

ÖZGÜR CBS YAZILIMI GELİŞTİRME: QUANTUM GIS ÖRNEĞİ

Volkan Osman KEPOĞLU¹, Şebnem DÜZGÜN²

¹ ODTÜ, Fen Bilimleri Fakültesi, Jeodezi ve Coğrafi Bilgi Teknolojileri Ana Bilim Dalı, ykepoglu@gmail.com,

² ODTÜ, Mühendislik Fakültesi, Maden Mühendisliği, ve Fen Bilimleri Fakültesi, Jeodezi ve Coğrafi Bilgi Teknolojileri Ana Bilim Dalı, duzgun@metu.edu.tr

ÖZET

Özgür yazılımlar, tüm kullanıcılara, yazılımı istediği gibi çalıştırma, dağıtma, değiştirme ve değişmiş halini tekrar dağıtma özgürlüğü sağlamaktadır. Özgür yazılım hareketi, yazılım geliştirme metodunu topluma yayma ve bu işi toplumsal bir üretim biçimi haline getirmeye çalışmaktadır. Bu hareket o kadar hızlı ve başarılı bir şekilde yürütülmektedir ki 20 Eylül 1983 yılında duyurulan özgür yazılım hareketi aradan geçen 25 yıllık geliştirmeler ile yeni bir ekonomi oluşturmıştır. Günümüzde bu gelişim sayesinde özgür yazılımlar, sunucularda, kişisel sistemlerde, gömülü sistemlerde, internet ve veritabanı uygulamalarında, mobil cihazlarda vb. uygulama alanlarında geniş bir yelpazede ürün çeşitliliğine ulaşmıştır. Bu farklı uygulama alanlarından biri de Coğrafi Bilgi Sistemleridir. Bu bildirinin amacı, masaüstü Coğrafi Bilgi Sistemi (CBS) dalında geliştirilmekte olan özgür yazılımlardan biri olan Quantum GIS (QGIS) üzerinden özgür CBS yazılımlardaki geliştirmelerin nasıl yapılabileceğini göstermektir.

Bu bildiride özgür yazılımın tanımı ve özgür yazılım kullanımından doğan avantajlar ile dezavantajlar hakkında bilgi verildikten sonra özgür yazılım geliştirmenin yolları aktarılacaktır. Daha sonra Özgür CBS yazılımlarından biri olan QGIS üzerinde bir geliştirmenin nasıl yapılacağı anlatılacaktır. Bu geliştirme yeni bir CBS yazılımı geliştirmek olabileceği gibi QGIS arayüzü içinde yeni bir eklenti yazmak da mümkün olabilmektedir. Bu yazılım örnekleri, QGIS ve Qt yazılım geliştirme ortamını kullanmakta olup C++ veya Python programlama dili kullanılarak geliştirilebilmektedir. Python dilinin hızlı yazılım geliştirme stratejisine uygunluğu nedeniyle bildiride anlatılan yeni geliştirmeler Python dili kullanılarak yapılmıştır. Sonuçta, bildiride özgür bir CBS yazılımı geliştirme imkanının olduğu ve geliştirmeyi kişisel olarak sahiplenmek için hangi koşulların oluşması gerektiği özetlenmekte ve QGIS örneğinde geliştirmenin nasıl yapılabileceği gösterilmektedir.

Anahtar Sözcükler: Özgür Yazılım, Coğrafi Bilgi Sistemi, Özgür CBS Yazılımı Geliştirilmesi, Quantum GIS, Python

ABSTRACT

DEVELOPMENT OF FREE GEOGRAPHIC INFORMATION SYSTEM SOFTWARE: IN THE EXAMPLE OF QUANTUM GIS

Free software provides for all users the rights to run the software for any purpose, to distribute the copies, to adapt the software and to redistribute the modified versions. Free software movement tries to spread the software development method to public and to adopt this work as a form of public production. This movement has implemented so much fast and successful that after the announcement of free software movement in September, 20 1983 all free source developments in 25 years have established a new economy. Today free software industry has reached enormous variety in the applications of server, personnel computer, embedded system, internet, database, mobile device, etc. Geographic Information System (GIS) is one of different application fields that a lot of free software is developed. Quantum GIS (QGIS) is one of the free desktop GIS software that is chosen to show that how free GIS software can be developed in terms of a new software and a new plug-in.

In this article, the definition of free software and the advantages of using free software are explained. The way of developing free GIS software in the example of QGIS is demonstrated. This can be done by developing a new GIS software or by a new plug-in in the existing interface of QGIS. These new development studies use QGIS and Qt software development frameworks and can be developed with either using C++ or Python programming language. Python is used as the primary development language for the sample applications shown in the article due to the benefits of Python for the rapid application development strategy. As a conclusion, in this article, the opportunity of developing a free GIS software and the conditions in order to own the development of such a free software are summarized and how a GIS application can be developed is shown in the example of QGIS.

Keywords: Free Software, Geographic Information System, Development of Free GIS Software, Quantum GIS, Python

1. ÖZGÜR YAZILIM

Özgür Yazılım Vakfı'nın (Free Source Foundation) kurucusu Richard Stallman "özgür yazılım" kavramını tanımlarken dört tip özgürlükten bahsetmektedir (Free Software Foundation 2008)(Altunergil 2003);

Özgürlük-0. Her türlü amaç için yazılımı çalıştırma özgürlüğü

Özgürlük-1. Yazılımın nasıl çalıştığını inceleme ve kendi gereksinimleri doğrultusunda değiştirme özgürlüğü

Özgürlük-2. Yeniden dağıtma ve toplumla paylaşma özgürlüğü

Özgürlük-3. Yazılımı geliştirme ve gelişmiş haliyle topluma dağıtma özgürlüğü

Özgürlük 1 ve 3 için yazılımın kaynak koduna erişim bir ön şarttır. Özgürlük 0 sayesinde yazılım istenen sayıda bilgisayara kurulum çalıştırılabilmektedir. Özgür Yazılım Vakfı, bahsi geçen özgürlüklerin devam ettirilebilmesi için türev olarak üretilen veya değiştirilen yazılımın lisansı önceki yazılımın lisans ile aynı olması gerektiğini vurgulamaktadır. Bu zorunluluk telif hakkından vazgeçildiği (copyleft) anlamına gelmektedir. Özgürlük 2 de ise yazılım dağıtılırken yazılımın lisansı kullanıcıya açıkça belirtilmesi gerektiği ve yazılımın kaynak kodunun da yayınlanmasının bir zorunluluk olduğu ifade edilmektedir.

Kısaca, özgür yazılımlar, tüm kullanıcılara, yazılımı istediği gibi çalıştırma, dağıtma, değiştirme ve değişmiş halini tekrar dağıtma özgürlüğü sağlamaktadır. Bu bildiride özgür yazılım kavramı, Stallman'ın tanımını ve GNU Genel Kamu Lisansı (GPL) ile lisanslanmış ürünlerini ifade etmektedir. Bu tanım, kaynak koda nasıl erişileceği ile değiştirme hakkına ve değiştirilmiş kaynak kodunun tekrar nasıl dağıtılacağını vurgulamaktadır. Kaynak kod, yazılımın nasıl çalışacağını, ne işler yapacağını belirleyen ve insanın okuyabileceği bir yazılı haldir. Bir programcı ancak kaynak kodu değiştirerek yazılıma yeni işlevler kazandırabilir. Başka bir deyişle kaynak koda erişmeden yazılımı geliştirmeye devam etmek mümkün değildir. Bu yüzden kaynak koda erişim diğer özgürlükler için temel bir koşuldur.

Özgür yazılım dünyasında, yazılım yapabilen bir kişi, istediği özellikleri hiç bir yasal engele takılmadan tamamen ücretsiz olarak yazılıma entegre etme hakkına sahiptir. Programcılar başkalarının yazdığı kodu kullanıp üzerine geliştirme yapıyorsa, başka geliştiriciler de o programcının yazdığı kodları kullanarak geliştirme yapabilmelidir. Bu yaratılan özgür yazılım ekosisteminde devamlılığın sağlanabilmesi için gereklidir ve GPL sayesinde ilave kodun yayınlanması da zorunlu kılınmıştır. GPL lisanslı ürünlerin çoğu ücretsiz olarak dağıtılmaktadır. Bu durum lisansın getirdiği bir zorlamadan ziyade geliştiricilerin yazılımı dağıtırken seçtikleri bir yoldur, çünkü kaynak koduna erişilebilen bir yazılımı ücretlendirerek satabilmek zordur. Günümüzde özgür yazılım dünyasının sağladığı çeşitlilik nedeniyle kullanıcılar belirli bir ücret ödemek yerine ücretsiz olan benzer yazılımları seçmektedir. Bu yüzden belirli bir ücret talep etmek bir süre sonra projenin yaygınlaşmasını engellediği için yazılımın geliştirilmesini yavaşlatmaktadır. GPL tanımlaması açısından bakıldığında bir yazılım şu şekilde ücretlendirilebilir; fiziksel olarak (CD, DVD ortamı, vb.) yazılımın kopyaları dağıtılırken bir ücret talep edilebileceği gibi yazılıma garanti verirken veya kullanıcılara teknik servis hizmeti vermek için de bir ücret talep edilebilmektedir.

Kapalı kaynak kodlu yazılımlar veya ticari yazılımlarda ise durum bunun tam tersidir; kaynak kodu kapalıdır. Yazılımın kodu ikilik sistem ile dağıtılmaktadır. Bu durum, yazılımın kaynak koduna erişmeyi teknik olarak imkansızlaştırmaktadır. Ayrıca yetkisiz erişim de kanunlarla yasaklanmıştır. Bir programcı kapalı kaynak kodlu bir yazılıma bir özellik eklese bile bu eklentiye yasal olarak yeniden dağıtmaya hakkı yoktur. Kapalı kaynak kodlu yazılımlar, kullanıcılara lisans sözleşmeleri sayesinde bir çok sınırlama getirmektedir. Kapalı kaynak kodlu bir yazılım satın alındığında yazılımı kullanırken nasıl çalıştığı, bilgisayarda o anda neler yaptığı kesin olarak bilinemez. Bunun doğal bir sonucu olarak da bir sorun ile karşılaşıldığında bu sorunun nedenini tahmin etmek güçtür. Yazılımı geliştirmek sadece o yazılım sahibinin tekelindedir (Dalgıç 2007). Bu sınırlamaların olmadığı diğer dünyanın adı ise özgür yazılım dünyasıdır. Üstelik bu dünyada her yazılım için yüksek kullanım ücretleri ödemek gerekmemektedir.

Özgür bir yazılım projesi, yazılımı kullanan ve destekleyen kullanıcılardan, projeye katkı sağlayan programcılardan ve yazılımı yaşatan geliştiricilerden oluşmalıdır. Çoğunlukla mekandan bağımsız olarak sanal dünyada bir araya gelen bu gruplar, "topluluk" (community) kelimesi ile tanımlanmaktadır. Farklı hedef ve amaçları olan bu gruplar için özgür yazılım farklı anlamlar taşımaktadır. Özgür yazılım, bir "programcı" için yazılımın kaynak koduna erişebildiği, kod yazarak istediği özellikleri ekleyebildiği ve yazdığı eklenti kodlarını yayınlamak zorunda olduğu bir yazılımdır. Diğer yandan, bir "kullanıcı" için özgür yazılım; yazılımı internetten bedelsiz olarak indirebildiği, bilgisayarlara istediği sayıda kurup çalıştırabildiği, özgürce kullanabildiği, kopyalayabildiği ve isterse kopyalarını bedelli veya bedelsiz olarak dağıtabildiği bir yazılımdır. Bir "geliştirici" için ise, yazılım sektöründe tanınmış biri olmak, bu yazılımı akademik çalışmalarda kullanmak veya para kazanmak vb. nedenlerle bir yazılım geliştirip yayımlayabildiği, bir topluluğu organize ederek katkı sağlayabilecekleri bir ortam yaratabildiği ve bu topluluk çevresinde yazılımı büyütüp geliştirebildiği bir yazılım geliştirme aracıdır. Programcı yazılımın sürdürülebilirliği ile ilgilenmezken, geliştirici yazılımın devamlılığını

sürdürmeye ve yazılımı bir ürün olarak sunmaya çalışan kişidir. Her programcı potansiyel bir geliştirici olabileceken, özgür yazılımlardan yararlanma hedef ve amaçları farklı olduğu için bu potansiyel çoğunlukla geliştirici olma yönünde sonuçlanmamaktadır.

Aslında bir “programcı” ile bir “geliştirici” arasında yaptıkları iş ve bu işin amaçları açısından belirgin bir ayrım bulunmaktadır. Bu ayrımı ortaya koymak, özgür yazılımların büyütülüp geliştirilmesi açısından çok önemlidir. Özgür yazılım programcısı ihtiyaca cevap verebilecek en iyi çözümü sunan ürünü araştırır, kullanır ve istenilen talepler doğrultusunda yazılıma yeni özellikler kazandıracak şekilde programlar. Programcı, anlık ihtiyaca en hızlı ve etkili çözümler sunan kişi olarak ortaya çıkmaktadır. Geliştirici ise kendini ürünü geliştirmeye adanmış, amaç olarak bir yazılımdan yararlanmak yerine sürekli onu geliştiren, yeni sürümlerini çıkartan, topluluğun daimi bir üyesi olarak kendini kabul ettirmiş ve kaynak kodunun bulunduğu depolara yazma yetkisi olan bir kişi olarak tanımlanabilir.

Bir programcı tarafından yapılan yeni bir geliştirmenin ürünün bir parçası olabilmesi için iki koşul vardır. İlk koşul, programcının, geliştirmenin kaynak kodunu yayınlamak zorunda olmasıdır. Bu koşul yerine getirilmeden bir sonraki adıma geçilemez. İkinci koşul ise, ürünü geliştiren topluluk ile yeni geliştirmenin amaç ve hedef kitesinin uyuşmasının gerekliliğidir. Yeni geliştirme topluluk için işlevsel olmalı ve yazılımın performansını olumsuz etkilememelidir. Bunun gibi benzer durumlar kesiştiği zaman, programcıların yaptığı yeni geliştirmeler, sadece geliştiriciler tarafından yazılımın ana kodu içine entegre edilebilir. Programcı ile geliştirici arasındaki temel fark tam olarak budur. Programcı yazılımı anlık olarak kullanan ve geliştiren bir kişi iken, geliştirici ise yazılım geliştirme işini yazılımın sürekliliğini sağlayabilecek bir ortamda devam eden kişidir. Programcının yaptığı geliştirme yazılımın kodu içine entegre edilmediği, bir servis hizmeti olarak sürekli verilmediği veya özgür bir yazılım projesi olarak yayınlanmadığı sürece, bir süre sonra kaybolup gitmektedir. Özgür yazılımları geliştirmenin en kalıcı ve sürekli yolu yapılan geliştirmelerin bir yazılım projesi haline getirilmesi ve proje etrafında bir topluluğun oluşturulmasıdır.

Programcı yaptığı geliştirmeyi bir özgür yazılım projesi şeklinde yayınlarsa ürününü pazarlama şansı için bir fırsat yaratmış olmaktadır. Özgür yazılım dünyası bugün her programcıya her hangi bir geliştirmeyi özgür yazılım projesi haline dönüştürebilmek için gerekli tüm araçları bedelsiz olarak sunmaktadır. Bu sayede, farklı yerlerde bulunan gönüllüler ile açık ve genişleyebilir bir ekiple çalışabilmek mümkün olabilmektedir.

Özgür yazılımların getirdiği en önemli avantajlar (özgürlükler) ise maliyetleri azaltabilme, kişisel ihtiyaçlara göre yazılımı geliştirebilme veya bir programcıdan bunu kendi yerinize ucuza ve hızlı bir şekilde yapmasını isteyebilme, yazılımın nasıl çalıştığını veya neler yaptığını inceleyerek güvenli olup olmadığını anlayabilme, yüzlerce hatta binlerce programcı tarafından gözden geçirildiği için daha az hata içeren ve güvenli bir yazılım kullanabilme, dağıtımın internette özgürce yapıldığı için çok daha hızlı bir şekilde yamaların çıkartılarak kullanıcılara ulaştırabilme ve yazılımı güncelleyebilme imkanı ve şansdır. Ayrıca eğer kullanıcılar özgür yazılım ürünlerini kapalı kaynak kodlu işletim sistemlerinde çalıştırmak yerine Linux’da kullanmayı seçerlerse virüsler ile de mücadele etmek zorunda kalmamaktadırlar. Bu da ekstra bir güvenlik getirmektedir.

Bir kullanıcı için özgür bir yazılımın getirdiği en büyük fayda ücret avantajıdır. Özgür yazılımlar çoğunlukla ücretsizdir ya da ticari yazılımlara göre çok küçük bir ücret talep ederler. Bu durumun aynısı ticari işletmelerde kullanılan yazılımlar için de geçerlidir. Günümüzde özgür yazılımlar o kadar büyük bir çeşitliliğe ulaşmıştır ki ticari bir işletmenin ihtiyaç duyabileceği tüm ürünlerde işletim sistemleri, ofis uygulamaları, internet tarayıcılar, e-posta uygulamaları, veritabanları, programlama dilleri ve editörleri, resim, müzik ve grafik araçları vb. gibi günlük hayatta ihtiyaç duyulabilecek ve kullanılmak istenilen tüm araçlar günümüzde özgür yazılım dünyasında mevcuttur ve özgürce kullanıcıların ve şirketlerin hizmetine sunulmuştur.

Özgür yazılımların kapalı kaynak kodlu ürünlere göre sağladığı maliyet avantajlarına Toplam Sahip Olma Maliyeti (TSOM) dikkate alınarak incelenmesi gerekmektedir. Bu konuda 2001’de yapılan bir araştırma sonucuna göre, özgür yazılımlar, alternatiflerine kıyasla Toplam Sahip Olma Maliyeti daha düşük ve yatırımın geri dönüşü daha yüksek olan yazılımlardır. Almanya’nın Bavyera eyaletinin Mali Denetim Ofisi’nin 2001 yılında hazırladığı raporda özgür yazılım ve kapalı kaynak kodlu yazılım arasında yapılacak maliyet karşılaştırmalarının lisans ücretlerinin çok daha ötesinde olduğu vurgulanmaktadır. Doğrudan maliyetlerin (lisans ücretleri, kurulum maliyeti, eğitim, destek) yanısıra dolaylı maliyetlerin (donanım yükseltmeleri, eski formattaki verilere ulaşım maliyeti) de dikkate alınması gerekmektedir.

Cybersource Şirketi tarafından yapılan bir çalışmada TSOM değerleri hesaplanmıştır. Bu çalışmada, 250 bilgisayar kullanan personel, internet bağlantısına sahip sunucular ve iş istasyonları, e-iş sistemi, ağ yapısı için gerekli kablolar ve donanım, standart yazılımlar, altyapı sisteminin devamlılığını sağlayacak bilişimci personelin maaşları gibi kriterler dikkate alınarak bir şirket modellenmiştir. Bu model, iki seçenek üzerinden geliştirilmiştir. İlk seçenekte, şirketin

halihazırda bir sistemi olduğu varsayılmıştır. İkinci seçenekte ise sıfırdan bir yapılanmaya gidilmiştir (Gözükeleş 2004). Cybersource Şirketi tarafından yapılan çalışmanın karşılaştırmalı sonuçları Tablo 1’de aşağıda gösterilmektedir (Cybersource Company 2004). Özgür yazılımların toplam sahip olma maliyeti, kapalı kaynak kodlu yazılımlara göre ilk seçenekte %36 ve ikinci seçenekte %26 daha ucuzdur. Bu model sadece 250 bilgisayara göre hesaplanmıştır. Eğer kamu kurumlarında kullanılan on binlerce bilgisayar düşünülürse tasarruf daha çarpıcı boyutlarda olabilecektir.

(Üç yıllık TSOM)	Microsoft Çözümü	Linux Standart Çözümü	Açık Kaynak Kullanımındaki Tasarruf	Yüzde olarak tasarruf
Mevcut donanım ve altyapı kullanılıyor	\$1,066,712	\$682,090	\$384,622	% 36
Yeni donanım ve altyapı satın alınıyor	\$1,366,883	\$1,012,260	\$345,623	% 26

Tablo 1: Üç yıllık toplam sahip olma maliyetinin karşılaştırılması (Cybersource Company 2004).

