

# Yazılım Geliştirme Dersleri Öğrenci Projelerinin Birliktelik Kuralı İle Değerlendirilmesi

Pınar Cihan<sup>1</sup>, Oya Kalıpsız<sup>1</sup>, M. Özgür Cingiz<sup>1</sup> ve Mahir Doksöz<sup>1</sup>

<sup>1</sup> Bilgisayar Mühendisliği Bölümü, Yıldız Teknik Üniversitesi, İstanbul  
{pkaya, kalipsiz, mozgur}@yildiz.edu.tr  
mahirdoksoz@gmail.com

**Özet.** Günümüzde yazılım mühendisliği eğitimi oldukça önem kazanmış olup, öğrenciler mezun olmadan ve önemli tasarım ve uygulama sorumlulukları almadan önce mutlaka bazı pratik deneyimleri kazandırmalıdır. Bu çalışmada Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği bölümü Yazılım Mühendisliği ve Sistem Analizi ve Tasarım derslerinde öğrencilerin gerçekleştirdikleri projeleri ile ilgili anketler yapılmıştır. Bu çalışmadaki amaç, yazılımın daha verimli ve kaliteli sonuçlanabilmesi için projelerden elde ettiğimiz istatistiksel verilere dayandırarak bazı önerilerde bulunabilmektir. Yapılan çalışma sonunda elde edilen veriler, yazılım geliştirme ders projelerinde, analiz ve tasarımda dikkat edilmesi gereken noktaları göstererek süreç adımlarının takibini kolaylaştıracaktır.

Elde edilen bu verileri analiz etmek ve veriler arasındaki beklenmeyen ilişkileri bulmak, gizli bilgileri açığa çıkararak gelecekteki eğilimleri belirlemek için, veri madenciliğinde, birliktelik kuralı çıkarım algoritmalarından biri olan Apriori algoritması kullanılmıştır. Çalışmamızda uygulanan yöntem ile elde edilen sonuçlar doğrultusunda sorular arasındaki ilgili bağımlılar tartışılmıştır.

**Anahtar Kelimeler.** Yazılım Mühendisliği Eğitimi, Veri Madenciliği, Birliktelik Kuralı, Apriori Algoritması

## 1 Giriş

Yazılım mühendisliği eğitimi, geleneksel olarak üniversitelerin Bilgisayar Bilimleri ve Bilgisayar Mühendisliği bölümlerinde yer alan dersler ile gerçekleştirilmektedir. Bu programlarda “Yazılım Mühendisliği” adında bir ders bulunmakta, bazı programlarda ise buna ek olarak yazılım mühendisliği dersinin yanı sıra buradaki konuları işleyen dersler yer almaktadır. Ayrıca bazı üniversitelerde, Yazılım Mühendisliği Lisans Programları oluşturulmuştur [1].

Yazılımlarda yapılan hatalar kuruma yüklü bir maliyet getirmekte ve iş kaybına yol açmaktadır. Bu nedenle yazılım mühendisliği eğitimi oldukça önemli olup [2,3], öğrenciler mezun olmadan ve önemli tasarım ve uygulama sorumluluklara almadan önce mutlaka bazı pratik deneyimleri kazandırmalıdır [2].

Karmaşık yazılımları geliştirmek ve bakımını yapmak çok masraflı ve zordur. Bu yüzden, yazılımlar yazılım mühendisleri tarafından nizami olarak planlı bir proje şeklinde geliştirilmektedir. Yazılım Mühendisliği derslerinde proje ağırlıklı öğretimin önemi hakkında çeşitli makaleler yayınlanmıştır [4-7].

