

Tümleşik VoIP Sistemlerinde Gereksinim Analizi Ve Tasarım Maliyet Yaklaşımı

Fatih Ayvaz¹, Selçuk Mitmit¹, Aycan Demirsoy¹, Ayşe Belma Şahin-Kaya¹, Ali Yıldırım¹, Oğuzhan Yavuz¹

¹ Netaş Telekomünikasyon A.Ş., İstanbul, Türkiye
{fayvaz, smitmit, aycan, belmas, aliyil, oyavuz}@netas.com.tr

Özet. Bu çalışmada haberleşme sistemlerinde (VoIP, TDM) yazılım süreçlerinde gereksinim analizi yapılması ve oluşturulan yazılım mimari tasarım doğrultusunda tasarım maliyet hesaplarının yapılışına ilişkin Netaş'ın sahip olduğu bilgi birikimi ve tecrübe paylaşılmıştır. TÜBİTAK TEYDEB tarafından desteklenmiş olan “Yüksek Kapasiteli Yeni Nesil Merkezi Santral Tasarımı” isimli projenin tasarım maliyet tahmininin yapılması anlatılmıştır.

Anahtar Kelimeler: Yazılım Süreçleri, Gereksinim Analizi, Tasarım Maliyet Tahmini, VoIP Sistemleri, Proje Planlama

1 Giriş

Bir haberleşme ağının çalışması donanımsal alt yapının yanında onun üzerine çalışan yazılım ile sağlanmaktadır. Haberleşme sistemlerinde yapılan değişiklikler donanımdan çok yazılım üzerinde yapılmaktadır. Bunun nedeni operatörlerin sistemlerini yüksek maliyetleri nedeniyle donanımsal olarak değiştirmekten çekinmeleridir. Bu yüzden yeni nesil teknolojileri destekleyen donanımların yanında hala eski nesil teknolojileri destekleyen donanımlar da ağ içerisinde kullanılmaktadır.

Haberleşme sistemlerindeki ArGe faaliyetleri genel olarak mevcut donanım üzerinde yazılım geliştirmelerini kapsamaktadır. Mevcut kaynakların daha iyi kullanımı, pazar stratejileri, rekabet, teknolojik gelişmeler, büyüme isteği ve tüketici tercihlerinin değişmesi gibi nedenlerle yeni gereksinimlere ihtiyaç duyulur. İnternet Üzerinden Ses İletişimi (Voice over Internet Protocol, VoIP) sistemleri [1] karmaşık bir yazılım mimarisine sahiptir. Bunun sebebi Çekirdek (Core), Ağ Geçidi (Gateway, GW), Ağ Geçidi Denetleyicisi (Gateway Controller, GWC), Oturum Trunk Sunucusu (Session Server Trunk, SST) ve İşletim Yönetim, Bakım ve Yapılandırma Birimi (Operations, Administration, Maintenance and Provisioning, OAM&P) gibi bileşenlerden oluşan tümleşik bir sistem olmasıdır [2]. Bu sebeple yazılım geliştirme projelerinde gereksinimlerin açık ve net bir şekilde belirlenmesi ve mimariye uygunluğunun ölçüklendirilebilmesi büyük önem taşımaktadır. Oldukça zor bir alan olan yazılım geliştirme projelerinin gereksinimlerinin –zaman, çalışan- belirlenmesi ve risk analizlerinin yapılmasında Netaş zengin bir bilgi birikimine sahiptir. Bu çalışmada Netaş'ın yazılım geliştirme projelerinin uyguladığı yöntemler ve tecrübeleri paylaşılmıştır.

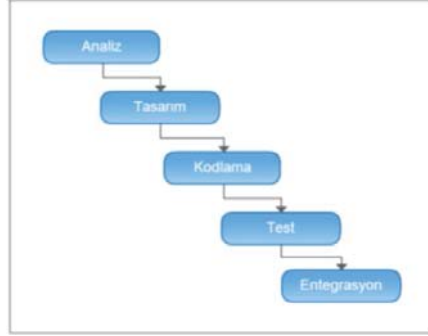
Bu yayın dört bölüme ayrılmıştır. Bölüm 2’de tümleşik VoIP sistemlerinde gereksinimlerin belirlenmesi ve yazılım geliştirme süreçleri anlatılmıştır. Bölüm 3’te tümleşik VoIP sistemlerinde gereksinim analizi ve tasarım maliyet tahmini deneyimine yer verilmiştir. Son bölümde ise genel değerlendirmelere ve sonuçlara değinilmiştir.

