

# Yazılım Mühendisliği Eğitiminde Takım Projelerinin Yönetimi ve Değerlendirilmesi

Kaya Oğuz, Sevcan Gül

İzmir Ekonomi Üniversitesi  
Bilgisayar Mühendisliği Bölümü  
Sakarya Cad. No:156, 35330, Balçova, İzmir

**Özet.** Yazılım mühendisliği eğitiminde öğrencilerin bireysel becerilerinin yanında, bir takım olarak çalışma becerileri de edinmeleri gereklidir. Takım çalışması içeren ders ve bitirme projeleri bu deneyimi sağlasa da, projelerin yürütülmesi ve değerlendirilmesinde çeşitli problemler ortaya çıkar. Bu problemlerden bazıları takım üyelerinin projeye farklı oranlarda katkıda bulunmaları, bu oranların öğretim elemanı tarafından takibinin güç olması ve bunun sonucu olarak öğrencilerin gerçek katkıların proje notlarına yansıtılmasındaki zorluklardır. Bu bildiri “Yazılım Proje Yönetimi” dersinde verilen bir projenin farklı ekiplerle yürütülmesi ve ekiplerdeki öğrencilerin dönem sonunda değerlendirilmesinde kullanılan Görev Puan Sistemi ve sonuçları sunulmaktadır. Bu sistem ile hem öğrenciler hem de öğretim elemanı projeye ait görevler açıp bu görevlere uygun puanları ve görevlerin değerlerini belirlerler. Dönem sonunda görev puanları bireysel katkıların hesaplanmasında kullanılır. Görev Puan Sistemi “Yazılım Proje Yönetimi” dersine kayıtlı 96 öğrencinin yer aldığı 20 ekip ile birlikte dönem boyunca yürütülmüştür. Ekiplerdeki her öğrencinin takımına verdiği katkı somut olarak elde edilmiş ve notlara yansıtılmıştır.

**Anahtar Kelimeler:** takım çalışması, değerlendirme, yazılım proje yönetimi, yazılım mühendisliği eğitimi, görev puan sistemi

# Managing and Evaluating Team Projects in Software Engineering Education

Kaya Oğuz, Sevcan Gül

Izmir University of Economics  
Department of Computer Engineering  
Sakarya Cad. No:156, 35330, Balçova, İzmir

**Abstract.** The students should also acquire team working skills along their individual skills during their software engineering education. Even though courses and senior projects that require team work provide such experience, various problems emerge during their execution and evaluation. Most common problems come from the varying degrees of contributions from team members and the difficulty for the lecturer to measure and reflect these contributions to the grades. This paper presents the Task Point System and its results during its application to a project developed by different teams and the evaluation of the students in these teams in the “Software Project Management” course. With this system, both the students and the lecturer can create tasks and set appropriate points for them. At the end of the term the task points are used to calculate individual grades. Task Point System has been applied to 20 teams made up of 96 students for a semester during the “Software Project management” course. The tangible contributions of team members are acquired and reflected to the grades for each student in the teams.

**Keywords:** team work, evaluation, software project management, software engineering education, task point system

## 1 Giriş

Yazılım mühendisliği eğitiminde takım çalışması içeren projeler, öğrencilerin bir arada çalışma deneyimi kazanmalarına olanak sağlar. Bu proje süreçlerinde hem öğrenciler hem de öğretim elemanı çeşitli problemlerle karşılaşır. Öğrenciler belki ilk kez uyguladıkları yazılım geliştirme süreçlerindeki deneyimsizlikleri nedeniyle proje takvimine uymakta zorlanırlar. Proje büyüklüğünü ve geliştirme için gereken süreyi yanlış tahmin etmeleri takvim ile yapılan işin arasının açılmasına neden olur. Proje geciktikçe artan telaş yüzünden takım içerisinde anlaşmazlıklar doğar.

Öğretim elemanının özellikle kalabalık sınıflarda her öğrencinin takımına katkısını takip etmesi güç olabilir. Öğretim elemanı, öğrenciler kendi aralarında farklı bir platformda iletişim kuruyorlarsa takım içerisindeki dinamikleri tam anlayabilir. Takıma dahil olmadıkları için de olası problemlerin önüne geçemez.

Sonuç olarak proje tamamlandığında, öğretim elemanı öğrencilerin projeye katkılarını değerlendirmekte zorlanır.

Derslerde yapılan projelerin en belirgin özelliği öğrencilerin farklı oranlarda katkı sağlamalarıdır [1]. Projelerde daha çok programlama kısmına ağırlık verildiği için takım içerisinde iyi programlama yapan bir ya da iki öğrenci bu işin tamamını veya tamamına yakını yüklenirler. Takımdaki diğer öğrenciler de yazılımın belgeleme süreçlerine odaklanırlar. Bu iş bölümü bir takım çalışması gibi gözüke de mühendislik eğitimi süresince belgeleme süreçlerine maruz kalan öğrenciler programlama deneyimi kazanamazken, aynı şekilde programlama yapan öğrenciler de belgeleme süreçlerinde deneyim kazanamazlar.

