

43.Yazılım yerelleştirmede kitle kaynaklı çeviri etkinliği**Sevda PEKCOŞKUN GÜNER¹****APA:** Pekcoşkun Güner, S. (2022). Yazılım yerelleştirmede kitle kaynaklı çeviri etkinliği. *RumeliDE Dil ve Edebiyat Araştırmaları Dergisi*, (Ö11), 600-621. DOI: 10.29000/rumelide.1146742.**Öz**

Kitle kaynaklı çeviri, bir hizmet ya da ürünle ilgili içeriğin aynı anda birçok dile çevrilmesi amacıyla, gönüllü olarak bir araya gelmiş insanların gerçekleştirdiği bir yerelleştirme uygulaması olarak tanımlanabilir. Çevrimiçi platformlar aracılığıyla gerçekleştirilen kitle kaynaklı çeviri işlerinde yer alan gönüllüler, çevirmen ve/veya düzeltmen olarak projeye katkı sağlayabilirler. Kitle kaynaklı projelerde çok sayıda dile eş zamanlı olarak hızlı bir şekilde çeviri yapılabilmesi ve çeviri etkinliğinin gönüllülük esasına dayanması ya da düşük ücretlerle gerçekleştirilmesinin sonucu olarak maliyetten tasarruf edilir. Bu nedenle, bir çeviri projesinin kitle kaynaklı yapıda yürütülmesi şirketlere avantaj sağlar. Geleneksel yerelleştirme projelerinde “şelale modeli” adı verilen bir yöntem kullanılmaktaydı. Bu yöntemde, yazılım ya da uygulama geliştirildikten sonra yerelleştirme aşamasına geçiliyor; tüm metin unsurlarının bir araya getirilmesi ve çeviri işlemlerinin başlaması, ürünün piyasaya sunulmasından hemen önce gerçekleştiriliyordu. Günümüzde ise, yerelleştirme çalışmaları çoğunlukla yinelemeli ve iş birliğine dayalı yöntemlerin öne çıktığı çevik yazılım geliştirme yaklaşımlarıyla yürütülmektedir. Çevik yerelleştirme döngüsü içerisinde yer alan çevirmenler, ürün geliştirme çalışmaları devam ettikçe iş akışı içinde sürekli olarak görev alır. Geleneksel şelale modelinin aksine, çevik yerelleştirme sürecinde, çeviriler, erek kitlenin/son kullanıcıların beklenti ve gereksinimlerine yanıt verebilecek şekilde, yenilikçi çözümler üretilerek ve yazılım güncellemeleri dikkate alınarak gerçekleştirilir. *Mozilla Firefox* gibi açık kaynaklı yazılımlar, içerik çevirilerinde makine çevirisi sistemleri ve kendi çeviri bellekleri yanında, kitle kaynaklı çeviri olanaklarını da kullanmaktadır. Sistemde, farklı dil ve lehçelere çeviri yapan 200’ü aşkın ekip yer almakta, gün içinde binlerce metin dizgisi birçok dile çevrilmekte ve gerekli düzeltmeler gerçekleştirilmektedir. Bu çalışma kapsamında, ilk olarak yazılım yerelleştirme sürecinde çevirmenin konumu sorgulanacak; ardından, kitle kaynaklı yerelleştirme çalışmalarında çevirmenin ne gibi görevler üstlenebildiği *Mozilla Pontoon* yerelleştirme yönetim sistemi özelinde örneklerle açıklanmaya çalışılacaktır.

Anahtar kelimeler: Kitle kaynaklı çeviri, çevik yerelleştirme, açık kaynaklı yazılım, yerelleştirme yönetim sistemi

Crowdsourced translation in software localization**Abstract**

Crowdsourced translation can be defined as a localization practice performed by people who voluntarily participate in translating the content of a service or product into many languages simultaneously. Volunteer translators involved in crowdsourcing contribute to the project as translators and/or editors via online platforms. The advantages of crowdsourced translation for companies can be summarized as: source content is translated into many languages at the same time,

1 Dr. Öğr. Üyesi; Kırklareli Üniversitesi, Fen-Edebiyat Fakültesi, İngilizce Mütercim ve Tercümanlık Bölümü (Kırklareli, Türkiye), sevda.pekcoskun@gmail.com, ORCID ID: 0000-0003-2750-3217 [Araştırma makalesi, Makale kayıt tarihi: 21.06.2022-kabul tarihi: 20.07.2022; DOI: 10.29000/rumelide.1146742]

and translation activities are on a voluntary basis, or participants work at low fees, which results in cost savings. The “Waterfall model” was used in traditional localization projects. In this method, the localization phase is started after the software or application is developed. All text components are combined, and translation activities are carried out shortly before the launch of the product. Today, localization practices are mostly performed with agile software development approaches in which iterative and collaborative methods are prominently used. Translators in the agile localization cycle are constantly involved in the workflow while product development continues. Unlike the traditional waterfall model, in agile localization, the relevant content is translated in accordance with the expectations and needs of the target audience/end users by producing innovative solutions and taking software updates into account. Besides machine translation systems and translation memories, open-source software products such as *Mozilla Firefox* use crowdsourced translation. In the system, there are more than 200 teams that translate into different languages and dialects. Thousands of text strings are translated into many target languages each day, and necessary editing is done. Within the scope of this study, first of all, the translator's position in software localization processes will be questioned. Then, the tasks of the translator will be explained with examples taken from the *Mozilla Pontoon* localization management system.

Keywords: Crowdsourced translation, agile localization, open-source software, localization management system

Giriş

Yazılım yerelleştirme, bir yazılım ürününün hedef pazarın dilsel, kültürel ve teknik özelliklerine göre uyarlanması anlamına gelmektedir. Bir uygulama birden fazla pazarda tanıtılacak ve kullanıma sunulacaksa, yazılım geliştirme süreci başlar başlamaz yerelleştirme konusu gündeme gelir (Collins, 2001, s. 36). Ürün yerelleştirme aşlında pazarlama stratejisinin de bir parçasıdır. Bir yazılım yerelleştirme işinde öncelikle hangi dillere çeviri yapılacağına karar verilir. Bu karar, ürünün hangi bölge ya da ülkelerde piyasaya sunulacağıyla doğrudan ilişkilidir. Ardından, ne tür bir yerelleştirmeye (tam ya da kısmi) gereksinim duyulduğu üzerine düşünülür. Bir ürünün tüm yazılım paketlerine (kullanıcı arayüzü, yardım ve destek belgeleri, kişiselleştirme ayarları, vb.) ilişkin içerikler ya da maliyetten tasarruf edilmesi amacıyla yalnızca gerekli görülen kısımlar yerelleştirilebilir. Yazılım yerelleştirmenin temel amacı son kullanıcı deneyimini iyileştirmektir. Bu nedenle, hataların düzeltilmesi amacıyla, ürün, geliştirme aşamalarının tümünde teknik açıdan sınırlanırken, dil bağlamında da çeşitli testlere tabi tutulur.

Yazılım yerelleştirme sürecinde, çevrilmesi gereken kaynak dizgilerinin² tümünün çevrilip çevrilmediğine, uygun dil kesitinin kullanılıp kullanılmadığına, terim birliğinin sağlanıp sağlanmadığına, standart biçimsel özelliklere uyulup uyulmadığına dil odaklı testler sayesinde karar verilir ve gerekli düzeltmeler yapılır. Aynı zamanda, dilbilgisi ve yazım kurallarına uygunluk ve akıcılık gibi unsurlar da test aşamalarında gözden geçirilir.

Yerelleştirme testleri “dile ilişkin testler” ve “kullanıcı arayüzüne (görselliğe) ilişkin testler” olarak ikiye ayrılabilir. Dile ilişkin testler, yerelleştirilmiş bir ürünün tüm dilsel yönlerine odaklanırken; kullanıcı arayüzüne ilişkin testler, yerelleştirilmiş ürünün, diyalog kutucukları, menüler ve iletiler gibi kullanıcı

² Dizgi (en. *string*), programlamada kullanılan bir veri türüdür. Harfler, sayılar, semboller, boşluklar gibi bir dizi karakter içerir. Yerelleştirme kapsamında, “kaynak metinler” yerine dil dışı unsurları da içeren “kaynak dizgilerin” çevirileri yapılır. Örneğin, “{user-name} sisteme giriş yaptı.” bir dizgidir. Bu yapılarıyla dizgiler, çeviri birimlerini andırır.

tarafından görülen özelliklerine odaklanır. Örneğin, metin içerisindeki çoklu boşluklar çeviri sırasında kaldırıldığında, metnin arayüz görünümünde hatalar oluşup oluşmadığı dilsel test aşamalarında kontrol edilmelidir. Yerleştirilmiş versiyonun, özgün versiyonla aynı sayıda menü, seçenek ve komuta sahip olup olmadığı ise arayüz testleriyle sınanmalıdır (Esselink, 2000, ss. 150-152).

Kalite değerlendirme aşamalarından da anlaşıldığı gibi, yazılım yerleştirme sürecinde gerçekleştirilen çeviri etkinliği, geleneksel çeviri etkinliğinden oldukça farklıdır. Bu nedenle, ilgili projelerde yer alan çevirmenlerin yazılım yerleştirme aşamaları ve yerleştirme sürecinin bileşenleri hakkında bilgi sahibi olması, çeviri sürecinin daha sorunsuz bir şekilde ilerlemesini sağlayabilmektedir. Bert Esselink (2003, s. 24), İrlanda’da yerleştirme uzmanı olarak istihdam edilmek üzere, Avrupa’nın çeşitli bölgelerinden gelen çevirmenler, yerleştirme mühendisleri ve proje yöneticilerine eğitim verildiğini ifade etmiş, çevirmenlerin çoğunun yazılım yerleştirmeye ilişkin kavramlarla bu eğitimler sırasında ilk kez karşılaştığının altını çizmiştir. Yerleştirme, çeviri etkinliğini içine alan bir süreç olup, dil ve iletişim becerileri yanında, dijital ve araçsal becerileri de gerektiren bir olgudur.

Yazılım yerleştirme projelerinde, çevirmenler iki şekilde görev alabilir: Ürünü geliştiren şirketin oluşturduğu çeviri ekibinde istihdam edilebilirler (en. *in-house translator*) ya da şirketin kitle kaynaklı hizmet alması durumunda, çevrimiçi platformlar aracılığıyla çeviri yapabilirler (en. *crowdsourced translator*). Şirketin kendi yerleştirme ekibi varsa, görece daha az sayıda kişi ilgili projede çalışır ve ekip içi iletişim kolaylıkla sağlanır. Bu durum, iş akışının sektöre uğramadan yönetilmesini mümkün kılar. Ancak günümüzde, yazılımların sürekli güncellenmesi, ürünlere ait içeriklerin devamlı olarak artması ve tüm bunlara bağlı olarak çeviri maliyetlerinin de yükselmesi nedeniyle birçok şirket kitle kaynaklı çeviriye yönelmiş, gönüllü çevirmen ve editörlerden oluşan dil ekipleri oluşturmuştur.

Kitle kaynaklı projelerde, çevirmenlerin motivasyon kaynağı “para kazanmak” değildir. Temel motivasyonlar; bir topluluğun parçası olma, kaliteli ve fazla miktarlarda dizgi çevirisi yaparak ün kazanma, yeteneklerini başkalarının faydası için kullanma, ilgi duyulan alanda bilgi sahibi olma isteği olarak sıralanabilir. Gönüllülük esasına dayanan kitle kaynaklı çeviri etkinliği, teknoloji ve bilişim alanındaki gelişimlere paralel olarak potansiyelini gün geçtikçe artırmaktadır. Bu çalışma kapsamında, çeviribilim alan yazınında görece yeni bir kavram olarak karşımıza çıkan kitle kaynaklı çeviri, yazılım yerleştirme bağlamında ele alınmaya çalışılmıştır.

