

Tümleşik VoIP Sisteminde Alt Katman Yazılım Geliştirme Deneyimi ve Mimari Tasarım Yaklaşımları

Fatih Ayvaz¹, Mehmet Yunus Dönmez¹

¹ Netaş Telekomünikasyon A.Ş., İstanbul, Türkiye
{fayvaz,ydonmez}@netas.com.tr

Özet. Alt katman yazılımlarında yapılan yapısal değişikliklerin diğer katmanlara olan etkileri bazı projelerde tahmin edilemeyen boyutlara ulaşabilmektedir. Bu durum proje maliyetinin kestirilememesine, öngörülemeyen teknik ve zamanlama risklerine neden olabilmektedir. Bu çalışmada çok katmanlı yazılım mimarisine sahip olan, yazılım ürün sahipliğini Netaş'ın yaptığı tümleşik VoIP (Voice over IP) sisteminin alt katman yazılımlarında yapılan yapısal değişiklikleri içeren bir projede yaşanan deneyim ve mimari tasarım yaklaşımı paylaşılmıştır. Alt katman değişiklikleri nedeni ile sistem davranışına yüksek ölçüde etkisi olması beklenen düğüm konum bilgisi genişletme projesinin beklenenden düşük maliyetle ve riskle yapılabilmesi için uygulanan çözüm yaklaşımı anlatılmıştır.

Anahtar Kelimeler: VoIP, Yazılım Mimarisi, Proje Planlaması, Tasarım Maliyeti

Lower Layer Software Development Experience and Architectural Design Approaches in Integrated VoIP System

Fatih Ayvaz¹, Mehmet Yunus Dönmez¹

¹ Netas Telecommunications, Istanbul, Türkiye
{fayvaz,ydonmez}@netas.com.tr

Abstract. The effects of the structural changes made in the lower layer software to other layers can reach unpredictable dimensions in some projects. This can lead to both unpredictability of project cost and unpredictable technical and timing risks. In this study, the experience and architectural design approach of a multi-layered software architecture with a structural change in the underlying software of the integrated VoIP (Voice over IP) system of software product owned by Netas is shared. The solution approach of the node location expansion project which is expected to have a high impact on system behavior due to the substrate changes to be made has been described.

Keywords: VoIP, Software Architecture, Project Planning, Design Estimate

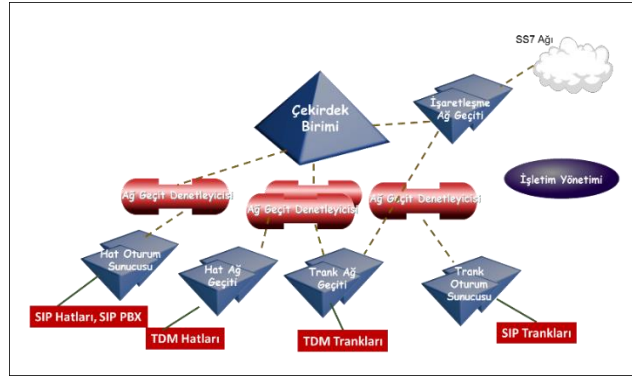
1 Giriş

Telekomünikasyon ağlarında uçtan uca haberleşmenin sağlanabilmesi için, bünyesinde çok sayıda yazılımsal ve donanımsal bileşeni barındıran tümleşik VoIP sistemleri kullanılmaktadır [1]. Bu bileşenlerin birbirleri ile haberleşmeleri, gerçek zamanlı olarak kesintisiz hizmet verebilmeleri ve telekom operatörlerinin müşterilerine sunmuş olduğu yüzlerce servisin devamlılığı; milyonlarca kod satırı içeren büyük ve karmaşık yazılımlar ile sağlanmaktadır. Bu yazılımlar işlevselliklerine ve uygulama alanlarına göre yazılım katmanlarına ve alt sistemleri ayrılmıştır. Yazılım katmanları ve alt sistemler gereksinimler doğrultusunda diğer yazılım katmanları ve alt sistemler tarafından kullanılabilir. Örneğin alt yazılım katmanlarında tasarlanmış olan bazı veri yapıları üst katmanlarda yer alan yüzlerce alt sistem tarafından kullanılabilir. Veri yapılarında yapılan yapısal değişiklikler, bu veri yapılarını kullanan tüm kodları etkilemektedir. Bu durum yazılım projelerinin teknik ve zamanlama risklerini arttırmakta ve projelerin daha maliyetli bir şekilde yapılmasına neden olmaktadır.