Coppit, takım projelerindeki öğrencileri yedi farklı profile ayırır [2]. Projeye az katkı verirken alacağı notu yüksek tutmak isteyen ile çok katkı verip diğer öğrencilerin anlamlı katkı sağlamasına engel olan öğrenci profilleri en sık karşılaşılanlar arasındadır. Diğer bir profil ise projeden düşük ama geçer not bekleyen, bu yüzden aldıkları görevleri tam yapmayan öğrencilerdir. Bazı öğrenciler ise arkadaşlarına yardım etmek isterler ve yüksek not almaları için onların payına düşen görevleri de yaparlar.

Bu öğrenci profilleri dönem projesi içeren bütün derslerde görülebilir. Özellikle bitirme projelerinde daha belirgin şekilde ortaya çıkarlar. Bunun nedeni bitirme projelerinin dönem projelerine göre daha büyük olması ve sonucunda alınacak notun mezuniyeti etkileyecek olmasıdır.

Mühendislik eğitimi süreci göz önüne alınarak projeler, bütün süreçleri boyunca, gerçekçi olarak yürütülmelidir [3]. Projenin gerçekleşmesi dışında projenin yürütülmesi ile ilgili diğer kısımlara da zaman ayırmak gerekmektedir. Bu kısımların başında takım oluşturma, iletişim, proje planlaması, maliyet tahminleri ve projenin takibi gelmektedir [4].

Projenin gereksinim analizi sırasında mühendislerin müşteriyle olan iletişimi, yazılım tasarımı sırasında da takım içi iletişimlerinin iyi olması gereklidir. Tasarımın ardından, hazırlanacak proje takvimi için maliyet tahminleri yapılmalıdır. Proje takvimi oluşturulurken bu tahminlerin isabetli olması gereklidir. Tahminler, proje yürütülürken de takvime müdahale etmeyi kolaylaştırır. Mühendislik öğrencilerinin bu tahminleri doğru yapabilmeleri için yapılacak işin ne olduğunu bilmeleri kadar, kendilerini de iyi tanımaları gerekir.

Bütün bu koşullar altında proje yönetiminin ders olarak yürütülmesi hem öğrenciler için hem de öğretim elemanı için çaba gerektiren bir süreçtir. Öğretim elemanı için zor bölümlerin başında öğrencilerin değerlendirilmesi gelmektedir. Öğretim elemanı projenin yürütülmesini ve planlamasını takip etmeli ama öğrencilerin proje deneyimi kazanabilmeleri için çok fazla müdahale etmemelidir.

Bu bildiri de proje yönetim ve yazılım geliştirme derslerinde bireysel değerlendirme için uygulanan görev puan sistemi ve ilk dönem sonrası elde edilmiş sonuçlar sunulmaktadır. Bu sistem sayesinde proje derslerinde takım olarak yapılan çalışmaların yürütülmesinin ve değerlendirilmesinin iyileştirilmesi hedeflenmiştir.

## 2 İlgili Çalışmalar

Ekip çalışmalarında bireysel değerlendirme için öğrencilerin yaptıkları sunumlar, yapılan toplantıların notları, öğrencilerin sundukları yazılı raporlar ve e-posta listelerindeki yazışmalar kullanılsa da bu bilgiler bireysel katkıların ne kadar olduğunu öğretim elemanına tam olarak gösteremez. Wilkins ve Lawhead öğretim elemanının bütün toplantılara katılıp herkesin ne yaptığını gözlemlemesi durumunda da varlığının öğrencilerin davranışlarını değiştirmelerine neden olacağını belirtir [5].

Yapılan araştırmalar [5, 6, 7], takım projelerindeki bireysel değerlendirmede bu yöntemlerin yanısıra, öğrencilerin kendileri de dahil olmak üzere tüm takım arkadaşlarını değerlendirmesini ve bu değerlendirmenin ders notu hesaplanmasına dahil edilmesi gerektiğini göstermektedir. Öğrencilerin notlama sistemine dahil olmaları ise farklı eleştirileri ve problemleri beraberinde getirmektedir. Herbert notlandırmada sadece öğrencilerin verdiği puanların kullanıldığı bir sistemde öğrencilerin notları arkadaşları arasında eşit olarak paylaştıklarını görmüştür [7]. Bunun sebepleri arasında öğrencilerin katkıların gerçekten eşit olduğunu düşünmeleri, formu acele doldurmaları ya da önemsememeleri, kendi katkılarını -az ya da çok- saklamaya çalışmaları, notlandırma işinin öğretim elemanının işi olduğunu düşündüklerinden tepki vermek amacıyla eşit dağıttıklarını, ya da takım olarak eşit dağıtılması üzerine anlaşmalarını vermiştir. Bu durumda sadece öğrenci notlarının kullanılmasının sakıncalı olduğunu ve farklı yaklaşımlar kullanılmasını gerektiğini belirtmiştir.