1. Kitle kaynaklı çeviri etkinliğine genel bir bakış

1.1. Kitle kaynak kullanımı

Kitle kaynak kullanımı (en. *crowdsourcing*), bir sorunun çözümüne yönelik olarak geniş bir insan topluluğuna görev verilmesi şeklinde tanımlanabilir. Kitle kaynaklı projelerde gönüllülük esastır ve işi başlatan kişi ya da kurum internet üzerinden bir çağrı yaparak, projesinde çalıştırmak üzere konuyla ilgili kişilere ulaşmaya çalışır. Kitle kaynaklı işlerde çalışana ödeme yapılabildiği gibi, çalışan verdiği hizmet karşılığında herhangi bir ücret almayabilir. Kavram, 2006 yılında Amerikalı gazeteci Jeff Howe tarafından ilk kez bir terim olarak kullanılmıştır. Howe, kitle kaynak kullanımını, “bir şirket veya kurumun, daha önce çalışanları tarafından yapılan bir işin duyurusunu, bireysel olarak tanımadıkları insanlardan oluşan geniş kitlelere açık çağrı yöntemini kullanarak yapması ve bu kitleleri taşeron olarak çalıştırması” şeklinde tanımlamaktadır. Söz konusu iş, “akran üretimi” (en. *peer production*) yöntemiyle; başka bir deyişle, ortaklaşa olarak yapılabileceği gibi, tek kişi tarafından da üstlenilebilir (Estellés-Arolas, vd., 2015; Howe, 2006).

Kitle kaynaklı işlerde, şirketler ve kullanıcılar internet üzerinden (sosyal medya ve mobil uygulamalar gibi platformlar aracılığıyla) birbirleriyle iletişime geçerler. Çalışan, geleneksel iş akışında, işverenin verdiği görevi belli bir süre içinde yerine getirmekle yükümlüdür; kitle kaynaklı çalışma sisteminde ise işverenin yaptırmak istediği işle ilgili çağrı, işi yapmaya istekli olan ve hazır bulunan kitlelere ulaştırılır. Kitle kaynak kullanımı, iş birliği odaklı çalışma süreçlerinin önünü açan yeni ve çağdaş bir model olarak kabul edilebilir.

Estellés-Arolas ve González-Ladrón-de-Guevara kitle kaynak kullanımının internet tabanlı herhangi bir girişimden neden farklı olduğuna ilişkin kapsamlı bir inceleme gerçekleştirmiş ve kitle kaynaklı bir işe yönelik olarak başlatılan bir girişimin özünde sekiz temel bileşen barındırdığını bildirmiştir: (1) Sınırları net olarak belirlenmiş olan bir kitlenin bulunması, (2) Net amacı olan bir görevin tanımlanmış olması, (3) Kitlenin yaptığı iş karşılığında alacağı mükafatın açıkça belirlenmesi, (4) Kitle kaynaklı işi başlatan kişi ya da kurumun açık bir şekilde tanımlanması, (5) İş başlatanın elde edeceği faydanın net olarak belirlenmesi, (6) Sürecin çevrimiçi şekilde, aktif katılıma açık olarak işlenmesi, (7) Ne zaman sonlandırılacağı kesin olarak belli olmayan açık çağrı yönteminin uygulanması ve (8) Tüm süreç boyunca kesinlikle internetin kullanılması gerekmektedir (2012, s. 10).

*Amazon Mechanical Turk*³ adlı dijital platform kitle kaynaklı bir ortam olarak kabul edilebilir. Bu platformda ilan edilen açık pozisyonlara yönelik uygun iş gücünün görevlendirilmesi, bireyler ve işletmelerin bir araya gelmesiyle sağlanır. Böylece, şirketlerin kolektif zekâ, beceri ve deneyimlerden en üst düzeyde faydalanmasının önü açılmış olur. Kitle kaynak kullanımı, iş süreçlerinin düzene sokulması, veri toplama ve analiz çalışmalarının artırılması ve makine öğrenmesi (en. *machine learning*) süreçlerinin hızlanması için de başvuru bir yöntemdir.

Kitle kaynaklı işler, iş birliği ruhu ve paylaşımcı yaklaşımla başarılı bir şekilde sonuçlandırılabilir. Kitle kaynaklı projelerin temelinde “birlikten kuvvet doğar” anlayışı yatmaktadır; söz konusu iş, birden fazla kişi tarafından yüklenilir ve süreç boyunca bu kişilerin tüm bilgi, beceri ve deneyimlerinden yararlanır. Dolayısıyla, ortaya çıkan ürün, ilgi veya uzmanlık alanları benzer olan bireylerden oluşan grubun kolektif çalışmasının sonucudur. James Surowiecki, *The Wisdom of Crowds* (tr. *Kitlelerin Bilgeliği*) adlı kitabında, “uygun koşullar altında çalışan gruplar, oldukça zeki bir profil çizerler; grubun zekâ düzeyi, grupta yer alan en zeki insanların zekâ düzeylerinden çok daha yüksektir,” şeklinde bir saptamada bulunmuştur (2004, s. xiii). Bu bağlamda, internet milyonlarca farklı ve bağımsız fikri bir araya getirebilen mükemmel bir altyapı sağlar (a.g.e., s. xix). Kitle kaynaklı projeler, devingen bir yapıdadır ve projelerin devamlılığında topluluk içi etkileşim önemli rol oynar.

İnternet sayesinde dünyanın farklı yerlerinden, farklı kültürel altyapılara sahip birçok insan bir araya gelir ve görüş alışverişinde bulunur; dolayısıyla genel ağ, barındırdığı birçok kitle kaynaklı uygulama çözümüyle farklı düşüncelerin bir araya gelmesini sağlayan faydalı bir araç olarak düşünülebilir (Brabham, 2008, s. 81). Brabham, kitle kaynak kullanımını “sorun çözmeye yönelik bir model” olarak nitelendirir. İnternetin çalışma mantığına paralel şekilde, kitle kaynaklı projelerde merkeziyetçi bir yapı bulunmamaktadır. Bu durum belli oranda fırsat eşitliğini de beraberinde getirir. Örneğin, Amerika Birleşik Devletleri’nde yer alan bir yazılım şirketinin açtığı çağrıya, dünyanın neresinde yaşadığı fark etmeksizin, ilgili programlama dilini bilen bir yazılımcı başvuruda bulunabilir.

Kitle kaynak kullanımıyla müşteriler ve kullanıcılar potansiyel ortaklar haline geldikçe, iş modellerinin tanımı da büyük ölçüde değişmiştir. Böylece ortaklar, satın almak, satış yapmak ya da tüketmek gibi

³ <https://www.mturk.com/>

eylemlerin bir ya da birkaç aşamasında bulunmak yerine, tüm iş akışına dahil olmuşlardır. Kitle kaynak kullanımıyla ilgili yaklaşımlar, piyasanın ve içinde bulunulan zamanın koşullarına göre güncellenmektedir. Kitle kaynaklı çalışmalarda bir makinenin ya da motorun kullanımı (makine çevirisi yazılımları gibi) zorunlu değildir, ancak Web 2.0⁴ ilgili süreçleri önemli ölçüde kolaylaştırmaktadır (Anastasiou ve Gupta, 2011, s. 1). Web 1.0 çağında sadece büyük şirketler içerik üretebilmekte ve web sayfası oluşturabilmekte iken, Web 2.0'da internet kullanıcısı olan tüm bireyler içerik üretebilmeye başlamış ve sosyal medya kullanıcılığı bu döneme damgasını vurmuştur. Blog yazma ve forumlara katılma da bu dönemde ortaya çıkan internet kullanımlarıdır. Genel ağın durağan yapıdan devingen bir yapıya evrilmesiyle kitleler arası etkileşim mümkün olabilmiş, kitle kaynak kullanımı da tüm dünyada yaygınlık kazanmıştır.

Bir yöntem olarak kitle kaynak kullanımı, internetin yaygınlaşması ile birlikte oldukça farklı biçimler ve görev tanımlarıyla kendini göstermeye başlamıştır. Kitle kaynak kullanımı, katılımcıların kendi rızalarına ve emeklerine dayalı şekilde katkıda buldukları bir çalışma şekli olabildiği gibi, farkında olmadan da sürece katkı sağlayabildikleri bir yöntem olarak da karşımıza çıkabilir. Örneğin, mobil navigasyon uygulamalarının kullanımı yaygınlaştıkça, uygulamayı kullananlardan toplanan konum ve hareket verileri belirli bölgelerde oluşan trafik yoğunluklarının belirlenmesini sağlamaktadır. Bu uygulama örneğinde, kullanıcılar uygulamadan yararlanırken aynı zamanda uygulamaya katkıda bulunmaktadır. Navigasyon sırasında trafik yoğunluğu olan bölgenin ne kadar sürede geçilebileceği gibi bilgilerin hesaplanması da aynı bölgede bulunan çok sayıdaki kullanıcının sağlamış olduğu veriler kullanılarak gerçekleştirilmektedir. Bu tarz uygulamalar, kullanıcıların doğrudan emek sarf etmediği kitle kaynak kullanımına örnektir.

LEGO şirketi, kullanıcıların yeni LEGO setleri için fikirlerini paylaşabilecekleri *LEGO Ideas* (tr. *LEGO Fikirler*) platformunu kurmuştur. Platformda, tüketiciler hem yeni fikirler sunmakta hem de gönderilen diğer fikirler için oy verip, geri bildirimde bulunabilmektedir. 10.000'den fazla oy alan herhangi bir fikir LEGO tarafından incelemeye alınır. Fikri uygun bulunan kullanıcı, fikrini gerçeğe dönüştürmek için LEGO ekibiyle birlikte çalışır ve satışlardan telif ücreti alır⁵. Bu uygulama da gönüllü katılımcıların sürece katkıda bulunduğu kitle kaynaklı bir proje örneğidir.

Sonuç olarak, iş süreçlerinde kitle kaynak kullanımı yöntemiyle bir topluluğun bilgisine ve deneyimine başvurulur. Topluluk içinde yer alan bireyler; yani kitle kaynak, sosyal medya, mobil uygulamalar veya çevrimiçi platformlar aracılığıyla projelere katılım sağlar; katılımcılar projelerde gönüllü olarak ya da düşük bir ücret karşılığında çalışırlar. Kitle kaynaklı projelere katılımın ön koşulu dijital okur yazar olmaktır; katılımcının, ilgili platformlarda yer alan profilinde uzmanlık alanlarını net biçimde belirtmesi ve etkili iletişim becerilerine sahip olması da süreci olumlu yönde etkileyebilecek diğer bileşenlerdir.

1.2. Kitle kaynaklı çeviri

Kitle kaynaklı çeviri, kendi anadili ile birlikte en az iki dil bilen kişilerin bir ürün ya da hizmetle ilgili içeriği gönüllü olarak çevirmesi anlamına gelmektedir. Kitle kaynaklı çeviri yapan çevirmenler,

⁴ Web 2.0, Web 1.0'a kıyasla, kullanıcılar tarafından oluşturulan daha fazla içeriğe sahip internet anlamına gelir. Web 2.0 sayesinde, kullanıcılar yalnızca internetten bilgi alan değil, aynı zamanda bilgi üreten ve paylaşan duruma gelmişlerdir. Dijital dönüşüm faaliyetlerinin yaygınlaşmasıyla, internet kullanıcıları, "izleyen" olmaktan çıkıp, "içerik üreten" eyleyenler olmuştur.

⁵ <https://ideas.lego.com/>

sundukları hizmet karşılığında hiçbir ücret almayabilir ya da düşük bir ücret karşılığında çeviri yapabilir. Çeviri, halihazırda mevcut olan kitle kaynaklı bir platform üzerinden çevrimiçi şekilde yapılır.

