



ÇAĞ ÜNİVERSİTESİ
ÇAĞ UNIVERSITY

29.
yıl

TRN 414 – TRANSLATION AND LOCALIZATION



Asst. Prof. Dr. Mustafa Kartal ÇELİKAY



The Sociological and Historical Context of Globalization (Post-1991)

The Paradigm Shift: The end of the Cold War and the dissolution of the Soviet Union in 1991 marked the collapse of the bipolar world order. The political and economic barriers that heavily divided the globe were dismantled, paving the way for a single, interconnected global market.

The Technological Revolution: The early 1990s coincided with the commercialization and widespread public use of the World Wide Web. Information, which was previously restricted by national borders, became universally accessible almost instantly.

Time-Space Compression: Sociologically, the world experienced what David Harvey termed "time-space compression." Rapid advancements in telecommunications and digital infrastructure effectively "shrunk" the globe, making real-time cross-cultural interaction the new norm.

Economic Integration and Expansion: Multinational corporations began expanding aggressively beyond their domestic borders. Capital, services, and products started flowing freely across continents to reach newly opened emerging markets.

The Intersection of Globalization and Translation

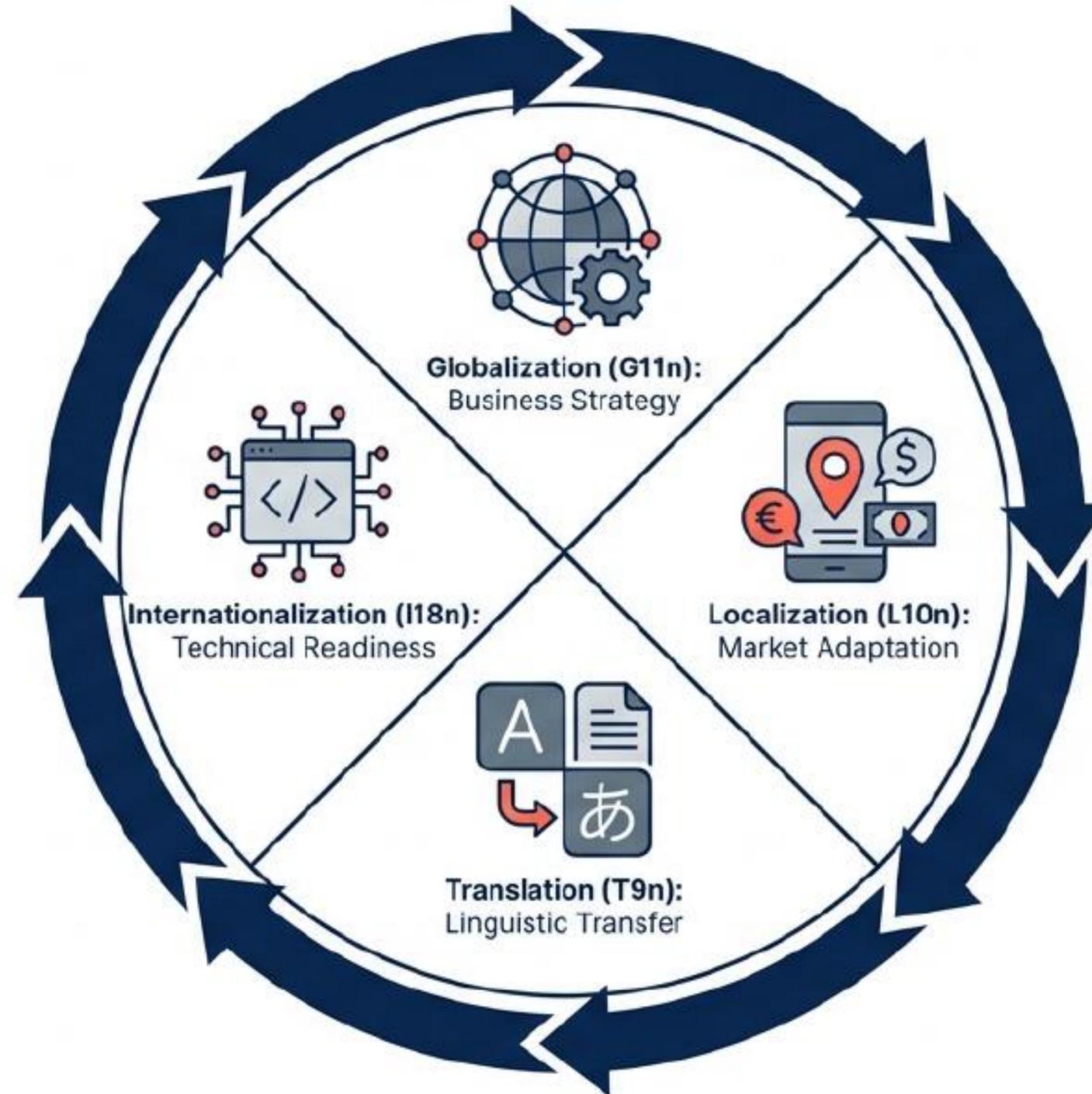
The Multilingual Barrier: As global economic integration accelerated, multinational entities encountered a fundamental obstacle: Language and Culture. A product designed in Silicon Valley could not simply be exported to Tokyo or Istanbul without addressing the local linguistic reality.