Clark ve arkadaşları, notlandırma tablosu olarak bireysel %40, takıma %60 oranda katkı veren bir sistem önermişlerdir [6]. Bu oranlarda öğrenci sadece takımdan gelecek nota güvenmemesi için de her ikisinden en az %40 alma koşulunu getirilmiştir. Öğrencilere dönemin başından itibaren kendilerinin ve arkadaşlarının değerlendirilmelerinin notlandırmaya büyük bir katkısı olacağını bildirilmiş ve web tabanlı araçlarla kendilerini ve arkadaşlarını değerlendirecekleri anketler, bireysel katkı raporları ve nicelik raporlarını doldurmaları istenmiştir. Bu araçlardan gelen verilerle öğrencilere de geri dönüşüm yapılması için anketler tamamlandığında, öğrenci anket sorularına verdiği cevaplarla, arkadaşlarının ona verdiği notların ortalamasını ve öğretim elemanının yorumlarını görebilmesi sağlanmıştır.

Farrell ve arkadaşları Takım Katkı Sistemi adı verdikleri ve takım içinde özet değerlendirmelerden ve geri bildirimlerden oluşan bir sistem önermektedirler [8]. Bu sistem ile öğrencilere kendi katkılarını gösteren bir form, takım arkadaşlarını değerlendirdikleri başka bir form ve toplantı notlarını içeren son bir form ile veri girişi yapmalarını sağlayıp bu verileri notlandırmada kullanılmaktadırlar. Takımın teslim ettiği raporlar değerlendirilip, bireysel katkılar göz önünde bulundurularak son not hesaplanmaktadır. Bu sistemi bir sonraki çalışma ile web tabanlı bir uygulama haline getirmişlerdir [9].

Vasilevskaya ve arkadaşları ise üç boyutta kategorilendirdikleri değerlendirme etkinlikleri ile projeleri notlandırmaktadırlar [1]. Bu boyutlar takım ve bireysel değerlendirme, öğretim elemanı ve öğrenci katkısı, biçimsel ve toplumsal değerlendirmedir.

### 3 Yazılım Proje Yönetiminde Değerlendirme

Yapılan çalışmalar ve edinilen deneyimler sonucu, dönem ya da bitirme projesi olarak yapılan çalışmaların yürütülmesinde ve değerlendirilmesinde yardımcı olacak araçlara ihtiyaç vardır. Kullanılacak sistemin

- projenin yürütülmesinin ve gerçekleşmesinin önüne geçmemesi,
- hem öğrencilerin, hem de öğrencilerin yaptığı işlerin öğretim elemanı tarafından değerlendirilmesini içermesi [5, 6, 7],
- adil ve açıklanabilir olması [8],
- takım içerisinde, Aggarwal e O'Brien'in deyiimiyle, kaytaranlara izin vermemesi [10],

gerekmektedir. Öğretim elemanının sürece dahil olması öğrencileri daha iyi değerlendirmesine olanak verir ve desteğe ihtiyaç duyulduğunda doğru yönlendirebilir. Fakat öğrenciler projede kendi başlarına kalmalı, kararları kendileri vermeleri ve projeyi kendileri yönetmelidirler.

#### 3.1 Proje Yönetimi ve Görev Puan Sistemi

Dönem içinde verilen projenin takibi ve değerlendirilmesi için Coppit'in uyguladığı yöntem temel alınarak bir sistem tasarlanmış, ve bu sistem 2016/2017 öğretim yılı Bahar döneminde İzmir Ekonomi Üniversitesi Mühendislik Fakültesinde verilen "SE315 Yazılım Proje Yönetimi" dersinde uygulanmıştır [2]. SE315 dersi dönem içinde iki ayrı şube olarak toplam 96 öğrenci ile işlenmiştir. Öğrenciler daha önce programlama ve yazılım geliştirme yöntemlerini anlatan dersleri almışlardır.

Coppit dönem içinde bir projeyi 20 - 30 kişiden oluşan bir sınıf ile birlikte gerçekleştirmiştir. Yapılacak projenin büyüklüğünün ayarlanmasında gerçekleştirilmesinin yanında test ve belgeleme gibi diğer yazılım süreçlerini de hesaba katılması gerektiğini belirtmiştir. Seçilen projenin esnek gereksinimlere sahip olması ve öğrencilerin paralel çalışabilmeleri için bu gereksinimlerin olabildiğince bağımsız olması gerektiğini vurgulamıştır.

SE315 dersinde sayıları 4 ile 6 öğrenci arasında değişen 20 takıma bir Trafik Benzetim Projesi verilmiştir. Öğrenciler kendi ekiplerini oluşturabildikleri gibi, isterlerse öğretim elemanı tarafından da bir ekibe dahil edildiler. Seçilen proje, Coppit'in önerdiği kadar büyük olmasa da, öğrencilerin kendilerinin iyi olduğu ortamları seçebilmeleri için gereksinimleri esnek ve bağımsız tutulmuştur.