2 Tümleşik VoIP Sistemlerinde Gereksinimlerin Belirlenmesi ve Yazılım Geliştirme Süreçleri

Gereksinim, bir sorunu çözmek ya da bir hedefe ulaşmak için müşteriler tarafından ihtiyaç duyulan bir koşul ya da yetenektir. Diğer bir ifadeyle gereksinim, bir sözleşmeyle tanımlanan ve bir ürün veya ürün bileşeni tarafından karşılanması ya da sahip olunması gereken bir durum, yetenek, standart, şartname ya da diğer resmi belgelerdir [4].

Tümleşik VoIP sistemlerinin müşterileri dünya genelindeki büyük operatör ve hizmet sağlayıcı firmalardır. Bu firmalar mevcut kaynakların daha iyi kullanımı, pazar stratejileri, diğer firmalarla rekabet, teknolojik gelişmeler, büyüme isteği ve tüketici tercihlerinin değişmesi gibi nedenlerle yeni gereksinimlere ihtiyaç duyarlar. Tümleşik VoIP sistemleri karmaşık bir yazılım mimarisini sahiptir ve bu sebeple yazılım geliştirme projelerinde gereksinimlerin açık ve net bir şekilde belirlenmesi ve mimariye uygunluğunun ölçülebilirliği büyük önem taşımaktadır.

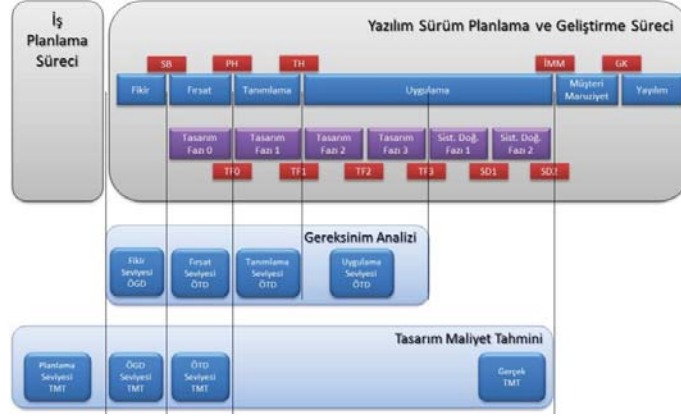
Bu sebeple tümleşik VoIP sistemlerinde, geleneksel yazılım geliştirme süreçlerinden biri olan Şelale süreci (Waterfall process) [5] uygulanmaktadır. Şelale süreci, Şekil 1’de görüldüğü gibi birbirini takip eden ardışık fazlardan oluşmaktadır ve sonraki faz bir önceki faz tamamlanmadan başlamamaktadır.



Şekil 1. Şelale Süreci

Şekil 2’de bir tümleşik VoIP sisteminde yer alan yazılım süreçleri gösterilmiştir. Bu süreçlerde iş geliştirme (business development) ekipleri geliştirilecek ürünün pazarda tercih edilebilirliğini ve pazar payını artırmanın yollarını ararlar. Ayrıca rakip firmalar içerisinde öncü ve yenilikçi olmak ve şirketin kazancını ve sürekliliğini ar-

tırmak için süreç boyunca çalışmalarını sürdürürler. Tasarım ekipleri ise müşterilerden gelen gereksinimlere uygun mimari tasarımlar yapar, uygular ve sistem doğrulamalarını gerçekleştirirler.