The Commercialization of Translation: Translation underwent a massive paradigm shift. It rapidly evolved from being primarily a literary, diplomatic, or academic endeavor into a core, technology-driven commercial industry. The "source texts" were no longer just books or treaties; they became software interfaces, websites, user manuals, and digital campaigns.

The Birth of the GILT Framework: This absolute commercial necessity gave birth to the GILT framework. In this context, **Globalization (G11n)** is defined as the overarching macro-level business strategy; the corporate decision to design, market, and distribute a product worldwide.

Translation as the Engine of Global Economy: In this post-1991 globalized era, translation ceased to be **an afterthought**. It became an integral component of the global economic machinery. Without translation and localization, globalization would physically halt, as users fundamentally cannot adopt or purchase digital products they do not understand.

The GILT Framework: A Cycle of Global Development



Key Insight

Critical:
Internationalization (I18n) must occur before the Localization (L10n) process begins.

The Sociological and Structural Context of Internationalization (I18n)

Moving Beyond Ethnocentrism: In the early days of software and the internet, digital systems were inherently ethnocentric. They were built primarily for an English-speaking, Western audience using restricted frameworks (e.g., *ASCII-American Standard Code for Information Interchange* character encoding, rigid left-to-right alignments, and US-specific date, time, and currency formats).

Digital Democratization: Sociologically, internationalization represents the democratization of the digital environment. It is the structural acknowledgment that the global village consists of diverse cultures with fundamentally different linguistic and visual architectures.

Creating a Culturally Neutral Infrastructure: Before a digital product can effectively speak to a specific local culture, it must first be stripped of its native biases. I18n is the engineering philosophy of generalizing a product so that it can handle multiple cultural conventions **without requiring a complete redesign of the core source code.**

The Intersection of Internationalization and Translation

The Prerequisite for Translation: While Translation (T9n) deals with the transfer of meaning and culture, Internationalization (I18n) builds the technical stage where this transfer can physically exist. **If a system is not internationalized, a translator's work simply cannot be integrated into the product.**

String Externalization: Translators are not software developers. Therefore, all translatable text must be separated from the software's source code. I18n extracts these texts into **external resource files (strings)**, ensuring that translators only interact with the language data and do not accidentally alter the **programming logic.**

Text Expansion and UI Flexibility: A translated text often changes in physical length. For instance, translating a simple button from English to German can result in significant text expansion. An internationalized User Interface (UI) is designed to be elastic, preventing the translated text from overflowing, truncating, or breaking the visual layout.