Trafik benzetim projesi ile öğrencilerden bir masaüstü yazılımı geliştirmeleri istenmektedir. Bu proje ile hayali bir belediyede çalışan sinyalizasyon mühendisleri, trafik ışıklarının konumlarını ve zamanlamalarını test edebileceklerdir. Yazılımın kullanıcıları bir trafik senaryosu oluşturmak için yollar, dönel kavşaklar, yaya geçitleri, üç renkli ve tek renkli trafik ışıkları çizebileceklerdir. Yolların kapasiteleri ve yoğunlukları belirtilince, yazılım belli bir süre sonra ne kadar trafik sıkışıklığı olacağını gösterecektir. Kullanıcı bu senaryoyu bir dosya olarak kaydedebilecektir.

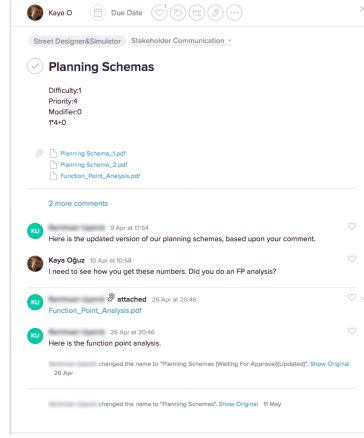
Coppit sınıfı yöneticiler ve geliştiriciler olarak ayırmıştır. Takımlar kendilerine lider seçebilmektedir. Bu yaklaşımın yerine, SE315 dersindeki proje takımlarındaki her öğrenci eşit sürelerde proje yöneticisi konumuna getirilmiştir. Bu süre, takımdaki öğrenci sayısına göre 10 gün ile iki hafta arasında değişmektedir. Böylece her öğrenci bir süre proje yöneticiliği yapma fırsatı kazanmış ve projenin her parçasının tasarlanmasında, gerçekleştirilmesinde ve yürütülmesinde yer almışlardır.

Coppit öğrencileri değerlendirmek için, isim vermese de, bizim Görev Puan Sistemi olarak adlandırdığımız bir yöntem önermektedir. Bu görev puan sistemine göre projedeki herkes görev açabilir ve bu göreve 1-10 değerleri arasında üç ayrı puan verir. Bu puanlar öncelik, zorluk ve değiştirici olarak isimlendirilmiştir. Öncelik ve zorluk puanları görevin önceliğini ve zorluğunu belirtirken, değiştirici puanı yöneticiye ve öğretim elemanına görevin sonucuna göre ek bir not vermesini sağlar. Örneğin iyi yapılan işlerde bu puan yönetici ve öğretim elemanı tarafından yükseltilebilir.

Görev açıldıktan sonra yönetici veya öğretim elemanı tarafından puanlar güncellenebilir. Görev puanı öncelik,  $p$ , ile zorluk,  $d$ , değerlerinin çarpımına değiştirici,  $m$ , değerinin eklenmesi ile hesaplanır. Görev tamamlandığında ise öğrenci görev durumunu “Yönetici Bekleniyor” olarak ayarlar. Yönetici görevi gözden geçirir ve kapatır. Coppit’in öğrencileri web tabanlı Issue Tracker (<http://issue-tracker.sf.net/>) yazılımının görev puan sistemi için özelleştirilmiş bir sürümünü kullanmışlardır.

Görev puan sistemi SE315 dersinde benzer şekilde uygulanmıştır. Issue Tracker yazılımının özelleştirilmiş sürümüne sahip olmadığımız için öğrenciler istedikleri bir web tabanlı proje yönetim sistemine yönlendirilmişlerdir. Kullanılacak sistemde yeni bir görevin açılıp, bu görevin geliştiricilere atanabilmesi koşulu aranmıştır. Görevlere yapılan yorumlarla da puanların belirtilmesi istenmiştir. Büyük çoğunluk Trello (<http://trello.com>) yazılımını kullanırken, GitHub’ın (<http://github.com>) hata takip sistemini seçenler de olmuştur. Bir ekip ise Asana (<http://www.asana.com>) yazılımını kullanmıştır. Örnek bir ekran görüntüsü Şekil 1 ile verilmiştir.

Coppit, projelerin değerlendirilmesinde takım ve bireysel görevlerin puanlarını kullanmıştır. Gruba ait geçerli görevlerin puanları toplanmış, ve bu puanlardan tamamlanan görevlere ait puanlarla oranlanmışlardır. Örnek bir görev puan tablosu, Tablo 1 ile verilmiştir. Bu tabloda proje grubu geliştiricileri toplamda 125 puanlık görev açmışlardır. Bu görevlerden 110 puan toplayabilmişlerdir. Bu oran,  $(110/125) \times 100 = 88$ , takım notu olarak belirlenmiştir. Bireysel puanların hesaplanmasında, her öğrencinin eşit miktarda katkı yapması beklenmektedir. Buradaki 125 puan, her öğrencinin  $125/3 = 41.66$  puan toplaması gerektiğini göstermektedir. Tablo 1 ile OGR1, OGR2 ve OGR3 öğrencilerinin sırayla 58, 39, 13 puan topladıklarını görebiliyoruz. Bu notları beklenen 41.66 notuyla oranlanarak hepsinin bireysel notu hesaplanabilir. Bu notlar sırayla, 139, 94 ve 31’dir. Projeye çok fazla katkı sağlayan OGR1, 139 puan yerine bireysel olarak 100 puan alacaktır. Coppit, takım ve bireysel notların ortalamasını öğrencinin son notu olarak belirlemiştir. Son notlar, 94, 91 ve 60 olarak he-