Şekil 2. Tümleşik VoIP Sistemlerinde Yazılım Geliştirme Süreçleri

Şekil 2’de görüldüğü gibi tümleşik VoIP sistemlerinde yazılım geliştirme ardışık iki temel süreçte ilerlemektedir:

2.1 İş Planlama Süreci

İş planlama sürecinde, ürün müdürleri, satış müdürleri ve çözüm mimarları, müşterilerle görüşerek, gereksinimleri ve müşterilerin ihtiyaçlarına katkıda bulunacak yeni teknolojik çözümleri ve fikirleri belirlerler. Bu süreçte “planlama” seviyesi tasarım maliyet tahmini (TMT) yapılır ve belirlenen tüm gereksinimlerin teknik olarak yapılabilir olup olmadığı; eğer yapılabilir ise insan kaynağı ve donanım maliyet dereceleri düşük (2 adam-yıldan az), orta (5 adam-yıldan az) veya yüksek (5 adam-yıldan fazla) olmak üzere ortaya çıkarılır. Bu süreçte tasarım ekipleri rol almamaktadır.

2.2 Yazılım Sürüm Planlama ve Geliştirme Süreci

İş planlaması yapılan gereksinimler sürüm planlama ve geliştirme sürecine sokulur. Bu süreçte iş geliştirme ve tasarım ekipleri birlikte çalışır ve her iki ekibin de kendilerine özgü ardışık süreç fazları ve her fazda elde ettikleri çıktılar vardır.

Fikir Fazı (Idea Phase): Müşteri ve pazarın ne istediğinin ve problemin iş (business) bakış açısıyla tanımlandığı fazdır. Ürün müdürleri tarafından fikir seviyesi özellik gereksinim dokümanı (ÖGD) hazırlanır. Bu fazda tasarım ekiplerinde yer alan yazılım mimarları tarafından ÖGD’de yazılan gereksinimlerin “anlaşılmış” (understood) olması gerekmektedir. Tanımlanan gereksinimlerin müşteri istekleri ve beklenti-

leri doğrultusunda önceliklendirilmesi yapılır. Önceliklendirilmiş gereksinimler iş paketlerine dönüştürülür ve çıkarılacak olan yazılım sürümüne “aday bir içerik listesi” (candidate content list, CCL) olarak dâhil edilir.

Fikir fazı tamamlandığında Sürüm Başlangıcı (SB) deklare edilmiş olur. Bu fazda yazılım mimarları tarafından yapılan analizler sonucu $\pm\%50$ yanılma oranlı adam-ay (staff month, SM) ÖGD seviyesi TMT dokümanını hazırlanır. Çıkan maliyete göre aday içerik listesinde yer alan gereksinimlerin yazılım sürümünde yer alıp almamasına tasarım ekiplerindeki insan kaynağının uygunluğuna göre karar verilir ve önceliği düşük olanlar listeden çıkartılır.

Fırsat Fazı (Opportunity Phase): Tasarım ekipleri gereksinimler üzerinde mimari tasarım ve analiz çalışmalarına başlarlar. Gereksinimler ürün özelliklerine göre şekillendirilir ve yeni türetilmiş (derived) gereksinimler oluşturulur. Bu fazda tüm gereksinimlerinin tasarım ekibi tarafından “tanımlanmış” (defined) olması ve ürünle uyumlu olup olmadığının belirlenmiş olması gerekmektedir. Tanımlanması yapılmış ve ürünlere uyumluluğu incelenmiş gereksinimler fırsat seviyesi özellik teknik dokümanında (ÖTD) toplanır. Yazılım sürümüne dâhil edilecek iş paketlerinin listesi (Plan of Intent, POI) bu fazda belirlenmiş olur.

Yazılım mimarları tarafından oluşturdukları mimari tasarım alternatifleri ile birlikte ileri seviye tasarım (İST) dokümanında belgelenir ve çözüm mimarları ve sistem mimarları ile tartışılır. Sürecin tamamlanması için doküman onayının alınması gerekmektedir. Yazılım mimarları onay alınan mimari tasarım için $\pm\%20$ yanılma oranlı adam-ay (staff month, SM) tasarım maliyet tahminini (design estimation) belirlerler. Bu amaçla ÖTD seviyesi tasarım maliyet tahminini (TMT) dokümanını hazırlanır. Proje müdürleri tarafından ÖTD seviyesi TMT’deki maliyete göre proje planları oluşturulur ve tasarım müdürleri ile koordineli çalışılarak tasarım yapacak olan yazılım mühendisleri belirlenir.

Tasarım ekipleri bu fazı Tasarım Fazı 0 (TF0) olarak adlandırır. Fırsat Fazı tamamlandığında iş geliştirme ekiplerinde Pazara Hazırlık (PH) tasarım ekiplerinde ise TF0 deklare edilmiş olur.