Bu tasarruf maliyetini sağlayabilmek için toplumsal bir üretim modeli gerçekleştirmek gerekmektedir. Bu modeli gerçekleştirebilmek için kullanılması gereken yazılım geliştirme araçları bulunmaktadır. Bu araçlar sayesinde özgür yazılım geliştirme ortamı kurulabilmektedir. Aşağıda sıralanan maddeler özgür bir yazılım projesinde olması gereken ortak özelliklerdir. Bu özelliklerin var olması özgür bir yazılım projesinin devamlılığını sağlayan ve hayatta kalmasını kolaylaştıran unsurlardır. Bu özellikleri sunan bir kaç geliştirme aracı da örnek olarak aşağıda verilmiştir (Wikimedia 2008b):

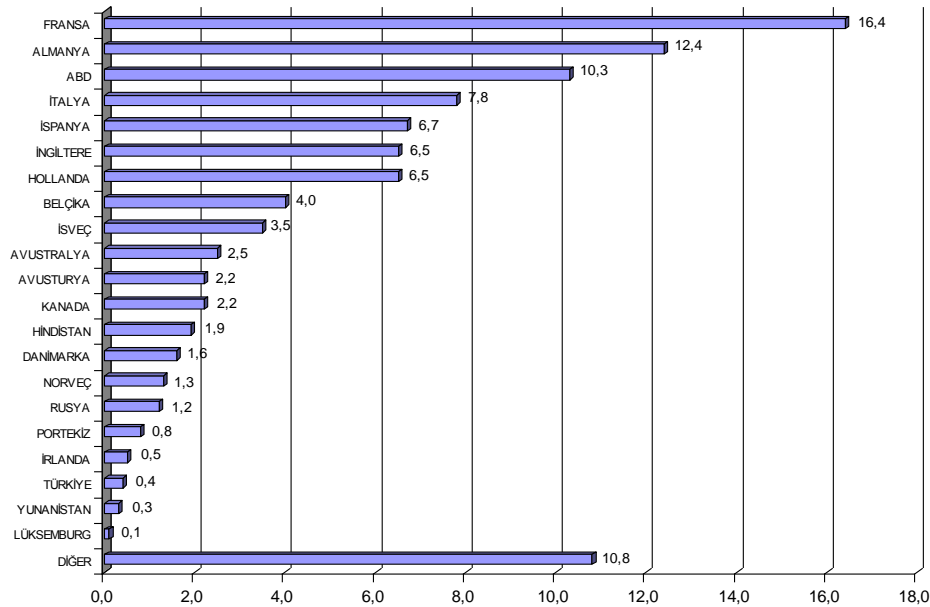
- **Sürüm Kontrol Sistemi:** Özgür bir projede zaman ve mekan birlikteliği olmayan geliştiricilerin aynı kod üzerinde çalışabilmesi, yazılımın kaynak kodunda bir çok geliştirici tarafından yapılan değişikliklerin izlenebilmesi, sürüm takibinin yapılabilmesi (örnek; Subversion, Concurrent Versions System)
- **Hata Takip ve Raporlama Sistemi:** (örnek; Mozilla’nın gelişmiş sistemi Bugzilla, Mantis; web tabanlı PHP ve MySQL kullanan bir hata takip sistemi, Trac; wiki entegrasyonuna sahip ve Subversion sürüm kontrol sistemine bağlantı sağlama özelliklerine sahip vb.)
- **İletişim Kanalları:** Geliştiricilerin anlık iletişim sorununa mesajlaşma yoluyla çözümler getirilebilmesi, kullanıcıların haberdar edilebilmesi, kullanıcılardan geri bildirim ve yorumların alınabilmesi. Bu ihtiyacın elektronik ortamda en kolay çözümlenebildiği yollar; e-posta, haberleşme ve IRC programları (örnek: kopete, xchat, vb.). E-postalar eğer e-posta listeleri halinde düzenlenirse, kullanıcılar tarafından araştırılabilen bilgi kaynakları haline gelmektedirler. (örnek; e-posta listeleri yayınlamak için GNU Mailman, Hypermail, MhonArc, vb.)
- **Test Araçları:** Tinderbox vb.
- **Paket Hazırlama Sistemleri:** Red Hat Package Manager (RPM), Advanced Packaging Tool (DEP) vb.
- **Döküman Hazırlama:** lyx, latex, vb.
- **Hata Ayıklama:** GNU Debugger vb.
- **Bellek Hata Yönetimi:** XPCOM, Splint vb.
- **Projenin İnternette Yayınlanması:** web sayfası hazırlama araçları, forumlar, bloglar veya bu hizmetlerinin bir kaçını birden bir arada ve web üzerinden sunan depolar (örnek: GNU Savannah, SourceForge, BountySource, vb.) (Wikimedia 2008c)
- **Projenin Duyurulması:** Freshmeat, directory.fsf.org, O’Reilly Net, Linux Weekly News, IBM developerworks, vb.

Bu ve benzeri özellikleri sağlayan geliştirme araçlarını kullanmak özgür yazılım geliştirme işinin teknik yolunu kolaylaştırmaktadır. Sosyal ve ekonomik boyutları dikkate alındığında ise bir özgür yazılım projesini geliştirmeye devam ettirmek zahmetli ve yorucu bir iştir. Özgür bir yazılım projesi geliştirmenin en önemli zorluğu belirli bir sabır ve disiplin içinde sürekli olarak yazılıma devam edebilmektir. Ülkemizde pek çok özgür yazılım projesi başlatılmasına rağmen belirli bir olgunluğa ulaşamadan sonlanmıştır. Turkuaz Linux, Turanid Linux ve Turkix Linux genel olarak kullanıcılardan yeteri kadar ilgi görmediği için geliştirilmesine son verilen birkaç özgür Linux dağıtım projesidir. Özgür bir yazılım projesini başlatıp geliştirmek çok uzun bir süre gerektirmektedir. Duruma, şartlara, yapılan yazılıma göre bu süre değişiklik göstermekte olup 5 ile 10 yıl arasında değişebileceği tahmin edilmektedir. Bu sürelerden daha kısa bir zaman içerisinde özgür bir yazılım projesi etrafında onu destekleyen güçlü bir topluluk oluşturmak pek olası değildir.

Gereken zamanın uzunluğu yanında işin niteliği de özgür bir yazılım projesini devam ettirmeyi zorlaştırmaktadır. Yazılım geliştirmek nitelikli insan gücü gerektirmektedir. Günümüzde yazılım sektörümüzde gönüllü katkı

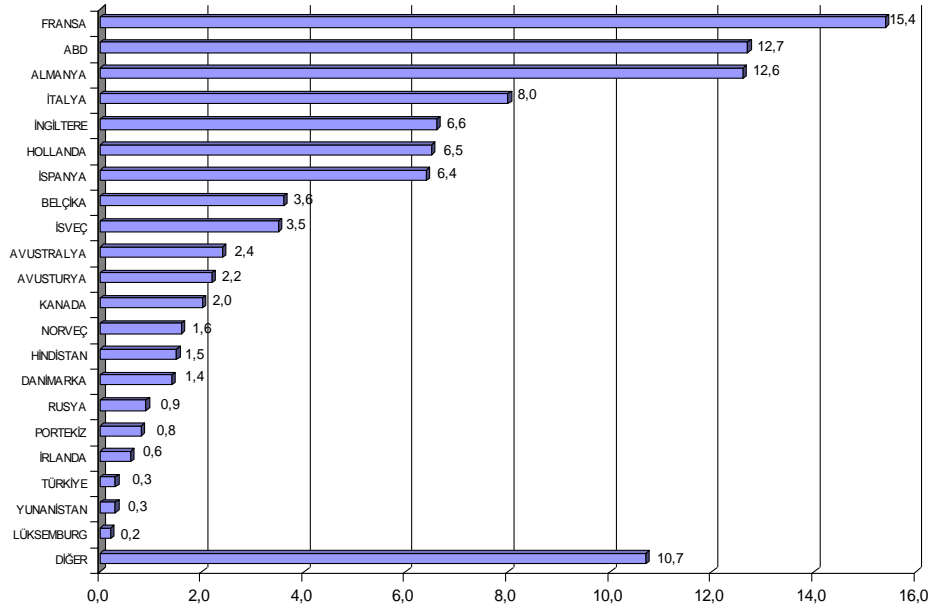
sağlayabilecek nitelikli işgücü ve programcı sayısı azdır. Özgür bir yazılımı geliştirmek ve etrafında bir topluluk oluşturmak ise çok uzun süre isteyen zahmetli bir iştir. Bu yüzden sadece gönüllü katkıları ile uzun bir süre bu nitelikli işgücünün bedavaya sunulması ülkemizde pek mümkün olmamaktadır. Bu durum, özgür yazılımların geliştirilmesinde ülke olarak aktif ve önemli bir rol alamamamızın nedenlerinden biridir. Özgür yazılım projelerinin devam ettirilebilmesi için bir topluluktan, şirketten veya kamu tarafından bir desteğin olması bir süre sonra kaçınılmaz olmaktadır.

Ülkemizdeki özgür programcıların nitelikleri ve sayıları kesin olarak bilinmemektedir. Bu konu ile ilişkilendirilebilecek bir çalışma 2002 yılında International Institute of Infonomics tarafından yapılmıştır. (International Institute of Infonomics ve Berlecon Research GMBH 2007). Bu çalışma sonucunda yayınlanan Özgür ve Açık Kaynak Kodlu Yazılımlar (ÖAKKY) hakkındaki raporun bulguları, 2002 yılında bir yıl boyunca yaklaşık 500 özgür programcının katıldığı bir anketin sonuçlarına dayanmaktadır. İnternet üzerinden yapılan bu anket, özgür yazılım dünyasında tanınan ünlü internet portalları ve e-posta listeleri ile duyurulmuştur. Çalışmanın amacı ÖAKKY'nın günümüz ekonomisi içerisindeki önemi ve rolünü belirleyecek temel veriyi toplamaktır. Raporda yayınlanan bulgulardan birine göre ankete katılanlar arasında Türk olan özgür programcı yüzdesi 4'tür. Programcı sayısının milliyetlerine göre dağılımı aşağıdaki şekilde verilmiştir. Anketteki özgür programcıların %71'i Avrupa kökenli olup, bunun %13'ü Kuzey Amerikalı ve geri kalanın %17'sini diğer ülkeler oluşturmaktadır. Türkiye, sırasıyla Lüksemburg ve Yunanistan'dan sonra sondan 3. sıradadır (Şekil 1).



Şekil 1: Özgür programcıların milliyetlerine göre dağılımı (yüzde) (International Institute of Infonomics ve Berlecon Research GMBH 2007).

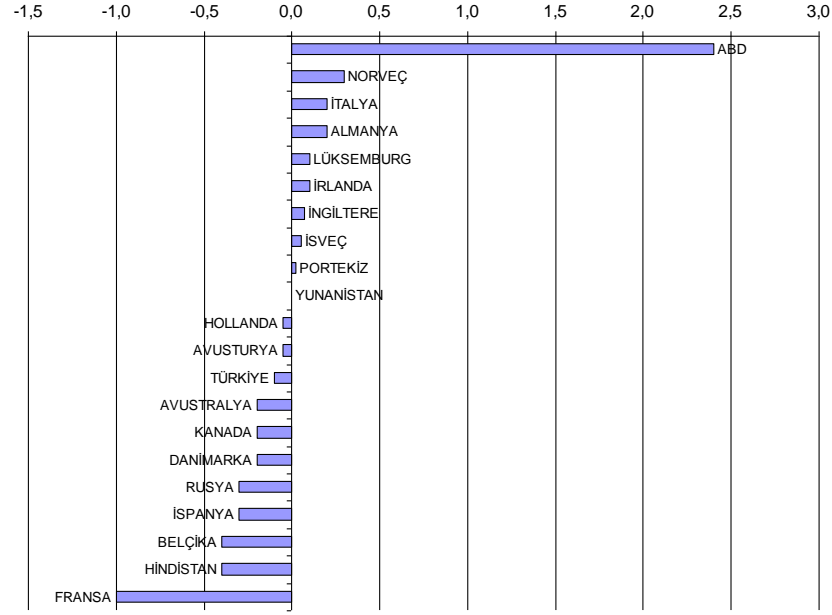
Bu programcıların nerede çalıştığı veya yaşadığı ile ilgili dağılıma bakıldığında ülkemiz açısından benzer bir sonuç çıkmaktadır (Şekil 2). Türkiye'de çalışan veya yaşayan programcı yüzdesi 3'tür. Aynı şekilde, ülkemiz, sırasıyla Lüksemburg ve Yunanistan'dan sonra sondan 3. sıradadır.



Şekil 2: Özgür programcıların çalıştığı veya yaşadığı ülkelere göre dağılımı (yüzde) (International Institute of Infonomics ve Berlecon Research GMBH 2007).

Özgür programcıların milliyeti ve çalıştıkları ülkelerin bilinmesi sayesinde özgür programcıların çalışmak için daha çok tercih ettikleri ülkelerin dağılımını ve hareketlilik desenlerini çıkartmak mümkün olabilmektedir (International Institute of Infonomics ve Berlecon Research GMBH 2007). Bu desen, Şekil 1 ile Şekil 2 karşılaştırılarak çıkarılabilmektedir. Özgür programcıların milliyeti ile çalıştığı ülkeler arasında büyük bir farklılık görülmemiştir. Ülkesi dışında başka bir ülkede çalışan programcı sayısı %10'dur. Şekil 2'deki ülkelerin sıralaması Şekil 1'e göre değişiklikler görülmektedir. Almanya ve ABD sıralamada yer değiştirirken, İspanya ve Rusya Federasyonu göreceli olarak özgür programcının daha az çalışmak istediği bir ülke konumunda gözükmemektedir (Şekil 3). Çalışmak için daha çok tercih edilen ülkelerin dağılımından da açıkça anlaşılacağı üzere ABD en çok tercih edilen ülke konumunda iken, Fransa özgür programcılar tarafından en az tercih edilen ülke olmuştur.

Ülkemiz de az tercih edilen ülkeler grubu içinde yer almakta olup, Yunanistan, Hollanda ve Avusturya'dan sonra 4. sırada yer almaktadır. Özgür programcıların çalışmak üzere seçtiği ülke olan ABD'nin çekiciliğini Avrupa ülkelerinden hiç biri geçememiştir. Bunun nedenleri ankette sorulmadığı için kesin olarak bilinmemektedir. ABD, özgür yazılımların en çok kullanıldığı ve yaygınlaştığı ülkelerden biridir. Bu yüzden özgür programcılar bu ülkeyi daha çok tercih etmektedirler (International Institute of Infonomics ve Berlecon Research GMBH 2007).



Şekil 3: Özgür programcıların çalışmak için daha çok tercih ettikleri ülkelerin dağılımı (yüzde) (International Institute of Infonomics ve Berlecon Research GMBH 2007).

Ülkemizde özgür yazılımların yaygınlaşması ile ilgili gelişmeler diğer ülkelere göre yavaş bir seyir izlese de, ulusal politikalar sayesinde bu gelişmeler hız kazanabilir. Özgür ve açık kaynak kodlu yazılımları ulusal bir politika olarak benimseyerek destekleyen ülkeler arasında Çin, Almanya ve Brezilya bulunmaktadır. Bu ülkelerdeki kamu desteği sayesinde özgür yazılım uygulamalarının giderek daha fazla yaygınlaşmaya başladığı görülmüştür. Ülkemizde ise, özgür yazılımların ulusal bir politika ile benimsenerek desteklendiğinden bahsetmek oldukça zordur. Kamu bünyesi içinde geliştirilen ve bu sayede popüler olan ulusal işletim sistemimiz Pardus'un varlığı, özgür yazılımların kamudan tümüyle dışlanmadığının temsili bir örneğidir.

Pardus'un kullanımı bir gün kamuda yaygınlaşabilirse, bu durum diğer gelişmeleri de tetikleyebilecek bir adım olacaktır. TÜBİTAK Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü (UEKAE) bünyesinde 2004 yılında geliştirilmeye başlanan ulusal dağıtım sistemimiz Pardus'un ilk sürümü Aralık 2005'te çıkarılmıştır (UEKAE 2007). Pardus'un sürüm adları ve yayınlandıkları tarihler aşağıda sıralanmaktadır (TIOBE Software):

1. Pardus 1.0 sürümü, Aralık 2005,
2. Pardus 2007 sürümü, Aralık 2006,
3. Pardus 2007.1 FetişChaus sürümü, Mart 2007,
4. Pardus 2007.2 Caracal Caracal sürümü, Temmuz 2007,
5. Pardus 2007.3 Lynx lynx sürümü, Kasım 2007,
6. Pardus 2008 sürümü, Haziran 2008'de çıkartılmıştır.

2004 yılından bu yana geliştirilmesi süren Pardus, Ağustos 2008 itibarıyla Pardus 2008 lynx sürümü ile hizmetine devam etmektedir. Pardus, GNU GPL (Genel Kamu Lisansı) lisansı ile lisanslanmış olup, www.pardus.org.tr adresinden ücretsiz olarak indirilip kurulabilen linux tabanlı bir dağıtım / işletim sistemidir.

Pardus sadece ülkemizde değil, tüm dünyada tanınan bir linux dağıtım projesi haline gelmeye başlamıştır. DistroWatch.com, Linux dağıtımlarının listesini ve kullanım sıklığını ait istatistikleri tutan bir web portalıdır. Her ay en az bir veya iki tane yeni çıkan dağıtım da listeye dahil edilmektedir. Temmuz 2008 itibarıyla kayıtlı 349 tane Linux dağıtımı arasından, Pardus Linux, son bir aydaki kullanım sıklığı istatistiklerine bakıldığında 14. sıradadır (Distrowatch 2008). Başlangıç dönemindeki bir proje için tüm dağıtımlar arasından ilk 50'ye girebilmek oldukça önemli bir başarıdır.

TÜBİTAK-UEKAE tarafından geliştirilen Ulusal Linux Dağıtımı, Pardus, kamunun desteği sayesinde özgür yazılım geliştirme ortamında başlatılan örnek bir projedir. Pardus kadar popüler olmasa da geliştirilmeye çalışılan ve hala aktif olan yerel Linux dağıtımları aşağıda sıralanmaktadır;

- Gelecek Linux: Gelecek A.Ş. tarafından geliştirilmektedir. Desktop Vision ve Enterprise Linux olarak iki ayrı ürün ile hizmet vermektedir. Bu linux dağıtımları küçük bir ücret karşılığında satın alınabilmektedir. (<http://www.gelecek.com.tr/>)
- Armador Linux: (<http://www.armadorbilisim.com/>)
- Truva Linux: Nisan 2004'ten beri faaliyet gösteren Slackware tabanlı yerel bir Linux dağıtımıdır ve Mart 2008 tarihi itibarıyla 2.0 RC1 sürümü çıkmıştır. (<http://www.truvalinux.org.tr/>)

DistroWatch.com listesinde, Gelecek Linux ve Truva Linux dağıtımları da listede yer almaktadır (Distrowatch 2008). Dünya'da sonlanan bir çok linux dağıtım projesi olduğu gibi, ülkemizde de bu örneklerle rastlamak mümkündür. Örneğin;

- Turkuaz Linux: İstanbul Teknik Üniversitesi ve Yeditepe Üniversitelerinden bir grup öğrenci tarafından geliştirilmeye çalışılmıştır ve 1996 ile Nisan 1998 yılları arasında faaliyet göstermiştir (Wikimedia 2008a).
- Turanid Linux: Daha sonraları Everest Linux olarak isim değiştirmiştir ve Ekim 2004 ile Nisan 2007 yılları arasında faaliyet göstermiştir (Wikimedia 2007a). (<http://www.turanid.info/>)
- Turkix Linux: Türkçe ve Azeri dillerinde yapılandırılmış Mandrake tabanlı bir Linux dağıtımıdır (Wikimedia 2007b). (<http://www.turkix.org/>)

Bu ve benzeri projeler, genel olarak, gerekli desteği göremediği ve geniş katımlı bir topluluk oluşturamadığı için faaliyetlerine son vermek zorunda kalmışlardır. Turkuaz Linux projesine son verilme nedeni şu şekilde açıklanmaktadır; “Türk Sermayesi bu projeye gerekli ilgiyi zamanında gösterip başarıya giden yolda çabalarını birleştirememiştir. Halbuki, Cisco, SUN ve Google hep birer üniversite bilişim projesi olarak başlamış ve daha sonraları bilişim dünyasına yön veren şirketlere dönüşmüşlerdir” (Çayır 2004). Özgür yazılım projelerinin devamlılığını belirleyen en önemli unsur aldığı katkılardır. Bu katkılar, kamudan ve bir şirketten gelebileceği gibi o yazılımı kullanan topluluk sayesinde de oluşabilir.