Yıldız Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü, Sistem Analizi ve Tasarımı dersi ile Yazılım Mühendisliği dersi kapsamında öğrenciler gruplar halinde proje gerçekleştirmektedirler. Bu gruplar öğrencilerin kendilerinin belirlediği 2 ile 5 arası öğrenciden oluşmaktadır. Proje konuları dersin yürütücü tarafından ilan edilir ve gruplar istedikleri konuları seçerler. Gerçekleştirilen bu projeler ileride iş hayatlarında onlara önemli tecrübeler kazandıracaktır. Bu nedenle öğrencilerin projelerde hangi alanlarda eksik olduklarının bilinip, bunların iyileştirilmesi oldukça önemlidir. Öğrenci projelerinden çıkarımlar yapabilmek için bu yazılım geliştirme ders projeleri ile ilgili öğrencilere anketler uygulanmıştır. Yapılan çalışmada [8] anket soruları incelenmiş olup, öğrencilerin sosyal yeteneklerinin zayıf olduğu gözlemlenmiştir. Eksik oldukları bu yetkinlik tespitinin yanı sıra anket soruları arasındaki ilişkilerin saptanması da son derece önemlidir. Çünkü bu sorular arasındaki ilişki bize öğrenci profilleri hakkında bilgi vermektedir. Bu nedenle sorular arasındaki gizli bilgileri ortaya çıkarmak için veri madenciliği yöntemlerinden olan birliktelik kuralı algoritmalarında olan Apriori algoritması kullanılmıştır. Apriori algoritması en bilindik, kolay ve performans açısından diğer algoritmalarından daha iyi olduğu için tercih edilmiştir.

Son yıllarda Veri Madenciliği eğitim alanında büyük önem kazanmış olup bu alanda birçok çalışma yapılmıştır [9-13]. Veri madenciliğinde kullanılan ilk tekniklerden birisi de birliktelik kurallarıdır [14]. Birliktelik kuralı, geçmiş verilerin analiz edilerek bu veriler içindeki birliktelik davranışlarının tespiti ile geleceğe yönelik çalışmalar yapılmasını destekleyen bir yaklaşımdır.

Bu çalışmada, Yazılım Mühendisliği ve Sistem Analizi ve Tasarımı derslerinde öğrencilere yapılan projeleri ile ilgili anket verileri üzerinde birliktelik kuralı algoritmalarından Apriori algoritması uygulanarak, anket soruları arasındaki ilişkileri saptamaktır.

Bu çalışmadaki amaç, yazılımın daha verimli ve kaliteli sonuçlanabilmesi için projede elde ettiğimiz istatistiksel verilere dayandırarak bir takım önerilerde bulunabilmektir. Ders projelerine önerilen bazı değişiklikler, hem projelerin takibini kolaylaştırmak hem de projeleri daha kaliteli hale getirmektir.

## 2 Sistem Tasarımı

Bilgisayar ve Yazılım Mühendisliği programlarında öğrencilerin yazılımla geliştirme süreçleriyle tanışması ve yazılım geliştirme aşamalarını öğrenmesinde *Sistem Analizi ve Tasarımı*, *Yazılım Mühendisliği*, *Yazılım Kalite ve Test Süreci* gibi dersler önemli rol oynamaktadır. Çalışmamız kapsamında öğrencilerin yazılım geliştirme sürecine adaptasyonlarının yazılım mühendisliği ile ilgili aldıkları derslerin geri beslemeleri göz önüne alınarak değerlendirilmiştir. Bu değerlendirmeleri yapabilmek için yazılım mühendisliği alanındaki derslerin projeleri ile ilgili farklı anket soruları oluşturularak

öğrencilerin bu derslerde yaptıkları proje çalışmalarından elde ettiği süreç kazanımları yorumlanabilir.

Çalışmamızda Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü'nde *Yazılım Mühendisliği* ve *Sistem Analizi ve Tasarımı* derslerinde öğrencilere ders kapsamında gerçekleştirdikleri projeleri ile ilgili anket yapılmıştır. Bu anketlere Sistem Analizi ve Tasarımı dersinde 86 öğrenci ve Yazılım Mühendisliği dersinde 70 öğrenci katılmıştır. Anketler çoktan seçmeli sorular olup Yazılım Mühendisliği ile ilgili 20 soru, Sistem Analizi ve Tasarımı dersi ile ilgili 19 soru öğrencilere yöneltilmiştir. Her iki ankette ortak olan soru sayısı ise 11'dir. Ortak soruların amacı, yazılım geliştirme süreçlerini öğrenmeyle ilgili Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği'nde alınan ilk ders olan *Sistem Analizi ve Tasarımı* dersinin sonundaki edinimleriyle bir sonraki yıl aldıkları Yazılım Mühendisliği dersinin sonunda yazılım geliştirme süreçleriyle ilgili edinimlerinin karşılaştırılmaya çalışılmasıdır. Bu bölümde *Sistem Analizi ve Tasarımı* ve *Yazılım Mühendisliği* dersleriyle ilgili anket soruları genel olarak ele alınmıştır.