Tanımlama Fazı (Definition Phase): Tasarım ekipleri belirlenen mimari tasarım ile ilgili detaylı teknik analizler ve araştırmalar yaparlar. Bu fazda yapılan detaylı çalışmalar neticesinde gereksinimlerin müşteriye “tahaddüt” (committed) edilmesi gerekmektedir. Fırsat fazında yazılmış olan ÖTD güncellenir. Yazılım sürümüne dâhil edilecek “onaylı” iş paketlerinin listesi (Plan of Record, POR) bu fazda belirlenmiş olur.

Tasarım ekipleri tarafından müşteriye sunulmak üzere İşlevsel Tanım (Functional Description, FD) dokümanı yazılır. Daha önceki fazlarda tanımlanmış gereksinimlerle ilgili herhangi ve değişiklik veya yeni bir istek olursa bu doğrultuda içerik listesi ve tahmini tasarım maliyeti $\pm\%5$ yanılma oranlı olarak güncellenir. Yazılım test ekipleri bu fazda devreye girerler ve İşlevsel Tanım dokümanında belirlenen kapsama göre test stratejilerini (laboratuvar ortamı, kurulum, teçhizat tedarik vb.) belirlerler.

Tasarım ekipleri tarafından tanımlama fazı Tasarım Fazı 1 (TF1) olarak adlandırılır. Tanımlama Fazı tamamlandığında iş geliştirme ekiplerinde Ticari Hazırlık (TH) tasarım ekiplerinde ise TF1 deklare edilmiş olur.

Uygulama Fazı (Implementation Phase): Bu fazda tasarım ekipleri proje müdürlerinin liderliğinde kodlama çalışmalarına başlarlar. Yazılan ve kod kapsam (code

coverage) testleri yapılmış kodlar yazılım mühendisleri tarafından organize edilen toplantılarda (code inspection) yazılım mimarları tarafından detaylı bir şekilde incelenir; onaylanan kodlar kod deposuna (code repository) yüklenir. Kod deposuna yüklenen kodlar haftalık olarak derlenir ve laboratuvar ortamında kullanılmak üzere yazılım yükü oluşturulur. Yazılım mühendisleri yazdıkları kodların birim testlerini (unit test) bu laboratuvarlarda koştururlar. Yazılım test mühendisleri test edecekleri alanları ve senaryoları detaylandırarak işlevsel doğrulama (Functional Verification, FV) dokümanlarını hazırlarlar.

Tasarım ekipleri tarafından iş paketindeki kodlama ve birim testler tamamlandı-ğında Tasarım Fazı 2 (TF2) deklare edilir. Yazılım test mühendisleri işlevsel doğrulama testlerini başlatarak kapalı kutu (black box) olarak iş paketinin işlevselliğini test ederler. Testler sonucu raporlanan problemleri çözmek için yazılım mühendisleri tarafından yeni kod değişiklikleri yapılır.