Özgür yazılım ekonomisi gönüllü katkılar sayesinde gelişmektedir ve her geçen gün biraz daha büyümektedir. Uzmanların gönüllü katkılarına her zaman ihtiyaç duyan bir ekonomidir. Karşılığı katkı yapana öncelikle prestij daha sonra da para olarak geri dönebilen bir sektördür. Prestij veya kazanılacak gelir, ne kadar çok katkı yapıldığı ile ilişkilidir. Özgür yazılımı kullananlar, yazılımı ikincil satış yapma hakkına, bağımsız olarak geliştirme, destek verme ve yan hizmetlerin sağlanması haklarına doğrudan sahiptirler. Bu haklar, gelir getirici bir faaliyet olarak dönüş sağlayabilmektedir. Özgür yazılım felsefesi temel olarak sosyal bir olgu olarak kabul edilmektedir. Gelir elde etme amacı ikincil veya üçüncül bir amaç olarak düşünülmelidir.

Özgür yazılım projelerinin devamlılığını etkileyen bir diğer nokta ise projenin etrafında bir topluluğun oluşturulamamasıdır. Bu topluluğu oluşturabilmek programcılık yeteneğinden ziyade farklı bir yöneticilik anlayışını da gerektirmektedir. Genel olarak kullanıcı ve gönüllü katkı sağlamak isteyenlerden oluşan bu topluluğun idare edilmesi için de ayrıca zaman ayırmak gerekmektedir. Yazılım geliştirme ortamı öyle bir şekilde tasarlanmalıdır ki o ortama ilk defa gelen bir kullanıcı bile isterse katkı sağlayabilmelidir. Yazılım geliştirme işlemlerinin tümü kullanıcıların gözü önünde takip edilebilir bir biçimde yapılmalıdır. Yazılım geliştirme kararları açık ve ulaşılabilir olmalıdır. Örneğin, yazılımı hangi yönde geliştirileceğine dair stratejik kararlar alırken kullanılan haberleşme (chat) programının çıktıları bile internette yayınlanabilmelidir.

Bir topluluğun oluşmasını engelleyen ve tahmin edilemeyen pek çok neden olabilir. Kullanıcısı olmayan bir yazılım varlığını uzun süre sürdüremez. Eğer yazılımın kendisi veya geliştirme ortamı kullanıcıların ilgisini çekebilirse katkı alabilir. Bunun için kullanıcılara nasıl katkı yapabileceklerini gösteren talimatlar hazırlanmalıdır. Bu sayede hem yazılımın etrafında bir topluluk oluşturulabilir hem de yazılım geliştirme işlerinin rutin kısmı kullanıcılara devredilerek geliştiricilerin iş yükü hafifletilmiş olur. Her kullanıcı iyi bir katkı sağlayıcı haline getirilerek projenin gelişmesinde önemli bir rol alabilir. Bir programcı veya kullanıcı çeşitli yollarla geliştiricilere katkı sağlayabilirler. Bu yollardan bazıları, dökümantasyon hazırlamak, paketleme yapmak, yerelleştirme ve tercüme çalışmalarına katılmak, programı test etmek, hataları raporlamak vb. şeklinde sıralanabileceği gibi doğrudan bağış yapmak da etkili bir yoldur. Bu katkılarının amacı kişisel tatmin, bir topluluğa ait olma duygusu, kendini geliştirme, sektörde tanınmış biri olma, akademik çalışma yapma veya yazılım geliştirmeyi öğrenme vb. nedenler olabilir. 2006 yılında Gözükeleş'in yaptığı araştırmaya göre özgür Türk programcılarının gönüllü katkı sağlama nedenleri Tablo 2'de sıralanmıştır (Gözükeleş 2006):

Motivasyon Nedenleri	Yüzde (%)
Bilgi paylaşmak, iş birliği yapmak ve daha iyi bir dünya için katkıda bulunmak isteği	67,2

Programlama yaparken eğlendiği ve hoş vakit geçirdiği için	21,8
Yeni beceriler öğrenmek ve kendini geliştirmek için	10,9
Yazılımların özel mülk sahiplerine ait olmadığını düşündüğü için	7,8
Milliyetçilik duygularıyla	4,7
Ticari yazılımların çözemediği bir problemi çözmek için	4,7
İş fırsatları geliştirmek için	1,6

Tablo 2: Özgür Türk programcılarının motivasyon nedenleri (Gözükeleş 2006).

Özgür yazılım ürünlerini kullanarak gelir elde etmenin en temel yolu müşteriye teknik destek hizmeti sağlamaktır. Türkiye’de özgür yazılımları kullanarak teknik destek hizmeti veren kurumsallaşmış firma sayısı çok fazla olmasa da özgür yazılım geliştirme metodunun sağladığı faydalar nedeniyle giderek daha fazla yaygınlaşacağı düşünülmektedir.

Bu konuda karşıt bir düşünce ise özgür yazılımlara teknik destek hizmetinin kurumsallaşmış firmalar yolu ile sağlanamayacağını ileri sürmektedir. Bu görüş, işin doğası nedeniyle, bu teknik destek hizmetinin kurumsallaşmış firmalar yerine birkaç kişiden oluşan küçük firmalar tarafından sağlanmaya devam etmesi gerektiğini iddia etmektedir. Özgür yazılım sektöründe aktif projelerde ürünlerin geliştirilmesi ticari uygulamalara göre çok hızlı bir şekilde gerçekleştirilebilmektedir. Bu hız sürümlerin ve yamaların çıkartılması aşamalarında gözlemlenebilir. Dolayısıyla, teknik desteği, hiçbir kurumsallaşmış büyük bir firmanın, küçük bir firma kadar hızlı, etkin ve ucuza sunması mümkün olamayacağı düşünülmektedir. Kurumsallaşmış firmalar büyüklükleri nedeniyle hantaldırlar. Hızla gelişen teknolojiye uyum sağlama konusunda yavaş hareket ederler ve isteksiz davranırlar. Bu noktada yazılım sektöründe küçük firmaların üretebileceği teknolojinin gücü, özgünlüğü ve yeniliği ile yarışmaları mümkün olamamaktadır. Küçük firmaların yeniliklere uyum sağlayabilme gücü, yani hareket hızı ve manevra kabiliyeti büyük firmalara göre çok daha fazladır. Özgür yazılım geliştirme araçları, küçük firmalara bu gücü sağlamaktadır.

Büyük firmalar ise, güçlerini geniş müşteri yelpazesine sahip olmalarına borçludurlar. Yazılımı bir marka haline getirmek ve pazarlamak ise yazılım alanı dışında bir yatırım gerektirir ki bu husus küçük firmaları, kurumsallaşmış büyük firmalara mahkum hale getirmektedir. Bu bağımlılığı internet üzerinden dağıtım yaparak azaltmak her ne kadar mümkün olsa da reklam yapmak bir mecburiyettir. Bu da belirli bir yatırım yapabilme gücü gerektirmektedir. Bu sebeplerden dolayı teknik destek hizmeti verme ve teknoloji pazarlama konularında küçük firmalar ile büyük firmalar arasında her zaman kıyasıya bir yarış ve etkileşim yaşanmaktadır. Teknik destek hizmetinin nasıl ve kim tarafından verileceği biraz da özgür yazılım ürünlerinin kullanımının ne kadar yaygınlaşacağı ile ilgili bir konudur. Bu açıdan bakıldığında gelecekte her iki durumunda mümkün olabileceği düşünülmektedir.

Günümüzde, bir çok farklı alanda özgür yazılımlar, ticari ürünlerin sunduğu kalitede hizmet sunabilen ürünler olarak ön plana çıkmaktadır. Günümüzde özgür yazılımlar ticari ürünlerle kıyasıya yarış halindedir. Özgür yazılımın Coğrafi Bilgi Sistemi (CBS) alanında daha fazla büyümesi ve gelişmesi için gerek kullanıcı ve gerekse hizmetini pazarlayabilecek imkanlara sahip geliştiricilere ve programcılara büyük görevler düşmektedir.

Sağladığı özgürlükler ve toplumsal faydalar nedeniyle, daha fazla özgür yazılım projelerinin yerel geliştiriciler tarafından geliştirilmeye başlanacağı ve kullanıcıların da katkı sağlayarak özgür yazılım ürünlerini daha sık kullanacakları ve bu ürünleri tercih edecekleri umulmaktadır. Bu konu, ülkemizdeki Bilişim Teknolojileri sektörünün gelişimi için ön bir koşuldur. Açık kaynak kod bildirgesinin sloganı bu ilkeyi şu şekilde açıklamaktadır: “Türkiye Bilişim’le, Bilişim Özgür Yazılımlarla Gelişir!” (Açık Kaynak Kod Platformu 2008).

2. ÖZGÜR CBS YAZILIMLARI

CBS dünyasında özgür yazılımların kapalı kaynak kodlu veya ticari yazılımlara göre sunduğu en önemli avantajı, lisans ücreti ödeme zorunluluğunun olmamasıdır. Bu avantaj ile yukarıda anlatılan özgür yazılımların sağladığı diğer avantajlar nedeniyle özgür CBS yazılımları kullanmak aynı zamanda CBS dalının da giderek daha fazla yaygınlaşacağı anlamına gelmektedir. Özgür CBS yazılımları, CBS dünyasının yaygınlaşmasındaki engellerden biri olan yazılım fiyatlarını ortadan kaldırmaktadır.

Gerek özgür yazılımların gerekse CBS dalının daha fazla yaygınlaşmasını sağlayan etkili yollardan biri üniversitelerde hangi CBS yazılımlarının kullanıldığı, uygulamalı CBS derslerinde hangi CBS yazılımlarının öğretildiği ve araştırmacıların çalışmalarında hangi CBS yazılımları üzerinde geliştirme yaptıkları ile ilgilidir. Ticari CBS yazılımlarını üreten firmalar, CBS ürünlerini yaygınlaştırma katkısını kendi lehlerine kullanabilmek için ürünlerini kendi

ülkelerindeki üniversitelere bedelsiz veya çok düşük ücretler ile pazarlamak gibi bir ücret politikası izlemektedirler. Yalnız aynı ücret politikasını geliştirmekte olan ülkeler için daimi olarak uygulamamaktadırlar. Bu durum ülkemizdeki üniversitelerin CBS kullanımını lisans ücretlerinin yüksekliği nedeniyle ya birkaç adet ile sınırlandırmakta ve ürün çeşitliliğini azaltmakta ya da tümüyle bu teknolojiler ile tanışılmasını engellemektedir. Öğrenciler açısından da kısıtlı sayıdaki lisans sayısı nedeniyle ürünler lablarda yeteri kadar öğrenilememektedir ya da korsan kopyalar ile suç işlemeye itilmektedirler. CBS kullanımının yaygınlaşmasını engelleyen yüksek lisans ücretleri üniversiteleri, kamu sektörünü ve özellikle özel sektörü olumsuz olarak etkilemektedir. Kamu sektörü ve üniversiteler için nispeten indirimler uygulanırken özel sektör için bu tür bir ücret düzenlenmeleri yapılmamaktadır. Bu durum, CBS dalının gelişimi açısından kamuya bağlılığı zorunlu hale getirmektedir.

Özgür yazılımların yaygınlaşması öncelikle üniversitelerden başlamalıdır. Özgür yazılım dünyası araştırmacılara büyük kolaylıklar sağlamaktadır. Özgür yazılımlar araştırmacıları uygulanmış bir algoritmayı tekrar yapmaktan kurtarmaktadır. CBS alanında çalışan araştırmacılar için özgür yazılımlar, mekansal veri analiz tekniklerinin aslına uygun bir şekilde nasıl çalıştıklarını öğrenebilmek için kullanabilecekleri muazzam bilgi kaynaklarıdır. Ayrıca özgür yazılımları kullanan araştırmacılar yeni teknikleri deneyebilecekleri ve bu sayede yeni yöntemler keşfedebilecekleri gelişmiş bir test ortamına sahip olmaktadır. Araştırmacılar için özgür yazılımlar iyi bir dökümantasyona sahiptirler. Algoritmaların ve optimizasyon yöntemlerinin nasıl çalıştığını anlatan en iyi kaynak, yazılımın kaynak kodudur. Özgür yazılımların kaynak kodu isteyen herkes tarafından okunabilmektedir. Özgür yazılımlar herhangi bir sınırlama olmaksızın araştırmanın ihtiyaçlarına göre geliştirilebilmektedirler. Araştırmacılar tarafından özgür yazılımlarda yapılan yeni geliştirmeler ve keşfedilen teknikler anında yayınlanabilmekte ve diğer araştırmacılar ile paylaşılabilmektedir. Başka bir deyişle, yeni geliştirmelerin kullanıcılarla buluşması için ticari yazılımlarda olduğu gibi yazılımın yeni sürümünün çıkmasını beklemeye gerek yoktur. Bazen bir araştırmanın çözümü başka bir alan ile ilişkilendirildiği zaman çözümlenebilmektedir. Sonuca ulaşmak için diğer alanda geliştirilen bir yazılım ürünü CBS yazılımı ile entegre edilmesi gerekmektedir. Bu entegrasyonun doğru bir çözüm olup olmadığı test edilmeden bilinmemektedir. Ticari ürünleri özgür yazılımlara göre elde etmek, hem yüksek bir ücreti ödemeyi hem de yazılımın ulaşması için daha uzun bir zaman beklemeyi gerektirmektedir. Üstelik test sonucu olumsuzsa gereksiz bir harcama yapılmış olunmaktadır. Özgür yazılımlarda ise başka bir özgür yazılımı test etmek için harcanacak tek çaba, o yazılımı internette indirmektir.

ODTÜ Jeodezi ve Coğrafi Bilgi Teknolojileri (JCBT) anabilim dalı bünyesinde ticari CBS yazılımları ile birlikte özgür CBS yazılımları da kullanılmaya başlanmıştır. Bu amaçla bir QGIS eklentisi örnek olarak geliştirilmektedir. Eklentinin kaynak koduna web aracılığıyla ulaşılabilen ve GNU GPL lisansı ile lisanslanmıştır. Bu çalışmada QGIS ile özgür istatistik yazılımı olan R entegre edilerek sistemin yapılandırılması tamamlanmıştır. Örnek olarak bir adet mekansal veri analiz aracı da oluşturulmuştur. Çalışma ile ilgili detaylı bilgiye http://www.ggit.metu.edu.tr/full_time_members/research_files/v_kep_webSite/SDATool.htm adresinden ulaşılabilir. Mekansal veri analiz alanında kullanılan diğer analiz teknikleri de özgür yazılım felsefesi kuralları içerisinde belirtilen sistem yapısı altında entegre edilmeye devam edilmektedir. Özgür CBS yazılımların kullanımını yaygınlaştırmak için ODTÜ JCBT bölümünün imkanları kullanılarak geliştirilen bu QGIS eklentisi ücretsiz olarak kullanıcıların kullanımına sunulmaktadır. Bahsedilen çalışma şu an için yalnızca özgür yazılım ürünlerini bir sistem bütünlüğü içerisinde kullanan ve yapıları bir web sayfasında sergileyen örnek bir geliştirmedir. Bunun gelişerek daha büyük ve özgür bir CBS projesi haline dönüşebilmesi için CBS kullanıcılarından ve araştırmacıların gelecek katkılar ve talepleri önemli olacaktır.

CBS dünyası kendi özelliklerini çeşitli ürünler aracılığıyla sunmaktadır. CBS yazılım ürünleri, masaüstü uygulamaları, web harita uygulamaları, sözel veritabanı uygulamaları, sözel veritabanına coğrafi veri özelliği kazandıran arayüzler, geliştirme kütüphaneleri, 3 boyutlu analizler, mekansal modellemeler ve ağ analizi uygulamaları gibi özelleşmiş alanlar için geliştirilen eklentilerden oluşmaktadır. Şu ana kadar tüm özellikleri tek bir üründe bulmak mümkün olamamıştır. Bahsi geçen her ürün ayrı ayrı lisanslanarak ücretlendirilebilmektedir. Bu ücretlere her kullanıcı için ilave edilecek lisans ücretlerini de kullanıcı sayısını çarparak eklemek gerekmektedir. Bu toplam ticari CBS yazılımı sahip olma maliyetine birkaç yılda bir çıkarılan yeni sürümlerin de lisans ücretlerini ilave etmek gerekmektedir. Tüm bu maliyetler dikkate alındığında CBS alanının yaygınlaşmasındaki önemli maliyet engellerinden birinin yazılım lisans ücretlerinin yüksek olduğu daha kolay anlaşılmaktadır. Özgür CBS yazılımlarının yapabilecekleri ticari yazılımlara göre bazı alanlarda nispeten daha kısıtlı olduğu düşünülse de toplam sahip olma maliyetinin düşüklüğü nedeniyle fark edilmesi gereken somut bir olgu olarak öne çıkmaktadır. Kaldı ki özelliklerin kısıtlılık durumu yazılımın geliştirilebilirliği özgürlüğü sayesinde gereksinilen özellikler yazılıma eklenmesiyle çözümlenmektedir. Bu ekleme işlemi bir programcı tarafından yapılabileceği gibi bir hizmet olarak satın alınabilir. Yazılıma istenilen özellikleri eklemek için gereken maliyet ile ticari yazılımda bu özelliklerin bulunup bulunmadığı ve toplam sahip olma maliyetinin miktarları karşılaştırılması gereken bir durumdur.

CBS uygulamaları bir kurumda kullanıma yaygınlığı dikkate alınarak değerlendirildiğinde CBS kullanımını dört olası gruba ayırmak mümkün olabilmektedir. Gruplar arasındaki ayrım kesin olmamakla birlikte sıralanan özelliklerden bir kısmı diğer gruplar içerisinde de bulunabilir. Burada amaç CBS kullanım düzeylerinin üst ve alt seviyelerdeki kullanım özelliklerine göre gruplandırarak bir kurum içinde ürün çeşitliliği edinilebildiği zamanlarda ne tür bir kullanım düzeyine çıkabileceğini gösterebilmektir.

İlk grup en basit düzeydeki CBS kullanımıdır. Bu grup kurumdaki bir birimde bir veya bir kaç tane CBS kullanıcısının bulunduğu bir yapıyı temsil etmektedir. CBS kullanıcıları yazılımı kullanarak istenilen işi yapmakta ve genellikle CBS verisi kişisel bilgisayarlarında bulundurulmaktadır. Bu durumda, CBS kullanımı kurum içinde oldukça sınırlıdır. İkinci grupta ise, kurum içinde CBS ile ilgili işlerin yapılması için görevlendirilen bir bölüm kurulmuştur. CBS bölümü kurumdaki diğer birimlerin ihtiyaçlarına cevap verebilmek için hizmet etmektedir. CBS verileri sunucu üzerinde merkezi ve tek bir veritabanı ortamında bölüm içinde paylaşılarak tutulmaktadır. İkinci grup alt ile üst düzey arasındaki geçişi simgeleyen bir gruptur. Bu gruptaki CBS kullanım düzeyi birinci gruba göre kurumun sahip olunan CBS yazılım lisans sayısı ve ürün çeşitliliği bakımından daha fazladır ve bu sayesinde bahsedilen sorumlu bir birim kurulabilmiştir.

Üst seviyedeki CBS kullanımı olan üçüncü grupta ise görevli bir bölüm ile birlikte diğer bölümler içinde bir veya birden fazla personel CBS yazılımı kullanabilmekte ve coğrafi veriye ulaşabilmektedir. Sunucularda dağıtık mimaride birden fazla mekansal veritabanında tutulan coğrafi verilere farklı birimlerdeki CBS kullanıcıları erişebilmektedir. CBS, tek sorumlu bir birim yerine birkaç birimle birlikte kullanılabilir. Bu grupta CBS kullanımı kurumun çoğuna yayılmıştır. İsteyen her personel intranet yolu ile hızlı bir şekilde coğrafi veriye ulaşabilmekte ve mekansal analizler yapabilmektedir. Bir CBS yazılımı kurumun ihtiyaçları doğrultusunda özelleştirilmiş ve diğer ERP ve CRM gibi sistemlerle birlikte entegre bir şekilde çalıştırılmaktadır. Müşteriler için de web aracılığıyla coğrafi veri sunulmaktadır.