Kitle kaynaklı çeviri sisteminde iş akışına geçmeden önce, benzer bir kavram olan dış kaynak kullanımı (en. *outsourcing*) iş akışından söz etmek uygun olacaktır. Dış kaynak kullanımına yönelik iş akışında, proje yöneticisi, müsaitlik durumlarına bağlı olarak, mevcut akredite çevirmen havuzunda uzmanlık düzeyi ve çeviri yapabildiği dil çiftleri kayıtlı olan çevirmenlerden projeye uygun olanları seçer ve bu çevirmenlere proje kapsamında çevrilmesi gereken metin kesitlerini yollar. Çevirmenler, çevrimdışı ya da çevrimiçi olarak gerçekleştirdikleri çevirileri proje yöneticisine gönderir ve yönetici, erek metinleri düzeltmenlere ve son okuma uzmanlarına iletir. Süreç tamamlandığında müşteriye gönderilecek olan son sürüm de hazır olur. Ancak, klasik dış kaynak kullanımı yönteminde iş akışında aksayan ya da süreci yavaşlatan birtakım kısıtlar meydana gelebilmektedir. Örneğin, günümüzde özellikle teknik metinler tek bir dilden ziyade birçok dile eş zamanlı olarak çevrilmektedir. Bu durum, sürecin her bir dil çifti için tekrarlanmasını ve artan yönetsel karmaşıklıkla beraberinde getirir. Başka bir deyişle, aynı süreç her dil çifti için ardışık şekilde yinelenir ve bu da zaman kaybına yol açar. Ek olarak, proje planlaması, ücretlendirmeler, risk yönetimi gibi konular her bir süreçte tekrarlandığından, proje yöneticisinin ilgilenmek zorunda olduğu karmaşıklık düzeyi de artar.

Öte yandan, standart bir kitle kaynaklı çeviri iş akışında ilgili web sitesinin ana sayfasına girilerek kayıt olunur ve çeviriye başlanır. Sisteme kayıtlı kişiler; çevirmen, düzeltmen, müşteri ve yönetici olmak üzere çeşitli görev tanımlarıyla yetkilendirilir. Sistem kullanıcısı, yetkisine bağlı olarak sisteme giriş yaptığı anda uygun panel ekranına yönlendirilir. Kullanıcılar, yetkilerine göre yalnızca çevirmen olarak görev alabileceği gibi hem çevirmen hem de düzeltmen/kontrolör veya son okuma uzmanı olarak da sistem içinde konumlanabilir. Kitle kaynaklı projelerde çeviri yapanların çoğu profesyonel olmayan kişilerdir, ancak profesyonel çevirmenler de çeviri yaparlar. Kitle kaynaklı çeviri, çeviri süresinin kısaltılması, maliyetlerin düşürülmesi ve birden fazla dil çiftinde eş zamanlı olarak çeviri yapılabilmesine olanak sağlaması açısından dış kaynak kullanımına göre daha avantajlıdır. Günümüzde birçok şirket kitle kaynaklı çeviri uygulamalarına yönelmiştir. Bu durumun nedenlerini kısaca şöyle özetleyebiliriz:

- Bu yöntemle aynı kaynak metnin birden fazla erek dile çevirisi oldukça kısa sürede yapılabilir.
- Her ne kadar ilk aşama, yani kitle kaynaklı platformun oluşturulması adımı (mobil uygulama, web sitesi, vs.) maliyetli olsa da projelerde çoğunlukla gönüllü çevirmenler yer aldığından, şirketler orta ve uzun vadede maliyetten tasarruf eder.
- Projelerde görev alan çevirmenler halihazırda konuyla ilgili olduklarından belli düzeyde özel alan ve terminoloji bilgisine sahiptir, bu nedenle görece kaliteli erek metinler üretebilir.

Ancak, kitle kaynaklı çeviri uygulamalarında birtakım olumsuz sonuçlarla da karşılaşılabilir. Çevirmenlerin çoğunun “gönüllülük” esasına dayalı olarak çalışması ve çevirmenin sahip olması gereken dilsel, kültürel ve iletişimsel becerilere yeterince sahip olmaması nedeniyle düşük kaliteli çeviriler ortaya çıkabilir. Aynı içeriğe ait kesitler farklı çevirmenler tarafından erek dile aktarıldığından, terim birliğinin sağlanmasında zorluklar yaşanabilir. Ayrıca, akademik ya da profesyonel dil kesitine sahip metinlerde terim yoğunluğu oldukça fazla olduğundan ve amatör çevirmenler terim çevirisine yönelik yöntemleri (açıklama, ödüncleme, vb.) tam olarak bilemeyebileceğinden erek metnin tutarlılığı ve anlaşılabilirliği zarar görebilir.

Profesyonel çevirmenlerin konumu ve çevirmenlik mesleğinin geleceği açısından da birtakım endişeler dile getirilmektedir. Kitle kaynaklı çevirinin yaygınlık kazanmasıyla profesyonel çevirmenlerin işlerini ve statülerini kaybedeceğine ilişkin görüşler de vardır (European Commission, 2012, s. 23).

Kitle kaynak kullanımı, farklı hedef ve iş akışlarına sahip teknolojik platformlarda kendini göstermeye başlamış ve günümüzde kitle kaynaklı süreçler belli bir gelişmişlik düzeyine erişmiştir. Kitle kaynak kullanımını çeviribilim odaklı bir bakış açısıyla sınıflandırmak zor olsa da sürecin amacı, uzmanlık alanı, işin kâr amacı güdüp gütmeyeceği gibi ölçütler göz önünde bulundurularak genel bir sınıflandırma yapılabilir (*Jiménez-Crespo, 2017, ss. 83-85*):

1. *Genel amaçlı kitle kaynak platformları*: Topluluğa mikro görevler vererek minik kesitlerin çevirilerinin yapılmasına olanak sağlayan *Crowdfower* ve *Amazon Mechanical Turk (AMT)* gibi platformlardır. Bu platformlarda çevirmene yapılan ödemeler düşüktür.

2. *Dil endüstrisi alanında çalışan ve kâr amacı güden şirketler tarafından geliştirilen kitle kaynak platformları*: Bunlar, profesyonellerin ve gönüllülerin firmalar tarafından kitle kaynağı olarak kullanılabilmesini sağlamak amacıyla oluşturulmuş platformlardır. *Crowdin*, *Smartling*, *Lingotek*, *Lingohub* ve *Translation Workspace* bu platformlara örnek olarak verilebilir. Bu kategoriye giren platformlar, çoğunlukla çok dilli içeriklerin çevrilmesine yönelik olarak oluşturulmuş bulut tabanlı yerelleştirme yönetim sistemleridir.

Bu platformların en bilinenlerinden olan *Crowdin*, temel özelliklerini “otomasyon”, “iş birliği” ve “kalite” olarak açıklar⁶. Otomasyon özelliği ile çok dilli içerik eş zamanlı olarak çevrilebilir, çevirmenler oluşturulan en son içeriğe (son versiyon) erişebilir ve tamamlanan çeviriler otomatik olarak yazılıma aktarılabilir. İş birliği özelliğiyle, işi başlatan kişi birlikte çalışmak istediği kişilere davet gönderebilir, görevler oluşturabilir, iş akışını gerçek zamanlı olarak izleyebilir. Daha kaliteli erek metinlerin tasarlanması için ise, çeviriler üzerinden çevrimiçi tartışmalar yapıp gerekli düzeltmeler gerçekleştirilebilir. Bu platformlar, işverenlerle çevirmenleri bir araya getiren ve çevrimiçi olarak çalışan dijital ortamlardır.

3. *Açık kaynaklı kitle kaynak platformları*: Bunlar, açık kaynaklı yazılım yerelleştirme platformlarıdır. Yerelleştirme görevini çevirmenlere verecek olan firma, bu platformları kendi sunucularında çalışır hale getirebilir. *Pootle* ve *Translate Wiki* örnek olarak verilebilir.

4. *Şirketler tarafından özel olarak geliştirilmiş platformlar*: Birçok şirket kendi kitle kaynak platformunu geliştirmiştir. Sosyal medya sitesi *Facebook* bu kategoriye örnektir. *Facebook*, 2008 yılında kitle kaynaklı çeviri seferberliğini başlatmış, internet kullanıcıları, web sayfalarının dilsel içeriğini birçok dile eş zamanlı olarak çevirmiştir. Örneğin, 2008 yılında, yalnızca 24 saat içinde *Facebook*'un tüm içeriği 8000 gönüllü çevirmen tarafından Fransızcaya çevrilmiştir⁷.

Facebook, 2022 yılı Nisan ayında kitle kaynaklı çeviri çalışmalarına son vermiş, çeviri ekibi bu kararı almalarının ardında yatan sebepleri, gönüllü çevirmen sayısında ciddi bir düşüş yaşanması ve yerleştirilmiş *Facebook* kullanıcı arayüzlerinin kullanımında kademeli bir azalma gözlemlenmesi olarak açıklamıştır. Ayrıca, bundan sonra gönüllü çevirmenlerle değil profesyonel çevirmenlerle çalışacaklarını, böylece daha eksiksiz bir kullanıcı deneyimi sunmak istediklerini belirtmişlerdir⁸.

5. *Görsel-işitsel çeviri platformları*: Sınırları belli iş akışlarına sahip olan görsel-işitsel çeviri platformlarında, kaynak içeriğe farklı erek dillerde altyazı oluşturabilecek nitelikte yazılım becerilerine sahip olan hayran kitleleri gönüllü olarak çeviri yapar. *Dotsub* bu alanda oldukça popüler bir platformdur. *Dotsub* video çeviri yönetim sistemi sayesinde ilgili içerik birçok dile kolayca çevrilebilir.

⁶ <https://crowdin.com/>

⁷ <https://www.web-translations.com/blog/crowdsourced-translation/>

⁸ <https://tr-tr.facebook.com/TranslateFacebookTeam/>

6. *Kâr amacı gütmeyen platformlar*: Kâr amacı olmayan, çeviri faaliyetlerini insani yardım olarak gören platformlardır. *Translators without Borders*⁹ (tr. Sınır tanımayan çevirmenler) platformu, 1993 yılında kurulmuş, toplum çevirmenliği ekseninde şekillenmiş, farklı diller konuşan insanları sağlık, eğitim, beslenme gibi konularda bilgilendirme amacıyla çalışan bir oluşumdur. İhtiyacı olan insanlara finansal destek sağlamak için kurulmuş, kâr amacı gütmeyen bir kuruluş olan *Kiva*¹⁰ da gönüllü çevirmen ve düzeltmenlerle çalışmaktadır.

7. *Dil öğrenimine yönelik platformlar*: *Duolingo* gibi uygulamalar oylama yöntemi ile kitle kaynaklı çeviriden yararlanmaktadır.

8. *Akıllı telefon uygulamalarıyla erişilebilen kitle kaynaklı platformlar*: Masaüstü uygulamalardan, bulut tabanlı sistemlere geçilmesiyle masaüstü bilgisayarların yerini tabletler ve akıllı telefonlar almıştır. Bu cihazlar aracılığıyla erişilen uygulamalar üzerinden çeviri yapılabilmektedir. *Unbabel* art alanında yapay zekâ teknolojisini kullanan ve kullanıcılara kaynak metnin makine çevirisi çıktılarını sunan bir uygulamadır. Çevirmenler, çıktılar üzerinde düzelttiler gerçekleştirerek ilgili görevleri tamamlarlar.

9. *Makine çevirisi (MÇ) tabanlı kitle kaynak platformları*: *Google Çeviri*, *Yandex Çeviri*, *Microsoft Bing Çeviri* gibi MÇ sistemleriyle yapılan çeviriler üzerinde art düzenleme (en. *post-editing*) işlemleri gerçekleştirilerek, sistemlere çeviri önerilerinde bulunulabilir. Çeviri önerileri, bu sistemlerin daha kaliteli çeviriler gerçekleştirmesi için oldukça önemlidir.

10. *Çeviri sürecini destekleyen dilsel görevler*: Kitle kaynaklı süreçler, çeviri sürecine destek olabilecek kaynakların hazırlanmasında da geçerli olabilir. Terimce ve sözlükçelerin hazırlanması bu kategoriyeye örnektir. *TermWiki pro* sık kullanılan bulut-tabanlı bir terminoloji yönetim sistemidir.