Şekil 1. Asana ile oluşturulmuş bir görevin ekran görüntüsü.

saplanabilir. Aynı sistem SE315 dersi için de kullanılmıştır. Fakat son notun hesaplanmasında bireysel nota %60, takım notuna %40 oranları verilmiştir.

Tablo 1. Örnek bir görev puan tablosu.

ID	Özet	Puan ( $p \times d + m$ )	Durum	Atanan
1	Dönel Kavşaklar	$4 \times 6 + 5 = 29$	Tamamlandı	OGR2
2	Yaya Geçidi	$4 \times 4 + 2 = 18$	Tamamlandı	OGR1
3	Benzetim	$6 \times 6 + 4 = 40$	Tamamlandı	OGR1
4	Kullanıcı El Kitabı	$3 \times 2 + 4 = 10$	Tamamlandı	OGR2
5	Kurulum Programı	$4 \times 3 + 3 = 15$	Devam ediyor	OGR3
6	Yazılım Testi	$5 \times 2 + 3 = 13$	Tamamlandı	OGR3
Biten / Toplam		110 / 125		

### 3.2 Projenin Yürütülmesi

Dönemin ikinci haftasında Trafik Benzetim Projesi öğrencilere sunulmuştur. Aynı zamanda görev puan sistemi anlatılmış ve öğrencilerden hem bu sistemi kullanabilecekleri, hem de proje yönetimi sırasında kullanabilecekleri, öğretim elemanın da katılabileceği çevrim içi bir proje yönetim yazılımı belirlemeleri istenmiştir. Proje içi iletişimin buradan yürütülmesi gerektiği vurgulanmıştır.

Proje yöneticiliği görevini herkesin yapacağını, bu görevin puan dağılımının zorluk olarak 2, önem olarak 10 ve değiştirici olarak 5 olduğu belirtilmiştir. Proje yöneticisi olmanın diğer öğrenci arkadaşlarına göre daha üst bir konum olmadığı, sadece farklı bir görev olduğu belirtilmiştir. Bu görevin gereklilikleri anlatılmış, örnek olarak toplantıların yapılması ve yürütülmesi, projenin plana uygun şekilde gidip gitmediğinin kontrol edilmesi ve görev puanlarının gerektiğinde

düzenlenmesi verilmiştir. Yöneticinin bu görevleri olduğu için, görev sırasında başka bir görev almamaları istenmiştir.

Görev puan sisteminde puanlandırma için şu şekilde bir yaklaşım izlenmesi istenmiştir. Eğer görev yazılım geliştirme sürecindeki adımlardan biriye, öncelik değeri olarak 4-6 arasında bir puan verilmesi; 1-3 arasında ek özellikler, 7-10 arasında da önemli hatalar için kullanılması önerilmiştir. Zorluk puanı için de daha önce sık yapılmış görevler için 1-3, yazılım geliştirme süreçlerinde ilk kez uygulanan görevler için 4-6, karmaşık algoritmalar gibi durumlarda ise 7-10 arası değerler verilmesi istenmiştir. Değiştirici değerinin 0 ile başlaması, daha sonra geliştiricileri motive etmek, ya da iyi yapılan işleri ödüllendirmek için kullanılması önerilmiştir.

Projenin geliştirme evresi için bütün öğrencilerden bir GitHub hesabı edinmeleri ve burada herkese açık olarak projelerini geliştirmeleri istenmiştir. Projenin gerçekleştirme evresinde bir Git kullanımı sunumu yapılmış ve sürüm kontrol sistemleri tanıtılmıştır. Git sunucusuna herkesin kaynak kod katkısı yapması istenmiş ve her hafta laboratuvar saatlerinde yapılan proje toplantılarında kontrol edilmiştir.

Takımlar proje geliştirme yöntemleri konusunda serbest bırakılmışlardır. Proje sürecinde bütün takımlardan bir gereksinim analizi belgesi istenmiştir. Bu belgenin amacı, herkesin projeyi doğru anladığından emin olmaktır. Daha sonraki belgelerin hiçbiri zorunlu tutulmamış, öğrencilerin kendilerinin belirlediği yazılım mühendisliği süreçlerini takip etmeleri beklenmiştir. Projenin geliştirilmesi için de, masaüstü uygulaması olması koşulu ile, programlama dili ve kütüphaneleri öğrencilerin seçimine bırakılmıştır. Öğrenciler daha çok C++ ile Qt Kütüphanesi (<https://www.qt.io/>), Java Swing ve Python Tkinter ortamlarını tercih etmiştir.