Dördüncü grup son grup olup en gelişmiş düzeydeki CBS kullanımını ifade etmektedir. CBS, bu grupta bir araç değil, bir amaç haline gelmiştir. CBS, mekansal araştırma ve geliştirme projeleri yapmak, ulusal veya bölgesel coğrafi veri altyapıları hazırlamak ve sunmak, yeni bir CBS analiz metodu bulmak, yeni bir CBS yazılımı geliştirmek, mekansal teknolojiler keşfetmek veya yeni CBS uygulamalarını denemek için kullanılmaktadır. Bu grup için sıralanan özelliklerden birkaçı ülkemizde ancak üniversitelerde yapılan çalışmalar sayesinde gerçekleştirilebilmektedir.

Ülkemizdeki CBS kullanımı genellikle birinci gruptadır. İkinci gruptaki CBS kullanımı yaygınlığına sahip olabilmek için güçlü mali yapısı sayesinde altyapısını sağlamlaştırabilen ve mekan ile doğrudan ilgili işler yapan bir kurum olmak gerekmektedir. Ticari CBS yazılım ücretlerinin pahalılığı, kesintisiz bir ağ altyapısına ve donanımsal özellikleri itibarıyla hızlı masaüstü ve sunucu bilgisayarlarına sahip olmak gibi altyapı gereksinimlerini karşılama zorluğu, yetişmiş CBS personeli istihdam etme güçlüğü gibi sebeplerle kurumlardaki CBS kullanımı ikinci gruptaki seviye ulaşmakta zorluklar çekilmektedir. Özgür yazılımlar CBS alanın yaygınlaşmasında sıralanan engellerden biri olan yazılım ücretlerini düşürülmesi konusunda dikkate değer düşüşler sağlamaktadır. Kurumsal CBS uygulamaları için gerekli olan donanımsal altyapı sistemlerini yönetebilecek yazılım ürünleri benzer bir biçimde ucuza özgür yazılım dünyasında karşılanabilmektedir. Kurumsal CBS uygulamalarında gerekli sunucu ve masaüstü işletim sistemleri, veritabanı, web ve masaüstü CBS yazılımları için kendilerini yıllardır kanıtlamış özgür yazılım ürünleri kullanılabilir. Özgür yazılım ürünleri kullanıldığında sistemin toplam yazılım maliyetindeki azalma sayesinde sistemi bir bütün olarak birçok küçük ve orta ölçekli kurumlara da kurabilecek düzeyde fiyat avantajı yakalanmaktadır.

Açık standartları destekleyen ve yazılımın kullanılmasına bir sınır getirmemiş özgür yazılım ürünleri tercih edildiği zaman, bu ürünlerin CBS dünyasına sağladığı avantajlar ticari ürünlere göre daha fazla olmaktadır. Ulusal veya bölgesel mekansal veri altyapıları kurmanın amacı tüm CBS kullanıcılarına hitap etmektir. Bu amaca ulaşabilmek için mekansal veri altyapı sistemi açık standartları ve ortak kullanılan coğrafi veri formatlarını desteklemeli ve farklı yazılım türlerine çeşitli çözümler sunabilmelidir. Özgür CBS yazılımların birçoğu Open Geospatial Consortium (OGC) tarafından tanımlanan açık standartları ve uluslararası coğrafi veri formatlarını desteklemektedirler.

Yazılımının kullanılmasına herhangi bir sınır getirilmemesinin anlamı ise iki şekilde tanımlanabilmektedir. İlki, yazılımın herhangi bir bilgisayarda özel veya ticari bir amaç için istenildiği şekilde kullanılabilmesidir. Sınırsız kullanımın ikinci anlamı ise yazılımın sınırsız sayıda bilgisayara kurulum çalıştırılabilmesidir. Bu özellik de özgür CBS yazılımlarında bulunmaktadır.

CBS kullanıcılarının ticari yazılımdan özgür yazılıma geçerken bazı hususlara dikkat etmesi gerekmektedir. Bu hususlar aynı zamanda özgür yazılımların ticari ürünlere göre dezavantajlarıdır. Yeni bir CBS yazılımı kullanmak farklı bir CBS uygulama mantığını öğrenmeyi zorunlu kılabilir. Örneğin GRASS GIS’de olduğu gibi yeni bir CBS veri formatı ve veritabanı dosya yapısı öğrenmek, “location”, “mapset” gibi yeni uygulama özelliklerini kabullenmek

gerektirmektedir. Yeniden öğrenmek emek ve zaman harcanması anlamına gelmektedir. Ayrıca bu durum eğitim masrafına yol açabileceği gibi özgür bir yazılımın kurulması birkaç bilgisayar becerisi de gerektirmektedir. Yazılım kurmak özellikle masaüstü CBS yazılımları için deneyimsiz bir kullanıcının da yapabileceği basitlik düzeyine indirilmiş olsa da sunucular üzerinde web CBS yazılımlarını kurmak, bu yazılımlara ekstra özellikler kazandırmak için kurulması gereken ilave yazılımlar, sözel veritabanlarını mekansal veri formatı ile birlikte kullanmak gibi işler bazen kullanıcılara sorunlar yaratabilmektedir. Eğer özgür yazılım projesi yeterli bir olgunluğa erişmemiş ise, geniş katılımlı bir topluluk tarafından desteklenmiyorsa veya özellikle dökümantasyonun hazırlanmasına projede önem verilmiyorsa bu tip sorunlarla baş etmek için profesyonel bir servis hizmeti satın almak gerekebilmektedir. Bu yüzden hangi özgür CBS ürünün tercih edileceği çok büyük önem arz etmektedir.

Özgür yazılım dünyasında değişik alanlar için özelleştirilmiş birçok ürün vardır. Bu ürün yelpazesi incelenirken geniş katılımlı topluluk destekli bir proje ve yeterli dökümantasyonun olup olmadığına dikkat edilmelidir. Ayrıca çoğunlukla geniş katılımlı topluluk destekli projeler özel sektör firmaları tarafından da desteklenmektedir. Özgür CBS ürünlerine teknik destek hizmeti veren ticari firmalardan bu tip gereksinimler için hizmet satın alınabilmektedir. Çoğu ürünün ücretsiz olduğu bir dünyada hizmet satın almak ülkemizdeki kullanıcılara değişik gelmektedir. Aslında bu durum olağandır ve özgür yazılım hareketinin kurucusu Richard Stallman sloganı ile bu duruma vurgu yapmaktadır. Ünlü deyişinde (free as in free speech, not as in free beer) dediği gibi buradaki özgürlük, ifade özgürlüğündeki özgürlük gibidir. İfade özgürlüğü nasıl sınırsız değilse özgür yazılım dünyasındaki özgürlüğünde bazı kuralları ve sınırları vardır. Bu sınırlar GNU GPL lisansı ile çizilmiş olup hizmetin bedelsiz sunulmasını herhangi bir koşula bağlamamıştır. Bir başka deyişle yazılım geliştirmek bir haktır ve bu hakkı bu sistemde isteyen herkes kullanma özgürlüğüne sahiptir. Bu kullanma özgürlüğü ise şu anlama gelmektedir. Yeni veya türev bir yazılım geliştirirken herşeyi sıfırdan programlamaya gerek yoktur. Başkalarının geliştirdikleri yazılımlardan yararlanabilmeli, yeni özellikler ekleyip bir yazılım üretebilmeli ve başkaları da bu yazılımı isterlerse yeniden kullanabilecekleri şekilde tekrar dağıtılabilmelidir. Bu sistem GNU GPL lisansı ile yazılımın geliştirilme haklarını koruma altına almaktadır. Özgür yazılım ekosistemi varlığını bu sistem yapısına borçludur.

Daha önceden de belirtildiği gibi CBS alanı için geliştirilen oldukça fazla sayıda özgür yazılım uygulaması vardır. Özgür CBS yazılımları, www.freegis.org ve www.opensourcegis.org sayfalarında oldukça uzun bir liste halinde yayınlanmaktadır. Listelerde işlevi sınırlı yazılımlardan çok gelişmiş yazılımlara kadar çeşitli yazılım örnekleri bulmak mümkündür. Eylül 2008 itibarıyla freegis.org sayfasında 339 adet özgür CBS yazılımı sıralanmıştır. Opensourcegis.org sayfasında ise bu rakam Mayıs 2007 itibarıyla 238 proje iken 1 yıl sonra bu sayı Eylül 2008'de 256'ya çıkmıştır. Bu sayılar ve artış oranı özgür yazılım sektörünün CBS alanına ne kadar önem verdiğini ve özgür CBS yazılımı geliştirebilmek için ne kadar çok çabanın sarf edildiğini göstermektedir.

Bu kadar çok açık kaynak kodlu CBS uygulamaları arasından hangi özgür CBS yazılımının geliştirilebileceğine karar verebilmek için özgür CBS uygulamaları hakkında bilgi veren web portallarına ve araştırmalara bakılarak karar verilmeye çalışılmıştır. İlk olarak, Wikimedia sayfasında yayınlanan ve diğerlerine göre daha çok kullanılan özgür CBS yazılımları, masaüstü CBS ve diğer CBS araçları olarak iki grup altında toplanarak aşağıdaki şekilde listelenmiştir (Wikimedia 2008d):

- Masaüstü CBS (GRASS GIS, SAGA GIS, Quantum GIS, MapWindow GIS, ILWIS, uDig, gvSIG, JUMP GIS)
- Diğer CBS Araçları (FIST, Chameleon, GeoNetwork opensource, GeoTools, OpenMap, MapGuide Open Source, OpenLayers, PostGIS, TerraView, Xastir)

Özgür CBS projeleri hakkındaki gelişmeleri haber olarak veren ve bu uygulamalardan birkaçına ev sahipliği yapan başka bir web CBS portalı olan MapTools.org sayfasında sıralanan önemli uygulamalardan birkaçı üç grup altında şu şekilde toplanmıştır (MapTools 2008):

- Web Araçları (CartoWeb, Chameleon, Fusion, ka-Map, Mapbender, MapBuilder, MapServer, MapStorer, OpenLayers, OWTChart, PHP MapScript)
- Masaüstü Araçlar (GRASS GIS, OpenEV, Quantum GIS, uDig)
- Yardımcı Programlar ve Kütüphaneler (AVCE00, DGNLib, GDAL, GeoTiff, MITAB, OGR, Proj 4, Shapelib)

Diğer bir web CBS portalı olan gisdevelopment.net'in özgür CBS yazılımlar kategorisi altında Open Geospatial Konsorsuyumu tarafından tanımlanan standartlara uygun olarak geliştirilen bir kaç popüler özgür CBS yazılımı herhangi bir sınıflandırma yapılmadan alfabetik olarak listelenmektedir (GISDEVELOPMENT 2008):

- OSSIM Image Processing, degree, GRASS GIS, gvSIG, ILWIS, JUMP GIS, MapWindow GIS, PostGIS, Quantum GIS, SAGA GIS, Topology Framework .NET, uDig

Bu listede ilk sırada yer alan “OSSIM Image Processing” Uzaktan Algılama dalı için geliştirilen kısıtlı GPL (LGPL) lisanslı özgür bir yazılım ürünüdür. “Degree”, sunucu tabanlı bir web CBS uygulaması, “PostGIS”, PostgreSQL veritabanına mekansal veri özelliği kazandıran bir eklenti ve “Topology Framework .Net”, bir yazılım geliştirme kütüphanesi olup bu ürünler hariç diğerleri özgür masaüstü CBS yazılım uygulamalarıdır.

FreeGIS.org ve opensourcegis.org listelerinde yer alan yüzlerce özgür CBS uygulaması arasından zengin içeriğe sahip, geliştirilmesi kolay ve geniş topluluk desteğine sahip bir ürün olarak bilinen ve kullanılan bir masaüstü CBS uygulamasını seçebilmek için başvurulacak diğer bir kaynak ise CBS yazılımlarını detaylı bir şekilde çeşitli özelliklerine göre kıyaslayan ve 2008 yılında Steiniger tarafından yapılan bir araştırmadır. Bu araştırmada değerlendirmeye alınan özgür masaüstü CBS yazılımları şu şekilde sıralanmaktadır (Steiniger 2008):

- GRASS GIS, Quantum GIS, uDig ve türevleri; JGrass ve DivaGIS, gvSIG, Kosmo, SAGA GIS, ILWIS, MapWindow GIS, JUMP / Open Jump ve türevleri; JCS, Single JUMP, Pirol JUMP, DeeJUMP, SkyJUMP

Başka bir araştırma ise Ramsey (2007) tarafından yapılmıştır. Açık kaynak kodlu CBS ürünleri geliştirilen programlama dili ile bir uygulama veya kütüphane olup olmamasına göre sınıflandırılmıştır. Bu sınıflandırma aşağıda verilmiştir:

1. C ile geliştirilen projeler
 - a. Kütüphaneler (GDAL/OGR, Proj4, GEOS, Mapnik, FDO)
 - b. Uygulamalar (MapGuide, MapServer, GRASS GIS, Quantum GIS, OSSIM, TerraLib, GMT, PostGIS)
2. Java ile geliştirilen projeler
 - a. Kütüphaneler (JTS Topology Suite, GeoTools)
 - b. Uygulamalar (GeoServer, deegree, JUMP / OpenJUMP, Kosmo, gvSIG, OpenMap, uDig)
3. .Net ile geliştirilen projeler
 - a. Kütüphaneler (NTS, Proj.Net, SharpMap)
 - b. Uygulamalar (WorldWind, MapWindow)
4. Web CBS (internet üzerinden coğrafi veri yayınlayan ürünler)
 - a. Kütüphaneler (MapBuilder, ka-Map, OpenLayers, Mapbender, Cartoweb)
 - b. Sunucular (TileCache, FeatureServer)

Web portallarında ve araştırmalarda bahsedilen özgür CBS yazılımların bazıları değişiklik göstermekle birlikte bazıları da tüm listelerde yer almaktadır. Bu ve benzeri bilinen projeleri zengin içerik ve daha fazla işlevsellik elde etmek açısından seçmek daha uygun olacaktır. Önerileri kesin bir kural olarak kabul etmek yerine bu tür sınıflandırmalarda yer almayan ama istenilen ihtiyaca cevap verebilecek düzeyde geliştirilmiş özgür bir CBS yazılımı da seçilebilmektedir. Hangi ürün seçilirse seçilsin unutulmaması gereken nokta şudur ki; eğer kullanıcılar özgür bir CBS yazılımını kullanmayı seçerlerse, kullanıcılar yazılımı istedikleri şekilde çalıştırma ve kaynak kodunu görüp değiştirme haklarına sahiptirler.

Son olarak başvurulacak önemli bir bilgi kaynağı ise Open Source Geospatial Vakfı’dır. Bu vakfın kurulma amacı kaliteli açık kaynak kodlu mekansal yazılımlar üretmek ve topluluk destekli özgür CBS projelerinin kullanımını ve ortaklaşa geliştirilmesini sağlamaktır. Vakıf, web portalı aracılığıyla üye kullanıcılarına ve geliştiricilerine projelerin geliştirilmesine katkı yapabilecekleri ve fikirlerini paylaşabilecekleri bir ortam sağlayarak hizmet etmektedir. Vakfın kurulma amacı, özgür yazılım geliştirme hareketinin felsefesi ile aynı olduğu ve bu felsefeyi CBS alanında yaşatmaya çalıştığı için bu bildiride hangi CBS yazılımının kullanılacağı vakfın şemsiyesi altında projenin desteklenip desteklenmediğine göre karar verilmiştir. Eylül 2008’de Vakfın desteklediği projeler sınıflandırılmış bir şekilde aşağıda gösterilmektedir (OSGEO Foundation 2008):

- Web Haritalama (deegree, Mapbender, MapBuilder, MapGuide Open Source, MapServer, OpenLayers)
- Masaüstü Uygulamaları (GRASS GIS, OSSIM, Quantum GIS, gvSIG)
- Mekansal Kütüphaneler (FDO, GDAL/OGR, GEOS, GeoTool)
- Metadata Kataloğu (GeoNetwork)
- Diğer Projeler (Public Geospatial Data, Education and Curriculum)

ODTÜ Jeodezi ve Coğrafi Bilgi Teknolojileri (JCBT) ana bilim dalında yapılacak bir geliştirmede kullanılacak özgür CBS ürününe Ocak 2008’de karar verilmiştir. O tarihte Vakfın masaüstü uygulamalarında desteklediği ürünler arasından sadece GRASS GIS, Quantum GIS ve OSSIM bulunmaktadır. Ocak ile Eylül 2008 tarihleri arasında ise “degree” ve “MapGuide Open Source” projeleri web haritalamada, “gvSIG” masaüstü uygulamalarında ve “GEOS” mekansal kütüphaneler kategorisi içerisinde Vakfın şemsiyesi altına eklenen yeni projeler olarak göze çarpmaktadır.

Masaüstü uygulamaları kategorisinde yer alan OSSIM yazılımı ise özgür bir uzaktan algılama uygulamasıdır. Belirtildiği gibi gvSIG projesi Vakfın şemsiyesi altına daha sonradan eklenmiştir. Bu nedenlerle bu iki yazılım seçilmemiştir. 2003 yılının sonunda İspanyol bir firma tarafından yönetilen ve üniversitelerin desteği ile geliştirilmeye başlanan gvSIG projesi İspanya devletinin finansman desteği sayesinde hızlı bir gelişim göstermektedir. Bölgesel idare tüm bilgisayar altyapısını Linux ortamına taşımak istemektedir. Bu yazılımı geliştirmenin amacı ESRI Firması'nın ArcView 3.x serisinin özelliklerine sahip bir CBS yazılımı geliştirmek ve böylece belediyelere Linux işletim sistemi ortamında bahsi geçen CBS uygulaması kadar zengin bir uygulama sunabilmektir. Bu amaca kimi eksikliklere rağmen büyük ölçüde ulaşılmıştır. Proje her ne kadar tek bir ülke tarafından yönetiliyor olsa da topluluğa katılım uluslararası düzeydedir. Bu yazılımın dezavantajı ise geliştiriciler için yeterli dökümantasyona sahip olmamasıdır. Ayrıca yaklaşık 100 adet C++ ve Java kütüphanesine bağlıdır. Bu kadar yüksek bir bağımlılık yığına sahiplik nedeniyle gvSIG üzerinde özelleşmiş bir yazılım geliştirmek hayli zorlaşmaktadır.

Vakfın masaüstü uygulamaları kategorisinde geriye kalan iki yazılımdan biri olan GRASS GIS yerine Quantum GIS yazılımı seçilmiştir. GRASS GIS'in gücüne ve yaygın kullanılmasına rağmen neden seçilmediğini anlamak için yazılımının tarihçesine bakmak yeterlidir. GRASS GIS, US Army Construction Engineering Research lab (CERL) tarafından 1982 yılında UNIX ortamında geliştirilmeye başlanmıştır. 1980 ve 1990'larda CERL tarafından geliştirilmeye devam edilen GRASS GIS'in geliştirilme amacı o zamanlardaki ticari CBS yazılımlarına bir alternatif yaratmak ve fiyatların ucuzlamasını sağlamaktır. ESRI firmasının ArcView 3.x serisini çıkartmasıyla ucuzlayan fiyatlar ve kolay kullanım nedeniyle GRASS GIS'in geliştirilmesine son verilmiştir. 1997 yılında ise ilk açık kaynak kodlu CBS yazılımı olarak özgür yazılım ortamında geliştirilmeye başlanmıştır. GRASS GIS ve QGIS yetenekleri açısından karşılaştırıldığında üstün gelen yazılım GRASS GIS'dir. Yalnız bu kadar uzun bir yazılım geliştirme sürecine sahip GRASS GIS'in yazılım mimarisi doğal olarak eski ve karışıktır. Grafiksel arayüzünün görünümü ve pencere sistemi demode olan UNIX temellidir. Windows işletim sistemine alışmış kullanıcıların UNIX tarzındaki birçok pencere parçasından ve komut satırından oluşan bu sisteme alışması daha uzun zaman almaktadır. Gerçi GRASS GIS topluluğu bu sorunu gidermek için sürüm 6.3 ile birlikte wxPython geliştirme ortamı kullanarak kullanıcı dostu ve çarpaz platform desteği sağlayan bir sürüm geliştirmeye çalışılmaktadır. Ayrıca bu sürüm ile birlikte GRASS GIS Windows işletim sistemi ile tek başına uyumlu halde çalışabilen taşınabilir bir yazılım haline getirilmeye çalışılmaktadır. Şu ana kadar Windows işletim sistemindeki taşınabilirlik cygwin adlı bir emülator aracılığıyla sağlanabilmektedir. Bu tür yenileştirmeler, GRASS GIS kütüphanesindeki yapıyı ve fonksiyonları etkileyebileceği için üzerinde geliştirme yapan programcıların değişiklikleri daha fazla yakından izlemelerini bir zorunluluk haline getirmektedir.