Bidirectional Support and Encoding: I18n ensures the foundational architecture supports right-to-left (RTL) reading patterns required by languages like Arabic or Hebrew. Furthermore, it mandates universal encoding standards like Unicode (UTF-8) so that non-Latin alphabets and special characters (like Turkish 'ş', 'ğ' or Japanese Kanji) are displayed flawlessly instead of **appearing as corrupted symbols.**

The Fundamental Difference Between Globalization (G11n) and Internationalization (I18n)

The fundamental difference between Globalization (G11n) and Internationalization (I18n) in a business or project process is the distinction between **"business strategy"** and **"technical infrastructure."**

Globalization (G11n): The Decision and Strategy Phase

It is the **commercial, managerial, and macro** dimension of the business. The decision-makers are typically company management, marketing, finance, and legal departments.

It is the strategic vision where questions like "Should we introduce our product to overseas markets (e.g., Arab and European markets)? What is the market potential? What are the legal procedures?" are answered.

In other words, the process involves the company deciding to cross borders and establishing the corporate structure to **manage this operation.**

Internationalization (I18n): The Engineering and Design Phase

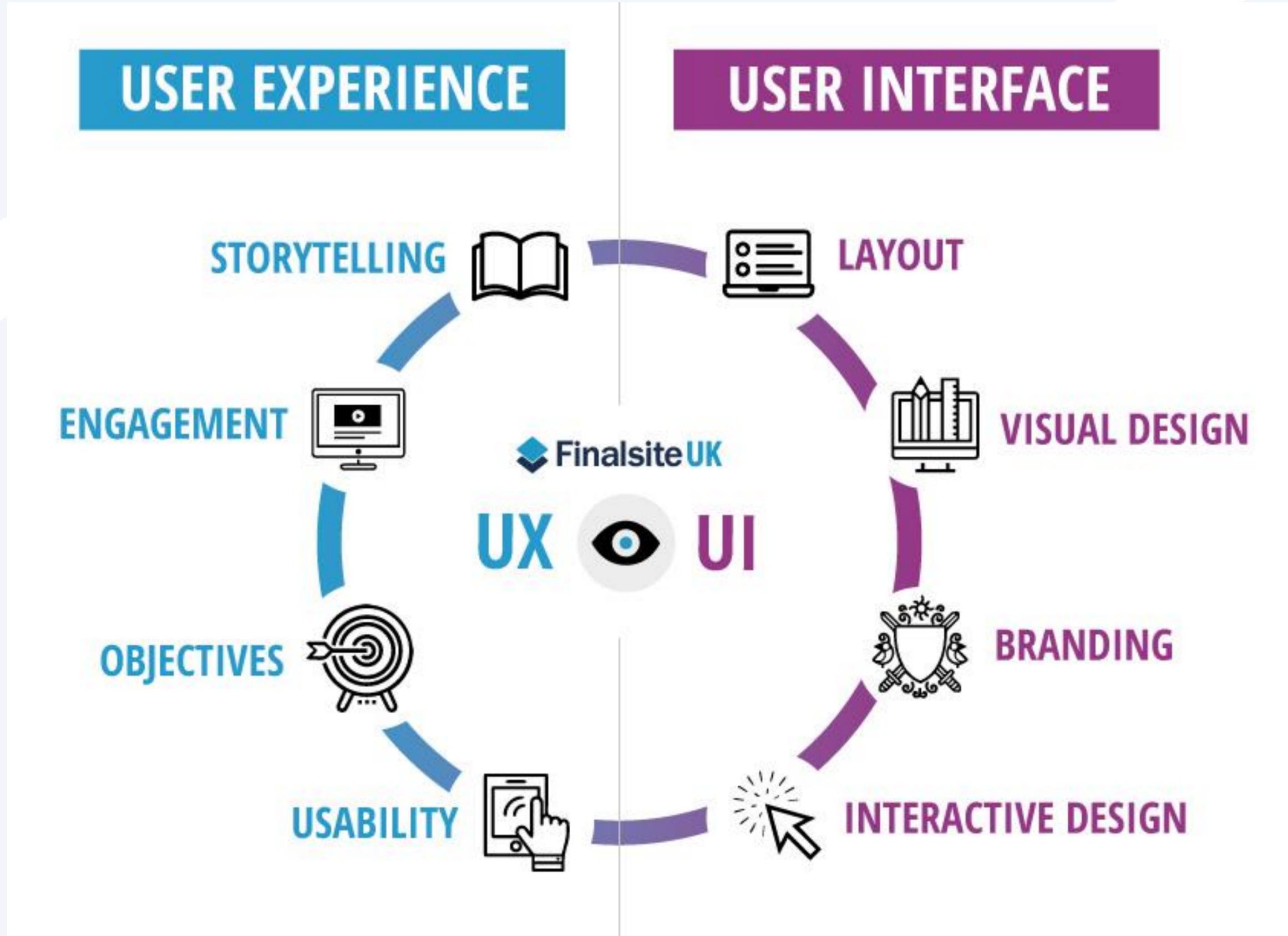
It is the **technical, coding, and architectural** dimension of the work. The actors involved are software developers, UI/UX designers, and system engineers.