## 2.1 Veri Kümesi ve Ön İşlem Adımları

*Yazılım Mühendisliği*, *Sistem Analizi ve Tasarımı* ve bu iki dersle ilgili ortak olarak anketlerle öğrencilere yöneltilen ve önemli olarak görülen sorular Tablo 1, Tablo 2 ve Tablo 3'de gösterilmiştir. Bu çalışmada veri setlerinde yazı kalabalığı oluşmaması ve değerlendirme aşamalarında kolaylık sağlamak için her veri setindeki sorular "Soru1", "Soru2" ve "Soru3" gibi değerlerle gösterilmiştir. Tablolarda hangi sorunun hangi değerle ifade edildiği de görülebilmektedir.

**Tablo 1.** Sistem Analizi ve Tasarımı dersi anketinin önemli görülen soruları

<b>SİSTEM ANALİZİ VE TASARIMI SORULARI</b>	
Soru1	Yazılım geliştirici analiz ve tasarımdan elde edilen süreç ve raporları ne ölçüde kullanmıştır?
Soru4	Geliştirdiğiniz yazılım projesinin size en büyük katkısı hangi alanda olmuştur?
Soru10	Yazılıma ait fizibilite analizi yapmak için kullandığınız bir araç ve/veya yöntem var mı?
Soru11	MS Project ile belirlediğiniz proje geliştirme zamanı ile sürecin tamamlanmasından sonra elde edilen proje geliştirme zamanı birbirleri ile paralellik göstermekte midir?
Soru14	Veri akış diyagramında tasarladığınız proje ile geliştirdiğiniz projenin varlık, modül ilişkileri ve veri akışları ne ölçüde paraleldir?
Soru15	Yazılım geliştirici, uygulama geliştirme esnasında sistem analiz ve tasarım argümanlarının değişimini gerekli görmüş müdür?
Soru16	Veri tabanı oluşturulurken ER diyagramından yararlandınız mı?
Soru18	Projenizde en az süre çalışan ekip üyeleri hangisi/ hangileridir?
Soru19	Sizce çalışanlardan en önemli görev hangi çalışana aittir?

**Tablo 2.** Yazılım Mühendisliği dersi anketinin önemli görülen soruları

<b>YAZILIM MÜHENDİSLİĞİ SORULARI</b>	
Soru2	Yazılım geliştirici, uygulama geliştirme esnasında sistem analiz ve tasarım argümanlarının değişimini gerekli görmüş müdür?
Soru3	Müşterinin sürecin ilk başında istediği yazılım ile süreç sonunda oluşturulan yazılım birbiriyle ne ölçüde örtüşmektedir?
Soru4	Proje sürecinde hangi role daha çok kişi atamanız gerekti?
Soru6	Proje içi ekip uyumunuz ne ölçüdeydi?
Soru8	Proje sonunda elde ettiğiniz ürünü nasıl değerlendiriyorsunuz?
Soru11	Yazılıma ait fizibilite analizi yapmak için kullandığınız bir araç ve/veya yöntem var mı?
Soru13	Aşağıdaki diyagramlardan hangisi modelleme ve tasarımı ifade etmek için daha faydalıdır?
Soru15	Aşağıdaki diyagramlardan hangisi yazılım süreçlerinde ilk oluşturduğunuz diyagramdır?
Soru18	Modelleme ve tasarım aşamasında sınıflarda yer alan metod ve değişkenlere bakıldığında fazla karmaşık sınıf var mıdır?