QGIS ise 2002 yılında geliştirilmeye başlandığı için göreceli olarak daha güncel ve gelişmiş bir mimariye sahiptir ki bu özellik QGIS'i daha kolay ve hızlı bir şekilde yazılım geliştirmeyi mümkün kılmaktadır. Bu sayede GRASS GIS'daki gibi yazılım mimarisinde köklü değişiklikler yapmaya da gerek kalmadığı için GRASS GIS'e göre geliştirme ortamının daha kararlı olduğu söylenebilmektedir. Ayrıca GRASS GIS'in sahip olduğu tüm özellikleri barındıran araç setine bir eklenti sayesinde QGIS içinden çağrılıp çalıştırılabilmektedir. Bu eklenti ile GRASS GIS'in tüm araçlarına QGIS içinde de kullanılabileceği anlamına gelmektedir. Bununla birlikte QGIS, MapTools, gisdevelopment gibi web CBS portallarında, 2008 yılındaki Steiniger ve 2007 yılındaki Ramsey'in yaptığı çalışmalarda listelenen özgür CBS yazılımları arasında da yer almaktadır. Özetle, QGIS bu bildiride aşağıda sıralanan şu özelliklere sahip olduğu için seçilmiştir:

- Open Source Geospatial Vakfı tarafından desteklenmesi
- Geniş katılımın sağlandığı uluslararası bir topluluk tarafından geliştirilmesi
- GNU GPL lisansı olması
- Modüler bir yapıda geliştirilebilmesi
- API Kütüphanesi ile ilgili tanımlamalara webden ulaşılabilmesi
- Yeni bir CBS yazılımı veya yeni bir eklenti geliştirilebilmesi
- GRASS GIS'in tüm araçlarını QGIS içinden de kullanılabilmesi
- Detaylı bir kullanım kılavuzu olması ve dökümantasyon zenginliği
- Hızlı bir yazılım geliştirmeyi mümkün kılan Python programlama diline destek vermesi
- GRASS GIS'e göre yazılım mimarisinin daha yeni olması ve grafiksel arayüzünün daha fazla kullanıcı dostu olması
- Çarpaz platform desteği sayesinde Linux, Windows ve Mac OS X işletim sistemlerinde aynı kod ile çalıştırılabilmesi
- Aktif bir proje olması ve hızlı bir şekilde geliştirilmesi

Bu bildirinin son iki bölümünde anlatılan yazılım geliştirme örnekleri, ODTÜ Jeodezi ve Coğrafi Bilgi Teknolojileri (JCBT) bölümünde QGIS üzerinde gerçekleştirilen geliştirmelerin basit bir özetidir. Bildiride özgür masaüstü CBS

yazılımları arasından geliştirme yapmak için seçilen QGIS üzerinde örnek olarak yeni bir geliştirmenin nasıl yapılacağı sonraki bölümlerde anlatılmaya çalışılacaktır. Python programlama dili ile önce QGIS kütüphaneleri kullanılarak yeni bir CBS yazılımının nüvesini oluşturabilecek bir arayüzün nasıl geliştirileceği örnek verilerle gösterilecek daha sonra QGIS içinde çalışabilen yeni bir eklentinin geliştirilmesi açıklanacaktır.

2.1. Özgür CBS Yazılım Örneği: Quantum GIS (QGIS)

Quantum GIS (QGIS), Linux, Unix, Mac OSX ve Windows'ta çalışan kullanıcı dostu ve açık kaynak kodlu özgür bir coğrafi bilgi sistemi yazılımıdır. QGIS, GDAL kütüphanesi desteği sayesinde coğrafi raster formatları (GeoTIFF, Erdas Imagine Images, Arc/Info ASCII Grid, vb.), OGR kütüphanesi ile de vektör formatlarını (ESRI Shapefile, MapInfo, GML, vb.) desteklemektedir. QGIS, Genel Kamu Lisansı (GNU GPL) ile lisanslanmıştır. QGIS, web haritalama servisi (Web Map Service) ekleme, PostgreSQL veritabanına Postgis eklentisi sayesinde coğrafi veritabanı bağlanma, farklı projeksiyona sahip verileri otomatik olarak arka planda dönüştürüp üst üste bindirme (on the fly projection) ve ESRI shapefile formatında veri yaratma ve editleme vb. özellikler sunmaktadır.

QGIS, Açık Kaynak Kodlu Mekansal Yazılım Destekleme Vakfı'nın (Open Source Geospatial Foundation) şemsiyesi altında faaliyetlerini devam ettirmektedir. Bahsi geçen vakıf, kaliteli açık kaynak kodlu coğrafi yazılımları hazırlamak ve bunlara destek vermek için kurulmuştur. Vakfın amacı topluluk destekli projelerin kullanımını ve ortak geliştirilmesini teşvik etmektir. Vakfın web portalı kullanıcılara ve geliştiricilere fikirlerini paylaşmak ve proje gelişimine katkı sağlamak için hizmet etmektedir (OSGEO Foundation 2008).

QGIS, 2002 yılından itibaren C++ programlama dili kullanılarak geliştirilmektedir. 25 Eylül 2008 tarihindeki Ohloh portalının istatistiklerine göre QGIS'i geliştirmek için sarf edilecek tahmini iş gücü ortalama aylık maaşı 4500\$ olan 50 personelin bir yıl boyunca çalıştırılması gerekmekte olup, tahmini maliyeti ise 2,757,345 \$'dır (Ohloh 2008). Her geçen gün sarf edilen çabalar nedeniyle bahsi geçen rakamlar hızla artmaktadır. QGIS, geniş bir topluluk tarafından desteklenmekte ve hızlı bir şekilde geliştirilmektedir.

QGIS yazılımı modüler yapıda tasarlanmıştır. Yazılıma pek çok özellik, eklenti (plug-in) yapısı sayesinde genişletilebilmektedir. Bu modüler eklenti yapısı sayesinde, özgür CBS yazılımlarından biri olan GRASS'nın (Geographic Resources Analysis Support System) tüm araçları QGIS içinde çalıştırılabilmekte ve kullanıcı dostu grafiksel bir arayüz ile birlikte kullanılabilmektedir. Başka bir eklenti sayesinde, QGIS, web CBS dalında geliştirilen güçlü ve özgür bir yazılım olan MapServer için coğrafi verilerin internetten nasıl yayınlanacağını düzenleyen bir proje dosyası üretmektedir. Eklentilerin bazıları yazılım ile birlikte varsayılan olarak gelmekte iken bazı eklentiler de üçüncü şahıslarca geliştirilebilmektedir. Bu eklentiler internetten yayınlanan yazılım depolarına bağlanarak indirilip çalıştırılabilmektedir. Projenin resmi eklenti deposu (<http://spatialserver.net:3001/>) haricinde programcılar da kendi geliştirdikleri eklentileri XML formatındaki eklenti depolarını yaratarak kullanıcıların hizmetine sunabilmektedirler. Örneğin, Carson Farmer'ın QGIS eklenti deposundaki eklentilerine <http://www.geog.ubc.ca/spar/carson/cfarmerQgisRepo.xml> adresinden ulaşılabilir.

QGIS, iki yolla ve iki farklı programlama dili kullanarak özelleştirilebilmektedir. Birinci yol yeni bir CBS yazılımı üretmek iken, diğer yol ise QGIS içinde yeni bir eklenti geliştirmektir. İlk yolda, bir CBS yazılımı için gerekli olan temel yönlendirme araçlarından yakınlaşma, uzaklaştırma, kaydırma, sorgulama, tümünü göster vb. genel işlevler QGIS kütüphanesindeki ilgili sınıflar çağrılarak eklenip kullanılabilmektedir. QGIS kütüphanesinin API tanımlarına <http://doc.qgis.org/index.html> adresinden ulaşılabilir. İkinci yol ise, QGIS'in sunduğu hazır tüm araçların bulunduğu arayüz içerisinde ihtiyaç duyduğunuz bir özellik için bir eklenti geliştirmektir. Bu geliştirme ya bir programcı tarafından ya da QGIS geliştiricilerinden talep edilebilir. Talebin zamanında ve istenilen şekilde karşılanması isteniyorsa ya o topluluğa uzun bir süre katkı sağlayarak tanınan biri olmak, belirli bir ücret ödemeyi baştan kabul etmek ya da beklemeyi göze almak gerekmektedir.

QGIS üzerinde geliştirme iki farklı programlama dili kullanarak yapılabilmektedir; bunlar Python ve C++ programlama dilleridir. Python programlama dili kullanılarak geliştirme desteği sürüm 0.9 ile birlikte gelmiştir. Eylül 2008 itibari ile QGIS'in resmi sürümü 0.11'dir. 2008 yılının sonuna doğru sürüm 1.0'in yayınlanması planlanmaktadır. Sürüm 1.0 ile birlikte QGIS daha kararlı ve olgun bir seviye ulaşacağı tahmin edilmektedir.

Python dili ile bu geliştirmeyi mümkün kılan özgür ürünlerin neler olduğu ilerleyen bölümlerde anlatılmaktadır. Gerekli olan ürünler öncelikle PyQGIS, Qt, Python ve PyQt'dir. PyQGIS, QGIS topluluğu tarafından geliştirilen ve Python dili ile QGIS API'sini kullanarak yeni bir CBS yazılımı geliştirmek ve QGIS içinde yeni bir eklenti yapmak için gerekli bir bağdaştırıcıdır. Bu aracın doğru şekilde çalışabilmesi için bağımlı olduğu birkaç özgür yazılım ürünü şöyle sıralanabilir;

Nokia'nın satın aldığı Trolltech Firması tarafından geliştirilen QGIS'in çoğunlukla bağımlı olduğu yazılım geliştirme ortamı olan Qt, Riverbank Computing Limited Firması tarafından geliştirilen ve Qt'yi Python ile bağdaştırmaya yarayan PyQt4 bağdaştırıcıları ve ticari kullanımda bile lisansı özgür olarak sunulan Python programlama dilidir. Sonraki bölümlerde bu ürünler hakkında bilgi verilecektir.

2.2. Python Dili ile QGIS Geliştirme Ortamı; PyQGIS

QGIS, çoğunlukla Qt kütüphanelerine bağımlıdır. Qt kütüphaneleri ileriki bölümlerde ayrıntılı bir şekilde açıklanmıştır. PyQt 4, Qt için Python bağlayıcısıdır. Bu destek Qt sürüm 4.2'de Ekim 2006'da eklenmiştir. Bu tarihten sonra QGIS'i Python ile programlamak mümkün olmuştur. O günden beri QGIS kütüphanesi Python ile çağrılabilmek için topluluk üyeleri tarafından SIP aracı kullanılarak sürekli güncellenmektedir. Şu anda, Python QGIS API yapısı %99 oranında C++ QGIS API yapısı ile bire bir aynıdır. QGIS sürüm 0.9 ile birlikte bu özelliğe kavuşmuştur.

PyQGIS bağdaştırıcısı SIP aracına ve PyQt 4 kütüphanelerine bağımlıdır. Bu bağdaştırma işi için daha çok bilinen ve kullanılan SWIG aracı yerine Riverbank Computing Şirketi tarafından açık kaynak kodlu ortamda geliştirilen SIP aracı kullanılmaktadır. PyQt, SIP aracı kullanılarak Qt ile bağdaştırıldığı için kesintisiz bir entegrasyon ve tam anlamıyla uyumluluğu devam ettirmek için SIP aracı tercih edilmiştir. SIP aracı ile Qt PyQt'ye dönüştürülürken aynı araç ile QGIS de PyQGIS'e dönüştürülmektedir (Dobias 2008). Böylece her iki kütüphane, Python programlama dilinde buluşturulmaktadır.

Bu bağdaştırmalar sayesinde Python ile özel bir CBS uygulamaları geliştirilebilmek, QGIS içinde bazı görevleri otomatik yapabilmek ve QGIS'de eklenti oluşturabilmek mümkündür. Böylece C++ kodu kullanmadan hızlı bir şekilde yeni bir CBS yazılımı geliştirilebilir. Ayrıca, Python, QGIS içinde bir uçbirim olarak gömülü halde gelmektedir. Bu uçbirim erişimi kullanılarak, programcılar Python komutlarını QGIS içinde deneyip sonuçlarını görebilmektedirler (Dobias 2006).

QGIS için Python bağdaştırıcıları sürüm 0.9'dan itibaren standart kurulum paketi ile birlikte gelmektedir. QGIS çalıştırıldığı zaman, "Eklenti" menüsü altında Python uçbirim komutu bulunmaktadır. Eğer bu komut yoksa, Python desteği kurulmamış demektir. Python desteğini ayrıca kurmak için gerekli olan SIP sürüm 4.5 veya üzeri ve PyQt sürüm 4.2 veya üzeri yazılımlarının da kurulması gerekmektedir. Bu bağdaştırıcılar sayesinde Python ile QGIS ve Qt kullanarak, QGIS'de yeni bir Python eklentisi veya özel bir CBS uygulaması geliştirilebilir.

2.3. Qt Geliştirme Ortamı

PyQGIS bağdaştırıcısının temelini oluşturan Qt geliştirme ortamının geçmişi Linux dünyasında çok eskilere dayanmaktadır. Linux dünyasında geliştirilen uygulamaların büyük bir çoğunluğu iki farklı grafik kütüphanesi kullanılarak yazılmaktadır. Bunlardan biri GTK, diğeri ise Qt kütüphanesidir (Vardar 2006). Gnome masaüstü ortamı geliştirmek için kullanılan kütüphane GTK iken, "K" Geliştirme Ortamının temelini (K Development Environment [KDE]) Qt kütüphanesi oluşturmaktadır. Aynı kod ile farklı işletim sistemlerinde çalışabilme özelliğine sahip Qt, C++ geliştiricileri için gelişmiş ve hızlı performans sağlayan bir geliştirme ortamı sunmaktadır. Qt, Trolltech Firması tarafından 1994 yılından beri Norveç Oslo kentinde yaklaşık 250 çalışan tarafından geliştirilmektedir (Riverbank 2008a). Firmanın yazılımları 10 milyondan fazla cihazla birlikte ihraç edilmek üzere yüklenmiştir (Trolltech 2008b).

Qt'nin desteklediği işletim sistemleri arasında Microsoft Windows, Mac OS X, Linux, Sun Solaris, tüm ana ticari Unix türevleri ve gömülü Linux sistemleri sayılabilir (Trolltech 2008a). Qt ile yazılmış bir ürün, kaynak kodu üzerinde herhangi bir değişiklik yapmadan hem masaüstü işletim sistemlerinde hem de gömülü cihazlarda çalışabilmektedir. Qt'nin müşterileri arasında Adobe, Google, Skype, Lucasfilm, NASA, Walt Disney Feature Animation, IBM, Sharp, Siemens ve Mathematica gibi ünlü firmalar bulunmaktadır (Trolltech 2008b). Qt, bir geliştirme ortamı olarak tecrübesi ve geliştirilen uygulamaların zenginliği ile gücünü kanıtlamıştır.

Qt, tamamen nesneye dayalı, genişleyebilen ve modüler bir geliştirme ortamıdır. Qt, karışık ve yüksek performanslı grafiksel arayüz ve konsol uygulamaları geliştirebilmek için bir programcının ihtiyaç duyabileceği çoğu işleve sahiptir. Bu işlevler arasından veritabanı, betik, ağ, OpenGL, XML, çoklu ortam (multimedia) ve web uygulamaları sayılabilir. Qt, hızlı grafiksel uygulama, çoklu dil desteği, yerelleştirme, dokümantasyon ve çoklu platform derleme için gerekli entegre geliştirme araçlarını programcılara sağlamaktadır. Qt, Visual Studio ve Eclipse gibi yazılım geliştirme editörleriyle de entegredir. Bununla birlikte, Qt, görsel olarak arayüz tasarlama işlemlerini yerine getirebilen bir araç olan Qt Designer ürününü de kendi geliştirmekte olup programcılarının kullanımına sunmaktadır. C++ programlama dili ile geliştirilen Qt, Jambi ürünü ile de Java programlama dili ile geliştirme yapmayı mümkün kılmaktadır.

Qt kütüphanesi ilk olarak tamamen ücretli bir ürün olarak piyasa sürülmüş olup daha sonra GPL sürümü de yerini almıştır (Vardar 2006). Qt, çift lisanslıdır. Açık kaynak kodlu projelerde GNU GPL ile kullanılabilirken kapalı kaynak kodlu projelerde kullanmak için lisans ücreti ödenmek zorundadır. Qt Ticari Sürümü, geleneksel ticari yazılım dağıtımına, bedava güncellemelere ve teknik desteğe izin vermektedir. Qt Açık Kaynak Kodlu Sürümü ise Q Kamu Lisansı ve GNU GPL koşulları altında ücretsiz olarak sunulmaktadır. Yukarıda bahsedilen tüm Qt yazılım geliştirme araçları (Qt Designer, Qt Linguist, Qt Assistant, qmake) açık kaynak kodlu uygulamaları geliştirmek için de ücretsiz olarak kullanılabilirlerdir.

QGIS, Qt kütüphanesi kullanılarak geliştirilmektedir. Qt'nin farklı platformlarda çalışabilme özelliği sayesinde QGIS, hem masaüstünde hem de portatif taşınabilir el cihazlarında kullanılabilir. Qt, CBS için geliştirilmiş veya özelleştirilmiş bir kütüphane değildir. Bu nedenle, coğrafi verileri idare etmek, projeksiyon tanımları ve dönüştürmeleri yapmak, CBS'ye özgü mekansal uygulamalar gerçekleştirebilmek vb. işlevleri yapabilmek için açık kaynak kodlu diğer uygulamalar ve kütüphaneler de projeye dahil edilmiştir. QGIS'in içinde yer alan Qt haricindeki uygulama ve kütüphaneler Flex, Bison, GRASS, GEOS, GDAL/OGR, SQLITE, GSL, EXPAT, POSTGRES, Proj4, PostGIS, Expat'dır. Bunlar, başka topluluk üyeleri tarafından geliştirilen ürünler olup, QGIS fonksiyonel yapısını zenginleştirmek için kullanılmaktadır.

C++ ile geliştirilen QGIS'1, sürüm 0.9 ile birlikte Python programlama dili kullanarak da geliştirilebilmek mümkün olmuştur. Python programlama dili önceleri bir betik dili olarak geçse de yapılan geliştirmeler sayesinde günümüzde üst düzey bir programlama dili haline gelmiştir. Java gibi üst düzey programlama dillerinin avantajı geniş kütüphane desteği sayesinde az kod ile çok daha fazla iş yapabilmeleridir. QGIS'de olduğu gibi birçok kütüphanenin kullanıldığı bir yazılımda hızlı ve kolay bir şekilde ürün geliştirebilmek için Python gibi üst düzey bir programlama dili kullanmak daha avantajlı olmaktadır. Bu yüzden, QGIS'de geliştirme yaparken C++ yerine Python programlama dilini kullanılması daha uygun olacağı için bu bildirinin ilerleyen bölümünde bu programlama dili hakkında bilgi verilecektir.

2.4. Python Programlama Dili

Python geçmişte betik (script) programlama dili olarak tanımlansa da günümüzde gösterdiği gelişim sayesinde nesne temelli, çoklu platform desteği sunan, kolay anlaşılabilir söz dizilimine sahip, üst düzey bir programlama (high level programming language) dilidir.

Python programlama dili bir çok amaç için kullanılabilir; web programlama, veritabanı, grafiksel arayüz geliştirme, bilimsel uygulamalar, eğitim, ağ, yazılım ve oyun geliştirme vb.. Python, NASA, Google, YouTube, Industrial Light & Magic, Firaxis Games, Infoseek, Yahoo, RedHat, TÜBİTAK UEKAE (Pardus Geliştiricileri) vb. kurumlar tarafından kullanılmaktadır. Python'ın ana modülü C ile yazılmıştır. Modülün kısalığı ve C'nin getirdiği avantajlar nedeniyle dil hızlı çalışmaktadır. Daha sonraki kodlamalar Python dili kullanılarak yapılmıştır.