Bu çalışma beş bölümden oluşmaktadır. Bölüm 2’de Netaş’ın müşterilerine sunduğu tümleşik VoIP sistemi çözüm mimarisi, Bölüm 3’te tümleşik VoIP sistemlerinde yazılım geliştirme süreçleri, Bölüm 4’te çekirdek birimi çok katmanlı yazılım mimarisi ve Bölüm 5 ve 6’da çekirdek birimi alt yazılım katmanlarında yapılan bir proje deneyimi ve mimari tasarım yaklaşımı anlatılmıştır. Son bölümde ise genel değerlendirmelere ve sonuçlara değinilmiştir.

2 Tümleşik VoIP Sistemi Çözüm Mimarisi

Netaş’ın müşterilerine sunduğu tümleşik VoIP sistemi çözüm mimarisi Şekil 1’de verilmiştir. Tümleşik VoIP sistemi çözümü değişik görevleri yerine getiren çok sayıda alt bileşenden oluşmaktadır ve 1300 civarında servisi müşterilerine sunabilmektedir [2].



Şekil 1. Tümleşik VoIP Sistem Mimarisi

Tümleşik VoIP sistemi ITU-T, ETSI ve IETF gibi standart organizasyonları tarafından oluşturulmuş nerdeyse tüm haberleşme standartlarını desteklemektedir. Bu

sistemi oluşturan bileşenlerden bazıları şunlardır [2]:

Çekirdek Birimi: Çağruların kurulması, yönlendirilmesi ve sonlandırılması esnasında gerçekleşen işaretleşme ve arama servisleri ile ilgili tüm denetimleri sağlamaktadır. Çekirdek birimi yaklaşık 33,5 milyon kod satırından oluşan çok katmanlı bir yazılım mimarisine sahiptir.

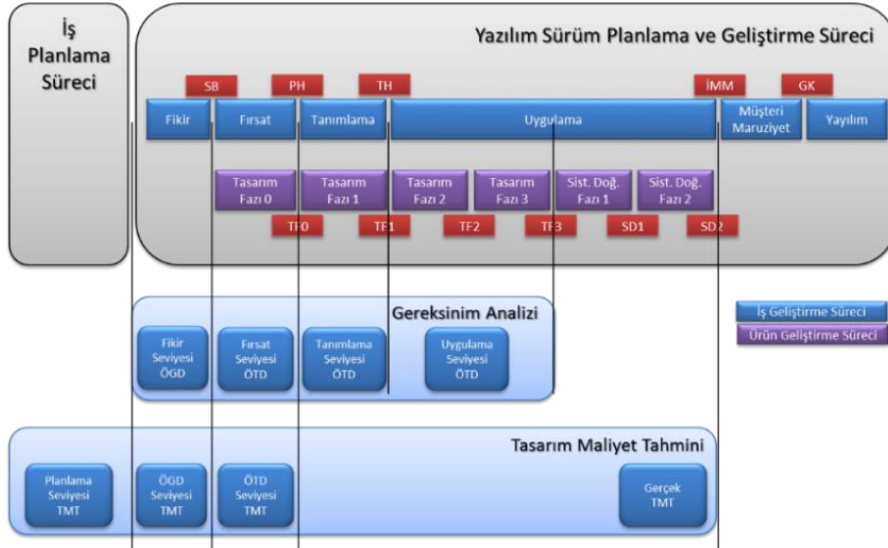
Ağ Geçidi Birimi: Transfer katmanında TDM hatları, TDM trunkları ve No:7 (SS7) işaretleşme sistem ile IP tabanlı işaretleşme protokolleri arasında dönüşüm yapmaktadır.