**Tablo 3.** Ortak sorulardan oluşan veri setinden önemli görülen sorular

<b>ORTAK ANKET SORU ÖRNEKLERİ</b>	
Soru1	Yazılım geliştirici analiz ve tasarımdan elde edilen süreç ve raporları ne ölçüde kullanmıştır?
Soru4	Proje sürecinde hangi role daha çok kişi atamanız gerekti?
Soru5	Geliştirdiğiniz yazılım projelerinin size en büyük katkısı hangi alanda olmuştur?
Soru8	Proje sonunda elde ettiğiniz ürünü nasıl değerlendiriyorsunuz?
Soru10	Yazılım projelerinizde, sistem analiz ve tasarım aşamaları projenizin yüzde kaçını oluşturmaktadır?
Soru11	Yazılıma ait fizibilite analizi yapmak için kullandığınız bir araç ve/veya yöntem var mı?
Soru12	MS Project ile belirlediğiniz proje geliştirme zamanı ile sürecin tamamlanmasından sonra elde edilen proje geliştirme zamanı birbirleri ile paralellik göstermekte midir?
Soru6	Proje içi ekip uyumunuz ne ölçüdeydi?

Tablo 1-2 ve 3'te yer alan sorular öğrencilere sorularak ilgili dersler hakkında öğrencilerin kazanımları değerlendirilmiştir. Anket sorularına cevap veren öğrencilerin bir kısmı sorulara cevap vermemiş veya bir soruya birden fazla cevap vermiş olabilir bu nedenle sorulardan çıkarım yapılabilmesi için anket verilerine ön işlem adımlarının uygulanması şarttır. Öğrencilerin anket sorularına cevap vermemesi durumunda ilgili sorulara cevap atanması kayıp değer yerleştirme (Missing Value Replacement) işle-

mine göre yapılmıştır. Kayıp değer yerleştirme yaklaşımına göre cevabı boş olan soruya cevap olarak ilgili soru için en çok cevaplanan şıkkın ataması gerçekleştirilmiştir. Aynı işlem bir soruya birden fazla cevap verilmiş sorular için de gerçekleştirilmiştir ve kayıp değer yerleştirme işlemiyle ön işlem adımı tamamlanmıştır.

## 2.2 Birliktelik Kuralı ve Apriori Algoritması

Veri kümesinde yer alan kayıtların birbiriyle olan ilişkileri ve bağıntılarını inceleyerek verilerin eş zamanlı olarak birlikte gerçekleşebilmelerini kurallarla ortaya koyan veri madenciliği yöntemine birliktelik kuralı denir. Birliktelik kuralı yaklaşımlarında verilerin her biri parça (item) ve bu verileri birlikte oluşturduğu veri kümesi ise işlemleri, parça kümelerini, (transaction veya item set) oluşturmaktadır. İşlemler kümesi ise tüm veri kümesini tanımlamaktadır. İşlemlerde birlikte geçen veri parçalarının birbirleriyle olan meydana gelme ilişkisini çıkarmak için kullanılan parametrelerden ilki destek (support) değeridir. A ve B veri parçalarının ilişkileri  $A \Rightarrow B$  şeklinde gösterildiğinde destek değeri A ve B parçalarının tüm işlemlerde geçme sıklığının tüm işlem sayısına oranı olarak hesaplanmaktadır.

$$\text{Destek } (A \Rightarrow B) = (\text{A ve B'nin birlikte geçtiği işlem sayısı}) / (\text{toplam işlem sayısı}) \quad (1)$$

Birliktelik kuralları aynı zamanda aşağıdaki gösterimdeki gibi çok boyutlu olarak da tanımlanabilmektedir.

$$\text{Destek } (A \vee C \Rightarrow B) = \frac{(\text{A,B ve C parçalarının birlikte geçtiği işlem sayısı})}{\text{toplam işlem sayısı}} \quad (2)$$

Birliktelik kurallarının çıkarımında destek değerinin yanı sıra güven (confidence) değeri de parçaların birlikteliklerinin çıkarılmasında önemli bir kriterdir. Güven değeri A veri parçasının geçtiği yerde B veri parçasının geçme sıklığının tüm işlem sayısına oranını vermektedir.