Python adını "Monty Python's Flying Circus" adlı BBC komedi serisinden almıştır. Bu komedi serisini bir programlama diline ad verebilecek kadar çok seven kişi, Python'un geliştiricisi, Guido van Rossum'dur. Rossum, Python'u 1989 yılı sonunda Hollanda'da Centrum voor Wiskunde en Informatica (CWI) isimli araştırma enstitüsündeki Amoeba dağıtık işletim sistemi üzerinde çalışırken geliştirmeye başlamıştır. Rossum, Enstitüde kullanılan ABC dili genişletilemediği ve işletim sistemi için C veya Bourne shell betikleri yerine daha iyi bir yol yaratma gereksinimine ihtiyaç duyduğu için yeni bir programlama dili geliştirmeye karar vermiştir. Ona göre, bu dil hem çok yüksek düzeyde veri tiplerini barındıran hem de sistem yönetimi düzeyinde müdahale edebilecek bir dil olacaktır. Bu yüzden, Python dili, C, C++, ABC, Modula-3, Icon ve diğer dillerin özelliklerini barındırmaktadır (Lutz 2001). Bununla birlikte C gibi sistem düzeyinde hareket etme ve farklı dillerle birlikte kullanılabilir (extensible and embedable) yeteneğine sahip olacaktır. Python dili, Şubat 1991'de USENET'te duyurulmuştur (Python Software Foundation 2008).

Lisansı 2.0 sürümünden itibaren Genel Kamu Lisansı (GPL) ile uyumlu hale getirilen Python'u kullanmak hiç bir sınırlamaya tabi değildir. Ticari bir amaç için dahi olsa ücretsiz olarak kullanılabilir. Temmuz 2008 itibarıyla yayınlanan resmi sürümü 2.5.2'dir. Bu sürümün kaynak koduna veya kurulum dosyalarına <http://www.python.org/download> adresinden ulaşılabilir. 17 yaşında yeterince kararlı, olgun, güçlü ve esnek bir dildir. Öğrenme kolaylığı nedeniyle öğrenilmesi gereken ilk programlama dili olarak önerilmektedir. Destekleyen kitle sayısının büyüklüğü tahmin bile edilememektedir. Bir çok Linux dağıtımında (Fedora, Debian, Pardus vb.) kurulu olarak gelmektedir. TIOBE Programlama Topluluğu'nun her gün güncelleştirdiği dillerin popülerlik listesine göre Python ilk on dil içerisinde. 2005 yılında 8., 2006 ile 2008 yılları arasında 7. sırada yer almaktadır. Temmuz 2008 itibarıyla sıralama şu şekildedir: 1. Java, 2. C, 3. Visual Basic, 4. PHP, 5. C++, 6. Perl, 7. Python, 8. C#, 9. Ruby ve 10. Delphi (TIOBE Software).

Python, özgür programlama dili olduğu için kaynak kodu incelenerek nasıl çalıştığı öğrenilebilir. Kaynak koduna erişilebildiği için diğer programlama dilleri ile birlikte iki şekilde kullanılabilir. Birinci yol, farklı dillerde yazılmış (C, C++, Fortran, Java (Jython ile), .Net (IronPython ile)) kütüphaneleri, Python dili ile çağrılabilir. Bu genişletilebilirlik özelliği, Python programlama dilinin en belirleyici geliştirilme amaçlarından biridir. İkinci yol ise, Python derleyicisini başka bir uygulama içerisinde gömülü olarak çalıştırmaktır. Bu işlemin faydası, bir uygulama içinde Python programlama dilinin getirdiği faydaları başka bir dil ile birlikte kullanabilmektir.

C veya C++ kütüphanelerini Python ile birlikte kullanabilmek için programcının işini kolaylaştıran hazır özgür yazılımlar arasında SIP ve SWIG sıralanabilir. Genişletilebilirlik özelliği için SWIG en sık kullanılan yazılımlardan biri olup, <http://www.swig.org> adresinden hizmet vermeye devam etmektedir. SIP ise Riverbank Computing Limited firması tarafından Qt yazılım geliştirme ortamını Python'un erişimini sağlamak için geliştirilen özgür bir yazılımdır.

Python programlama dili Türkiye'de giderek daha fazla yaygınlaşmaya başlamıştır. Örneğin, TÜBİTAK UEKAE'nin geliştirdiği Pardus, Python programlama dili kullanılarak geliştirilmektedir. CBS dalında ise dünya devlerinden biri olan ESRI firmasının geliştirdiği kapalı kaynak kodlu ArcGIS ticari yazılımının betik dili Python'dur. ESRI betik dili olarak masaüstü uygulamalarında ArcGIS sürüm 8.0'dan itibaren Arc Macro Language'den (AML) vazgeçip, sahip olduğu potansiyel nedeniyle Python'u kullanmayı seçmiştir. Thuban ise, temel programlama dili olarak sadece Python'u kullanan özgür CBS yazılımlarından biridir.

Python programlama söz dizilimi kod yazmayı zorlaştıran noktalama işaretlerinden her hangi birini sıklıkla kullanmayı gerektirmemektedir. Bu yüzden anlaşılır ve kolaydır. İngilizce dil yapısına çok yakındır. C++ ile karşılaştırıldığında çok daha az kod ile fazla iş yapabilmeyi sağlamaktadır. Java gibi bellek yönetim (garbage collection) araçlarını ve çeşitli grafiksel arayüz geliştirme ortamlarını sunmaktadır. Zengin bir geliştirme kütüphanesi ile birlikte gelmektedir. Yazılan kodlar derlemeye gerek kalmadan çalışabilmektedir. Python, her türlü işletim sisteminde; Windows, Linux/Unix, Mac OS X, OS/2, Amiga, Palm Handhelds ve Nokia cep telefonlarında çalışabilmektedir.

Python, yorumlanan bir programa dili (interpreted) olduğu için derlenen dillere göre biraz daha yavaş çalışabilir. Bu dezavantaj, günümüzün güçlü bilgisayarları sayesinde 1-2 saniyeden daha fazla bir fark yaratmamaktadır. Büyük hacimli verilerle uğraşanlar, performans beklentisi içinde olanlar ve gömülü sistemlerde kullanmayı düşünenler için C'ye göre biraz yavaş çalışsa da Python, yazılım geliştirme konusunda programcılar için getirdiği kolaylıklar ve hız nedeniyle C/C++'a göre tercih edilen bir programlama dilidir. Söz konusu yavaşlık sadece programın çalışma performansı ile ilgilidir. Python modülleri en iyi şekilde performans sağlayacak biçimde defalarca test edilmekte ve binlerce programcı tarafından sürekli olarak gözden geçirilmektedir. Bu tecrübe birikimine karşı, bir kişi tarafından yazılan C kodunun, Python modüllerine göre performans avantajı yakalayabilme ihtimali düşüktür.

Python'un bir başka kullanım amacı ise hızlı prototip geliştirmektir. Karmaşık yazılımlar ilk önce Python ile hızlı bir şekilde yazılabilmekte daha sonra da C'ye çevrilmektedir. Python dilini kullanarak hızlı yazılım geliştirme (Rapid Application Development) metodolojisinin ilkeleri kolaylıkla gerçekleştirilebilmektedir. Günümüzde, bilgisayarların işlem yapabilme hızları düştüğünde, makinelerin çalışma verimliliği yerine programcının verimliliğine öncelik veren bir programlama dili ile yazılım geliştirmek daha akıllıca olmaktadır.

2.5. Python Dili ile Qt Geliştirme Ortamı; PyQt 4

PyQt, Qt yazılım geliştirme ortamındaki C++ kütüphanelerine Python programlama dili kullanarak erişimi sağlayan bağlayıcı bir kütüphanedir. PyQt, Riverbank Computing Limited Şirketi (www.riverbankcomputing.co.uk) tarafından geliştirilmektedir. Şirketin uzmanlık alanı açık kaynak kodlu yazılım teknolojileridir.

Qt gibi, PyQt de ikili lisans sistemine sahiptir. PyQt, özgür yazılım projeleri için GPL, kapalı kaynak kodlu projeler için ticari yazılım lisansı ile lisanslanmıştır. Başka bir deyişle, Python dili kullanılarak Qt kütüphanelerinde ticari bir ürün geliştirilecekse, PyQt gerekli olduğu için, hem Trolltech Şirketi'ne (Qt geliştiricisi) hem de Riverbank Şirketi'ne lisans ücreti ödenmelidir (Riverbank 2008b).

PyQt, Qt kütüphanelerin sağladığı tüm imkanları ve Python dilinin getirdiği tüm avantajları birleştirmektedir. PyQt, güçlü, esnek ve hızlı grafiksel arayüz geliştirmek için idealdir. PyQt, Python ve Qt'nin kurulduğu tüm işletim sistemlerinde çalışmaktadır. Windows 98'den Vista'ya kadar tüm sürümleri, çoğu Unix ve X11 ile çalışan türevleri, Mac OS X, Linux, Solaris ve HP-UX işletim sistemlerini destekler. PyQt ile geliştirilen ürünlerin genel görünimleri sıralanan

tüm işletim sistemlerinde kendi doğal görünüşleri ile uyum göstermektedir. Bu uyumu sağlamak için gerekli tüm işlemleri PyQt arka planda programcıya yük getirmeden kendisi üstlenmiştir (Riverbank 2008a).

PyQt geliştiricisi tüm programın arayüzünü kodlayarak oluşturabileceği gibi Qt bölümünde anlatılan araçları da kullanabilmektedir. Grafiksel arayüzü görsel olarak hazırlayan Qt Designer aracı XML yapısında bir proje dosyası üretmektedir. PyQt, bu XML dosyasını Python diline çevirirken, aynı durum simgeler, resimler ve çeviri dosyaları için de geçerlidir. Qt ortamında simge ve resimler XML kaynak koleksiyonu dosyasında tanımlanırken PyQt araçları ile Python koduna dönüştürülmektedir. Bunun için "pyrcc4", Qt Designer dosyalarını dönüştürmek için "pyuic4", çeviri dosyaları için "pylupdate4" araçları kullanılmaktadır (Riverbank 2008b).

PyQt geliştirme ortamı, Qt kütüphanesindeki 600'den fazla sınıfa Python dili kullanarak erişebilmeyi mümkün kılmaktadır. Ayrıca PyQt sayesinde geliştiriciler işletim sistemleri ve Windows sürümleri arasındaki farklarla uğraşmak zorunda kalmamaktadırlar (Riverbank 2008a). C veya C++ kütüphanelerini kullanma yeteneği sayesinde Python hızlı yazılım geliştirme metoduna uygun bir programlama dili olarak öne çıkmaktadır. PyQt ise Qt ve Python'un tüm avantajlarını birleştirmektedir. Bu sayede bir geliştirici Qt'nin tüm gücünü ve Python dilinin basitliğini bir arada kullanma şansını yakalamaktadır.

2.6. QGIS Eklentileri ve Eklenti Depoları

QGIS eklentiler ile genişleyebilecek şekilde modüler bir yapıda tasarlanmaktadır. QGIS'de birçok özellik eklentilerle hayata geçirilmektedir. Eklentiler ikiye ayrılmaktadır; ana eklentiler ve kullanıcı eklentileri. Ana eklentiler QGIS geliştirme ekibi tarafından geliştirilirler ve QGIS dağıtımının bir parçasıdır (QGIS 2008). Şu ana kadar bu ana eklentilerinin çoğu C++ kullanılarak yazılmıştır. Kullanıcı eklentileri ise hem C++ hem de Python ile geliştirilebilmektedir.

Yeni özellikler, QGIS geliştirme ekibine üye olmayan programcılar tarafından da geliştirilebilmekte olup QGIS'le birlikte kullanılabilir imkânına sahiptir. Bu yol ile geliştirilen eklentilerin dağıtımı, QGIS'in ana kodu içine dahil edilemediği için bu üçüncü şahıslara ait eklentiler standart kurulum ile birlikte gelmemektedir. Bu dağıtım engeleni aşmak için geliştirilen alternatif yollardan biri eklenti depolarının kurulması ve depolara internetten erişim sağlanmasıdır. Eklenti depolarını yayınlamak için basit bir XML dosyası oluşturup internete koymak yeterli olmaktadır. Bir programcıya ait QGIS eklenti deposunu yayınlarken kullandığı XML dosyası örnek olarak aşağıda gösterilmektedir (Şekil 4).

```
- <plugins>
  - <pyqgis_plugin name="Create Point Distance Matrix" version="0.01">
    <description>Measure distances between two point layers, and output results as a) Square
      distance matrix, b) Linear distance matrix, or c) Summary of distances. Can limit distances to
      the k nearest features.</description>
    <homepage>http://www.geog.uvic.ca/spar/carson</homepage>
    <file_name>pointdistance.zip</file_name>
    <author_name>cfarmer</author_name>
    <download_url>http://www.geog.uvic.ca/spar/carson/tools/pointdistance.zip</download_url>
  </pyqgis_plugin>
</plugins>
```

Şekil 4: QGIS eklenti deposunu oluşturmada kullanılan örnek bir XML kodu

Eklenti depolarına QGIS ile internetten bağlantı kurulup eklentiler indirilmekte ve sıkıştırma dosyası açılarak çalıştırılmaktadır. Güvenlik açısından riskli bir method olmasına rağmen kullanılıp kullanılmayacağı QGIS'de yeni özellikleri kullanma isteği ve eklenti deposunun geliştiricisine duyulan güven ile doğru orantılıdır. İnternette çalıştırılabilir bir kodu indirip kullanmak tehlike içermekte olup bu konuda ne QGIS topluluğu ne de depo sahipleri herhangi bir garanti vermemektedirler.

Python ile geliştirilen QGIS kullanıcı eklentilerini çalıştırmak için derlemeye ihtiyaç yoktur. QGIS açılırken, sistem Python eklenti kodlarını yorumlamaktadır. Eğer sistemde bir hata bulunursa, hatanın nerede olduğu otomatik olarak belirlenip eklenti çalışmaz. Python eklentilerini kaynak kodu ile veya sadece ikilik kod ile QGIS'de çalıştırabilmek mümkündür. Eklentileri sadece ikilik kod ile dağıtmak GPL lisansına aykırı bir durumdur. QGIS, eklentilerin hangi dosya tipi ile çalıştırıldığını kontrol eden bir mekanizmaya sahip değildir. GNU GPL lisanslı bir ürün kullanarak

geliştirme yaparken eklentinin kaynak kodunun dağıtılması gerektiği lisans sistemi ile zorunlu tutulmuştur. Eklentilerin kaynak kodunun dağıtılıp dağıtılmıyacağı programcılar için sosyal bir sorumluluk getirmektedir. Özgür yazılımcılar bu sosyal olguya geçmişten günümüze kadar sahip çıkmışlardır ve bunu devam ettirecekleri açıktır.

2.7. QGIS Kullanılarak Yeni CBS Yazılımı Geliştirme

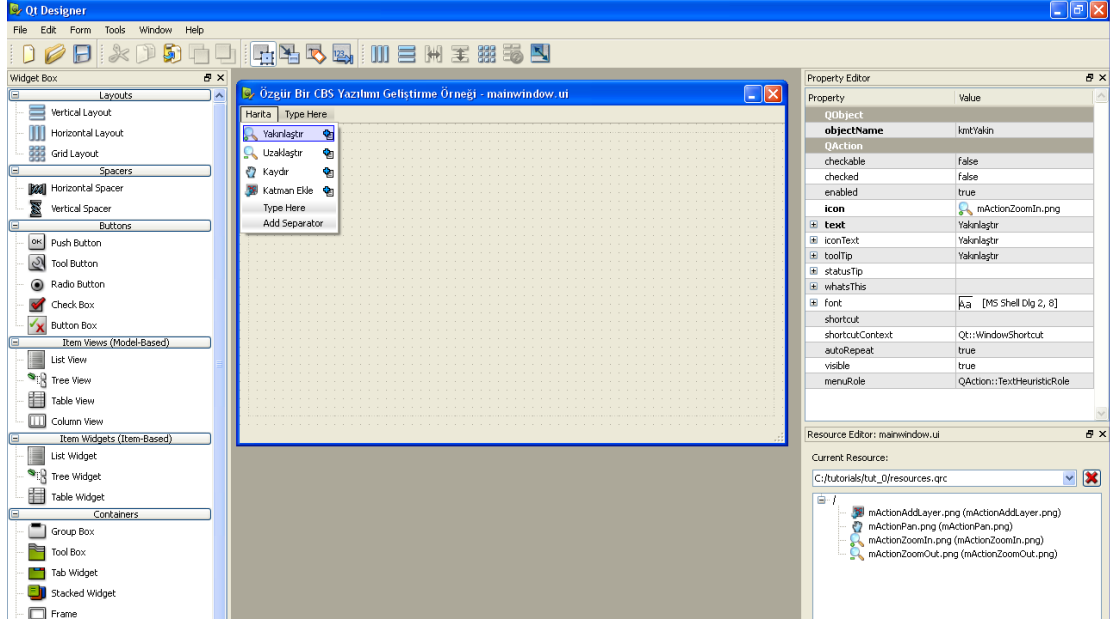
Bu bölümde Python dili ile QGIS kullanılarak nasıl yeni bir CBS yazılımı geliştirileceği basit bir örnek üzerinde anlatılacaktır. Bu yazılımı gerçekleştirebilmek için sırasıyla Qt, PyQt, Python ve QGIS ürünlerinin sisteme kurulması gerekmektedir. Bahsi geçen özgür yazılım ürünleri ücretsiz olarak dağıtıldığından kurmak için bir ücretin ödenmesi gerekmemektedir. Örnek olarak geliştirilen yazılımda birkaç yönlendirme aracı; “yakınlaş”, “uzaklaş”, “kaydır” ve “katman ekleme” araçları bulunmaktadır. Katman ekleme aracı ESRI SHP formatındaki coğrafi veriyi okuyup görüntülemektedir. Örnek veri olarak Türkiye’deki il merkezlerini gösteren nokta tabanlı SHP verisi kullanılmıştır. Bu araçlara hem menü aracılığıyla hem de araç çubuğu ile erişilebilmektedir.

Öncelikle böyle bir geliştirmeyi yapabilmek için iki adet dosyanın hazırlanması yeterli olmaktadır. Bu dosyalardan biri arayüzün nasıl oluşturulduğunu tarif ederken, diğeri ise yazılımın ne iş yapacağını belirlemektedir. Arayüzdeki komutları ikonlar ile gösterilmek için üçüncü bir dosya olarak da kaynak koleksiyon dosyası kullanılmıştır. Arayüz komut satırları ile yazılarak oluşturulabileceği gibi Qt Designer gibi arayüz geliştirme programı da kullanılarak yapılabilir. Bu yazılım ile ara yüz oluşturmak daha az vakit aldığı için bu yol tercih edilmiştir. Qt Designer çalıştırıldıktan sonra yeni form olarak Ana Pencere seçeneği seçilmiştir. Araç çubuğu üzerinde “Harita” adında yeni bir menü oluşturulduktan sonra menünün içine Yakınlaştır, Uzaklaştır, Kaydır ve Katman Ekle satırları eklenmiştir. Arayüzdeki her objeye tekil bir isim vermek gerekmektedir. Her komut için obje adı olarak sırasıyla “kmtYakin”, “kmtUzak”, “kmtKaydır” ve “kmtEkle” tanımlamaları verilmiştir. Qt Designer proje dosyası “mainwindow.ui” olarak kaydedilmiştir. Program proje dosyası olarak bir XML çıktısı üretmektedir. Proje dosyasını bir editör ile açıp bakılabilir. Arayüzün nasıl oluşturulacağını tarif eden mainwindow.ui adlı proje dosyasının içeriğinin bir kısmı Şekil 5’te gösterilmektedir. Dosyanın içeriği örnek olarak gösterildiği için kodun tümünü koyma gereği duyulmamıştır.

```
<ui version="4.0" >
<author></author>
<comment></comment>
<exportmacro></exportmacro>
<class>MainWindow</class>
<widget class="QMainWindow" name="MainWindow" >
<property name="geometry" >
<rect>
<x>0</x>
<y>0</y>
<width>579</width>
<height>330</height>
..... kod devam ediyor
```

Şekil 5: Qt Designer proje dosya kodun bir kısmı; mainwindow.ui

Qt Designer içinde yeni bir kaynak koleksiyon dosyası oluşturulmuştur ve resources.qrc adı verilerek kaydedilmiştir. Dosyanın içine komutları araç çubuğunda simgesel olarak temsil edeceği düşünülen dört adet ikon dosyası eklenmiştir. Komutlar ile ikonlar özellikler menüsü kullanılarak ilişkilendirilmiştir. Son olarak Qt Designer’daki “Çerçeve” objesini arayüzün üstünü kaplayacak şekilde yerleştirip obje adı olarak haritaEkrani verilmiştir. Çerçeve objesi coğrafi verileri göstermek için kullanılacaktır. Yeni CBS yazılımın arayüzünün son hali Şekil 6’da aşağıda gösterilmektedir.



