Yeni Bir Ders Tasarımı: Standartlarla Yazılım Geliştirme

Çağatay Çatal
İstanbul Kültür Üniversitesi, Bilgisayar Mühendisliği Bölümü, İstanbul
c.catal@iku.edu.tr

Özet: Yazılımların artan karmaşıklığı ile birlikte, mevcut standartlara bağlı olarak yazılım geliştirmenin önemi gün geçtikçe artmaktadır. "Standart" terim olarak Bilgisayar Mühendisliği ve/veya Yazılım Mühendisliği programlarının bazı derslerinde geçse de, endüstrideki ihtiyaçları karşılamaya dönük olarak yeterli düzeyde bilginin verilebildiğini söyleyebilmek güçtür. Öğretim programları konusunda rehber niteliğindeki en güncel raporlar olan; "Computer Science Curriculum" (CS2013) ve "Software Engineering 2004" (SE 2004) raporları incelendiğinde, özellikle yazılım geliştirme standartları bağlamında bazı eksikliklerin bulunduğu değerlendirilmiştir. Bu ihtiyaca bağlı olarak; İstanbul Kültür Üniversitesi (İKÜ), Bilgisayar Mühendisliği lisans programı içerisinde seçmeli ders düzeyinde, "Standartlarla Yazılım Geliştirme" ismiyle bir ders tasarlanarak, dersin içeriği saptanmış ve bu dönem içerisinde 4. sınıf lisans öğrencilerine verilmiştir. Bu bildiride; bu dersin nasıl tasarlandığını, hafta bazında anlatılacak konu içeriğini ve genel olarak elde ettiğimiz deneyimleri sunuyoruz. Dönemin tamamlanmasının ardından öğrenciler ile anketler yapılmış, anket sonuçlarına göre ders içeriğinde değişiklikler planlanmıştır.

Anahtar Sözcükler: Emniyet-kritik kodlama standartları, kalite standartları, süreç standartları, platform standartları, metrikler, yazılım geliştirme standartları.

A New Course Design: Software Development with Standards

Abstract: As software systems are becoming more and more complex, the importance of software development based on existing standards is increasing day by day. Although the term "standard" already exists in some of the courses in Computer Engineering and/or Software Engineering programs, it's difficult to say that satisfactory knowledge is taught to fulfill the requirements of software industry. After careful examination of the most recent reports that are guidelines for Computer Science and Software Engineering curriculums (CS2013 and SE 2014), it is evaluated that there are still some shortcomings in the scope of software development standards. Based on this need, an elective course called "Software Development Standards" was defined, designed and taught for 4th year undergraduate Computer Engineering students at the Department of Computer Engineering in Istanbul Kultur University. In this paper, we present how we designed this course, the content of the course for each week and the experience we got. After the semester finished, questionnaires were delivered to the students and then, course content has been updated based on the feedback received from the students.

Keywords: Safety-critical coding standards, quality standards, process standards, platform standards, metrics, software development standards.

1. Giriş

Yazılımların artan karmaşıklığı ile birlikte, kullanıcı beklentilerinin de artması, standartlara bağlı olarak yazılım geliştirmenin önemini hiç olmadığı kadar arttırmıştır.

Bilgisayar Mühendisliği ve Yazılım Mühendisliği öğretim programları kapsamında "standart" terimi kavram olarak;

- Yazılım Kalitesi
- Yazılım Geliştirme Süreçleri
- Yazılım İnşası (Construction)
- Yazılım Test ve Analizi
- Yazılım Metrikleri

gibi derslerde kısmen ele alınmakla birlikte, kapsamlı olarak yazılım geliştirme için bilinmesi gereken standartların tartışıldığı bir kapsamlı ders bulunmamaktadır.