### 3.3 Karşılaşılan Problemler

Dönem boyunca 96 öğrenci toplam 20 adet ekip oluşturmuştur. Görev puan sistemi istekli ve çalışkan ekiplerin kendi kendilerini idare etmelerini sağlamış olsa da takım sayısının fazla olmasından dolayı projeden uzaklaşan ekiplerin fark edilmesinde geç kalınmıştır. Dönem sonuna doğru ise dersten kopan öğrencileri geri kazanmak mümkün olmamıştır.

Proje tamamlandıktan sonra öğrencilerden bireysel bir proje sonu raporu istenmiştir. Bu raporda takım çalışması süresince kendilerini ve ekiplerini değerlendirmişlerdir. Takım olarak hatalarını ve doğrularını, nedenleri ile birlikte eleştirel bir biçimde raporlamışlardır. Edinilen bu bilgiler ışığında en çok karşılaşılan problemlerin başında takım içi tartışmalar gelmektedir. Bu tartışmalar hem projenin ilerlemesini, hem de bir takım olarak birlikte çalışmalarını engellemiştir. Projeyi gerçekleştirmek isteyen bazı öğrencilerin kendilerine güvenememeleri ya da masaüstü uygulama geliştirmede deneyimlerinin olmaması, projenin süreç tahminleme ve gerçekleştirme evresinde gecikmelere sebep olmuştur. Proje basit gözükse de, daha önce öğrenilen veri yapıları ve nesne yönelimli programlama yöntemlerinin probleme uygulamasında zorluklar yaşanmıştır. Problemin çözüle-



meyince iş bölümü de dengeli yapılamamıştır. Bu proje takvimini yaratırken de problem çıkarmıştır.

Öğrencilerden gereksinim analizi belgesi dışında proje ile ilgili bir belge istenmemiş ve yazılım geliştirme süreçlerini uygulamaları beklenmiştir. Tasarım yapılmadan ya da tasarım yapılsa bile tasarım belgesi hazırlanmadan geliştirilmeye çalışılan projeler gerçekleştirme evresinde problemler yaşamışlardır. Ekipteki her geliştiricinin projeyi farklı düşündüğü ofis saatlerinde projeyi tartışmak için geldiklerinde sordukları bireysel sorularla anlaşılmıştır. Belgelerin öneminin vurgulanması ve istenmese de hazırlanmasının gerekliliği açıkça ortaya çıkmıştır.

Dersin not dağılımında projenin katkısının %15 olması, birçok öğrenciyi proje yerine daha kolay gördükleri ödev ve sınav notlarına yöneltmiştir. Bir sonraki dönemlerde projenin oranının daha çok artırılması planlanmıştır. Coppit %40 gibi yüksek bir oran önermektedir.

Görev puan sisteminin farklı web servisleri ile kullanılması puanların toplanmasını zorlaştırmıştır. Öğrenciler bu servisleri görev puan sistemiyle birlikte kullanırken farklı yaklaşımlar sergilemişler, bu yüzden öğretim elemanı ile iletişimde de kopukluklar olmuştur. Bazı takımlar görevlere yüksek puan vermenin yine yüksek not olarak döneceğini düşünürken, bazı takımlar ise bir göreve bütün takımı dahil ederek bireysel katkıların takip edilmesini yine zorlaştırmışlardır.

### 3.4 Faydalı Durumlar

Öğrencilere takım içinde adil olarak not alacaklarını gösterdiği için görev puan sisteminin çalışmaya teşvik eden bir yapısı olduğu proje sonu raporlarından anlaşılmıştır. Alacakları nota takım katkısı kadar, bireysel katkı da olduğu için her öğrencinin eşit katkı yapması gözletilmektedir.

Proje ekipleri içerisinde yazılımı tamamlamak için çalışan ekipler olmuştur. Bu ekipler derslerde, ders aralarında ve öğretim elemanının uygun olduğu ders dışı saatlerde proje üzerine sorular sorarak sonuca ulaşmaya çalışmışlardır. Bu ekiplere hem programlama, hem yazılım tasarımı ve mimarisi, hem de uygulanabilecek algoritmalar açısından yol gösterilmiştir.

Projenin devamlı takibi, sürekli olarak projenin tamamlanması gerektiğinin belirtilmesi, proje gereksinimlerine uymayan, ama öğrencilerin işine gelen isteklerin sürekli reddedilmesi ile projeye gerçekçi bir ortam sağlamaya çalışılmıştır. Bu da takımların gerçek projelerde ve özellikle bitirme projelerinde karşılaşacakları problemler için bir deneyim olmuştur.