Yukarıda da belirtildiği gibi, çevirmenler birçok kanal üzerinden kitle kaynaklı projelere erişebilirler. Çeviri odaklı düşünüldüğünde, çevirmenlerin yer alabileceği iki temel kitle kaynak platformu karşımıza çıkmaktadır. Bunlardan ilki işveren ve çevirmenleri bir araya getiren kitle kaynak platformlarıdır. Bu platformlarda, farklı şirketlerden yetkili kişiler projeler başlatırlar. Çevirmen(ler) kendi uzmanlık alanına uygun projelere başvuru yapabilir ya da işi başlatan kurum, çevirmen(ler)e teklif sunabilir; başka bir deyişle, işi alması için çevirmen(ler)i davet edebilir (örneğin, *Crowdin*). İkinci grup ise, yalnızca kendi bünyesinde gerçekleştirilmesi gereken çeviriler için, belli bir şirketin kurduğu kitle kaynak platformlarıdır (örneğin, *Mozilla Pontoon*). Bu platformlara üye olan çevirmenler tarafından yazılım içerikleri çevrilir ve erek metinler üzerinde düzelttiler gerçekleştirilir.

Kitle kaynaklı çeviri uygulamaları, yerelleştirmeden bağımsız olarak düşünülemez; çünkü çeviriler çoğunlukla bir yerelleştirme yönetim platformu üzerinden gerçekleştirilmektedir. Bu nedenle, kitle kaynaklı çeviri ve yerelleştirme kavramları iç içe geçmiş yapıdadır ve projelerde görev alan çevirmenlerin yerelleştirme konusunda da bilgi sahibi olması gerekmektedir. 2. bölümde yerelleştirme olgusundan kısaca söz edilecek ve kitle kaynaklı çeviri odağında, yazılım yerelleştirme türlerine değinilecektir.

2. Yazılım yerelleştirme sürecine ilişkin kavramlar

Bilişim alanındaki gelişmelerin katkısıyla şekillenen dijital çağda, başta yazılım sektörü olmak üzere, birçok teknik alanda çeviriye olan gereksinim hızla artmıştır. Bu bağlamda, *GILT* (en. *Globalization, Internationalization, Localization, Translation*) kavramından kısaca söz etmek gerekir. Küreselleşme (en. *globalization*), uluslararasılaştırma (en. *internationalization*), yerelleştirme (en. *localization*) ve çeviri (en. *translation*) etkinlikleri iç içe geçmiş kavramlardır. Bilindiği gibi yerelleştirme, bir ürün, uygulama ya da hizmetin hedeflenen piyasanın dilsel ve kültürel yapısına uygun şekilde uyarlanması anlamına gelmektedir. Dolayısıyla, çeviri de yerelleştirme sürecinin önemli bir bileşenidir. Çeviri, diller




⁹ <https://translatorswithoutborders.org/about-us/>

¹⁰ <https://www.kiva.org/>

arası aktarımı mümkün kılarken; yerelleştirme, güncel teknolojileri kullanarak, ilgili içeriğin erek kültürün beklentilerine uygun şekilde tasarlanmasını sağlar.

Yerelleştirme süreci, renkler, yazı tipleri, resimler, simge ve grafikler, tarih biçimleri gibi metin dışı unsurların uyarlanması da kapsar. Örneğin, yeşil renk, batı toplumlarında para ve şans çağrıştırırken, Çin’de genellikle sağlık ve esenlik kavramlarına işaret eder. Çin toplumunda zenginlik ve refahla özdeşleştirilen renk kırmızı iken, Afrika’nın bazı bölgelerinde kırmızı renk ölümü çağrıştırır. Bu nedenle sağlıkla ilgili mobil uygulamalar Çin’de yeşil temalı renkleri kullanırken, ABD merkezli uygulamalarda, mavi veya Kızıl Haç da çağrıştırdığı gerekçesiyle, kırmızı temalı renkler tercih edilir¹¹.

Yerelleştirme süreci, yalnızca dilsel uyarlama aşamalarını değil; biçimsel uyarlamaları ve metin dışı unsurlara ilişkin uyarlamaları da barındırır. Örneğin, Covid 19 salgını sürecinde hükümetler, halkın yeni tip koronavirüs hakkında bilgilendirilmesi, temaslı kişilerin takibinin yapılması, risklerin en aza indirilmesi, hastalığın yayılımının önlenmesi ve aşı kartlarının görüntülenebilmesi gibi amaçlarla mobil uygulamalar tasarlamıştır. Türkiye Cumhuriyeti Sağlık Bakanlığı tarafından geliştirilen *Hayat Eve Sığar* adlı mobil uygulamada yeşil renk baskın şekilde kullanılırken, Hindistan hükümeti tarafından tasarlanan *Aarogya Setu* adlı uygulamada turuncu renk, İngiltere ve Galler’de kullanılan *NHS Covid 19* adlı mobil uygulamada ise mavi renk tercih edilmiştir. Tüm bu uygulamalar benzer yazılım altyapısını kullansa da hükümetler hedef kitleye uygun tercihlerle çağrışmsal bir işlevi yerine getirmeye çalışmıştır:

		
<i>Hayat Eve Sığar</i> (Türkiye)	<i>Aarogya Setu</i> (Hindistan)	<i>NHS Covid 19</i> (İngiltere ve Galler)

Tablo 1. Farklı ülkelere ait Covid 19 mobil uygulamaları

Yerelleştirilecek bir ürün, yeni pazarlar için dilsel ve kültürel açılardan uyarlanacağı ön kabulüyle tasarlanır ve geliştirilir. Böylece, her bir yerel pazar için ürünün yeniden tasarlanması ve geliştirilmesi gerekmez. Bu adım uluslararasılaştırma olarak adlandırılır. Küreselleşme ise yerelleştirmesi yapılacak ürünler üreten bir firmanın tüm süreçlerini bu duruma göre ayarlaması anlamına gelmektedir (Pym, 2010, s. 122). Geleneksel yerelleştirme projelerinde “bir dilden bir dile” (en. *one-to-one language*) çeviri modeli kullanılıyordu. Ancak bu yöntemin zaman içinde yetersiz kaldığı ve oldukça maliyetli olduğu sonucuna varıldı (Pym, 2010, s. 121). Pazar boyutlarının büyümesi ve yerelleştirilmesi gereken içeriğin artması nedeniyle “bir dilden çok dile” (en. *one-to-many-languages*) çeviri modeli benimsenmeye başlandı. Bu sayede, ürünün uluslararasılaştırılmış sürümü eş zamanlı şekilde birden fazla erek dile çevrilmeye başlandı.

Anthony Pym (2010, s. 123) ek olarak iki farklı yerelleştirme türünden de söz eder. İlki kısmi yerelleştirme (en. *partial localization*). Bu türde, yalnızca kullanıcı tarafından görülecek olan dilsel

¹¹ <https://www.languageintelligence.com/post/software-localization-and-color-cultural-implications-of-color-choices-in-ui>

malzemenin yerelleřtirmesi yapılır ve böylece maliyetten tasarruf edilmesi amaçlanır. Dięer kavram olan tersine yerelleřtirme (en. *reverse localization*) ise, Türkçeden İngilizceye çeviri gibi, minör bir dilden majör bir dile gerçekteřtirilen yerelleřtirme iřlemlerini içerir.

Günümüzde yerelleřtirme kavramı kapsamında çeviri etkinlięinden sıklıkla söz edilmiř; çeviri odaklı bir bakıř açısıyla yerelleřtirme sürecini deęerlendiren önemli çalıřmalar yapılmıřtır. Ancak, farklı yazılım yerelleřtirme türleri içerisinde çevirmenin konumuna deęinen bir çalıřmaya rastlanmamıřtır. Yazılım yerelleřtirme türlerine iliřkin çalıřmalar, genellikle bilgisayar bilimciler ve yazılım uzmanları tarafından yürütölmektedir. Oysa, yerelleřtirmenin omurgasını çeviri etkinlięi oluřturduęundan, yazılım yerelleřtirme süreçlerinin çeviribilim odaklı bir yaklařımla incelenmesi ve deęerlendirilmesi, söz konusu iřlemlerin daha iřlevsel bir řekilde yürütölmeleri açısından oldukça önemlidir.

3. Yazılım yerelleřtirme türleri

3.1. řelale modeli

řelale modeli (en. *waterfall model*) geleneksel yerelleřtirme projelerinde uygulanan bir yöntemdir. Bu modelde, kodlama ařamasına geçilmeden önce tüm gereksinimler belirlenir ve yazılımın geliřtirilmesinden sonra yerelleřtirme adımına geçilir. Model, analiz, tasarım, kodlama, sına ve bakım ařamalarından oluřur. Her proje adımı doęrusal bir sırada gerçekteřir ve her bir adım bir önceki adımın çiktısı üzerine kuruludur.

řelale modelinde risk faktörleri (Hijazi, vd., 2012, s. 9), yazılım geliřtirme sürecinde gereksinimlerin deęiřmesi, ařamaların birbiriyle örtüřmemesi, yetersiz kalite güvencesi ve görece uzun süren ařamalar olarak özetlenebilir. Oldukça idealist bir yaklařımla tüm gereksinimlerin önceden belirlenmesi, yazılım projelerinin sürekli güncellenen ve istikrarsız yapısına ters düşebilir. Bu modelde, geliřtirme sürecinde bir adım tamamlanmadan dięerine geçilemedięinden ařamalar arasındaki baę kopabilir; ayrıca, belli bir ařamada görevli olan yazılım ekibi, çalıřmaya bařlayabilmek için bir önceki ařamanın tamamlanmasını beklemek zorundadır. Bu durum ise maliyetin yükselmesine neden olur. Test ařaması son evre olduęundan, sorun, hata ve riskler çok geç fark edilir. Bunların çözümlenmesi için ayrıca zaman, emek ve para harcanır. řelale modelinde, son ařamaya kadar çalıřır durumda olan bir ürün yoktur ve son ařamaya gelindięinde ortaya çıkan ürün son kullanıcının beklentilerini karřılamıyorsa, sorunları çözmek çok zordur.

řelale modelinin doęasına uygun olarak, ürünle ilgili çeviri iřlemleri de geliřtirme süreci tamamlandıktan sonra, yani ürün piyasaya sunulmadan hemen önce gerçekteřtirilir. Modelin aksayan yönleri yerelleřtirme sürecine de doęrudan yansımaktadır. Örneęin, İngilizce hazırlanmıř bir yazılımın, ürün olarak piyasaya sürölmeden önce gerçekteřtirilecek olan yerelleřtirme ařamasında, kaynak dil baęımlı geliřtirilen metin kutucuklarının geniřlikleri duraęan olarak belirlenmiř olabilir. Bu durumda, erek dil ifadesinin, kaynak dil ifadesinden daha uzun olması görsel açıdan sorun oluřurmaktadır. Arayüzde deęiřiklik yapılabilmesi için yazılım geliřtirme ařamasına tekrar dönölmesi gerekecek, bu durum ise maliyeti artıracak ve yazılım geliřtirme sürecinin iřlevsellięini olumsuz yönde etkileyecektir. Bu nedenle yazılım geliřtirme sürecinin tüm ařamalarında çeviri etkinlięinin de eř zamanlı olarak gerçekteřtirilmesi daha kaliteli, düşük maliyetli ve daha az emek yoğunluklu bir proje yapısının oluřmasına katkı saęlayabilecektir.

3.2. Çevik ve sürekli yerelleştirme

Günümüzde, yazılım geliştirmede çevik yaklaşımlar tercih edilmektedir. Bu modelde, ürün, geliştirme sürecinin tüm aşamalarında sınındığından, çevik yazılım geliştirme yöntemlerini kullanan şirketler gerekli değişiklikleri zamanında gerçekleştirme şansını yakalar ve herhangi bir sorunla karşılaştığında ilgili yazılım ekibine anında bilgilendirme yapabilir. Çevik yazılım geliştirme, yazılım geliştirmenin esneklik gerektiren bir süreç olduğu gerçeğinden yola çıkarak, geliştirici ekipler arasında her aşamada iş birliğinin sağlandığı bir bakış açısıyla sürdürülür. Bu yönüyle, çevik yazılım geliştirme süreci kitle kaynaklı bir yapıyı andırmaktadır.