Yazılım geliştirme bağlamında kullanılabilecek standartlar üzerine yaptığımız araştırmalarda, standartlar konusunun 6 başlık altında değerlendirilebileceğini tespit ettik. Bu başlıklar aşağıda sunulmaktadır:

- Süreç standartları
- Kodlama standartları
- Kalite standartları
- Platform standartları
- Güvenlik standartları
- Metrik standartları

Süreç standartları; genel olarak yazılım geliştirme yaşam çevrimine (software development life cycle - SDLC) odaklanan standartlar olarak belirlenmiştir. Yazılımın ihtiyaç tanımından başlayarak, yazılım gereksinim analizi, yazılım ön tasarımı ve detaylı tasarımı, yazılım gerçeklemesi (implementation), yazılım testi ve yazılım bakım (maintenance) aktivitelerinin tümünün ele alındığı standartlardır.

Süreç standardı kapsamında, tasarlanan ders içeriğine **IEEE / EIA 12207** [1] standardı somut olarak eklenmiş bir standarttır. Bu standart, 1996 yılında ortaya konulmuş bir standart olup yazılım endüstrisinin kabul etmiş olduğu yazılım yaşam çevrimi süreçlerinin tanımlı olduğu bir standarttır.

1994 yılında Savunma Sanayi için ABD tarafından geliştirilmiş olan **MIL-STD-498** "Yazılım Geliştirme ve Dokümantasyon" standardı [2], IEEE/EIA 12207 standardıyla birçok benzerlikler içermektedir. Örneğin; dokümantasyon bağlamında benzer gereksinimler, kalite güvence ve konfigürasyon yönetim süreçleri anlamında benzerlikler, farklı yaşam döngüsü süreçlerinin önerilmesi ve yazılım geliştirme odaklılığı sayılabilir [3]. Farklı olan yönler açısından; dokümantasyon adlandırma farklılığı ve MIL-STD-498'de tedarik, bakım, operasyon süreçlerinin mevcut olmamasını sayabiliriz [3].

IEEE/EIA 12207 standardı ile ilişkili üç adet standart raporu mevcuttur:

IEEE/EIA 12207.0 Yazılım Yaşam Çevrimi Süreçleri: Bu belge 1996 yılında hazırlanmış olup 87 sayfadan oluşmaktadır. İçerisinde; temel süreçler, destekleyici süreçler ve organizasyonel süreçler açıklanmaktadır. Temel süreçler olarak; satın alma süreci, tedarik süreci, geliştirme süreci, bakım süreci ve operasyon süreci yer almaktadır. Destekleyici süreçler olarak; dokümantasyon, konfigürasyon yönetimi, kalite güvence, gözden geçirme, denetlemeler, doğrulama, geçerleme, problem çözümleme süreçleri verilmektedir. Organizasyonel süreçler ise, yönetim, altyapı, iyileştirme ve eğitim süreçleridir [3].

- **IEEE/EIA 12207.1 Yaşam Döngüsü Verileri:** Üretilen verilerin nasıl kayıt edileceği bu raporda açıklanmıştır.

- **IEEE/EIA 12207.2 Gerçekleme Konuları:** Standardın nasıl uyarlanabileceği başta olmak üzere standardın uygulanmasına dönük olarak gerekli olabilecek konuları içeren belgedir.

Tasarlanan ders kapsamında; daha güncel olması ve daha geniş uygulama alanı olması nedeniyle, **IEEE/EIA 12207** standardı bir derste ele alınmıştır.

Bu standart; Yazılım Mühendisliği öğretim programları için rehber niteliğindeki SE 2004 raporunun 37. sayfasında yer almaktadır. SE 2004 raporundaki; "Yazılım Süreci" (Software Process) bilgi alanı (knowledge area) altında bulunan, "Süreç Gerçeklemesi" (Process Implementation) bilgi biriminin (knowledge unit) en sonu konusu (topic) olarak, "Yazılım Yaşam Çevrim Süreci için Gereksinimler (örneğin; ISO/IEEE 12207)" yazılmıştır. Bu nedenle, Yazılım Mühendisliği programlarında bu standartın anlatılıyor olması beklenmektedir.