### 3.5 Öğrenci Anketi

Dönem sonunda, öğrencilerin de yorumlarını alabilmek için deneyimlerini sorgulayan bir anket hazırlanmış ve uygulanmıştır. Bu ankete sadece gönüllüler katılmış ve ders notuna herhangi bir katkı yapılmamıştır. Ankete anonim olarak 23 öğrenci katılmıştır. Katılımın az olmasının verilen cevapların bütün öğrencilerin görüşü olarak algılanmaması gerektiğini göstermektedir.

Anket soruları üç ana bölümden oluşmaktadır.

1. İlk bölümdeki deneyim soruları öğrencileri daha yakından tanımak ve önceki deneyimlerini göz önüne alarak ders projesini değerlendirmeyi amaçlamaktadır.
  - Bu derste ki proje dışında kaç tane ders projesi tamamladınız?
  - Ders dışında kaç tane proje tamamladınız?
  - Ders dışı ya da ders projesi olarak kaç projede bir takımın üyesi olarak çalıştınız?
  - Bu ders dışında yer aldığınız proje takımları en fazla kaç kişiydi?
  - Eğer yeni bir projeye başlayacak olursanız hangi programlama dilini kullanmak istersiniz?
2. İkinci bölümde proje ile ilgili sorular yer almaktadır.
  - Çoktan seçmeli soru olarak: Bu projeye paradigması (nesne yönelimli gibi), programlama dili ve ara yüzü olarak benzer proje geliştirdiniz mi? Her bir kategori için Aynı, Farklı ve Hiç seçenekleri verilmiştir.
  - Proje için kaç saat çalıştınız?
  - En çok hangisine zaman harcadınız? Organizasyon ve yönetim, iletişim, belgeler, gereksinim analizi, uygulama tasarımı, gerçekleştirme.
  - Hangi yazılım geliştirme yöntemini uyguladınız? Şelale, spiral, artımlı, çevik, prototip.
  - Kullandığımız geliştirme yöntemi yararlı oldu mu?
3. Son bölümde ise takım çalışması hakkında sorular sorulmuştur.
  - Bir takım olabildiniz mi?
  - Aynı takımla bir proje daha geliştirmek ister misiniz?
  - Projeye 5 üzerinden ne kadar katkı sağladığınızı düşünüyorsunuz?
  - Takım çalışmasının en zor kısmı nedir? İletişim, karar verme, başkalarını dinlemek ve onlarla çalışmak, Git gibi bir sürüm kontrol sistemi kullanmak, planlama ve programlama, gereksinim analizi, belgeler.

Anketin ilk kısmına verilen cevaplara göre, öğrencilerden %59,1'i daha önce yalnızca bir yazılım projesinde yer almış, %13,6'sı ise hiçbir projede yer almamıştır. Büyük çoğunluk, (%59,1) dersler dışında bir yazılım projesi geliştirmişlerdir. %27,3'ü ise ders dışı 1 proje geliştirmişlerdir.

İkinci kısımdaki cevaplara göre öğrencilerin yarısı projeye en fazla 40 saat, %27,3'ü 41-80 saat, %13,6'sı ise 81-120 saat arasında zaman ayırdıklarını belirtmişlerdir. Bu süreler öğrencilerin deneyimlerine bağlıdır ve sonraki dönemlerdeki projeler için fikir sahibi olmaya yarayacaktır. Proje yönetimi sırasında en fazla (%59,3) organizasyon ve yönetime, daha sonra (%50) projeyi gerçekleştirmeye ve belgelemeye, en az da uygulama tasarımına (%31,8) ve iletişime (%27,3) zaman ayırmışlardır. Büyük çoğunluk çevik geliştirme (%59,1) yaklaşımını kullanırken, diğerleri yine şelale yaklaşımını (%22,7) tercih etmişlerdir. Kullandıkları yaklaşımın büyük oranda (%54,5) yararlı olduğunu belirtmişlerdir.

Proje yönetimin en önemli kısımlarından biri öğrencilerin takım çalışması becerisini kazanmaları ve bir takımın üyesi olarak üzerlerine düşen görevleri yapabilmeleri gerekmektedir. Bu bağlamda, “Bir takım olabildiniz mi?” sorusunda evet ve hayır eşit not almasına rağmen “Aynı takımla bir proje daha geliştirmek ister misiniz?” sorusuna %68,2'si olumsuz yanıt vermiştir. “Projeye 5 üzerinden ne kadar katkı sağladığınızı düşünüyorsunuz?” sorusuna %45,5'i 4 değerini

işaretlerken, 3 (%22,7) ve 5 (%18,2) değerleri de birbirlerine yakın olması dikkat çekmiştir. Ankete katılan öğrenciler takım çalışmasının en zor alanının planlama ve programlama (%77,3) olduğunu, daha sonra iletişim (%63,6) ve karar verme (%54,5) olarak belirtmişlerdir.

## 4 Sonuçlar ve Gelecek Çalışmalar

Proje yönetiminde görev puan sisteminin kullanılması, proje sürecinde ve değerlendirilmesinde öğrencilerin katkılarının uygun şekilde değerlendirilmesine yardımcı olmaktadır. Aynı zamanda öğretim elemanını gerektiği yerde müdahale etmesine, onun dışında grubun kendi kendini idare edebilmesine olanak vermektedir.