Ağ Geçidi Denetleyicisi Birimi: Çekirdek ile ağ geçidi arasındaki bağlantıyı kurmaktadır.

Trank Oturum Sunucusu Birimi: Santrali IP altyapıya bağlayan ve diğer IP santraller ile bağlantıyı sağlayan birimdir. Oturum Trank Sunucusu, IMS ve diğer santraller arasındaki mesajlaşmalarda SIP [3] protokolü kullanılmaktadır.

Hat Oturum Sunucusu Birimi: Santralin IP tabanlı SIP hatları ve SIP PBX bağlantılarını sağlayan birimdir.

İşletim Yönetim Birimi: Çok bileşenden oluşan VoIP santraline Telekom operatörleri tarafından uzaktan erişilip denetim takibinin yapılmasını sağlayan birimdir.



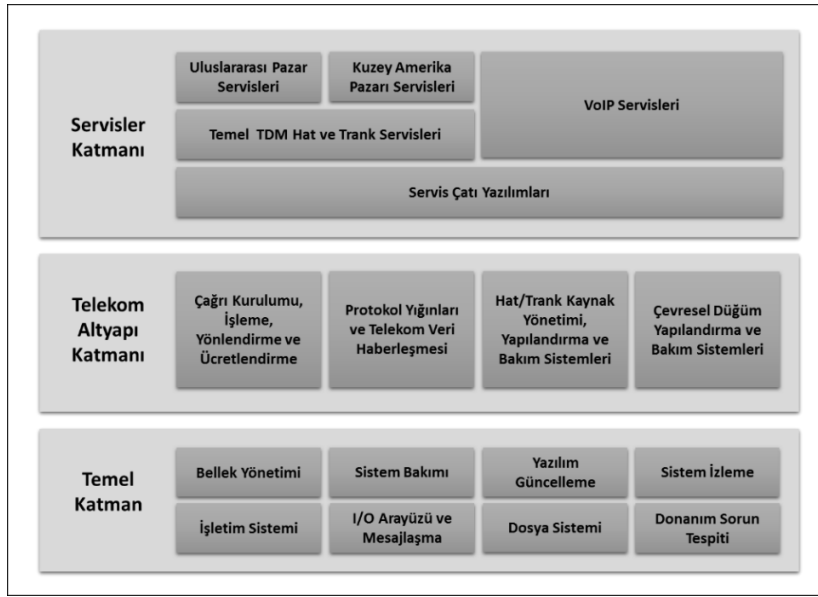
Şekil 2. Tümleşik VoIP Sistemlerinde Yazılım Geliştirme Süreçleri [5]

3 Tümleşik VoIP Sisteminde Yazılım Geliştirme Süreçleri

Tümleşik VoIP sistemi karmaşık bir yazılım mimarisini sahiptir ve yapılacak yazılım geliştirme projelerinin mimariye uygunluğunun ölçülebilirliği gerekmektedir. Proje planlaması yapılırken ardışık yazılım fazlarının uygulanması zorunludur. Bu nedenle tümleşik VoIP sisteminde Şekil 2'de görüldüğü gibi geleneksel yazılım geliştirme süreçlerinden birisi olan Şelale süreci (Waterfall process) [4] uygulanmaktadır.

İş planlama sürecinde iş geliştirme ekipleri (ürün müdürleri, çözüm mimarları vs.) müşteri ihtiyaçlarını ve gereksinimlerini önceliklendirirler ve yazılım sürüm planlama sürecine dahil ederler. Yazılım sürüm planlama ve geliştirme süreci ise analiz, mimari tasarım, kodlama, test ve sistem doğrulama fazlarını kapsayacak şekilde planlanır [5].