Computer Science Curricula 2013 (Ekim 2013 - Final Rapor 0.9 - sürüm öncesi versiyon) raporunu incelediğimizde, bu yönde bir standartın ders içeriklerinde yer almadığını tespit ettik. Kodlama standartları bazı ders içeriklerinde (Yazılım İnşası dersi içerisinde) yer almasına rağmen, geliştirme süreçleri bağlamında bir standart raporda sunulmamaktadır.

Kodlama standartları; özellikle emniyet-kritik yazılım geliştirirken olmazsa olmaz standartlardandır. Yüksek güvence gerektiren ve insan hayatını etkileyebilecek yazılımları düşündüğümüz zaman, gerek kodlama standartlarının uygulanması gerekse de statik analiz araçlarının kullanılması oldukça önemlidir.

Emniyet-kritik yazılım geliştirme için kullanılabilecek standartlar olarak; MISRA C/C++ [4], JSF++ [5], High Integrity C++ [6], CERT C Güvenli Kodlama Standardı [7], EC, Netrino Embedded C Kodlama Standardı, JPL "Power of 10" standartlarını sayabiliriz.

Ders kapsamında aşağıdaki standartların, yaygın şekilde kullanımı söz konusu olduğundan, ders kapsamında ele alınmasına karar verilmiştir:

- MISRA C/C++
- JSF++
- High-Integrity C++
- CERT Güvenli Kodlama Standardı

SE 2004 raporunu incelediğimizde (135 sayfa), kodlama standartları bağlamında herhangi bir içeriğe rastlamadık. Ayrıca, emniyet-kritik uygulama geliştirmede kullanılabilecek kodlama standartlarının programlarda yer alması gerektiği konusunda bir bilgiye de rastlamadık.

CS 2013 raporunda [8] (514 sayfa) ise Yazılım İnşası dersi altında kodlama standartlarının geçtiğini tespit ettik. Bu nedenle, CS 2013 raporu bu ihtiyacı şu anda karşılamakta iken SE 2004 raporunda [9] bu yönde bir eksiklik olduğunu değerlendiriyoruz. Programlarını sadece SE 2004 raporunu dikkate alarak tasarlamış olan Yazılım Mühendisliği lisans programı yöneticilerinin, bu yönde bir düzenleme yapmasının faydalı olacağını öneriyoruz.

Kalite standartları; yazılım kalitesi ile ilişkili olan standartlar bu kapsamda değerlendirilmiştir. Ayrıca, yazılım iyileştirme çerçevelerini de bu bağlamda ele aldık. Örneğin; CMMI (Capability Maturity Model Integration) [10] ve SPICE gibi süreç iyileştirme çerçeveleriyle ISO 9000 gibi Kalite Sistem Standartları bu grupta değerlendirilmiştir.

Bu tür standartlar, genellikle birçok programda anlatılan derslerde sunulmaktadır. Her iki raporda da (SE 2004 ve CS 2013) bu kapsamda içeriğe rastlanmıştır. SE 2004 raporunun 37. sayfasında Yazılım Kalitesi dersi içerisinde bu konunun ele alınabileceği ifade edilmektedir. CS 2013 raporunun 177. sayfasında ise Software Processes dersinde bu konuların ele alınabileceği yazılmıştır.

Tasarlanan ders içeriğinde, yazılım sektöründe son dönemde oldukça önem kazandığı için, **CMMI** konusunun bir hafta ele alınmasına karar verilmiştir.

Platform standartları; havacılık, medikal, tren yazılımı, otomotiv, nükleer santral gibi farklı platformlara özgü olarak gerekli olabilecek standartları içermektedir.