2001 yılında yayınlanan çevik yazılım geliştirme manifestosu da bu saptamayı doğrular niteliktedir. Manifestoya göre (Fowler ve Highsmith, 2001), yazılım geliştirme sürecinde, süreçler ve araçlardan önce bireyler ve bireyler arası etkileşime odaklanılmalıdır ve yazılım sistemlerini geliştiren ekipler (proje yöneticileri, programcılar, sınıama uzmanları, vb.) arasında sağlıklı iletişimin sağlanması gerekmektedir. Kapsamlı belgeleme çalışmalarından ziyade “çalışan” bir yazılım hedeflenmelidir. Başka bir deyişle, ilk amaç sorunsuz bir şekilde çalışan bir yazılımın tasarlanması olmalıdır. Müşteriyle iş birliğinin altı çizilmeli, müşteriyle diyalog, sözleşme görüşmeleriyle sınırlı kalmamalıdır. Yazılım geliştiriciler, müşterinin istek ve gereksinimlerine göre kararlar alabilmelidir. Son olarak, proje başlangıcında oluşturulan bir plana sıkı sıkıya bağlı kalmaktan ziyade, değişime açık olunması gerekmektedir. Zaman içerisinde paydaşların öncelikleri değişebilir, teknoloji de süreç içinde gelişir ve değişir. “Değişimin” yazılım geliştirme süreçlerinin merkezinde yer aldığı unutulmamalıdır.

Yazılım geliştirme süreci, oldukça devingen ve devamlıdır. Bu nedenle günümüzde şelale modelinden ziyade çevik yerelleştirme (en. *agile localization*) yaklaşımları benimsenmektedir. Bu yöntemde, çevirmen sürekli olarak yerelleştirme döngüsü içerisinde yer alır; çeviriler son kullanıcının beklentileri ve yazılım güncellemeleri dikkate alınarak yenilikçi bir yaklaşımla gerçekleştirilir. Böylece yerelleştirme sürecindeki eş paydaşlar olan yazılım geliştiriciler ve çevirmenler arasında da iletişim ve denge sağlanmış olur.

Çevik yerelleştirmenin yapılacağı yerelleştirme yönetim sistemi, yazılımın kaynak kodlarının bulunduğu veri havuzu ile doğrudan bağlantılı olmalıdır. Yerelleştirilmesi gereken yeni bir dizgi kaynak koda eklendiğinde, eklenen bu dizgi yerelleştirme yönetim sistemine otomatik olarak aktarılmalıdır. Elsa Sklavounou’ya göre (2021), çevik yaklaşıma uygun bir yerelleştirme yönetim sisteminde:

- Uygun esneklik düzeyi sağlanır,
- Modern bir kullanıcı deneyimi sunulur,
- Kolay dosya yönetimi olanağı mevcuttur ve
- Çevirmenin gerçekleştirmiş olduğu çeviriyle denetleyicinin kararı aynı ekranda, yan yana şekilde görüntülenir.

Sürekli yerelleştirme (en. *continuous localization*) ise, yerelleştirmeyi mümkün olduğunca otomatik ve hatasız duruma getirmek için uygulanan bir dizi yöntem içerir. Çevik yerelleştirmede çeviri işi “sprint” adı verilen önceden belirlenmiş belirli zaman dilimlerinde gerçekleştirilirken, sürekli yerelleştirmede devamlı olarak, hız kesmeden yapılır. Örneğin, ilgili yazılıma yeni bir dizgi eklendiği anda, bu dizgi doğrudan bir bildirimle çevirmene iletir. Çeviride süreklilik olduğundan, çevrilen dizgi de doğrudan son kullanıcıya ulaştırılır.

Sürekli yerelleştirme sürecinin barındırdığı zorluklar kısaca şöyle özetlenebilir (Veveris, 2020):

<p>Adres RumeliDE Dil ve Edebiyat Araştırmaları Dergisi Osmanağa Mahallesi, Mürver Çiçeği Sokak, No:14/8 Kadıköy - İSTANBUL / TÜRKİYE 34714 e-posta: editor@rumelide.com tel: +90 505 7958124, +90 216 773 0 616</p>	<p>Address RumeliDE Journal of Language and Literature Studies Osmanağa Mahallesi, Mürver Çiçeği Sokak, No:14/8 Kadıköy - ISTANBUL / TURKEY 34714 e-mail: editor@rumelide.com, phone: +90 505 7958124, +90 216 773 0 616</p>
---	---

a. **Çeviride tutarlılığın ve doğruluğun sağlanması:** Bir yazılım ürününün küçük bir kesiti için işlevsel bir çeviri gerçekleştirmek oldukça zor bir iştir. Bu nedenle, çevirmenin ürün hakkında bilgi sahibi olması ve bağlamı dikkate alması önemlidir. Yazılım ekipleri de çevirmenlerin doğru çeviriler yapabilmesi için ürün planlamasının ne kadar önemli bir aşama olduğunun farkında olmalıdır. Küçük boyutlu, bağlamdan kopuk dizgi çevirileri süreci daha da karmaşık hale getirmektedir. Kitle kaynaklı çeviri söz konusu ise ya da projede serbest çevirmenler görev alıyorsa, ekibe yeni katılan her bir çevirmenin ürünü tanıması gerekmektedir.

b. **Çevirilerin ücretlendirilmesi:** Kısa içeriklerin çevirisinin nasıl ücretlendirileceği ile çevirmenin sistemdeki kotasının ne kadarını kullandığının belirlenmesi ve takip edilmesi oldukça zordur.

c. **Etkin ve işlevsel iş birliği:** Yerelleştirme sürecinde yer alan tüm paydaşlar birbirlerinin gerçekleştirdikleri işlemlerden haberdar olmalı, yapıcı iletişim yaklaşımını benimseyerek hedefe yönelik bir sinerji oluşturmalıdır.

d. **İş akışının sekteye uğramaması:** Yazılım geliştirme sürecindeki her ekibin iş akışı kendine özgüdür ve ekip çalışmalarının sekteye uğramaması için, kullanılacak yerelleştirme yönetim sisteminin ekibe uyumlu şekilde düzenlenmesi önemlidir.

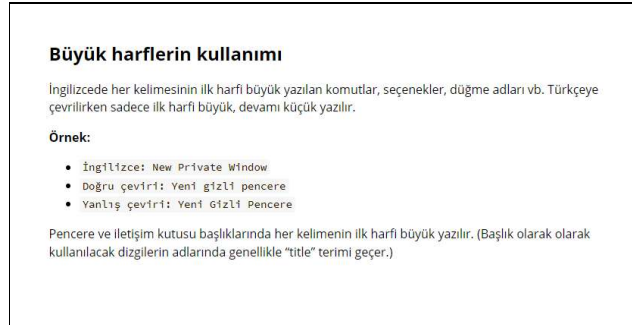
Günümüzde, çeviride kitle kaynak kullanımından faydalanan şirketlerin en sık kullandığı yaklaşımlar, çevik ve sürekli yerelleştirme modelleridir. Yazılım yerelleştirme projelerinde görev alan çevirmenler, çeviri süreçlerini desteklemek ve kolaylaştırmak için birtakım içerik ve araçlardan faydalanırlar.

4. Kitle kaynaklı yerelleştirmede çeviriye yardımcı araçlar

Yerelleştirme sürecinde, çevirmenler biçim kılavuzları, çeviri bellekleri, MÇ sistemleri, terimciler ve sistemde yer alan farklı dillerdeki erek metinlerden yararlanabilir.

4.1. Biçim kılavuzları

Biçim kılavuzları, bir yerelleştirme projesinde erek dile özgü yazım ilkelerini, terim birliğini sağlamak için hazırlanmış terimcileri, kullanılacak dil kesiti ile ilgili bileşenleri içeren, işveren şirket tarafından hazırlanmış yönergelerdir. Kitle kaynaklı çeviri sürecinde, çevirmenler biçim kılavuzlarından sıklıkla faydalanırlar.



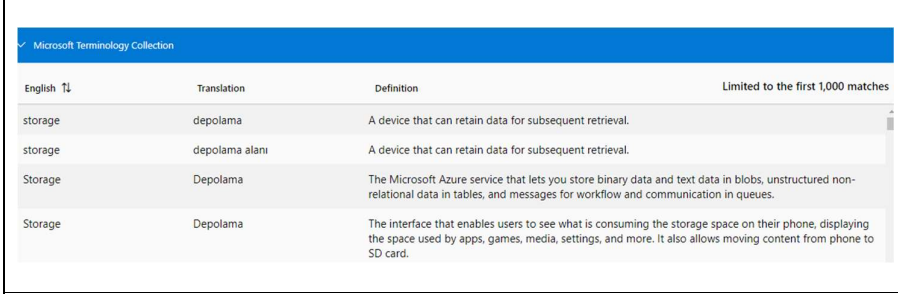
Resim 1. Biçim kılavuzunda yer alan bir yönerge¹²

Resim 1'de büyük harf kullanımına ilişkin bir yönerge görülmektedir. Anlaşıldığı üzere, İngilizce kaynak dizgide herhangi bir seçenek ya da komuta ait her bir sözcüğün baş harfi büyük yazılmıyken, erek dil Türkçede, bir söz öbeğinde yer alan sözcüklerden yalnızca ilkinin baş harfi büyük, takip eden sözcüklerin

¹² <https://mozilla-l10n.github.io/styleguides/tr/>

ise ilk harfleri küçük yazılmalıdır. Çevirmen bu konuda bir hata yaparsa, ilgili dizginin çevirisi erek dil editörü tarafından sistem üzerinden reddedilir ve gerekli düzelti yapıldıktan sonra çeviri onaylanır.

Yerleştirme sürecinde çevirmenler tarafından kullanılan dil bağımlı biçim kılavuzları, erek dile özgü biçimsel unsurların derlendiği kaynaklardır. Bu yönergeler hem biçimsel hem de dilsel açıdan daha erek odaklı çevirilerin yapılabilmesi için çevirmene destek olabilir. Örneğin, İngilizcede, dört basamaklı ve üzeri sayılar yazılırken, sondan başa doğru ilerlenerek, sayı üçlü gruplara ayrılır ve gruplar arasına virgöl konur¹³ (456,132 gibi). Türkçede ise, aynı durum söz konusu olduğunda virgöl yerine nokta kullanılır¹⁴ (2.309.876 gibi). Dolayısıyla, erek dile ait biçim yönergeleri göz önünde bulundurularak uyarlamaların yapılması gerekmektedir. Bazı biçim yönergeleri, çift dilli terimcileri de bünyesinde barındırır. Dizgi çevirileri yapılırken, terim düzeyinde tutarlığın sağlanması amacıyla terimcelere başvurularak daha akıcı ve ilgili şirketin dilsel standartlarına uygun bir erek metin oluşturulabilir. Resim 2’de *Microsoft Türkçe Biçim Kılavuzuna* entegre şekilde çalışan *Microsoft Dil Portalı*’nda yer alan terimceden örnek bir kesit sunulmuştur:



English	Translation	Definition
storage	depolama	A device that can retain data for subsequent retrieval.
storage	depolama alanı	A device that can retain data for subsequent retrieval.
Storage	Depolama	The Microsoft Azure service that lets you store binary data and text data in blobs, unstructured non-relational data in tables, and messages for workflow and communication in queues.
Storage	Depolama	The interface that enables users to see what is consuming the storage space on their phone, displaying the space used by apps, games, media, settings, and more. It also allows moving content from phone to SD card.