$$\text{Güven } (A \vee C \Rightarrow B) = \frac{(\text{A,B ve C parçalarının birlikte geçtiği işlem sayısı})}{\text{A ve B'nin birlikte geçtiği işlem sayısı}} \quad (3)$$

Birliktelik kuralında güven değeri çıkarımların gücünü verirken, destek değeri ise kurallardaki örneklerin frekansını vermektedir. Yüksek güven ve güçlü destek değerleri kullanılarak birbirleriyle bağlantılı olan yüksek ilişkili parçaların birliktelikleri çıkartılabilir. Bu nedenle güven ve destek değerlerinin önemli birliktelikleri vermesi için bu iki değer de belirli eşik değerlerinden yüksek olması beklenmektedir.

Apriori algoritması birliktelik kuralları ile öğrenmede en çok bilinen yaklaşımdır. Apriori algoritmasının yalancı kodu aşağıdaki şekilde gibidir.

```

1   Ck : k öđeli aday parça kümesi
2   Lk : k öđeli parça kümesinin sıklığı
3   L1= {1-öđeli parça sıklığı};
4   for(k = 1; Lk !=0; k++) do begin
5       Ck+1= adaylar Lk öđeli parça sıklığı bilgisini kullanılarak oluşturulur
6       for each veri kümesindeki işlem t için
7           t işlemindeki Ck+1 öđeli tüm adayların sayısını arttır
8           Lk+1= belirli bir minimum destek değerinin üstündeki adayları ata
9   end
10  return Uk Lk;

```

**Şekil 1.** Apriori Algoritması

Şekil 1’de yer alan Apriori algoritmasına göre parçaların birliktelikleri belirli bir destek değerinin üstünde sıklığa sahip veri kümesindeki tüm işlemler taranarak elde edilmektedir. Buna göre farklı sayıda birlikteliğe sahip işlemlerde yer alana parçaların oluşturulması hem işlemlerde yer alması hem de belirli bir destek değerinin üstünde olmasına göre çıkarılır.

### 3 Değerlendirme

Çalışmamızda Bölüm 2’de yer alan Yazılım Mühendisliği, Sistem Analizi ve Tasarımı ve bu iki dersle ilgili ortak sorulan önemli sorular öğrencilere yöneltilerek ilgili derslerde öğrencilerin yaptıkları proje çalışmalarından elde ettiği yazılım süreç kazanımları Apriori algoritması kullanılarak yorumlanmaya çalışılmıştır. Yazılım Mühendisliği, Sistem Analizi ve Tasarımı ve bu iki dersle ilgili ortak sorulan sorularda yer alan yazılım geliştirme süreçlerine ait kavramlar arasındaki birliktelikler belirlenmeye çalışılmıştır.

**Tablo 4.** Sistem Analizi ve Tasarımı Dersi Sorularının Birliktelik Kuralları

<b>Birliktelik Kuralı (destek değeri 1’den 0.1’e kadar delta 0.15)</b>	<b>Güven Değeri</b>
Soru1=a, Soru 10=b, Soru 16=c, Soru 18=d => Soru 14=c	1
Soru 10=b, Soru 11=c => Soru 15=b	0.96
Soru 15=b, Soru 16=c, Soru 19=b => Soru 10=b	0.95
Soru 4=b, Soru 10=b => Soru 15=b	0.95
Soru 7=a, Soru 18=d => Soru 14=c	0.94

Tablo 4-5 ve 6 ’da Apriori yaklaşımıyla sorular arasındaki ilişkiler güven değeri sırasına göre sıralanmıştır. Weka Tool[15] kullanılarak çıkartılan kurallar için destek değeri ilk olarak 1 olarak belirlenmiş daha sonra delta değeri kadar 0.1 değerine kadar azaltılarak daha fazla kural oluşturulması sağlanmıştır.

Sistem analizi ve tasarımı dersinin projelerine ait sorularının birliktelikleri aşağıdaki kurallarla açıklanmıştır.