SE 2004 ve CS 2013 raporları incelendiğinde, bu bağlamda gerekli olabilecek standartlara veya bu yönde içeriğin tanımlanması gerektiği konusunda bir bilgiye rastlanmamıştır. Bu alanlarda geliştirilecek olan yazılımların her birisi için önemli ölçüde farklı standartlar mevcut olduğundan, tasarlanan dersin birçok haftası bu gruptaki standartlara ayrılmıştır.

Konular aşağıda sunulmaktadır:

- **DO-178B**, Havacılık endüstrisi yazılımı
- **DO-278** ve **ED-109**, İletişim, Navigasyon, Gözetleme, Hava Trafik Yönetim Sistemleri (CNS/ATM)
- **EN 50128**, Tren Kontrol ve Koruma Yazılımı
- **ISO 26262,** Otomotiv için Emniyet Standardı
- **ISO 60880, N**ükleer Santraller için Standart
- **IEC 61508**, Elektrik/Elektronik/Programlanabilir Elektronik Sistemler için Fonksiyonel Emniyet

Metrik standartları; yazılım geliştirirken kod üzerinden toplanabilecek metriklerin karşılaştırılması gereken standart seviyeleri tanımlamaktadır. Başka bir ifadeyle, toplanan metrik değerlerinin, endüstri eşik seviyeleriyle kıyaslaması gerçekleştirilmelidir.

SE 2004 ve CS 2013 raporlarında, bu metriklerin eşik seviyeleri veya endüstri standartları konusunda bilgi verilmesi hususunda bir bilgiye rastlanmamıştır. Örneğin, çevrimsel karmaşıklık (Cyclomatic Complexity) metriğini toplayan bir kalite uzmanı, bunu hangi endüstri standardına göre kontrol etmelidir? Bu kapsamda, NASA ve SEI Yazılım Mühendisliği Enstitüsü'nün kabul ettiği standart eşik seviyeleri derste öğrencilere sunulmalıdır. Aksi halde toplanan bilgi uygulamaya geçmediği için fayda sağlamayacaktır. Tasarlanan derste bir hafta, toplanabilecek ürün metrikleri açıklanmakta ve endüstri standartları sunulmaktadır.

Güvenlik standartları kapsamında son dönemde en fazla önem kazanan standart, ISO/IEC 15408 Ortak Kriterler (Common Criteria) [11] olarak tespit edilmiştir. SE 2004 ve CS 2013 raporlarında bu standartla ilgili veya bilgisayar güvenlik sertifikasyonu konusunda bir bilgiye rastlanmamıştır. Tasarlanan dersin bir haftasında bu konunun ele alınmasına karar verilmiştir.

Tablo 1'de, ders kapsamında belirlenen yazılım geliştirme standartlarıyla ilgili konuların, SE 2004 ve CS 2013 raporlarında ele alınıp alınmadığı temsil edilmektedir. X işareti konunun ele alınmadığını, √ işareti ele alındığını göstermektedir.

	SE 2004	**CS 2013**
Süreç standardı	√	X
Kodlama standardı	X	√
Kalite standardı	√	√
Platform standardı	X	X
Metrik standardı	X	X
Güvenlik standardı	X	X

Tablo 1. SE 2004 ve CS 2013 raporlarının standartlar bağlamında değerlendirilmesi

Bir sonraki bölümde ders için belirlenen haftalık taslak program verilmiştir. 3. bölümde deneyimler, 4. bölümde sonuç ve öneriler sunulmaktadır.

2. Ders için Belirlenen Haftalık Program

Ders için belirlenen haftalık program aşağıda sunulmaktadır:

1. Ders Tanıtımı
2. IEEE/EIA 12207 Standardı
3. CMMI Süreç Çerçevesi
4. MISRA C/C++ ve JSF++
5. High-Integrity C++ ve CERT
6. Metrikler ve Eşik Seviyeleri
7. Ortak Kriterler
8. Vize Sınavı
9. DO-178B Havacılık Yazılımı
10. DO–278 / ED–109 Standardı

11. EN 50128 Tren Yazılımı
12. ISO 26262 Otomotiv Standardı
13. IEC 60880 Nükleer Santraller
14. IEC 61598 Standardı

Taslak mahiyetinde hazırlanan bu standart, ihtiyaçlara bağlı olarak değiştirilebilmektedir. Örneğin, platform standartları sayısı azaltılabilir ve kodlama standartları daha uzun süren haftalarda öğrencilere anlatılabilir.