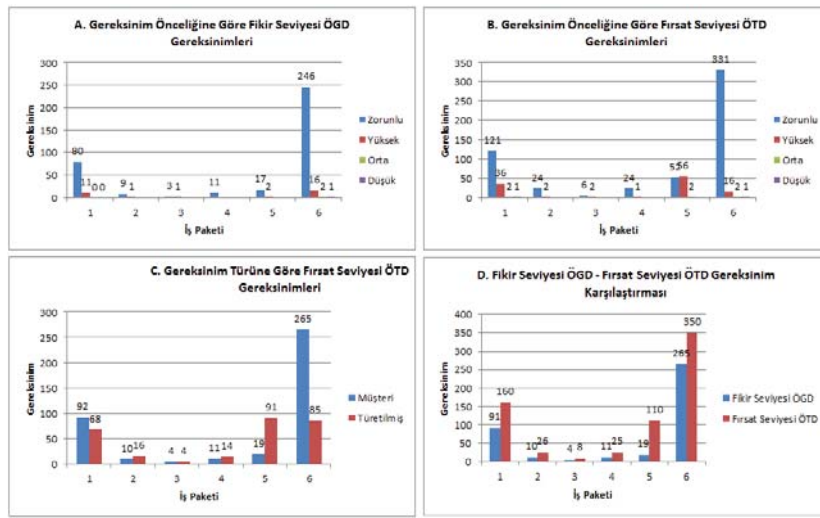
Tasarım ekipleri tarafından iş paketindeki işlevsel doğrulama tamamlandıktan sonra Tasarım Fazı 3 (TF3) deklare edilir. Sistem doğrulama ekipleri iş paketinin sürümdeki diğer iş paketleri ve mevcut sistem davranışına etkilerini test etmek için sistem doğrulama testlerine başlarlar ve buldukları problemleri, sürümdeki diğer iş paketleri ile etkileşimlerini ve sistem davranışlarına uyuşmayan durumları web tabanlı JIRA [6] izleme aracı üzerinden kayıt açarak yazılım mühendislerine raporlarlar. Yazılım mühendisleri raporlanan problemleri kod deposuna sistem doğrulama mühendisinin açtığı JIRA numarası ile girer. Sistem doğrulama mühendisi girilen kod ile derlenmiş yazılım yükünü laboratuvar ortamına yükleyerek test eder. Eğer sorun çözülmüş ise açılan JIRA kapatılır. Sistem doğrulama ekibi tüm testler tamamlandıktan ve problemler çözüldükten sonra test ilk geçiş (test first pass) ilan ederler. Test ilk geçiş sonrası raporlanan problemler ve kritik test senaryoları tekrar koşturulur. Sistem doğrulama problemsiz bir şekilde tamamlandığında ve tüm JIRA'lar kapandıktan sonra tüm yazılımın son derlemesi (Final Compile) yapılır ve sürüm kod girişine kapatılır ve tasarım ekipleri tarafından Sistem Doğrulama Fazı-1 (SD1) deklare edilir.

Sistem doğrulama mühendisleri son derleme sonrası oluşturulan yazılım yükü ile nihai değerlendirme (final assessment) testleri yaparlar ve bulunan problemler web tabanlı JIRA izleme aracı ile yazılım mühendislerine raporlanır. Yazılım mühendisleri raporlanan problemler için buldukları kod çözümlerini derleme yapılamadığı için sürüme yazılım yaması (software patch) olarak yazarlar. Nihai değerlendirme testleri problemsiz bir şekilde tamamlandığında ve yazılım yamaları yazılarak tüm JIRA'lar kapandıktan sonra, tasarım ekipleri tarafından Sistem Doğrulama Fazı 2 (SD2) deklare edilir ve uygulama fazı tamamlanmış olur.

Müşteri Maruziyet Fazı (Customer Exposure Phase) ve Yayılım Fazı (Deployment Phase): Uygulama fazının sonlanması İş geliştirme ekipleri tarafından İlk Müşteri Maruziyet (İMM) deklarasyonu ile birlikte müşteri maruziyet safhasına geçilir ve sistem doğrulaması yapılmış yazılım sürümünün müşterilere sunulması için hazırlık sürecidir (personel, teçhizat tedarik, saha destek vs.). Bu faz tamamlandıktan sonra yazılım sürümü için Genel Kullanılabilirlik (GK) deklare edilir ve müşterilere yayılımı (deployment) yapılır.

Bunun sebebi iş paketi 6'nın ağırlıklı olarak grafik arayüz tabanlı bir çözüm olması ve kullanıcıya görünür çok fazla gereksinimin belirlenebiliyor olmasıdır. Diğer iş paketlerinde yapılacak tasarımlar ilgili bileşenlerin altyapısında olduğundan iş (business) bakış açısıyla belirlenen gereksinimlerin sayısı azdır.

Şekil 4B'de verilen grafik önceliğe göre listelenmiş fırsat seviyesi ÖTD gereksinimlerini içermektedir. Zorunlu gereksinimlere sayıları iş paketi 1'de %51, iş paketi 2'de %166, iş paketi 3'te %100, iş paketi 4'te %118, iş paketi 5'te %205 iş paketi 6'da ise %34 artmıştır. Diğer gereksinimlerde de kayda değer artışlar olmuştur.