- Tablo 4'e göre yazılım geliştiricinin analiz ve tasarımdan elde ettiği süreç ve raporları büyük ölçüde kullandığını düşünenler, fizibilite analizi yapmak için MS Project kullananlar, veri akış diyagramında tasarladıkları ile geliştirdikleri projenin paralellik gösterdiğini de söyleyenler ve yazılım geliştirme ekibinde en az çalışan üyenin "veri tabanı tasarımcısı" olduğunu da söyleyenlerin hepsi veri akış diyagramında tasarlanan proje ile geliştirilen projenin varlık, modül ilişkileri ve veri akışlarının birbirlerine paralel olduklarını düşünmektedirler.
- Fizibilite analizi yapmak için MS Project kullanıp kestirilen yazılım geliştirme zamanı ile proje geliştirmenin gerçek zamanının paralellik göstermediğini düşünenler aynı zamanda yazılım geliştirme esnasında sistem analiz ve tasarım argümanlarının değişimini genel anlamda gerekli görmektedir.
- Yazılım geliştirici uygulama geliştirme esnasında sistem analiz ve tasarım argümanlarının değişimini kısmen gerekli görmüşken veri tabanı tasarlarken ER diyagramından yararlandığını söyleyenler ile projede yer alan çalışanlardan en önemli görevin sistem analistinin olduğunu düşünenlerin büyük bir kısmı fizibilite analizi yapabilmek için MS Project'den yararlanmıştır.
- Sistem Analizi ve Tasarımı dersinde geliştirilen yazılım projesinin en büyük katkısının ekip içi uyum olduğunu düşünüp ve MS Project kullananlar yazılım geliştiricinin uygulama geliştirme esnasında sistem analiz ve tasarım argümanlarının değişimini kısmen gerekli gördüğünü düşünmektedirler.
- Proje sonunda elde edilen ürünün iyi olduğunu düşünüp aynı zamanda projede en az göreve sahip olan kişinin Veri Tabanı Tasarımcısı olduğunu düşünenler veri akış diyagramında tasarlanan proje ile geliştirilen projenin varlık, modül ilişkileri ve veri akışları birbiriyle paralel olduğunu düşünmektedirler.

**Tablo 5.** Yazılım Mühendisliği Dersi Sorularının Birliktelik Kuralları

<b>Birliktelik Kuralı (destek değeri 1'den 0.1'e kadar delta 0.15)</b>	<b>Güven Değeri</b>
Soru 2=kısmen, Soru 3=50-75, Soru 16=orta => Soru 8=geliştirilmesi gereken	1
Soru 4=sistem analisti, Soru 8=geliştirilmesi gereken => Soru 3=50-75	0.96
Soru 11= MS Project Kullanıldı, Soru 13=Sınıf Diyagramı, Soru 15=Kullanım Senaryoları => Soru 18=az sayıda	0.95
Soru 3=50-75, Soru 6=orta => Soru 8= geliştirilmesi gereken	0.95
Soru 6=iyi, Soru 11= MS Project Kullanıldı, Soru 13= Sınıf Diyagramı => Soru 3=50-75	0.94

Yazılım mühendisliği dersinin projelerine ait sorularının birliktelikleri aşağıdaki kurallarla açıklanmıştır.