3. Deneyimler

Programlama konusunda yetenekli öğrencilerin, özellikle emniyet-kritik standartlar olan MISRA C/C++, JSF++ gibi kodlama standartlarındaki kuralları merakla takip ettiği ve öğrenmeye çalıştığı gözlemlenmiştir. Ayrıca, yazılım metrikleri ve bu metriklerin eşik seviyelerinin tartışıldığı konular, yine programlama konusunda meraklı öğrenciler açısından oldukça ilgi çekici olmuştur.

SE 2004 ve CS 2013 raporlarında, standartlar konusunda bazı eksikliklerin mevcut olduğu tespit edilmiş, bu kapsamda tespit edilen hususların raporları hazırlayan gruplara iletilmesine karar verilmiştir.

Emniyet-kritik kodlama standartlarına uyulup uyulmadığını kontrol etmek üzere çeşitli araçlar mevcuttur ancak bu araçların oldukça yüksek maliyetli olması nedeniyle laboratuarlarda öğrencilerin kullanımına açılması bir problem olarak karşımıza çıkmıştır. Understand ve Parasoft isimli ticari ürünlerin kısa süreli sürümleri makinelerde kurularak, MISRA C/C++ ve JSF++ gibi standartların kurallarının ihlal edildiği durumlar bu yazılımlarla otomatik olarak tespit edilmiştir.

Dersi alan öğrenciler, özellikle büyük ölçekli yazılım sistemi geliştirmede, çok sayıda sürecin mevcut olduğunu görerek, bu tür organizasyonlardaki farklı roller konusunda bilgi sahibi olmuştur. Sadece yazılım kodu geliştirmenin veya kodu çalışır hale getirmenin tek başına yeterli olmadığını görerek, bakım yapılabilirlik ve diğer kalite faktörlerini de dikkate almanın önemini, standartların geliştirme faaliyetlerindeki kritik rolünü kavramışlardır.

Dersin tamamlanmasının ardından, Final sınavı öncesinde öğrencilere ders içeriğini değerlendirebilmesi için anket sunulmuştur. Anketin ders içeriğiyle ilgili bölümü Şekil 1'de verilmektedir.

Öğrencinin 1 ve 2 verdiği puanlar Zayıf, 3 verdiği puan Orta, 4 ve 5 verdiği puanlar İyi olarak değerlendirilip, ilgili grafikler çizilmiştir. Şekil 2 ve Şekil 3'de öğrencilerin 1. ve 2. anket sorusu için verdiği yanıtlara göre oluşturulmuş grafikler sunulmaktadır. Özellikle bu 2 grafik, IEEE 12207 ve emniyet-kritik kodlama standartlarının öğrenciler tarafından yeterli seviyede anlaşıldığını göstermektedir. Üçüncü soruya verilen yanıtlar incelendiğinde CMMI konusunun 1 haftada anlaşılmadığı gözlemlenmiş ve bir sonraki dönemde bu konunun 2 hafta olarak işlenmesi planlanmıştır. Dördüncü soruya verilen yanıtlara bakıldığında, yazılım metrikleri konusunun yeterince anlaşılmadığı gözlemlenmiş ve sonraki dönemlerde 2 hafta anlatılarak uygulama yapılması planlanmıştır. Beşinci soruya verilen yanıtlar incelendiğinde, çok sayıda emniyet-kritik kodlama standardının ders içerisinde işlenmesinin öğrenciler açısından iyi olmadığı, standartların birbirine benzemesi nedeniyle 2'şer hafta MISRA C ve JSF++ standartlarının anlatılmasına karar verilmiştir. Son soruya verilen yanıtlara bakıldığında, çok sayıda standardın teorik olarak işlenmesinin öğrenciler açısından anlaşılabilirliği güçleştirdiği, bu nedenle sadece DO-178B ve MIL-STD-498 standardının 1'er hafta anlatılması planlanmıştır.

