best-translation-management-software/>.
- Kenner, T. A. 2006.** *Symbols and their hidden meanings*. místo neznámé : Carlton Books Ltd, 2006. 978-1844422937.
- Kent State University, Ohio. 2018.** [www.kent.edu/mcls/translation-ma/blog/software-localization](http://www.kent.edu/mcls/translation-ma/blog/software-localization). *www.kent.edu*. [Online] 2018. <https://www.kent.edu/mcls/translation-ma/blog/software-localization>.
- Labohá, Martina. 2017.** 13 běžně užívaných gest, na která si ve světě dávat pozor, Prima ZOOM. *Prima ZOOM*. [Online] 1. Prosinec 2017. <https://zoom.iprima.cz/zpravodajstvi/13-bezne-uzivanych-gest-na-ktera-si-ve-svete-davat-pozor>.
- Microsoft. 2025.** Microsoft:Strojový překlad. *Microsoft Translator*. [Online] Microsoft, 2025. [Citace: 24. Březen 2025.] <https://www.microsoft.com/cs-cz/translator/business/machine-translation/>.

**Richard Ishida, W3C, Susan K. Miller, Boeing. 2005.** Localization vs. Internationalization  
www.w3.org. *www.w3.org*. [Online] 12. Květen 2005.

<https://www.w3.org/International/questions/qa-i18n>.

**Schoening, Stephan. 2025.** Phrase: Machine Translation Software: The Best Tools to Try  
(and Use) in 2025. *Phrase*. [Online] Phrase, 20. Březen 2025. [Citace: 25. březem 2025.]

<https://phrase.com/blog/posts/machine-translation-tools/>.

—. **2024.** Phrase: What Makes a Translation Management System, and How to Choose the  
Right One for Your Needs. *Phrase Translation Management*. [Online] Phrase, 19. Únor 2024.

[Citace: 25. Březen 2025.] <https://phrase.com/blog/posts/translation-management-system-how-it-works/#what-is-a-translation-management-system>.

**SUMMA LINGVAE. 2022.** SUMMA LINGVAE - Top Localization Tools for Engineering  
& Quality Assurance. *SUMMA LINGVAE*. [Online] SUMMA LINGVAE, 25. březem 2022.

[Citace: 31. březem 2025.] <https://summalinguae.com/localization/localization-testing-tools/>.