Once the globalization decision is made, it is the process of designing the product's infrastructure to be flexible from the very beginning so that it can easily be adapted (localized) into different languages and cultures in the future.

For example; during the coding process, instead of embedding text directly into the source code (hardcoding), extracting it into external language files (strings), designing the interface to be flexible (expandable) for languages with longer words like German, providing right-to-left (RTL) support in the interface for languages like Arabic, or preventing character corruption by using Unicode (UTF-8) standards.

A Concrete Example in a Project Process: A Turkish e-commerce platform deciding to enter the Middle Eastern market, allocating a budget, and developing a marketing strategy is **Globalization**.

In line with this decision, the software team completely restructuring the website's code to support different currencies (Dinar, Dirham, etc.), different date/time formats, and the right-to-left reading layout with the Arabic alphabet is **Internationalization**. If internationalization is not done correctly, the localization (translation and cultural adaptation) phase cannot be initiated, or very costly technical crises will arise.



The History of Localization: How the Tech Giants Went Global

The 1980s: The "English-Only" Era In the early days of personal computing, the software industry was heavily centered in the United States. During the 1980s, products were designed with a single target audience in mind: English-speaking Americans. This meant the software was "ethnocentric."

From a technical perspective, text was "hardcoded," meaning the English words were buried directly inside the source code. The systems used basic ASCII encoding, which only supported the English alphabet. If you wanted to type in **Turkish, Russian, or Japanese, the computer simply couldn't understand it.**

The Catalyst: Microsoft, Adobe, and the Global Market (Late 80s - 1990s) As personal computers became popular worldwide, tech giants like Microsoft, Adobe, and IBM faced a massive problem. They wanted to sell their software in huge markets like Japan, Germany, and France, but **users in these countries didn't want to use English interfaces.**

When Microsoft wanted to launch the Windows operating system and Microsoft Office in Japan, they couldn't just "translate" the words. The Japanese language has thousands of Kanji characters. The old American code structure literally broke when trying to display them. Microsoft had to completely rebuild the underlying architecture of their software to support complex character sets.

Similarly, **Adobe** faced challenges with products like Photoshop and Illustrator. It wasn't just about translating the menus. Adobe had to adapt the software to different real-world standards. For example, the US uses "inches" and "Letter" size paper, while the rest of the world uses "centimeters" and "A4" size paper. If Adobe didn't localize these physical measurements in their software, designers in Europe or Asia couldn't use it properly.

The Birth of Internationalization (I18n) These tech giants quickly realized that rewriting the entire source code for every single country was a financial and engineering nightmare. They needed a new system.

This is how **Internationalization (I18n)** was born. Software engineers learned to separate the translatable text (strings) from the actual programming code. They created a flexible architecture. Once the software was "internationalized," it acted like a blank template. Then, translators could step in and do the **Localization (L10n)**, adapting the text, the images, the date formats, and the currencies without breaking the software's core functions.

The 1990s to Today: An Independent Industry By the late 1990s, especially with the rise of the internet, localization was no longer just a technical problem for software engineers; it had become a massive, independent industry. The GILT (Globalization, Internationalization, Localization, Translation) framework became the standard workflow. Translation was no longer just about books and documents; it was now the engine of global technology and digital communication.

<https://lokalise.com/blog/localization-examples/>