Anket sonuçlarına göre oluşturulmuş yeni ders içeriği aşağıda sunulmaktadır.

1. Ders Tanıtımı
2. IEEE/EIA 12207 Standardı
3. MIL-STD-498 Standardı
4. CMMI Süreç Çerçevesi

5. CMMI Süreç Çerçevesi
6. Görev-Kritik Kodlama Standartları
7. Vize Sınavı
8. MISRA C/C++
9. MISRA C/C++
10. JSF++
11. JSF++
12. Yazılım Metrikleri ve Eşik Seviyeleri
13. Yazılım Metrikleri ve Eşik Seviyeleri
14. DO-178B Havacılık Yazılım Standardı

4. Sonuç ve Öneriler

Yaptığımız incelemelere göre, bu derinlikte standartların tartışıldığı ve kodlama standartlarının ele alındığı bir derse gerek Türkiye'de gerekse de yurt dışında rastlanmamıştır. Yazılım geliştirme bağlamında söz konusu olan standart sayısının fazlalığı ve henüz bir taksonominin geliştirilmemiş olması nedenleriyle, sistematik olarak bir ders içeriğini oluşturmak oldukça güçtür. Bu bildiri kapsamında tasarlanan ders ilk kez bu dönem (2013-1014 Güz dönemi), İstanbul Kültür Üniversitesi, Bilgisayar Mühendisliği 4. sınıf öğrencilerine seçmeli ders olarak verilmiştir. Seçmeli dersler, 3. sınıf öğrencileri tarafından da alınabildiği için az sayıda 3. sınıf öğrencisi de bu dersi almıştır ancak dersin hedefinin son sınıf öğrencileri olduğu değerlendirilmektedir.

Öğrencilerle anketler yapılarak, dersin anlaşılma durumu analiz edilmiş ve sorun tespit edilen konularla ilgili olarak ders içeriği haftalık bazda güncellenmiştir.

CS 2013 ve SE 2004 raporlarında, standartlar kapsamında tespit ettiğimiz bazı eksikliklerin ilgili raporları hazırlayan gruplara iletilmesi planlanmaktadır. Bu sayede, ilgili kurulların uygun görmesi durumunda, bazı içeriklerin güncellenebilmesi söz konusu olabilecek veya bu ders benzeri bir dersin eğitim programlarına eklenebilmesi söz konusu olacaktır.

Özellikle ülkemizde Yazılım Mühendisliği alanında eğitim veren bölümlerin, bu ders özelinde veya dersin sahip olduğu içerikler konusunda mevcut öğretim programlarını güncellemesi önerilebilir. Sadece yazılım kalitesi veya yazılım geliştirme süreçleri gibi dersler içerisinde, standartlar konusunu işlemek konunun yeterli etkiyi sağlamasını engellemektedir. Bu nedenle, doğrudan somut standartlarla desteklenecek şekilde ders içeriklerinin güncellenmesi ve mümkünse bu tür bir dersin tanımlanması faydalı olacaktır. Kalite konusu, yazılım geliştiren organizasyonlarda sadece kalite uzmanlarını etkileyen bir konu olmayıp tüm geliştiricileri, yöneticileri, konfigürasyon uzmanlarını, iş paketi liderlerini, proje yöneticilerini ve diğer rolleri etkilediği için öğretim programlarında bu konuların ayrıntılı olarak anlatılmasında fayda olacağı değerlendirilmektedir.