Geçen ilk dönemin ardından, görev puan sisteminde çeşitli değişiklikler yapılması da planlanmaktadır. Zorluk ve öncelik değerleri verilirken 1-10 arası puan yerine değeri tanımlayan ve daha anlaşılır olan “kolay, normal, zor” ifadeleri kullanılacaktır.

Görev puan sisteminin en çok ihtiyaç duyduğu kısım ise proje evreleridir. Dönem içerisinde 20 ekibin büyük çoğunluğu projeyi tamamlayamamıştır. Projeyi evrelere ayırmak ve bu evrelere belli oranlarda puan vermek öğrencilerin kısmi notlandırmalarına yardımcı olacaktır.

Görev puan sisteminin Trello ya da Asana gibi var olan çevrim içi hizmetlerle birlikte kullanılması zorluklar çıkarmaktadır. Çevrim içi proje yönetiminin görev puan sistemi ile bütünleşik olması projelerin daha kolay yürütülmesini sağlayacaktır. Ayrıca takımların ilerlemelerinin çevrim içi izlenebilmesi projeden uzaklaşan takımların daha erken fark edilerek tekrar kazanılmalarına yardımcı olacaktır.

Görev puan sistemi projenin takip ve değerlendirilmesi için kolaylık sağlasa da anket sonuçlarında öğrencilerin yine de organizasyon ve yönetime çok fazla süre ayırırken yazılım tasarımına çok daha az süre ayırdıkları ve takım arkadaşlarından memnun olmadıkları, dolayısıyla iletişim için de az zaman harcadıkları ortaya çıkmıştır. Bunları projelerin büyük çoğunluğunun tamamlanamamasının nedenleri olarak da sayabiliriz. Görev Puan Sisteminin proje yürütülmesinin önüne geçmeden ve ek yük çıkarmadan işlemesi yönetime harcanan süreyi de daha az etkilemesini sağlayacaktır. İletişimin daha iyi olması da öğrencilerin takım oluştururken arkadaşlarını daha yakından tanımaları ile mümkün olacaktır. Öğrencilerin geçmiş projelerine ait bilgilerin tutulması, dönem başında takımlar oluşturulurken yardımcı olacaktır.

## Teşekkürler

SE315 Yazılım Proje Yönetimi dersini 2016/2017 Bahar döneminde alan iki şubedeki bütün öğrenci arkadaşlarımıza, katkılarından dolayı değerli hakemlerimize ve Damla Oğuz’a teşekkür ederiz.

## Kaynakça

1. Vasilevskaia, M., Broman, D., Sandahl, K.: Assessing large-project courses: Model, activities, and lessons learned. *Trans. Comput. Educ.* **15**(4) (December 2015) 20:1–20:30
2. Coppit, D.: Implementing large projects in software engineering courses. *Computer Science Education* **16**(1) (2006) 53–73
3. Ardis, M., Budgen, D., Hislop, G.W., Offutt, J., Sebern, M., Visser, W.: Se 2014: Curriculum guidelines for undergraduate degree programs in software engineering. *Computer* **48**(11) (Nov 2015) 106–109
4. Börstler, J., Hilburn, T.B.: Team projects in computing education. *Trans. Comput. Educ.* **15**(4) (December 2015) 16:1–16:5
5. Wilkins, D.E., Lawhead, P.B.: Evaluating Individuals in Team Projects. In: *Proceedings of the Thirty-first SIGCSE Technical Symposium on Computer Science Education. SIGCSE '00, New York, NY, USA, ACM (2000)* 172–175
6. Clark, N., Davies, P., Skeers, R.: Self and Peer Assessment in Software Engineering Projects. In: *Proceedings of the 7th Australasian Conference on Computing Education - Volume 42. ACE '05, Darlinghurst, Australia, Australia, Australian Computer Society, Inc. (2005)* 91–100
7. Herbert, N.: Quantitative Peer Assessment: Can Students Be Objective? In: *Proceedings of the Ninth Australasian Conference on Computing Education - Volume 66. ACE '07, Darlinghurst, Australia, Australia, Australian Computer Society, Inc. (2007)* 63–71
8. Farrell, V., Ravalli, G., Farrell, G., Kindler, P., Hall, D.: Capstone Project: Fair, Just and Accountable Assessment. In: *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education. ITiCSE '12, New York, NY, USA, ACM (2012)* 168–173
9. Farrell, V., Farrell, G., Kindler, P., Ravalli, G., Hall, D.: Capstone Project Online Assessment Tool Without the Paper Work. In: *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education. ITiCSE '13, New York, NY, USA, ACM (2013)* 201–206
10. Aggarwal, P., O'Brien, C.L.: Social Loafing on Group Projects: Structural Antecedents and Effect on Student Satisfaction. *Journal of Marketing Education* **30**(3) (2008) 255–264