Yazılım sürüm planlama ve geliştirme sürecini iş geliştirme ekipleri fikir, fırsat, tanımlama, uygulama, müşteri maruziyet ve yayılım fazları altında takip ederler. Tasarım ekipleri ise tasarım fazı 0-3 ve sistem doğrulama 1-2 fazları olarak takip ederler. Fikir fazında ürün müdürleri müşteri gereksinimlerini Özellik Gereksinim Dökümanı (ÖGD) altında toplarlar. Tasarım ekipleri içerisinde yer alan yazılım mimarları ise ÖGD seviyesi proje maliyetini içeren Tasarım Maliyet Tahmini (TMT) dökümanını hazırlar. Fırsat fazında (Tasarım Fazı 0) ise yazılım mimarları tarafından mimari tasarım oluşturulur ve İleri Seviye Tasarım (İST) dökümanı yazılır. Ayrıca yazılım mimarları tarafından gereksinimler ürün özelliklerine göre şekillendirilir ve yeni türetilmiş (derived) gereksinimler Özellik Teknik Dökümanında (ÖTD) toplanır. ÖTD gereksinimlerine göre proje maliyeti hesaplanır. ÖTD seviyesi Tasarım Maliyet Tahmini (TMT) dökümanı hazırlanır. Proje planlaması buna göre detaylandırılır [5].



Şekil 3. Çekirdek Birimi Çok Katmanlı Yazılım Mimarisi

4 Çekirdek Birimi Alt Yazılım Katmanlarında Yazılım Geliştirme Deneyimi ve Mimari Tasarım Yaklaşımları

Bu çalışmada Kuzey Amerika Pazarı için geliştirilmesi planlanan telekom ağ modernizasyonu projesi kapsamında çevresel düğüm konum sayısının 256'dan 4096'ya artırılması için yapılan yazılım mühendisliği faaliyetleri analiz edilmiştir. Bu çalışma-

nın ortaya koyduğu gereksinimin gerçekleştirilmesi için Şekil 3'te gösterilen Tümlerik VoIP sisteminin PROTEL dilinde [6] yazılmış olan ve yaklaşık olarak 40 milyon kod satırından oluşan çekirdek biriminde yer alan Telekom Altyapı katmanında yer alan çevresel düğüm yapılandırma ve bakım sistemleri biriminde yer alan kodlarda yapısal değişiklikler yapılması gerekmektedir.

4.1 Proje mimari analiz fazı:

Projenin mimari analiz fazında, proje kapsamında yapılması gerekli olan en temel değişikliğin Şekil 4'te gösterilen ve bünyesinde veri yapı elemanı olarak düğüm konum sayısı bilgisini içeren nodeName veri yapısının boyutunun artırılması olduğu belirlenmiştir. nodeName veri yapısı tüm çekirdek birimi tarafından kullanılan temel veri yapılarından biridir ve toplamda 32 bit bellek alanında saklanabilmektedir. Proje kapsamında nodeName içerisinde yer alan nodeLoc veri yapısı elemanının 256'dan (8 bit bellek alanı) 4096'ya (12 bit bellek alanı) çıkarılması gereklidir. Bu durum nodeName veri yapısı bellek kullanımının artmasına neden olmaktadır.

Projenin geliştirilmekte olduğu çekirdek biriminin alt katmanlarında bulunan veri yapılarında bir değişiklik yapılabilmesi için en uygun mimari tasarım yöntemi, değişiklik yapılacak olan veri yapısının etkilediği tüm global veri yapıları, veri dizileri, fonksiyonlar, sınıflar gibi yazılım parçalarını içerisinde bulunduran ve modül adı verilen kaynak kod dosyalarının incelenerek analiz edilmesi olarak belirlenmiştir. Bu inceleme ve analiz safhasında katmanlar, katmanlar arası sistemler, sistemler arası alt sistemler, en alt seviyede modüllerin olduğu hiyerarşik bağımlılık çizgesi oluşturulur.