5. Kaynaklar

[1] Gray, L. "Guidebook to IEEE/EIA 12207: Standard for Information Technology, Software Life Cycle Processes", (2000).

[2] DeWeese, P. R. "MIL-STD-498 Software Development and Documentation", (1994).

[3] Karasu, E. ve Tütüncü, M. Ş., "IEEE/EIA 12207 Standardı ve Uygulanması"**, İstanbul Bilişim Kongresi**, (2007).

[4] Montgomery, S. L., "MISRA C: Guidelines for the Use of the C Language in Critical Systems", (2012).

[5] www.phaedsys.com/standards/JSFPP/JSF-AV-rules.pdf, Lockheed-Martin, Erişim: 1 Aralık 2013

[6] http://www.codingstandard.com/section/index/

[7] Seacord, R. C., "The CERT C Coding Standard", (2008).

[8] http://ai.stanford.edu/users/sahami/CS2013/, Erişim 1 Aralık 2013.

[9] http://sites.computer.org/ccse/SE2004Volume.pdf

[10] Chrissis, M. B., Konrad, M., Shrum, S., "CMMI for Development: Guidelines for Process Integration and Product Improvement", (2011).

[11] http://www.bilgiguvenligi.gov.tr/bt-guv.-standartlari/ortak-kriterler-iso-iec-15408-standardi-ve-ulkemizdeki-uygulamasi.html, Erişim 1 Aralık 2013.

COURSE EVALUATION FORM

Istanbul Kültür University – Computer Engineering Department

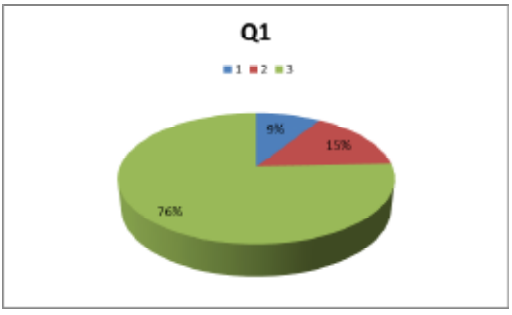
Course Name : Software Development with Standards	**Course Instructor : Assist. Prof. Dr. Çağatay ÇATAL**
Course Code : CSE0462	**Date : 10.12.2013**

Please indicate your opinion on each of the following items by circling the appropriate number.

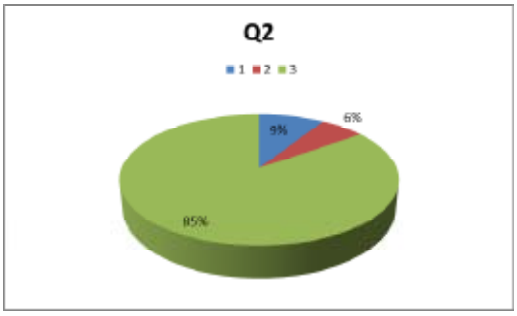
LEARNING OUTCOMES *1:Strongly disagree ....5: Strongly agree*

1.	I can describe IEEE 12207 standard.	1	2	3	4	5
2.	I can describe safety-critical coding standards.	1	2	3	4	5
3.	I can explain what the CMMI is.	1	2	3	4	5
4.	I can use some code measurement metrics for my programs.	1	2	3	4	5
5.	I can describe CERT C Secure Coding Standard, MISRA C standard, JSF++, High-Integrity C++ coding standard.	1	2	3	4	5
6.	I can explain DO-178B, DO-278, IEC 62304, EN 50128, ISO 26262, IEC 61508, IEC 60880 Standards	1	2	3	4	5

Şekil 1. Ders içeriği değerlendirme formu



Şekil 2. İlk soruya verilen yanıtların dağılımı [%76 İYİ, %15 ORTA, %9 ZAYIF]



Şekil 3. İkinci soruya verilen yanıtların dağılımı [%85 İYİ, %6 ORTA, %9 ZAYIF]