Resim 2. Çift dilli terimce örneği (*Microsoft Language Portal*)¹⁵

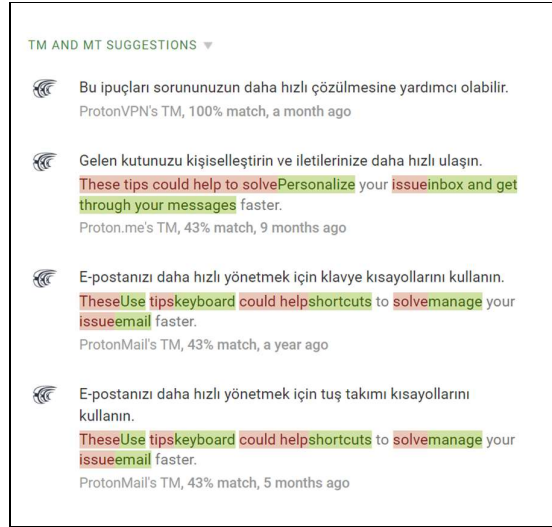
4.2. Çeviri bellekleri

Çeviri belleği, kaynak ve erek metin parçalarının (tümceler, paragraflar, metin kesitleri, vb.) hizalanmış şekilde depolandığı bir veri tabanıdır. Çeviri bellekleri, bilgisayar destekli çeviri araçlarının bir bileşeni olarak kabul edilir. İnsan çevirmenler tarafından çevrilen metin yığınları arttıkça, çeviri belleğinde depolanan paralel metin kesitlerinin miktarı da artar. Çeviri sürecinde, bellekte yer alan çeviri çözümleri çevirmene önerilir. Tam eşleşme (en. *exact match*), kısmi eşleşme (en. *fuzzy match*) ve herhangi bir eşleşmenin olmaması (en. *no match*) durumlarına göre çevirmen, belleğin önerdiği çeviriyi doğrudan onaylayabilir (tam eşleşme); uygun görülen karşılıklardan birini seçebilir veya bağlama göre yeni bir çeviri önerebilir (kısmi eşleşme) ya da bellek içinde söz konusu kaynak kesitin erek dilde herhangi bir karşılığı yoksa çevirisini yapabilir (eşleşme yok). Günümüzde, yerleştirme çalışmaları yürüten şirketlerin kendi çeviri bellekleri bulunmakta; bu bellekler sürekli güncellenmekte ve gelişmektedir. Çevirmenler de ilgili platformların çeviri belleklerinden yararlanabilmekte, aynı zamanda çevirileriyle belleğin gelişimine katkı sağlayabilmektedir. Resim 3’te bir çeviri belleği örneği görülmektedir:

¹³ Bkz. “Numerals, Fractions, and Dates”, <https://www.sussex.ac.uk/informatics/punctuation/misc/numeralsetal> ve “Expressing Numbers in English”, <https://www.thoughtco.com/expressing-numbers-in-english-1210097>

¹⁴ Bkz. “Noktalama İşaretleri (Açıklamalar)”, <https://www.tdk.gov.tr/icerik/yazim-kurallari/noktalama-isaretleri-aciklamalar/>

¹⁵ <https://www.microsoft.com/en-us/language/>



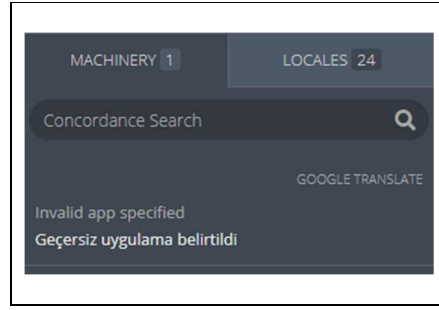
Resim 3. Çeviri belleği kesiti (Crowdin)¹⁶

4.3. Makine çevirisi sistemleri

MÇ sistemleri günümüzde çevirmenlerin iş istasyonlarında önemli bir yer tutmaktadır. İstatistiksel MÇ sistemlerinden sinir ağı tabanlı sistemlere geçilmesi bir paradigma değişimine neden olmuş, sinir ağı tabanlı yöntemlerle daha kaliteli MÇ çıktıları elde edilmeye başlanmıştır. Bu sistemler en gelişmiş MÇ yazılımları olarak kabul edilir ve kaynak metnin en olası çevirisinin belirlenmesi amacıyla yapay sinir ağlarının kullanıldığı bir çeviri yaklaşımını benimser.

Çok dilli sinir ağı tabanlı MÇ yaklaşımları son yıllarda geliştirilmeye başlanmış ve sıfır atımlı çeviri (en. *zero-shot translation*) fikri ortaya atılmıştır. Sıfır atımlı çeviri (Johnson, vd., 2017, ss. 339-351), sistemin eğitim materyalleri içerisinde hiç karşılaşmadığı dil kombinasyonları arasında çeviri yapabildiğini sağlayan bir modeldir. Örneğin, Portekizce-İngilizce ve İngilizce-İspanyolca dil çiftlerinde eğitilmiş çok dilli sinir ağı tabanlı bir sistem, Portekizce-İspanyolca dil çiftine ait herhangi bir veri ile karşılaşmamış olmasına rağmen, Portekizceden İspanyolcaya makul sayılabilecek çeviriler üretebilir. Sıfır atımlı çeviri yaklaşımında MÇ sistemindeki yapay sinir ağlarının öğrendiği modeller, sadece söz konusu kaynak ve erek dil için değil, sistem bünyesindeki tüm dil çiftleri için kullanılabilir. Bu nedenle, kural tabanlı ve istatistiksel yöntemlerle karşılaştırıldığında, sinir ağı tabanlı sistemler minör diller arasında daha kaliteli çeviriler gerçekleştirebildiği gibi, minör-majör diller arasında da daha iyi çeviriler yapabilmektedir. Sinir ağı tabanlı sistemlerin gelişimi sayesinde, özellikle teknik alanda (yazılım dizgi çevirileri gibi) MÇ sistemlerinden daha elverişli şekilde faydalanılabilmektedir. Kitle kaynaklı yerelleştirme projelerinde de Resim 4'te görüldüğü üzere, çeviri sürecinde MÇ çıktıları kullanılabilir.

¹⁶ <https://crowdin.com/>



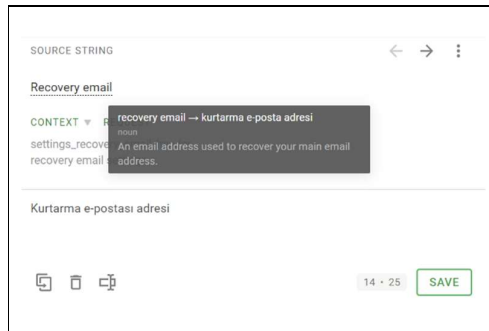
Resim 4. MÇ çıktısı (Mozilla Pontoon)¹⁷

4.4. Terimler

Özel alan çevirileri (teknik, tıp, hukuk, vb.) yapan çevirmenler, tek dilli, çift dilli ve çok dilli sözlükler, terimler ve terim bankalarından sıklıkla yararlanır. Yazılım yerelleştirme projelerinde de uzmanlık alanı terminolojisi, alana özgü dil, teknik terimler ve profesyonel bir dil kesiti ile karşılaşılır. Günümüzde, özellikle açık kaynaklı yazılım şirketleri olmak üzere, birçok yazılım şirketi kitle kaynaklı çeviri hizmeti almaktadır. Çoğunlukla gönüllü amatörler olarak yerelleştirme projelerinde yer alan çevirmenlerin sayısı da oldukça fazla olabilmektedir. Aynı dil ekibinde çalışan çevirmenler arasında her zaman sağlıklı bir iletişim sağlanamamaktadır. Bu nedenle, çevrilen binlerce dizgi arasında terim tutarlılığının sağlanabilmesi için özellikle çift dilli terimcelere gereksinim duyulmaktadır. Çift dilli terimlerde sorgulanan kavrama ilişkin tanımlamaların yapılması da çevirmenin bağlamı daha iyi anlamasını; böylece daha erek odaklı bir çeviri yapmasını sağlayabilmektedir.

Bilgen'in de ifade ettiği şekilde (2016, s. 18), kullanım kılavuzları ve destek belgelerinde olduğu gibi, bir yazılım arayüzünde yer alan terimler de kullanıcı deneyimini doğrudan etkileyen bileşenlerdir. Bir yazılım ürününün veya web sitesinin yerleştirilmesinde terminoloji tutarlılığının sağlanması çevirmenlerin birincil önceliğidir.

Resim 5'te, bir yerelleştirme yönetim sisteminde bulunan terimceden bir kesit sunulmuştur. Örnekte İngilizce *recovery email* teriminin, "kurtarma e-posta adresi" şeklindeki Türkçe karşılığı, sözdizimsel kategorisi ve terim açıklaması verilmiştir:



Resim 5. Terimce örneği (Crowdin)¹⁸

¹⁷ <https://pontoon.mozilla.org/tr/>

¹⁸ <https://crowdin.com/>

4.5. Farklı dillere yapılmış dizgi çevirileri

Yerelleştirme sürecinde aynı kaynak dizginin farklı erek dillere yapılmış çevirilerinden yararlanılabilir. Sistem içerisinde bu unsurlar “yerel dil”¹⁹ ya da “kullanıcı dili” (en. *locale*) olarak adlandırılır. Yerelleştirme bağlamında, yerel dil, “kullanıcının dilini, ülkesini ve tercihlerini tanımlamak için kullanılan bir dizi parametreyi içeren bir kavramdır” (Sandrini, 2008, s. 168). Başka bir deyişle, dilin ve ilgili coğrafi bölgenin tüm kültürel unsurlarının birleşimidir. Anthony Pym, “yerel dil”i, son kullanıcıya ulaşan ürüne ilişkin bir dizi dilsel, ekonomik ve kültürel parametrenin kombinasyonu olarak tanımlamıştır (2010, s. 121). Son kullanıcının yaşadığı bölgeye ait dilsel ve biçimsel unsurlar göz önüne alınarak, tarihlerin, sayıların, ölçü ve para birimlerinin çevrilmesi başarılı bir yerelleştirme olarak kabul edilebilir. Örneğin, İngilizce *Microsoft Office* yazılımında “Ctrl+B” kısayol komutu, seçilen metin parçasının kalın yazılmış biçime dönüştürülmesini sağlarken; yazılımın Türkçe sürümünde bu kısayol “Ctrl+K” olarak kullanılmaktadır. İngilizce komutta “B” harfi *bold* (tr. kalın) sözcüğünün kısaltılmış haliyken, Türkçe kısayolda bu komuta “K” harfi karşılık gelmektedir. Bu tarz yerelleştirmeler kullanıcı deneyiminin eniyileştirilmesi için önemlidir.

Farklı erek dillerden İngilizceye geri çeviri yaptırılarak, başka dillerdeki erek metinlerde yer alan dilsel seçimlerden esinlenilebilir. Özellikle çeviride yaratıcılık²⁰ gerektiren durumlarda, geri çeviriye başvurulur. Tablo 2’de bu durum örneklenmeye çalışılmıştır:

1	İngilizce kaynak dizgi	Download Firefox themes that feature the environment and the natural world. This category includes sunsets, beaches, illustrations, city skylines, and more.
2	Kaynak dizginin Türkçe MÇ çıktısı	Çevreyi ve doğal dünyayı öne çıkaran Firefox temalarını indirin. Bu kategori gün batımlarını, sahilleri, illüstrasyonları, şehir silüetlerini ve daha fazlasını içerir.
3	İtalyanca erek dizgi (kitle kaynaklı çeviri belleği)	Scarica temi per Firefox ispirati all’ambiente e al mondo della natura. In questa categoria troverai tramonti, spiagge, illustrazioni, orizzonti urbani e altri panorami mozzafiato.
4	İtalyanca erek dizginin İngilizceye geri çevirisi (MÇ aracılığıyla)	Download themes for Firefox inspired by the environment and the world of nature. In this category you will find sunsets, beaches, illustrations, urban horizons and other breathtaking views.
5	Türkçe erek dizgi	Çevreden ve doğadan ilham alan Firefox temalarını indirin. Bu kategoride, gün batımı, sahiller, çizimler, kent silüetleri ve çok daha fazlasını bulacaksınız.