Şekil 4. Yüksek Kapasiteli Merkezi Yeni Nesil Santral Tasarımı Gereksinim Analizleri

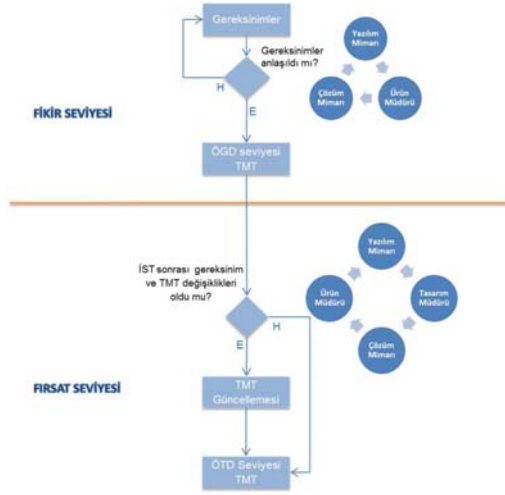
Şekil 4C'de verilen grafikte, türüne göre listelenmiş gereksinimler belirtilmiştir. Ürünün altyapısında tasarım yapacak olan iş paketlerinde türetilmiş gereksinimlerin oranının daha fazla olduğu dikkat çekmektedir. Özellikle iş paketi 1'de %74 iş paketi 5'te ise %378 oranında gereksinim artışı olmuştur.

Şekil 4D'de verilen grafikte, ÖGD-ÖTD karşılaştırması yapılmış ve yine ürün altyapısında tasarım yapılacak iş paketlerinde gereksinim artışı dikkat çekmektedir.

3.2 Tasarım Maliyet Tahminleri

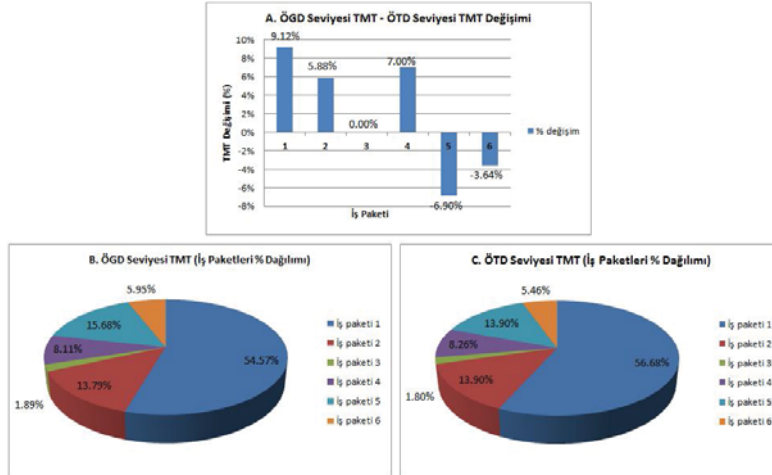
Çözüm mimarları iş planlaması sürecinde proje kapsamında yer alacak gereksinimlerin tahmini maliyetinin yüksek (5 adam-yıldan fazla) olduğu belirlenmiştir. Yazılım sürüm planlama ve geliştirme sürecinde ise yazılım mimarları tarafından Şekil 5'te belirtilen akış doğrultusunda tasarım maliyet tahmini belirleme çalışmaları yapılmıştır. Fikir seviyesinde yazılım mimarları, ürün müdürleri ve çözüm mimarları ile anlaş-tıkları gereksinimler doğrultusunda ÖGD seviyesi TMT dokümanını hazırlamışlardır.

Fırsat seviyesinde ise yazılım mimarları, İST sonrası oluşan gereksinim ve TMT değişikliklerini tasarım müdürleri, ürün müdürleri ve çözüm mimarları ile anlaşarak tasarım maliyet güncellemesi yaparak ÖTD seviyesi TMT dokümanını hazırlamışlardır.



Şekil 5. Fikir ve Fırsat Fazı Tasarım Maliyet Tahmini Belirleme Akış Diyagramı

Fikir ve fırsat fazlarında hazırlanan TMT dokümanlarında belirlenen tahmini tasarım maliyetlerinin iş paketi bazında yüzde dağılımları Şekil 6'da verilmiştir.