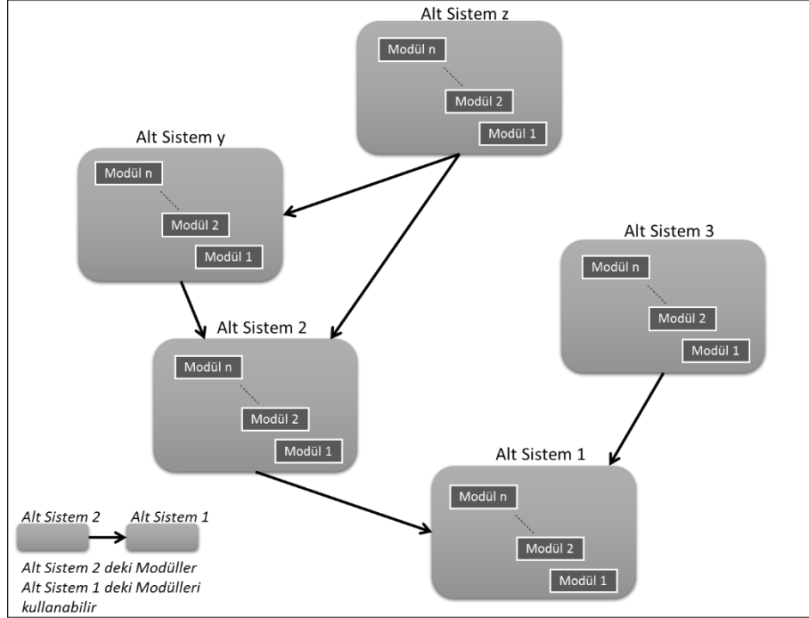
```
TYPE nodeName          ** 32 bits
STRUCT
  nodeLoc      nodeLocRng,  ** 8 bits
  nodeType     nodeTypeRng, ** 8 bits
  nodeNo       int         ** 16 bits
ENDSTRUCT;
```

Şekil 4. nodeName veri yapısı

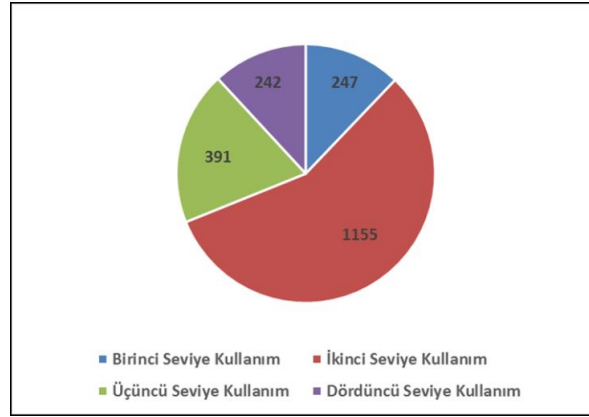
Şekil 5'te bu çizgenin bir örneği verilmiştir. Burada en altta yer alan Alt Sistem 1 grubunda yer alan Modül1'de (düğüm konum sayısının artırılması projesinde değişen nodeName veri tipinin bulunduğu modül) tanımlanmış olan bir veri yapısında değişik olduğunda üstte bulunan ve nodeName veri yapısından etkilenen modüllerde bulunan yazılımların analiz edilmesi gerekmektedir.

Yapılan analiz sonucunda nodeName veri yapısının bellek alanının genişletmesinin Telekom Altyapı katmanında ve Servisler katmanında yer alan yaklaşık 2035 modülü etkileyeceği tespit edilmiştir. Bu modüllerde nodeName veri yapısının diğer veri yapılarının elemanı olmasının yanında bellekte saklanan bazı veri dizilerinin elemanı olarak kullanılmakta olduğu saptanmıştır. Ayrıca nodeName veri yapısını kullanan diğer veri yapılarının ve veri dizilerinin başka veri yapıları ve veri dizileri tarafından kulla-

nıldığı belirlenmiştir. Şekil 6’da gösterildiği gibi nodeName bellek alanının genişletilmesinin en çok ikinci seviye kullanımdaki veri yapılarını etkilediği tespit edilmiştir.