Tablo 2. Çeviri sürecinde farklı dillere yapılan çevirilerden yararlanılması (*Mozilla Pontoon*)²¹

Tablo 2’yi incelediğimizde, ilk satırda İngilizce kaynak dizgi görülmektedir. İkinci satırda kaynak dizginin MÇ çıktısı yer almakta iken, üçüncü satırda İtalyanca erek dizgi bulunmaktadır. Dördüncü satırdaki İtalyancadan İngilizceye yapılan geri çeviriye baktığımızda *feature* (tr. karakterize etmek, belirtmek) kavramının *inspired by* (tr. ilham almak); *this category includes* (tr. bu kategori ... içerir)

¹⁹ *Locale* (en) sözcüğü için Türkçede “yerel dil” ya da “yerel ayar” karşılıkları önerilmiştir (bkz. Microsoft Language Portal, <https://www.microsoft.com/en-us/language/>).

²⁰ Çeviribilim alanında görece yeni bir kavram olan “yaratıcı çeviri” (en. *transcreation*), genellikle pazarlama içeriklerinin ve oyunların yerelleştirilmesiyle özdeşleştirilir. Çeviri sürecinde, kültürel unsurların göz önünde bulundurulması, son kullanıcının duygularına hitap edilmesi, anlamsal içerik korunurken, hedef kitleye yönelik birtakım uyarlamaların yapılması yaratıcı çeviri yaklaşımı kapsamında değerlendirilebilir. Yazılım yerelleştirme sürecinde de son kullanıcının beklentilerinin dikkate alınması gerektiğinden, çevirmenin “yaratıcı” kararlar alması, hedef kitlenin ortaya çıkan ürünü sorunsuz bir şekilde kullanabilmesi açısından önem taşımaktadır.

²¹ <https://pontoon.mozilla.org/tr/>

kısının ise, *in this category, you will find* (tr. bu kategoride ... bulacaksınız) şeklinde daha erek odaklı ve çağrışsal bir biçimde erek dile aktarıldığını görmekteyiz.

Türkçe çeviri ekibi, İtalyanca erek metinde karşılaşılan bu dilsel tercihleri, Türkçe erek dizgiye yansıtabilir; bu durumda Türkçede, “Çevreden ve doğadan ilham alan Firefox temalarını indirin. Bu kategoride, gün batımı, sahiller, çizimler, kent silüetleri ve çok daha fazlasını bulacaksınız” şeklinde daha işlevsel bir çeviri ortaya çıkacaktır.

5. Kitle kaynaklı yerelleştirme projelerinde iş akışı: Mozilla Pontoon örneği

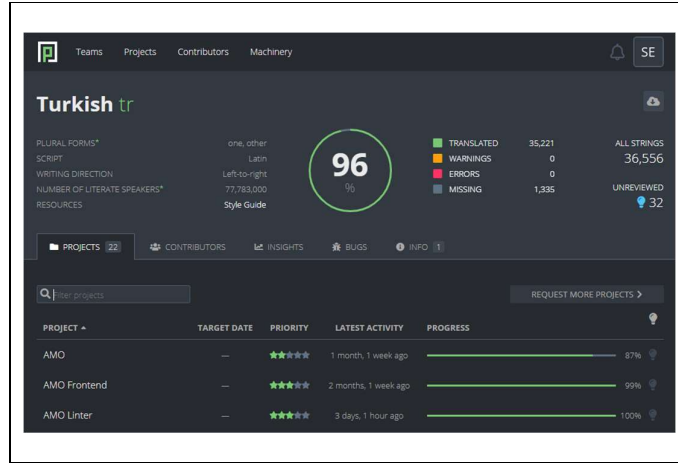
Bu bölümde, kitle kaynaklı yerelleştirme projelerinde iş akışından *Mozilla Pontoon* özelinde söz edilecektir. *Mozilla*, *Netscape* adlı bilişim şirketinin elemanları tarafından 1998 yılında kurulan özgür bir yazılım topluluğudur²². Topluluk, kurumsal düzeyde, kâr amacı gütmeyen bir kuruluş olan *Mozilla Vakfı* tarafından desteklenir. *Mozilla* topluluğu, açık kaynaklı yazılımlar²³ geliştirir. *Firefox* web tarayıcı ve *Thunderbird* e-posta uygulaması, *Mozilla* tarafından geliştirilen yazılımlara örnek olarak verilebilir. Topluluğun temel amacı, ayırım olmaksızın interneti herkes için açık ve erişilebilir kılmaktır.

*Pontoon*²⁴, *Mozilla* yerelleştirme ekibi tarafından kullanılan ve sürekli geliştirilen bir yerelleştirme yönetim sistemidir. Topluluk tarafından açık kaynaklı yerelleştirme yapılı ve çevirilerin depolanması için sürüm kontrol sistemleri kullanılır. Halihazırda, *Pontoon*'da 240 dil ekibi yer almaktadır. Dil ekipleri, gönüllü çevirmenler ve editörlerden oluşur. Geliştirilen yazılımların yerelleştirmelerini, gönüllü çevirmenlerden oluşan *Mozilla* yerelleştirme topluluğu yapar. Kullanıcı sisteme üye olduktan sonra kendi anadil ekibine kaydolar ve “eksik” olarak belirtilen dizgilerin çevirisini yapmaya başlar. Resim 6'da Türkçeye ilişkin veriler görülmektedir:

²² Stallman'a (2009: 23) göre, bir programın özgür bir yazılım olabilmesi için: (1) herhangi bir amaçla programı çalışma özgürlüğüne sahip olunması, (2) ihtiyaçlara uygun şekilde programı değiştirme özgürlüğüne sahip olunması, (3) kopyaları ücretsiz olarak ya da belli bir ücret karşılığında yeniden dağıtma özgürlüğüne sahip olunması, (4) toplumun ilgili gelişmelerden faydalanmasını sağlamak için, programın değiştirilmiş sürümlerini dağıtma özgürlüğüne sahip olunması gerekmektedir.

²³ Geliştirme modellerine ya da lisanslarına göre, yazılımlar temelde iki kategoride sınıflandırılabilir: Ticari yazılımlar ve açık kaynaklı yazılımlar. Bu ayırım, kaynak kodunun erişilebilirlik düzeyiyle ilgilidir. Kaynak kodu, aslında çoğu bilgisayar kullanıcısının hiç görmediği bir yazılım parçasıdır; programcılar bu kod üzerinde oynama yaparak bir yazılım bileşeni ya da uygulamanın şeklini değiştirebilirler. Ticari yazılımlar, bir şirkete ait olan ve şirketin fikri mülkiyetinin korunması için kaynak koda erişimin kısıtlandığı yazılımlardır. Kodun kullanımı, dağıtımı veya değiştirilmesi yasaktır ya da kısmen kısıtlanmıştır. *Microsoft Office* bu türe örnek olarak verilebilir. Açık kaynaklı yazılımlarda ise, kaynak kod herkes tarafından incelenebilir, değiştirilebilir, iyileştirilebilir ya da geliştirilebilir. Örneğin, *Libre Office* açık kaynaklı bir yazılımdır. Günümüzde, kitle kaynaklı yerelleştirme projeleri kapsamında gerek ticari gerekse de açık kaynaklı yazılım unsurları birçok dile çevrilir.

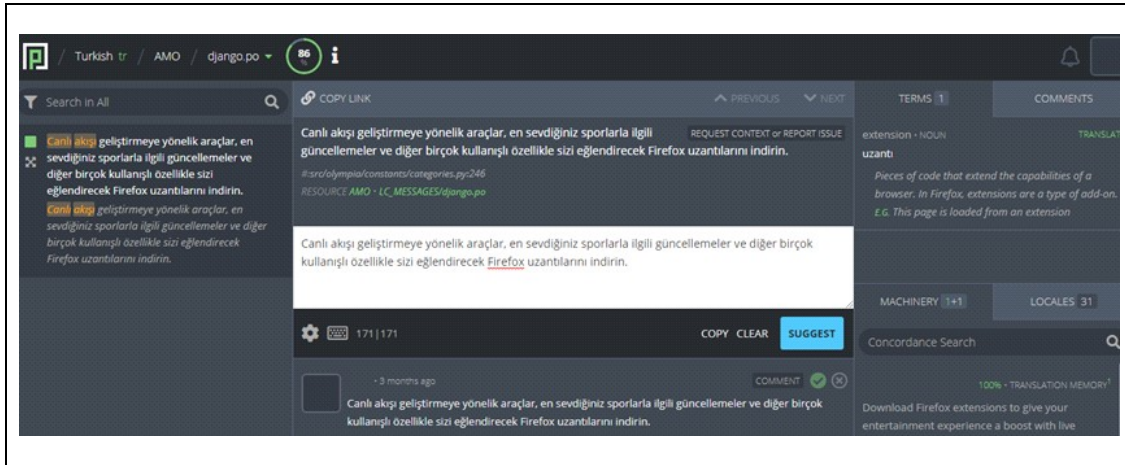
²⁴ <https://pontoon.mozilla.org/>



Resim 6. Mozilla Pontoon Türkçe verileri

Resim 6'da yer alan ekran görüntüsünü kısaca yorumlayacak olursak, İngilizce kaynak dizgilerin %96'sının halihazırda Türkçeye çevrilmiş olduğu görülmektedir. Ancak bu oran, durağan değildir, bir yandan bazı projelere ilişkin çeviriler tamamlanırken; diğer yandan yeni projeler açılmakta ve devingen çeviri süreci hız kesmeden sürmektedir. Bu bağlamda “proje önceliğinden” (en. *priority*) de söz etmek gerekir. Bazı projeler, içerdiği dizgi çevirileri tamamlandığında sonlandırılırken, bazı projeler için belirlenmiş bir son teslim tarihi bulunmaktadır. Süreli projelere katılım sağlayan çevirmenler proje bitiş tarihini dikkate almalıdır.

Çevirmen, seçtiği projeye dahil olarak kaynak dizgilerin çevirilerini yapmaya başlar. Çeviri esnasında, 4. Bölümde sözü edilen çeviri belleği ve MÇ sistemleri gibi eklentilerden faydalanabilir. Gönüllü çevirmenler tarafından dizgi dizgi çevrilen içerik yazılımın dilsel içeriğini oluşturur. Çeviri yapıldıktan sonra ilgili dil editörü, çeviriyi kabul edebilir, reddedebilir ya da çeviriye ilişkin yorum yapabilir. *Mozilla Pontoon* Türkçe ekibi, halihazırda gönüllü 100 çevirmen ve iki editörden oluşmaktadır. Resim 7'de *Mozilla Pontoon* platformundan alınmış bir çeviri ekranı görülmektedir:

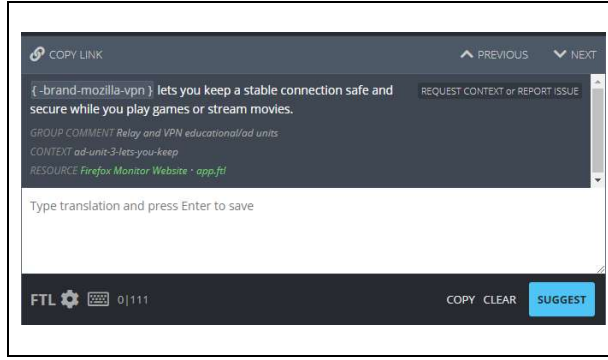


Resim 7. Çeviri ekranı

Resim 7'den de anlaşıldığı üzere, *Pontoon* platformu kullanıcı arayüzü açısından oldukça “çevirmen dostu” bir yapıda tasarlanmıştır. Katılım sağlanan proje içerisinde çevirisi henüz yapılmamış olan bir