- Yazılım geliştiricinin uygulama geliştirme esnasında sistem analiz ve tasarım argümanlarının değişimini kısmen gerekli görmüştür diyenlerin, müşterinin sürecin ilk başında istediği yazılım ile süreç sonunda oluşturulan yazılımın birbiriyle yüzde 50-75 örtüşmektedir diyenler ile yine proje içi ekip uyumunun orta olduğunu düşünenlerin hepsi proje sonunda elde ettikleri ürünün geliştirilmesi gerektiğini söylemişlerdir. Burada proje içi ekip uyumunun projenin sonucu için ne denli önemli olduğu görülmektedir. Çünkü frekans analizine yapıldığında da ekip içi uyumuna iyi diyenlerin çoğunun ürününün iyi olduğunu ekip içi uyumun orta derece olduğunu düşünenlerin çoğunun ürünün geliştirilmesi gerektiğini düşünmüştür. Ayrıca ekip içi uyumun kötü olmadığını düşünenlerden hiç kimsenin proje sonunda elde ettiği ürünün kötü olduğunu söylememiş olduğu görülmektedir.
- Proje sürecinde sistem analizi ve tasarımcısı rolüne daha fazla kişi atanması gerektiğini düşünenlerken proje sonunda elde ettikleri ürünün geliştirilmesi gerektiğini düşünenlerin çoğu müşterinin sürecin ilk başında istediği yazılım ile süreç sonunda oluşturulan yazılımın birbiriyle yüzde 50-75 örtüşmekte olduğunu söylemektedirler.
- Projelerinde MS Project kullananlardan Modelleme ve tasarımı ifade etmek için sınıf diyagramının daha faydalı olduğunu düşünenler yazılım sürecinde ilk oluşturulan diyagramın kullanım senaryoları olduğunu da düşünenlerin modelleme ve tasarım aşamasında sınıflarda yer alan metot ve değişkenlere bakıldığında az sayıda karmaşık sınıf olduğunu belirttikleri görülmektedir.
- Müşterinin sürecin ilk başında istediği yazılım ile süreç sonunda oluşturulan yazılımın birbiriyle yüzde 50-75 örtüşmekte olduğunu düşünürken proje içi ekip uyumunun orta olduğunu düşünenler proje sonucunda elde ettikleri ürünün geliştirilmesi gerektiğini düşünmektedirler. Hem burada hem de 1. sıraya bakıldığında ekip içi uyumun önemi görülmektedir. Ayrıca yüzde 50-75 örtüşmektedir cevabını verenlerin genel olarak proje sonucuna geliştirilmesi gereken cevabını vermiştir diyebiliriz.
- Proje sonucunda elde edilen ürünün iyi olduğunu düşünenlerin çoğunun MS Project kullandığı, durum böyleyken de modelleme ve tasarım için sınıf diyagramının daha faydalı olduğunu düşünenlerin yazılım ile süreç sonunda oluşturulan yazılımın birbiriyle yüzde 50-75 oranında örtüşmekte olduğu belirttikleri görülmektedir.

**Tablo 6.** İki Derse Ait Ankette Yer Alan Ortak Soruların Birliktelik Kuralları

<b>Birliktelik Kuralı (destek değeri 1'den 0.1'e kadar delta 0.15)</b>	<b>Güven Değeri</b>
Soru 4=a, Soru 12=c => Soru 11=b	0.94
Soru 5=d, Soru 12=a => Soru 12=Sistem Analiz	0.94
Soru 1=b, Soru 4=a, Soru 8=b => Soru 11=b	0.94
Soru 11=a, Soru 12=Sistem Analiz => Soru 1=a	0.94
Soru 7=e, Soru 10=c => Soru 6=c	0.94

İki dersin ortak sorularının birliktelikleri aşağıdaki kurullarla açıklanmıştır.



- Proje sürecinde sistem analistine daha çok kişi atanması gerektiğini düşünlerin neredeyse tamamı MS Project kullanıp belirlenen yazılım geliştirme zamanı ile sürecin tamamlanmasından sonra elde edilen proje geliştirme zamanının paralellik göstermediğini düşünmüşlerdir.
- Gerçeklenen projenin geliştirme sürecinde kişiye en büyük katkının geliştirilen ilk büyük proje olduğunu düşünenlerden MS Project kullanıp belirlenen yazılım geliştirme zamanı ile sürecin tamamlanmasından sonra elde edilen proje geliştirme zamanının paralellik gösterdiğini de düşünenler genellikle sistem analizi ve tasarımı dersi öğrencileri olduğu görülmektedir.
- Yazılım geliştiricinin analiz ve tasarımdan elde ettiği süreç ve raporları kısmen kullandığını düşünenlerden, proje sürecinde sistem analistine daha çok kişi atanması gerektiğini düşünlerin tamamına yakını proje sonunda elde ettiği ürünün geliştirilmesi gerektiği kanaatindedir. Aynı kişiler ise MS Project kullanıp belirlenen yazılım geliştirme zamanı ile sürecin tamamlanmasından sonra elde edilen proje geliştirme zamanının paralellik göstermediğini düşünmüşlerdir.
- Yazılıma ait fizibilite analizi tahmini olarak yapıldı diyenlerden sistem analizi ve tasarımı dersi öğrencileri olanları yazılım geliştiricinin sistem analizi ve tasarımlarından elde edilen süreç ve raporları büyük ölçüde kullandıkları görülmektedir.
- Proje tasarımı ve analiz sürecinde en çok zorlanılan konunun proje süresi kısıtlılığı olduğunu söyleyenlerden yazılım projelerinde sistem analizi ve tasarım aşamalarının tüm yazılım geliştirme sürecinin yüzde 50-75 oranında olduğunu düşünenler genellikle ekip içi uyumunun iyi olduğu gözlemlenmiştir.