Şekil 6. Yüksek Kapasiteli Merkezi Yeni Nesil Santral Tasarımı Tasarım Maliyet Tahminleri

Şekil 6A’da verilen grafikte ÖGD seviyesi TMT ile ÖTD seviyesi TMT arasındaki değişimler verilmiştir. ÖTD ile gelen türetilmiş gereksinimler ve yapılan teknik analizler neticesinde iş paketi bazında tahmini tasarım maliyetlerinde değişimler olmuştur. Tasarım maliyetinde yüzdesel olarak en fazla artırım iş paketi 1’de %9.12 oranında yapılırken en fazla azaltma ise iş paketi 5’te %6,90 oranında yapılmıştır.

Şekil 6B ve 6C’de iş paketlerinin ÖGD ve ÖTD seviyesi tahmini tasarım maliyetlerine yüzdesel dağılımı gösterilmiştir. Grafikten de görüleceği gibi ürün altyapısında tasarım gerektiren iş paketlerindeki maliyet oldukça yüksektir. Özellikle iş paketi 1’deki maliyet dikkat çekmektedir. Bunun sebebi iş paketi 1’deki yapılacak olan kodlamaların sistem altyapısında köklü değişiklikler yapmasıdır. Kodlama ile birlikte işlevsel ve sistem doğrulama anlamında da yüksek bir maliyet vardır. Çekirdek bileşeninin karmaşıklığı ve tüm sistemi etkileyen riskler nedeni ile tasarım maliyeti bu şekilde belirlenmiştir.

4 Sonuçlar

Bu çalışmada anlatılan “Yüksek Kapasiteli Yeni Nesil Merkezi Santral Tasarımı” projesinin proje planlaması, TF0 sonunda belirlenen adam-ay türünden belirlenen tasarım maliyet tahminine göre yapılmıştır. Fırsat fazında yazılım mimarları tarafından yapılan mimari tasarım, teknik analizler ve belirlenen riskler neticesinde; fikir fazından belirlenen yazılım tasarım maliyeti %5.06 oranında artış göstermiştir. Proje müdürleri müşterilerin beklentileri ve tasarım ekiplerindeki mevcut insan kaynaklarının uygunluğunu ve yazılım mimarları tarafından belirlenen teknik riskleri göz önünde bulundurularak, projenin başlangıcından SD2 deklare edilene kadar geçen süreyi 19 ay olarak planlamıştır. Bu proje sistem kapasitesinde ve altyapısında ciddi değişikliklere neden olduğu için; yapılan planlama, müşteriler tarafından makul karşılanmış ve sürüm geçiş planlamalarını bu doğrultuda yapmışlardır. Ancak teknolojik eğilimler ve pazarda rekabet doğrultusunda bazı gereksinimlerin daha hızlı bir şekilde sağlanmasını istenebilmektedir. Bu tarz projelerde farklı yazılım tasarım süreçleri uygulanmaktadır.

Bilgilendirme

Bu çalışmada adı geçen, “Yüksek Kapasiteli Yeni Nesil Merkezi Santral Tasarımı” TÜBİTAK-Teknoloji ve Yenilik Destek Programları TEYDEB tarafından 3120226 numaralı proje olarak desteklenmiştir.

Kaynakça

1. Goode, B., “Voice over Internet Protocol (VoIP)”, Proceedings of the IEEE, vol. 90 (6), pp. 1495-1517, (2002).

2. Aktürk, E., Demirkıran F., Uysal Ö., Hacıbeyođlu B., ve Yavuz O., “IMS-TDM Entegrasyonu: AGCF Çözümü” IEEE 22. Sinyal İşleme ve İletişim Uygulamaları (SİU 2014), pp.1495-1498, (2014).
3. Henning, S., ve Rosenberg, J., “The Session Initiation Protocol: Internet-centric signaling” IEEE Com. Magazine, vol. 38(10), pp.134-141, (2000).
4. IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 610.121990 (Sep. 28), Reaffirmed Sep. 2002, IEEE Press, New York, (2002).
5. http://tr.wikipedia.org/wiki/Waterfall_model.
6. <https://www.atlassian.com/software/jira>.
7. Cashin, P. M., Joliat, M. L., Kamel, R. F. ve Lasker, D. M., “Experience with a modular typed language: PROTEL”, Proceedings of the 5th international conference on Software engineering ICSE '81, pp.136-143, (1981).