Şekil 6: Yeni CBS yazılımının Qt Designer’da hazırlanmış arayüzünün son hali

Qt kaynak koleksiyon dosyasının çıktı formatı da XML’dir. Dosyada ikonların adları ve bulunduğu klasörlerin yolu listelenmektedir (Şekil 7). Bu iki dosya da PyQt’nin dönüştürme araçları kullanılarak XML formatından Python koduna dönüştürülmelidir. Bunun için ui dosyaları “pyuic4” komutu ile qrc dosyaları ise “pyrcc4” ile aşağıda gösterilen söz dizilimi ile Python koduna dönüştürülmektedir (Şekil 8).

```
<RCC>
<qresource prefix="/" >
  <file alias="mActionAddLayer.png">../images/mActionAddLayer.png</file>
  <file alias="mActionPan.png">../images/mActionPan.png</file>
  <file alias="mActionZoomIn.png">../images/mActionZoomIn.png</file>
  <file alias="mActionZoomOut.png">../images/mActionZoomOut.png</file>
</qresource>
</RCC>
```

Şekil 7: Qt kaynak koleksiyon dosya kodu; resources.qrc

```
pyuic4 -o mainwindow_ui.py mainwindow.ui
pyrcc4 -o resources_rc.py resources.qrc
```

Şekil 8: PyQt dönüştürme araçlarının kullanımı

Py dosyaları Python kodu olup içine bir editör yardımı ile açıp bakılabilmektedir. Aşağıdaki şekillerde önceki iki XML dosyasının Python koduna dönüşmüş halleri gösterilmektedir (Şekil 9 ve 10). Yerden tasarruf etmek için kodların sadece baştaki kısımları listelenmiştir.

```
from PyQt4 import QtCore, QtGui
class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(QtCore.QSize(QtCore.QRect(0,0,579,330).size()) ....
        self.centralwidget = QtGui.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.gridlayout = QtGui.QGridLayout(self.centralwidget)
```



```
self.gridlayout.setMargin(9)
```

..... kod devam ediyor

Şekil 9: Qt Designer proje dosya kodun Python koduna dönüştürülmüş halinin bir kısmı; mainwindow_ui.py

```
from PyQt4 import QtCore
qt_resource_data = "\
\x00\x00\x04\xe9\
\x89\
\x50\x4e\x47\x0d\x0a\x1a\x0a\x00\x00\x00\x0d\x49\x48\x44\x52\x00\
\x00\x00\x16\x00\x00\x00\x16\x08\x06\x00\x00\x00\x00\x00\x00\x00\
\x00\x00\x00\x04\x67\x41\x4d\x41\x00\x00\x00\x00\x00\x00\x00\
\x00\x00\x00\x19\x74\x45\x58\x74\x53\x6f\x66\x74\x77\x61\x72\x65\
..... kod devam ediyor
```

Şekil 10: Qt kaynak koleksiyon dosyasının Python koduna dönüştürülmüş halinin bir kısmı; resources_rc.py

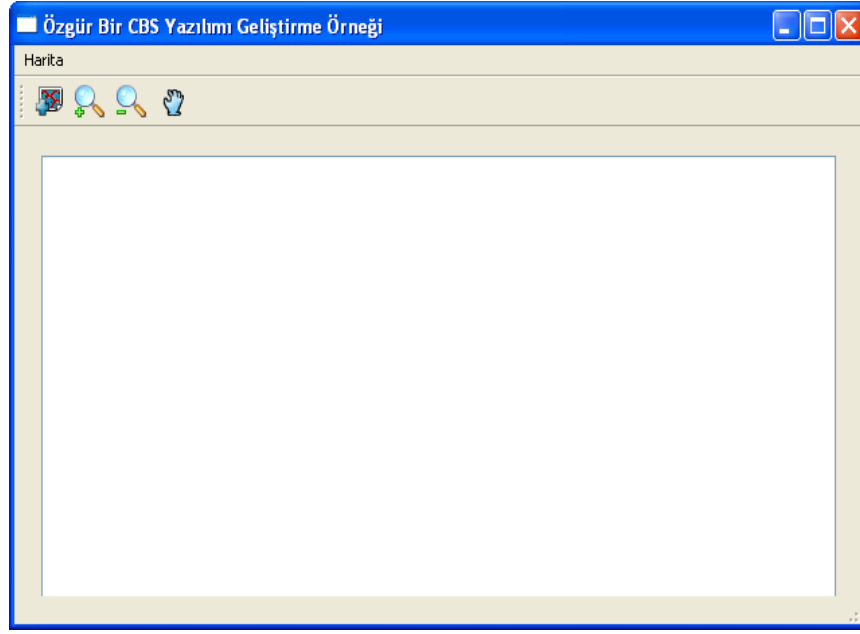
Buraya kadar tarif edilen tüm işlemler arayüzün oluşturulması için yapılmıştır. Bundan sonra yazılımın ana işlevini yapması sağlayan kod doğrudan Python dili kullanılarak geliştirilecektir. Python kodunun girintili yazılma özelliğine uymak gerekmektedir. Aksi takdirde yazılım çalışmayacaktır. Girintilerin doğru olup olmadığı bir editör yardımı ile kontrol edilebilmektedir. Aşağıdaki Şekil 11’de örnek yazılımın ana modülünü oluşturan Python kodu gösterilmektedir. Bu kod örneği QGIS Sürüm Müdürü Tim Sutton’un C++ ile QGIS’de özel bir CBS yazılımı geliştirmek için hazırladığı eğitici kodun Martin Dobias tarafından Python’a taşınması ve bazı yerlerinin benim tarafımdan değiştirilmesi ile oluşmuştur. Bu kodu mainwindow.py adlı bir dosya içine kaydettikten sonra Python derleyicisi kullanılarak çalıştırılmıştır. Çalıştırıldığında Şekil 12’de gösterildiği gibi bir görüntü ile karşılaşılacaktır.

```
1 from PyQt4 import QtCore, QtGui
2 from mainwindow_ui import Ui_MainWindow
3 from qgis.core import *
4 from qgis.gui import *
5 import sys
6 qgis_prefix = "C:/Program Files/Quantum GIS"
7 class MainWindow(QtGui.QMainWindow, Ui_MainWindow):
8     def __init__(self):
9         QtGui.QMainWindow.__init__(self)
10        # Qtiyi hazırla
11        self.setupUi(self)
12        # harita ekranını oluştur
13        self.canvas = QgsMapCanvas()
14        self.canvas.setCanvasColor(QtGui.QColor(255,255,255))
15        self.canvas.enableAntiAliasing(True)
16        self.canvas.useQImageToRender(False)
17        self.canvas.show()
18        # düğmeleri ana pencere içinde yerleştir
19        self.layout = QtGui.QVBoxLayout(self.haritaEkranı)
20        self.layout.addWidget(self.canvas)
21        # düğmelerin davranışlarını belirle
22        self.connect(self.kmtEkle, QtCore.SIGNAL("triggered()"), self.addLayer)
23        self.connect(self.kmtYakin, QtCore.SIGNAL("triggered()"), self.zoomIn)
24        self.connect(self.kmtUzak, QtCore.SIGNAL("triggered()"), self.zoomOut)
25        self.connect(self.kmtKaydir, QtCore.SIGNAL("triggered()"), self.pan)
26        # araç çubuğunu oluştur
27        self.toolbar = self.addToolBar("File");
28        self.toolbar.addAction(self.kmtEkle);
29        self.toolbar.addAction(self.kmtYakin);
30        self.toolbar.addAction(self.kmtUzak);
31        self.toolbar.addAction(self.kmtKaydir);
32        # gösterim araçlarını oluştur
```

```
33 self.toolPan = QgsMapToolPan(self.canvas)
34 self.toolPan.setAction(self.kmtKaydir)
35 self.toolZoomIn = QgsMapToolZoom(self.canvas, False)
36 self.toolZoomIn.setAction(self.kmtYakin)
37 self.toolZoomOut = QgsMapToolZoom(self.canvas, True)
38 self.toolZoomOut.setAction(self.kmtUzak)
39 def zoomIn(self):
40     self.canvas.setMapTool(self.toolZoomIn)
41 def zoomOut(self):
42     self.canvas.setMapTool(self.toolZoomOut)
43 def pan(self):
44     self.canvas.setMapTool(self.toolPan)
45 def addLayer(self):
46     """add a (hardcoded) layer and zoom to its extent"""
47     layerPath = "../data/City_Centres.shp"
48     layerName = "test"
49     layerProvider = "ogr"
50     # katmanı oluştur
51     layer = QgsVectorLayer(layerPath, layerName, layerProvider)
52     if not layer.isValid():
53         return
54     # katmanı registrye ekle
55     QgsMapLayerRegistry.instance().addMapLayer(layer);
56     # harita ekranı katman alanı olarak ayarla
57     self.canvas.setExtent(layer.extent())
58     # harita ekranına katmanı ayarla
59     cl = QgsMapCanvasLayer(layer)
60     layers = [cl]
61     self.canvas.setLayerSet(layers)
62 def main(app):
63     # qgis kütüphanesini ayarla
64     QgsApplication.setPrefixPath(qgis_prefix, True)
65     QgsApplication.initQgis()
66     # ana pencereyi oluştur
67     wnd = MainWindow()
68     wnd.show()
69     # çalıştır
70     retval = app.exec_()
71     # çık
72     QgsApplication.exitQgis()
73     sys.exit(retval)
74 if __name__ == "__main__":
75     # Qt uygulamasını oluştur
76     app = QtGui.QApplication(sys.argv)
77     main(app)
```

Şekil 11: Yeni CBS yazılımın ana Python kodu; mainwindow.py

Komutları deneyerek programın nasıl çalıştığı test edilebilmektedir. Katman ekle komutuna basıldığı zaman ise yazılım Türkiye'deki il merkezlerinin konumunu gösterecektir.



Şekil 12: Yeni CBS yazılımının görünümü

Yukarıda anlatılan işlemler biraz uzun ve zor görünse de böyle özel bir CBS yazılımını geliştirmek için sadece 62 satırdan oluşan (yorumlar hariç) bir kodun kullanıldığının farkına varılması gerekmektedir. Özelliği bu kadar ile sınırlı bir yazılımı bu şekilde geliştirmek için gereken emek ile hiç bir kütüphane kullanmadan sıfırdan yazmaya başlayarak harcanacak emek karşılaştırıldığında ortaya çıkacak büyük fark nedeniyle ne kadar doğru bir yol izlenmiş olduğu daha kolay anlaşılacaktır.

Bu sistemi anlamak için harcanacak zaman ile sıfırdan başlayarak yazmak için gereken zaman karşılaştırma bile kabul etmez. Bir kere bu sistemin nasıl geliştirildiği öğrenildiği zaman istenilen özellikleri yazılıma kazandırmanın ne kadar etkili ve çabuk olacağı aşikardır. Özgür yazılımların sağladığı en büyük avantaj istenildiği şekilde yazılımı yeni baştan geliştirebilme esnekliğidir.

2.8. QGIS İçinde Yeni Eklenti Geliştirme

QGIS'in kişiselleştirilmesi eklenti bazındaki geliştirmeler sayesinde yapılmaktadır. Geliştirilen eklentiler QGIS arayüzünün içinde çalıştırılarak kullanılmaktadır. Bu bölümde anlatılan eklenti QGIS içinden Qt'nin mesaj kutusu sınıfını çağırıp bir karşılama mesajı gösterecektir.

Python eklentilerinin çalıştırılabilmesi için belirli bir klasör yolunun altında bulunması gerekmektedir. Python eklentilerinin varsayılan klasör yolu Windows işletim sistemi için "C:\Program Files\Quantum GIS\python\plugins" dir. Kurulumun nasıl yapıldığı değişmekle birlikte Python eklenti klasör yolu Linux için "/usr /share /qgis /python /plugins/" veya "/usr /local /share /qgis /python /plugins/" olabileceği gibi "~/qgis /python /plugins/" klasörü de kullanılabilir. Eklenti çalıştırmak için bahsi geçen klasör yollarından birinin altına yeni bir klasör açılıp Şekil 13'te gösterildiği gibi hazırlanan en az beş dosya bu klasörün altına kopyalanması gerekmektedir (Juergen 2008).

Bu klasör içindeki dosyalardan biri olan "icon.png" adlı dosya resim formatındaki herhangi bir ikon olabilir. Önceki bölümde anlatıldığı gibi "resources.qrc" dosyası ikonun dizin yolunu tutmaktadır ve PyQt araçları kullanılarak "resources.py" dosyası olarak Python koduna dönüştürülmüştür. Geriye kalan iki Python dosyasından ilki olan "__init__.py" dosyası eklenti yöneticisinin eklentileri aktif hale getirirken kullanılan ilk tanımları ayarlamak için hazırlanmaktadır (Şekil 14'deki ilk yedi satır). Bu dosya "iface" referansı ile QGIS API'sine ilk iletişimi kurmakta ve bu iletişimi son Python dosyası olan "plugin.py" göndererek iletişimi devam ettirmektedir (Şekil 14'deki satır 8 ile 11 arası). Ayar dosyası olan "__init__.py" dosyası ile eklentiye bir isim verilmekte, ne işe yaradığı yazılmakta ve sürüm numarası verilmektedir. Tüm bu tanımlamalar eklenti yönetici tarafından kullanıcıya bilgi verecek şekilde gösterilip kullanılmaktadır. Önceki bölümde anlatıldığı için yeni eklentide herhangi bir arayüz tasarlanmamıştır.


```
Python Eklenti Klasör Yolu/  
yeniEklenti/      __init__.py  
                  plugin.py  
                  resources.qrc  
                  resources.py  
                  icon.png
```

Şekil 13: Yeni eklentinin klasör içeriği

```
1      # -*- coding: utf-8 -*-  
2      def name():  
3          return "ilk Eklentim"  
4      def description():  
5          return "Deneme amaçlı bir eklentidir."  
6      def version():  
7          return "Sürüm 0.1"  
8      def classFactory(iface):  
9          # plugin.py dosyasından TestPlugin sınıfı yükleniyor.  
10         from plugin import TestPlugin  
11         return TestPlugin(iface)
```

Şekil 14: Yeni eklenti için __init__.py dosyasındaki Python kodu

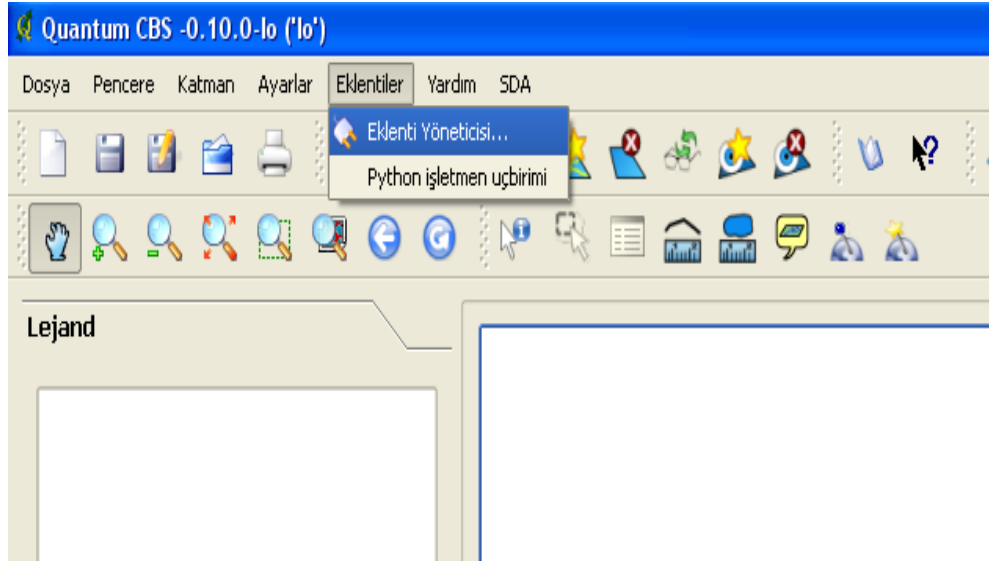
“Plugin.py” dosyasının içeriği aşağıdaki Şekil 15’te gösterilmekte olup eklentinin ana işlevi bu dosya tarafından tariflendirilmektedir. Şekil 15’teki 2 ile 4. satırlar arasındaki ifadeler QGIS ve PyQt aracılığıyla Qt kütüphanelerini çağırılmaktadır. Her ne kadar eklenti bir arayüze sahip olmasa da Qt’nin grafiksel arayüz kütüphanesi “ilk eklenti” adlı komuta bir ikon tanımlama işi bu kütüphane sayesinde yapıldığı için arayüz kütüphanesi de çağırılmıştır. TestPlugin sınıfının ilk ayarları yapılandırılırken QGIS arayüz referansı TestPlugin sınıfı içinde kaydedilmektedir (Satır 10). Daha sonra Qt terminolojisi olan Sinyal ve Slot bağlantısı ile düğmeye davranış şekli ve hangi fonksiyonun çalıştırılacağı belirlenmiştir (Satır 14). Mesaj kutusunu çağırma işi sağlanan düğme araç çubuğunda (Satır 15) ve bir komut satırı olarak da “Eklentim” menüsü içinde oluşturulmuştur (Satır 16). Ayarlamalar bittikten sonra düğmeye tıkladığında çalışan ve eklentinin ana işlevini belirleyen fonksiyon 22. satır ile birlikte başlamaktadır. Bu fonksiyonun yaptığı iş ise bir mesaj kutusu çağırıp “ ilk eklentimi başarıyla çalıştırdım” diye bir mesajı göstermektir. Bu mesajı görmek için geriye kalan tek şey eklenti çalıştırmaktır.

```
1      # -*- coding: utf-8 -*-  
2      from PyQt4.QtCore import *  
3      from PyQt4.QtGui import *  
4      from qgis.core import *  
5      # resources.py dosyasındaki ikonları çağır  
6      import resources  
7      class TestPlugin:  
8          def __init__(self, iface):  
9              # QGIS arayüz referansını kaydet  
10             self.iface = iface  
11             def initGui(self):  
12                 # eklentinin başlangıç ayarını yap  
13                 self.action = QAction(QIcon(":/icon.png"), "ilk eklenti", self.iface.getMainWindow())  
14                 QObject.connect(self.action, SIGNAL("activated()"), self.run)  
15                 # araç çubuğuna düğme ekle ve menüyü oluştur  
16                 self.iface.addToolBarIcon(self.action)  
17                 self.iface.addPluginMenu("&Eklentim", self.action)  
18             def unload(self):  
19                 # eklenti menüsünü ve ikonu araç çubuğundan kaldır  
20                 self.iface.removePluginMenu("&Eklentim", self.action)  
21                 self.iface.removeToolBarIcon(self.action)
```

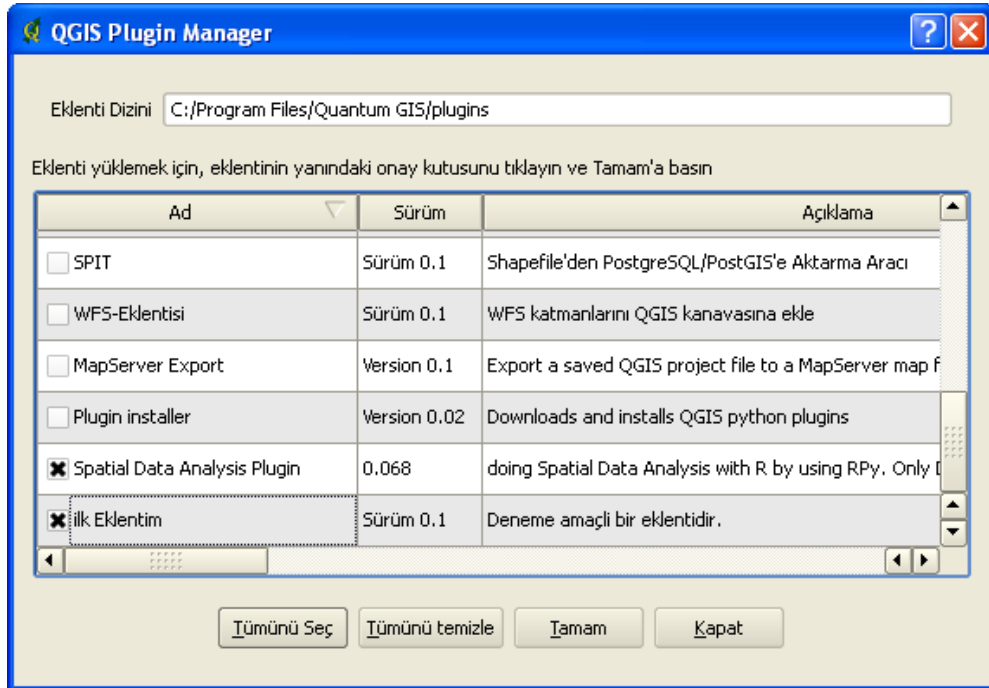
```
22 def run(self):
23     # eklentinin görevini oluşturmaya buradan başla
24     QMessageBox.information(self.iface.getMainWindow(), "Merhaba Dünya!", "İlk eklentimi  
başarıyla çalıştırdım.")
```

Şekil 15: Yeni eklenti için plugin.py dosyasındaki Python kodu

Eklenti kodunu yazma işi tamamlanmıştır. Eklenti çalıştırmak için yapılması gereken bir kaç adım daha vardır. Önce QGIS çalıştırılmaktadır. Sistem çalışırken aktif olan tüm eklentilerin kodunu kontrol etmektedir. Eğer eklentide bir hata bulursa eklenti kapatır. Eklentideki hata düzeltilmeden eklenti kullanılamaz. Eğer yeni eklentide bir hata yoksa, QGIS çalıştıktan sonra eklenti aktif yapmak için “Eklentiler” menüsü altındaki “Eklenti Yöneticisi” çalıştırılmaktadır (Şekil 16). Eklenti Yöneticisi’nde varsayılan olarak yüklenen diğer eklentilerle birlikte örnek olarak anlatılan “İlk Eklentim” adlı eklenti de sıralanmıştır (Şekil 17). Eklentinin üzeri tıklanarak aktif edilmekte ve pencere kapatılmaktadır.