Şekil 5. Alt Sistemler ve Modüller Arasındaki İlişiyi Gösteren Hiyerarşik Bağımlılık Çizgesi



Şekil 6. nodeName kullanım seviyesine göre etkilenen modül sayısı

4.2 Proje mimari tasarım fazı:

Projenin mimari tasarım fazına belirlenen analiz sonuçları doğrultusunda başlanmıştır. Bölüm 5’te daha detaylı olarak anlatıldığı gibi referans alınan analiz sonuçları-

nın proje maliyet hesaplamaları yapıldığında yapısal büyük çaplı değişikliklerden dolayı bir takım belirsizlikler mevcuttur. Bu durumda projenin risk analizi gözden geçirilmiş ve riskin en aza indirilmesi için ileri seviye mimari tasarım çalışmaları başlatılmıştır. Söz konusu çalışmalar İleri Seviye Tasarım (İST) dökümanı kapsamında nodeName veri yapısının bellek kullanımını arttırmanın haricindeki olası mimari çözüm önerileri yapılan mimari tasarım toplantılarında masaya yatırılmıştır. Yapılan ileri seviye analiz çalışmalarında sistem tarafından desteklenen çevresel düğüm sayısının 4096 (12 bit) olmasına rağmen nodeName veri yapısı içerisinde yer alan ve integer (16 bit) olarak tasarlanan nodeNo elemanın 4096'ya indirilmesinin mümkün olduğu ve bu değişikliğin sistem davranışına etkilerinin incelenmesi için prototip çalışması yapılmasına karar verilmiştir. Bu mimari tasarımın yöntemi nodeName veri yapısının boyutunu değiştirmeyeceği için (32 bit olarak kalmaya devam edecek) diğer modüllere etkilerinin düşük seviyede olacağı ve proje maliyetine pozitif etki edeceği düşünülmüştür.

Bu doğrultuda hazırlanan prototip çalışmasında nodeName veri yapısı Şekil 7'de gösterildiği gibi tasarlanmış ve gerçek laboratuvar ortamında test edilmiştir. Yeni tasarlanan veri yapısında nodeLoc belleğin 16 bit hizalı kullanım gereksimi nedeni ile nodeLocLS ve nodeLocMS şeklinde ikiye bölünmüştür.



Şekil 7. Dönüştürülmüş nodeName veri yapısı

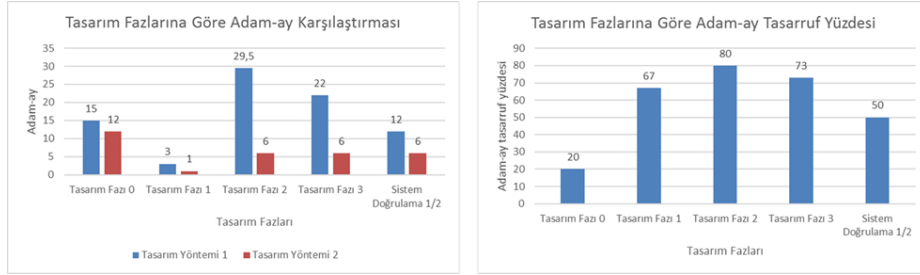
5 Mimari Tasarım Maliyetlerinin Karşılaştırılması

Projenin mimari tasarım sürecinin başlarında tasarım yöntemi olarak nodeName veri yapısının bellek alanının arttırılması önerilmiştir. Ancak mimari tasarım çalışmalarının başında proje maliyetine etkilerini tam anlamıyla belirlemek mümkün değildir. Karşılaşılan bu belirsizlik nedeniyle Tasarım Maliyet Tahmini (TMT) dökümanında proje maliyeti $\pm\%50$ yanılma oranlı olarak 81.5 adam-ay şeklinde hesaplanmıştır.