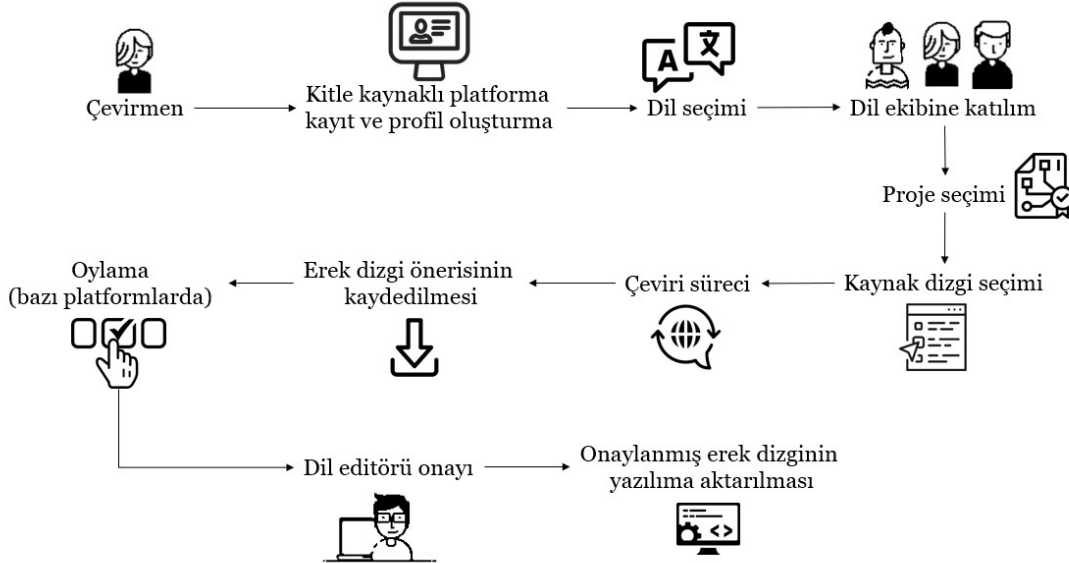
dizgi seçilerek çevrilir. Kaynak dizgi ve çeviri alanı yanında, tüm yardımcı araçlar tek bir arayüzde, bir arada görüntülenebilmektedir. Bu durum, çevirmene oldukça rahat bir çalışma ortamı sunmaktadır. Dizgi çevirisi ilgili dil editörü tarafından onaylandığında, platformun çeviri belleğine aktarılır. Çevirmen aynı zamanda sistemdeki diğer çevirmenlerin kabul edilmiş ya da reddedilmiş çevirilerini görüntüleyebilir.

Kitle kaynaklı yerleştirme projelerinde çevirmeni en çok zorlayan unsur *placeholder* adı verilen yer tutuculardır. Yer tutucular, kodun içerisindeki değişken ifadelerinin, yerleştirilecek dizgilerde kullanımını sağlayan metin parçalarıdır. Erek dizgi düzenlenirken yer tutucuların kaldırılmaması ya da bozulmaması gerekir. Aksi durumda, yazılımda hatalar ortaya çıkabilir. Resim 8’de bir yer tutucu örneği görülmektedir:



Resim 8. Yer tutucu örneği ({-brand-mozilla-vpn})

Kitle kaynaklı bir yerleştirme projesinde çeviri odaklı örnek bir iş akışı Şekil 1’de verilmiştir:



Şekil 1. Kitle kaynaklı yerleştirmede iş akışı

Sonuç ve deđerlendirme

Günümüzde hem ticari hem de açık kaynaklı yazılım yerelleřtirme süreçlerinde kitle kaynaklı çeviriye başvurulmaktadır. Yazılım yerelleřtirme, biliřim alanında *string* adı verilen dizgilerin çevrilmesini içerir. Yerelleřtirme projelerine gönüllü olarak katılım sađlayan çevirmenler kaynak dildeki dizgileri, bařta kendi anadilleri olmak üzere, bildikleri dillere çevirirler. Dizgi çevirileri, erek dil editörü tarafından gözden geçirilir. Yazılım yerelleřtirmede çevirmen ve/veya editör olarak görev alan gönüllülerin çabalarıyla, ilgili yazılım ürünü son kullanıcıya kendi anadilinde sunulmuş olur. Dolayısıyla, yazılım yerelleřtirmenin nihai amacının son kullanıcının kullanıcı deneyimini iyileřtirmek ve kullanıcının üründen beklentilerini karřılamak olduđunu söylememiz yanlış olmaz. Bu yönüyle, yazılım yerelleřtirme süreci, erek odaklı çeviri yaklařımlarına iliřkin parametrelerle özdeřtir.

Çevik ve sürekli yerelleřtirme yöntemleriyle ilerleyen projelerde, yazılım ekipleri gibi, çevirmenler ve editörlerden oluřan dil ekipleri devamlı olarak yerelleřtirme döngüsünde yer alır. Örneđin, sistemde yazılım üzerinde herhangi bir deđiřiklik gerçekteřtiđinde, çevirilerde de gerekli güncellemeler yapılabilir, ya da eklenen kaynak dizgiler varsa, bu dizgilerin çevirileri anlık olarak gerçekteřtirilebilir. Böylece, oldukça canlı ve devingen bir süreç olan yazılım geliřtirmenin tüm ařamalarına çevirmenler de dahil olur. Yazılım yerelleřtirme, ilgili yazılım ürününün son kullanıcının hem diline hem de kültürüne yönelik olarak uyarlanması anlamına geldiđinden, yerelleřtirme sürecinin yalnızca teknik çeviriden ibaret olduđu düşünülemez. Bađlama ve son kullanıcının özelliklerine göre çevirmenin yaratıcı kararlar alması gerekebilir, dolayısıyla işlevsel bir çeviri yaklařımı belirlemek önem taşımaktadır.

Çevirmenler, genellikle ilgili yazılım řirketleri tarafından oluřturulmuş olan yerelleřtirme yönetim sistemleri üzerinden, çevrimiçi olarak yazılım yerelleřtirme projelerine katılım sađlarlar. Kitle kaynaklı çeviriye açık olan bu platformlar çeviri süreçlerinde kullanabilecekleri bir araç kitini de çevirmenlere sunar. Bu araç kitinde, biçim kılavuzları, terimceler, řirketin çeviri belleđi, aynı kaynak dizginin farklı dillere daha önceden yapılmış olan çevirileri ve entegre MÇ sistemleri yer almaktadır. Çeviri yönetim platformlarında yer alan projeler sürekli güncellenmekte; bir yandan bazı projeler sonuçlanırken, diđer yandan yeni yazılım projeleri açılmaktadır. Geleneksel çeviri süreçlerinin aksine, yazılım yerelleřtirmede görev alan çevirmenler, sistemin kendilerine sunduđu dizgileri çevirir. Bu dizgiler, yazılım arayüzü bileřenlerinden gelebileceđi gibi, yazılımla bađlantılı çoklu ortam (multimedya) dosya veya bileřenlerinden de gelebilir.

Bu çalıřma kapsamında, *Mozilla* yazılım topluluđu tarafından geliřtirilmiş bir yerelleřtirme yönetim sistemi olan *Mozilla Pontoon* platformu incelenmiştir. Kitle kaynaklı çeviriden yararlanan platformda birçok dil ve dil deđiřkesinde çevirmen ve düzeltmenlerden oluřan ekipler yer almaktadır. Ekip üyeleri gerek kendi ekipleri içerisinde gerekse de bařka dil ekiplerinde yer alan katılımcılarla iş birliđi içinde gönüllü olarak çalıřırlar. *Mozilla Pontoon* platformu, kitle kaynaklı yerelleřtirme projelerinin mantıđını, işleyiřini ve motivasyonunu anlamak isteyen arařtırmacılara, çeviribilim öğrencilerine ve çevirmenlere oldukça verimli bir alan sunmaktadır. Zira kitle kaynaklı yerelleřtirme süreci içerisinde yer almadan söz konusu projelerdeki iş akıřım kavramak oldukça zordur.

Kitle kaynaklı yerelleřtirme projelerinde yer alan profesyonel çevirmen sayısının görece az olması dikkat çekicidir. Projelerde genellikle yazılım ve biliřim alanlarından uzmanlar, çevirmen ya da düzeltmen olarak görev almaktadır. Bu kişiler yazılım yerelleřtirme sürecinde disiplinli bir řekilde çalıřmaktadır; ancak dil, kültür ve iletiřim uzmanı olan profesyonel çevirmenlerin yazılım yerelleřtirme projelerinde daha etkin olarak yer alması, işleyiře oldukça olumlu katkılar sunabilecektir. Kitle kaynaklı yerelleřtirme

platformları, çeviribilim ve mütercim tercümanlık bölümü öğrencilerinin hem genel çeviri becerilerini hem de özel alan ve terminoloji bilgilerini geliştirmelerine yardımcı olabilir. Aynı zamanda, bu platformlar öğrencilerin araçsal becerilerinin de geliştirilmesi için fırsatlar sunmaktadır. Gerçek çeviri ortamlarında çeviri yapmak da öğrencilerin profesyonel hayata hazırlanması açısından önemlidir.

Kaynakça

- Anastasiou, D., & Gupta, R. (2011). Comparison of crowdsourcing translation with machine translation. *Journal of Information Science*, 37(6), 637-659.
- Bilgen, B. (2016). *Localization and Terminometrics: Measuring the impact of user involvement on terminology* (Yayımlanmamış Doktora Tezi). Ottawa Üniversitesi, Kanada.
- Brahman, D. C. (2008). Crowdsourcing as a model for problem solving: An introduction and cases. *Convergence*, 14(1), 75-90.
- Collins, R. W. (2001). Software Localization: Issues and methods. *Proceedings of the 9th European Conference on Information Systems* içinde, (ss. 36-44).
- Esselink, B. (2000). *A practical guide to software localization*. Amsterdam: John Benjamins.
- Esselink, B. (2003). The evolution of localization. *The Guide from Multilingual Computing & Technology: Localization*, 14(5), 4-7.
- Estellés-Arolas, E., & González-Ladrón-de-Guevara, F. (2012). Towards an integrated crowdsourcing definition. *Journal of Information Science*, 38(2), 189-200.
- Estellés-Arolas, E., Navarro-Giner, R., & González-Ladrón-de-Guevara, F. (2015). Crowdsourcing Fundamentals: Definition and typology. *Advances in Crowdsourcing* içinde, (ss. 33-48). Springer International Publishing.
- European Commission (2012). *Crowdsourcing Translation: Studies on multilingualism and translation*: Brüksel ve Lüksemburg: Directorate-General for Translation.
- Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development*, 9(8), 28-35.
- Hijazi, H., Khdour, T., & Alarabeyyat, A. (2012). A review of risk management in different software development methodologies. *International Journal of Computer Applications*, 45(7), 8-12.
- Howe, J. (2006). The rise of crowdsourcing. *Wired*, 14 (6), Erişim Tarihi: 15 Haziran 2022, <https://www.wired.com/2006/06/crowds/>.
- Jiménez-Crespo, M. (2017). *Crowdsourcing and online collaborative translations*. Amsterdam: John Benjamins.
- Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., Hughes, M., & Dean, J. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5, 339-351.
- Pulcino, A. (t.y.). Software Localization and Color: Cultural implications of color choices in UI. Erişim Tarihi: 15 Haziran 2022, <https://www.languageintelligence.com/post/software-localization-and-color-cultural-implications-of-color-choices-in-ui>
- Pym, A. (2010). *Translation theories explained. Londra ve New York: Routledge*.
- Sandrini, P. (2008). Localization and translation. *MuTra Journal*, 2, 167-191.
- Sklavounou, E. (2021). What a future talent would say about translation automation?. *Proceedings of the Translation and Interpreting Technology Online Conference* içinde, (ss. 1-6).
- Stallman, R. M. (2009). *Özgür yazılım, özgür toplum, Richard M. Stallman'ın seçme yazıları*. Serkan Çapkan, İzlem Gözükeleş, Tahir Emre Kalaycı, Çiğdem Özşar, Birkan Sarıfakıoğlu (Çev.), Ankara: TMMOB Elektrik Mühendisleri Odası.
- Surowiecki, J. (2005). *The wisdom of crowds*. New York: Anchor Books.

Veveeris, E. (2020). Continuous Localization 101: What it is and when it makes sense. Eriřim Tarihi: 15 Haziran 2022, <https://lokalise.com/blog/continuous-localization-101/>.