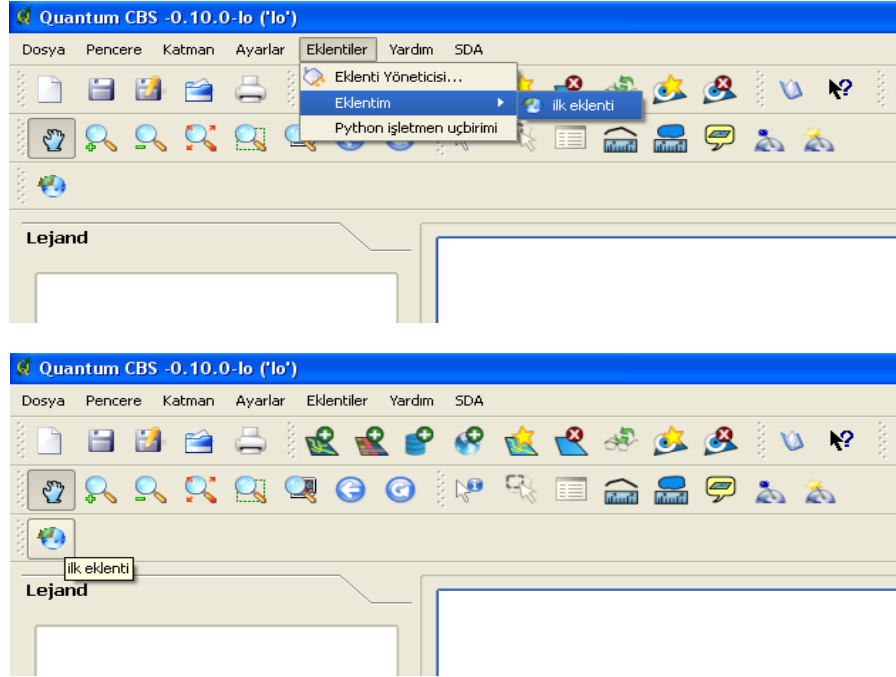
Şekil 16: Eklenti yöneticisinin çalıştırılması



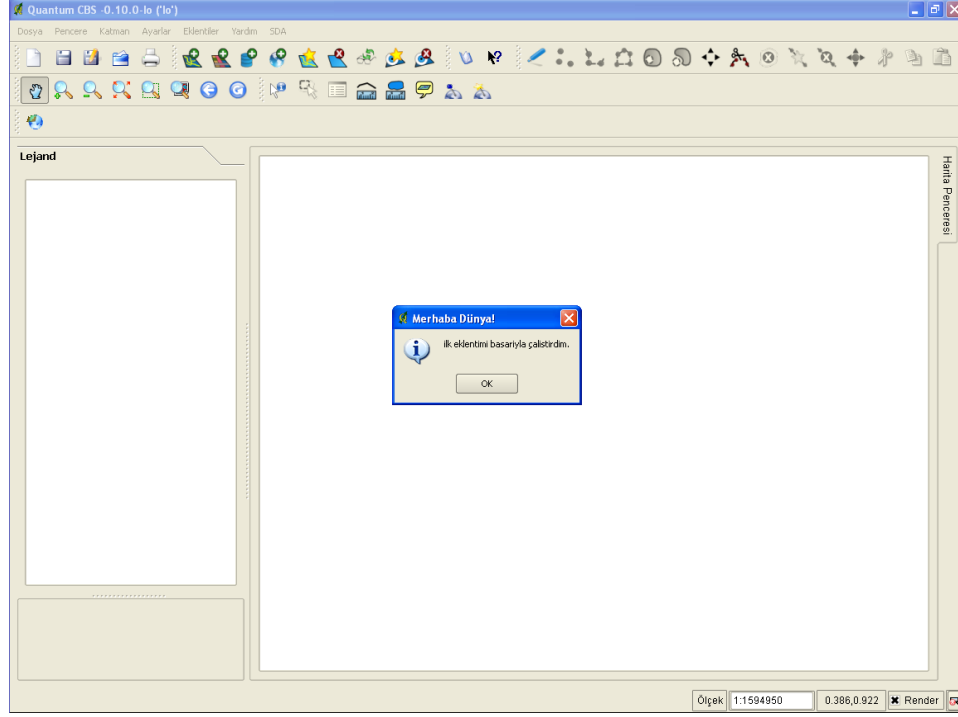
Şekil 17: Eklenti yöneticisindeki “ilk Eklentim” adlı yeni eklentinin aktif yapılması

Böylece “Eklentiler” menüsü altında “Eklentim” adlı yeni bir menü ve içinde “ilk eklenti” adlı komut satırı görülmektedir. Ayrıca eklentiler araç çubuğunda da koleksiyon dosyasında tanımlanan ikonun şeklinde yeni bir düğme oluşmuştur (Şekil 18). Bu düğmeye tıklandığında aşağıdaki Şekil 19’da gösterildiği gibi bir mesaj kutusu ile karşılaşılabilecektir.

Sonuç olarak QGIS içinde yeni bir eklenti geliştirilmeyi ve bu eklentinin nasıl çalıştırılacağı anlatılmaya çalışılmıştır. Burada amaçlanan QGIS içinde yeni bir eklentinin nasıl oluşturulduğunu göstermek olduğu için eklentinin işlevi basit bir şekilde tasarlanmıştır. QGIS ve Qt sınıfları kullanılarak istenilen özellikler anlatılan yapı sayesinde geliştirilebilir. Masaüstü CBS ortamı için geliştirilmek istenen tüm özellikler bu iki kütüphane; QGIS ve Qt kullanılarak eklenebilmektedir. Önceki bölümde anlatıldığı gibi bu özellikler ya yeni veya özel bir CBS yazılımı yapılarak oluşturulabileceği gibi bu bölümde gösterildiği gibi QGIS içinde çalışan bir eklenti olarak da geliştirilebilmektedir.



Şekil 18: Geliştirilen eklenti ile eklenen yeni düğme ve menünün görünümü



Şekil 19: Yeni eklentinin QGIS içinde çalıştırılması

3. SONUÇ YERİNE

Özgür yazılımın amacı tüketim değil üretimdir. Özgür yazılım felsefesi, kaynak kodun açıklığına (erişimine) vurgu yapmakta ve bunu toplumsal bir üretim aracı olarak nasıl kullanılabilirliğini örgütlemektedir. Bu kaynağı kullanmayı bilenler için özgür yazılım sürekli biriken toplumsal bir bilgi kaynağıdır.

Bir yazılımı geliştirmek için gerekli her türlü yasal ve teknik olanaklar özgür yazılım geliştirme ortamında geçmişte olduğu gibi günümüzde de mevcuttur. Kollektif ve gönüllü bir biçimde kod yazmayı yönetebilecek geliştiricilere her zaman ihtiyaç duyulmaktadır. Özgür yazılımların sunduğu araçlar sayesinde birlikte kod yazmak artık zamandan ve mekandan bağımsız olarak kollektif bir şekilde gerçekleştirilebilmektedir.

Yazılım geliştirme araçlarını kullanmak özgür yazılım geliştirme işini teknik olarak kolaylaştırmaktadır. Özgür yazılım geliştirme felsefesi sayesinde, bu ve benzeri araçları kullanabilme ve bir yazılım geliştirebilme şansına (özgürlüğünü) sahip olduğumuzun farkına varmamız gerekmektedir. Bu özgürlüğü korumak için bu felsefeyi yaşatmak gerekmektedir. Bu da en basit anlamıyla özgür yazılımları kullanmak ve yayılmasını sağlamak ile mümkün olabilmektedir. Yazılım geliştirme işinin maliyet boyutu ancak özgür yazılımların yaygınlaşması sayesinde çözümlenebilir.

Bahsi geçen özgür yazılım geliştirme olanakların var olması, bunlardan faydalanabileceği anlamına gelmemektedir. Bu olanaklardan yararlanabilmek için özgür yazılım dünyasının kuralları kavranmalıdır. Daha sonra bu kurallara uygun bir şekilde yazılım geliştirme işi yapan geliştiricilerin ve kullanıcı desteklerinin de yaygınlaşması gerekmektedir. Katkı sağlamayan kullanıcıların oluşturduğu bir ortamda programcıların birer geliştirici gibi çalışması yazılım geliştirme işini tek başına çözmeye yetmemektedir. Geliştiricilerin motivasyonu aldığı katkılara bağlıdır. Bir yazılımın varlığını devam ettirebilmesini sağlayan sürekli geliştirme işinin zorlukları ancak geliştiriciler ekonomik veya sosyal yönden tatmin edildiklerinde aşılabılır. Özgür yazılımların sağladığı toplumsal faydalar nedeniyle özgür yazılım felsefesine ülkemizde de giderek daha fazla sahip çıkılacağı düşünülmektedir.

Özgür yazılım dünyasının sağladığı en büyük fayda bir yazılımın fikri ve mülkü hakkını tek bir firmadan alıp toplumla paylaşabilme olanağını sunmaktır. Başka bir deyişle özgür yazılımlar herkese istediği şekilde yazılımı yeni baştan geliştirebilme özgürlüğünü sağlamaktadır. Bu bildiride bu imkanın varlığını ispatlamak için örnek olarak iki farklı yöntem kullanarak (yeni bir CBS yazılımı geliştirmek ve yeni bir eklenti oluşturmak) özgür bir CBS yazılımı olan

Quantum GIS üzerinde bir uygulamanın nasıl geliştirilebileceği gösterilerek anlatılmaya çalışılmıştır. Bu geliştirme imkanlarına kapalı kaynak kodlu ortamda erişebilmek göreceli olarak daha zordur. Kapalı kaynak kodlu CBS ürünlerinin bazılarında istenilen özellikler yazılıma eklenebilir. Bu ekleme işi bir kişiselleştirme olup çoğunlukla yeni bir yazılım geliştirmeye gidebilecek kadar açık uçlu değildir. Kapalı kaynak kodlu CBS dünyasında istenilen bir ad verilip geliştirilebilecek ve dağıtılabilecek bir yazılımı üretmek ya lisanslar aracılığıyla engellenmekte ya da yüksek maliyetler nedeniyle çoğunlukla gerçekleştirilememektedir.

Her ne kadar bu çözümlerden yararlanarak özgür yazılım dünyasında yeni bir CBS yazılımı geliştirmek nispeten alternatiflerine göre daha esnek olsa da, bir yazılım varlığını sürdürürebilmek için sürekli olarak geliştirilmeye devam edilmesi gerekmektedir. Değişen kullanıcı ihtiyaçlarına cevap verebilmek, donanımdaki yenilikler nedeniyle gelişen işletim sistemlerindeki yeniliklere ve yeni veri formatlarına uyum sağlayabilmek vb. sebeplerle her yazılım sürekli bir bakıma ve onarıma ihtiyaç duymaktadır. Bakım ve onarım yapmak için gerekli maliyeti karşılamak için yazılım ya bir ürün haline getirilip pazarlanmalı ya da özgür yazılım geliştirme yöntemi gibi gönüllü katkıları ile toplumsal bir üretim biçimi haline getirilmelidir. Bu toplumsal üretim biçimi kullanıcıların bilinçlenmesi, özgür yazılımları kullanması, geliştiricileri desteklemeleri ve katkıda bulunmaları ile gerçekleşebilmektedir. Türkiye’de bu aşamalara gelebilmek için genel olarak erken bir dönemde olduğumuz kabul edilirse özgür yazılımların giderek daha fazla yaygınlaştırılması gerekmektedir. Bu konuda çaba harcayan tüm bilişim uzmanlarının ve sivil toplum kuruluşlarının çabaları takdire değerdir ve önem arz etmektedir.

Hangi yolla olursa olsun yazılımlar kullanıcılarına daha iyi hizmet vermek için sürekli geliştirilmeye devam edilecektir. Türk CBS yazılım sektörü olarak bu yarıştan kopmamak için bu bildiride anlatılmaya çalışılan özgür yazılım dünyası benimsenerek özgür bir CBS yazılımının geliştirilmesi sahiplenilmelidir. Bu yolda aşılması gereken birçok adım olsa da CBS dünyasında uluslararası düzeyde bilinen ve tanınan özgür bir CBS yazılımının Türkiye’de geliştirilebilme olasılığının olduğunu bilmek insanı heyecanlandırmaktadır. Bir gün bu temenninin gerçekleşmesi dileğiyle...

4. KAYNAKLAR

AÇIK KAYNAK KOD PLATFORMU, (2008), ‘Açık Kaynak Kod Bildirgesi’, <http://acik-kaynak.org.tr/> (09.2008).

ALTUNERGİL, O., (2003), ‘Özgür Yazılım, Özgür Türkiye’, <http://www.belgeler.org/howto/ozguryazilim.html> (10.2003).

CYBERSOURCE COMPANY, (2004), ‘Linux vs. Windows TCO Report’, http://www.cyber.com.au/about/linux_vs_windows_tco_comparison.pdf (13.12.2004).

ÇAYIR, S., (2004), ‘Bir Yazılım Öyküsü: Turkuaz Linux’, <http://www.bidb.itu.edu.tr/?d=209> (30.11.2004).

DALGIÇ, S., (2007), ‘Özgür Yazılım Nedir?’, <http://e-bergi.com/2007/Nisan/Ozgur-Yazilim-Nedir> (04.2007).

DISTROWATCH, (2008), ‘Linux Distributions - Facts and Figures’, <http://distrowatch.com/stats.php?section=popularity> (07.2008).

DOBIAS, M., (2006), ‘The new era: QGIS and Python’, <http://blog.qgis.org/?q=node/29#comment-5185> (20.10.2006).

DOBIAS, M., (2008), ‘Python Bindings’, <http://wiki.qgis.org/qgiswiki/PythonBindings> (08.07.2008).

FREE SOFTWARE FOUNDATION, (2008), ‘The Free Software Definition’, <http://www.gnu.org/philosophy/free-sw.html> (20.03.2008).

GISDEVELOPMENT, (2008), ‘Free Open Source Software’, <http://www.gisdevelopment.net/downloads/opensource/#deegree> (09.2008).

GÖZÜKELEŞ, İ. İ., (2004), ‘Özgür/Açık Kaynak Kodlu Yazılım ve Ulusal Yazılım Politikaları’, http://www.emo.org.tr/resimler/ekler/99600eb275cc719_ek.pdf?dergi=7 (08.2004).

GÖZÜKELEŞ, İ. İ., (2006), 'Free and Open Source Software Hackers in Turkey',
<http://opensource.mit.edu/papers/WP.pdf> (31.07.2006).

INTERNATIONAL INSTITUTE OF INFONOMICS and BERLECON RESEARCH GMBH, (2002), 'Free/Libre and Open Source Software: Survey and Study', <http://www.infonomics.nl/FLOSS/report/> (06.2002).

JUERGEN, F., (2008), 'Developing QGIS plugins with Python HOWTO',
<http://wiki.qgis.org/qgiswiki/DevelopingPluginsWithPython> (06.09.2008).

LUTZ, M., (2001), Programming Python, O'reilly & Associates, Inc., USA.

MAPTOOLS, (2008), 'MapTools.org', <http://maptools.org/> (09.2008).

OHLOH, (2008), 'Quantum GIS', <http://www.ohloh.net/projects/3663> (08.2008).

OSGEO FOUNDATION, (2008), 'The Open Source Geospatial Foundation', <http://www.osgeo.org/home> (21.09.2008).

PYTHON SOFTWARE FOUNDATION, (2008), 'General Python FAQ', <http://www.python.org/doc/faq/general/> (07.09.2008).

QGIS, (2008), 'Quantum GIS User and Installation Guide Version 0.9.1 Ganymede',
http://download.osgeo.org/qgis/doc/manual/qgis-0.9.1_user_guide_en.pdf (09.2008).

RAMSEY, P., (2007), 'The State of Open Source GIS',
http://www.foss4g2007.org/presentations/view.php?abstract_id=136 (15.09.2007).

RIVERBANK COMPUTING LIMITED, (2008a), 'PyQt Whitepaper',
<http://www.riverbankcomputing.co.uk/static/Docs/PyQt4/pyqt-whitepaper-a4.pdf> (09.2008).

RIVERBANK COMPUTING LIMITED, (2008b), 'What is PyQt?',
<http://www.riverbankcomputing.co.uk/software/pyqt/intro> (09.2008).

STEINIGER, S., (2008), 'An Overview of Free & Open Source Desktop GIS', <http://www.spatialserver.net/osgis/> (06.09.2008).

TIOBE SOFTWARE, (2008), 'TIOBE Programming Community Index for September 2008',
<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> (09.2008).

TROLLTECH COMPANY, (2008a), 'About Qt', <http://doc.trolltech.com/4.3/aboutqt.html> (09.2008).

TROLLTECH COMPANY, (2008b), 'Qt Cross-Platform Application Framework', <http://trolltech.com/products/qt/> (09.2008).

UEKAE, (2007), 'Tarihçe', <http://www.pardus.org.tr/hakkimizda.html> (2007).

VARDAR M., A., (2006), 'Qt'ye Giriş', <http://qt.comu.edu.tr/dosyalar/malivardar-qt.pdf> (06.2006).

WIKIMEDIA, (2007a), 'Turanid Linux', http://tr.wikipedia.org/wiki/Turanid_Linux (03.10.2007).

WIKIMEDIA, (2007b), 'Turkix Linux', <http://tr.wikipedia.org/wiki/Turkix> (05.10.2007).

WIKIMEDIA, (2008a), 'Turkuaz Linux', [http://tr.wikipedia.org/wiki/Turkuaz_\(Linux_dağıtımı\)](http://tr.wikipedia.org/wiki/Turkuaz_(Linux_dağıtımı)) (03.01.2008).

WIKIMEDIA, (2008b), 'Open Source Software Development',
http://en.wikipedia.org/wiki/Open_source_software_development (13.09.2008).

WIKIMEDIA, (2008c), 'Comparison Of Open Source Software Hosting Facilities',
http://en.wikipedia.org/wiki/Comparison_of_free_software_hosting_facilities (16.09.2008).

WIKIMEDIA, (2008d), 'List of GIS software', http://en.wikipedia.org/wiki/List_of_GIS_software (23.09.2008).