## 4 Sonuçlar

Çalışmamızda *Yazılım Mühendisliği ve Sistem Analizi ve Tasarımı* dersini alan öğrencilerin ders projelerini gerçekleştirirken karşılaştıkları yazılım geliştirme adımları birliktelik kurallı öğrenme algoritması olan Apriori algoritmasıyla incelenmiştir. Birlikteliklerin çıkarımı için her iki dersin projesiyle ilgili farklı ve ortak sorulardan oluşan anket soruları öğrencilere sorulmuş ve değerlendirme iki derse ait farklı sorularla ayrı ayrı ve ortak sorular kullanılmak üzere üç farklı şekilde Bölüm 4'teki gibi gerçekleştirilmiştir.

Gelecek çalışmalarda projeler geliştirme süreçlerinde yer alan adımların proje sonucuna etkisi karşılaştırılarak proje sonucu sınıf etiketi olarak kullanılıp proje sonucuna süreç adımlarının etkisi incelenecektir.

## Kaynaklar

1. Mishra, A., Cagiltay, N. E., Kılıc, O.: Software engineering education: some important dimensions. In: European Journal of Engineering Education. Volume 32. (2007) 349-361.
2. Saiedian, H., Bagert, D., Mead, N.: Software Engineering Programs: Dispelling the Myths and Misconceptions. In: IEEE Software. Volume 19. (2002) 35-41.

3. Hilburn, T., Humphrey, W.: The Impending Changes in Software Education. In: IEEE Software. Volume 19. (2002) 22-24.
4. Gnatz M.: A Practical Approach to Teaching Software Engineering. In: 16th Conference on SE Education and Teaching. (2003)
5. Peslak A. R.: Teaching Software Engineering Through Collaborative Methods. In: Issues in Information Systems. Volume 5. (2004) 247-253.
6. Navarro E. O.: Teaching Software Engineering Using Simulation Games. In: International Conference on Software Engineering. (2004).
7. Stirewalt, R. E. K.: Teaching Software Engineering Bottom Up. In: American Society for Engineering Education Annual Conference. (2004).
8. Cihan, P., Kalipsiz, O.: Assessing the Human Factors in Software Development Courses Students Project. In: International Conference on Education and Educational Technologies. (2013).
9. Mohammed M. Abu Tair, Alaa M. El-Halees: Mining Educational Data to Improve Students' Performance: A Case Study. In: International Journal of Information and Communication Technology Research. Volume 2. (2012).
10. S.A. Kumar, Vijayalakshmi M.N.: Mining Of Student Academic Evaluation Records in Higher Education. In: IEEE. (2012).
11. Sushil, V., R. S. Thakur, Shailesh J.: Study of the Applications of Data Mining Techniques in Higher Education. In: International Journal of Computer & Communication Technology (IJCCT). Volume 3. ( 2012).
12. Hongjie, S.: Research on Student Learning Result System based on Data Mining In: IJCSNS International Journal of Computer Science and Network Security. Volume 10. (2010).
13. Yao-Te Wang, Yu-Hsin Cheng, Chang, Ting-Cheng, JEN, S.M.: On the Application of Data Mining Technique and Genetic Algorithm to an Automatic Course Scheduling System. (2008).
14. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules Between Sets of Items in Large Databases. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (ACMSIGMOD '93). (1993) 207-216.
15. "Data Mining Software in Java", <http://www.cs.waikato.ac.nz/ml/weka/>, Ocak 2013.