Mimari tasarım sürecinin ileri aşamalarında önerilen yeni tasarım yöntemi ile geliştirilen prototip çalışması sonrası yapılan test sonuçlarına göre proje tasarımının dönüştürülmüş yeni nodeName veri yapısı ile yapılabileceği kararı verilmiş ve prototipte oluşturulan yeni mimari tasarıma göre Tasarım Maliyet Tahmini (TMT) dökümanında proje maliyeti 31 adam-ay şeklinde hesaplanmıştır.

İki mimari tasarım önerisinin maliyetleri Şekil 8'de yer alan birinci grafikte yer almaktadır. Bu grafikte tasarım fazlarına göre ayrılaştırılmış maliyet gösterilmiştir. İkinci grafikte ise ikinci mimari tasarım yönteminin birinci mimari tasarım yöntemine göre sağladığı maliyet tasarrufu yüzdesel olarak gösterilmiştir. Proje maliyetindeki toplam tasarruf %62 olarak hesaplanmıştır. Bu tasarruf büyük oranda Tasarım Fazı 1,

2 ve 3 [5] süreçlerinde sağlanmıştır. Tasarım Fazı 0'daki oranın düşük olmasının nedeni yeni mimari tasarım önerisinin maliyetinin hesaplanabilmesi için Tasarım Fazı 0 boyunca geliştirilen prototip çalışmasının sonuçlarının beklenmesi ve bu süreçte oluşan maliyetin de proje maliyetine yansıtılmasıdır.



Şekil 8. Mimari Tasarım Maliyetlerinin Karşılaştırılması

6 Sonuçlar

Bu çalışmada tümleşik VoIP santralının çekirdek birimi çok katmanlı yazılım mimarisi anlatılmış ve alt katman yazılım bileşenlerinde geliştirilen telekom ağ modernizasyonu projesi kapsamında çevresel düğüm konum sayısının artırılması 256'dan 4096'ya çıkartılması projesindeki deneyim ve mimari tasarım yaklaşımı anlatılmıştır. Tasarım Fazı 0 sürecinde geliştirilen yeni mimari tasarım yöntemi ile toplam proje maliyetinde %62 oranında indirim sağlanmıştır.

Kaynakça

1. Yuan, Chnhui, and Hongli Zhao. "Implementing VoIP Voice Communication System Based on Soft-Switch Technology." *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 2016 International Conference on. IEEE, 2016.
2. Gürçan, F., Dönmez, Y., Ayvaz F., Mitmit, S., "AGCF Çözümü için Gerçek-Zamanlı Performans Optimizasyonu", *Proceedings of the 10th Turkish National Software Engineering Symposium*, pp.679-688, 2016.
3. Henning, S., ve Rosenberg, J., "The Session Initiation Protocol: Internet-centric signaling", *IEEE Com. Magazine*, vol. 38(10), pp.134-141, (2000)
4. Petersen, Kai, Claes Wohlin, and Dejan Baca. "The waterfall model in large-scale development." *International Conference on Product-Focused Software Process Improvement*. Springer, Berlin, Heidelberg, 2009.
5. Ayvaz, F., Mitmit, S., Demirsoy, A., Kaya, A. B. S., Yildirim, A., Yavuz, O."Tümleşik VoIP Sistemlerinde Gereksinim Analizi Ve Tasarım Maliyet Yaklaşımı." *Proceedings of the 8th Turkish National Software Engineering Symposium*, pp.501-510, 2014.
6. Foxall, D.G., Joliat, M.L., Kamel, R.F., ve Miceli, J.J. "Protel: a high level language for telephony", *The IEEE Computer Society's Third International Computer Software and Applications Conference*, 1979.