ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

YAZILIM PROJE RİSKLERİ İÇİN BİR KARAR DESTEK YÖNTEMİ VE YAZILIM PROTOTİPİ

Sahand VAHIDNIA

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ANKARA
2016

Her hakkı saklıdır

TEZ ONAYI

Sahand VAHIDNIA tarafından hazırlanan "**Yazılım Proje Riskleri İçin Bir Karar Destek Yöntemi ve Yazılım Prototipi**" adlı tez çalışması 21/12/2016 tarihinde aşağıdaki jüri tarafından oy birliği ile Ankara Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Prof. Dr. İman ASKERBEYLİ
Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı

Eş Danışman : Yrd. Doç. Dr. Ömer Özgür TANRIÖVER

Jüri Üyeleri :

Başkan : Prof. Dr. Onur DEMİRÖRS
Orta Doğu Teknik Üniversitesi Enformatik Enstitüsü

Üye : Prof. Dr. İman ASKERBEYLİ
Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı

Üye : Doç. Dr. Nizami GASİLOV
Başkent Üniversitesi Mühendislik Fakültesi

Üye : Yrd. Doç. Dr. Mehmet Serdar GÜZEL
Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı

Üye : Yrd. Doç. Dr. Ömer Özgür TANRIÖVER
Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı

Yukarıdaki sonucu onaylarım.
Prof. Dr. İbrahim DEMİR
Enstitü Müdür V.

ETIK

Ankara Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez içindeki bütün bilgilerin doğru ve tam olduğunu, bilgilerin üretilmesi aşamasında bilimsel etiğe uygun davrandığımı, yararlandığım bütün kaynakları atıf yaparak belirttiğimi beyan ederim.

21.12.2016

Shand VAHIDNIA

ÖZET

Yüksek Lisans Tezi

YAZILIM PROJE RİSKLERİ İÇİN BİR KARAR DESTEK YÖNTEMİ VE YAZILIM PROTOTİPİ

Sahand VAHIDNIA

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. İman ASKERBEYLİ
Eş danışman: Yrd. Doç. Dr. Ömer Özgür TANRIÖVER

Risk tanımlama ve değerlendirme, yazılım proje yönetimindeki kritik faaliyetler arasındadır. Bununla birlikte, risklerin ve belirsizliklerin tespit edilmesi ve değerlendirilmesi, özellikle küçük ve az gelişmiş yazılım kuruluşları ve takımları için zor bir süreçtir. Bu çalışmada, yazılım proje riskleriyle ilgili geniş kapsamlı bir liste oluşturulmuş ve öne sürülmüştür. Ardından, her bir risk faktörü için olasılık ve etki değerleri, yazılımcıların daha önceki zorlandıkları veya başarısız oldukları yazılım projeleri üzerine yapılan bir anket yardımıyla toplandı. Bu bilgilere dayanarak; nispeten az deneyime sahip, küçük yazılım kuruluşlar ve takımlarına yardımcı olmak amacıyla bir risk değerlendirme yöntemi ve prototip aracı geliştirilmiştir. Risk sıralama ve risk matrisi ile birlikte yöntem ve araç prototipi potansiyel başarısızlık türleri hesaplamaları ve toplam proje risk hesaplaması için bulanık mantığı kullanmaktadır. Yöntemin ve prototip aracın kullanılabilirliğini doğrulamak için küçük bir yazılım şirketlerinde proje risk değerlendirmesi üzerine bir vaka çalışması yapılmıştır.

Aralık 2016, 61 sayfa

Anahtar Kelimeler: Yazılım, Geliştirme, Risk, Bulanık Mantık, Risk Matris, Karar Destek, Araç

ABSTRACT

Master Thesis

A DECISION SUPPORT METHOD AND PROTOTYPE SOFTWARE FOR SOFTWARE PROJECT RISKS

Sahand VAHIDNIA

Ankara University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Prof. Dr. İman ASKERBEYLİ
Co-Supervisor: Assist. Prof. Dr. Ömer Özgür TANRIÖVER

Risk identification and assessment are amongst critical activities in software project management. However, identifying and assessing risks and uncertainties is a challenging process especially for emergent software organizations that lacks resources. In this study, a list with wider coverage of software project related risks is identified and put forward. Then by conducting a survey to software practitioners, probability and impact levels of each risk factor was collected based on a challenged or failed previous project. Based on this information, a risk assessment method and prototype tool is developed to assist small software organizations with relatively little experience. Along with a risk prioritization and risk matrix, the method utilizes fuzzy logic to provide the practitioners with predicted scores for potential failure types and aggregated risk score for the project. In order to validate the usability of the method and the tool, a case study is conducted for project risk assessment in a small software organization.

December 2016, 61 pages

Key Words: Software, Development, Risk, Fuzzy Logic, Risk matrix, Decision Support, Tool

TEŞEKKÜR

Bu çalışmanın gerçekleştirilmesinde değerli bilgilerini ve zamanlarını bana sunan tüm öğretim üyelerine, özellikle değerli danışman; Prof. Dr. İman ASKERBEYLİ (Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı) ve eş danısman hocam; Ömer Özgür TANRIÖVER'e (Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı) sonsuz teşekkürler ve saygılarımı sunarım.

Tez veri toplama aşamasında bana destek olan ve ankete katılan tüm katılımcılara sonsuz teşekkürlerimi sunarım

Sahand VAHIDNIA
Ankara, Aralık 2016

İÇİNDEKİLER

TEZ ONAYI
ETİK ..... **i**
ÖZET ..... **ii**
ABSTRACT ..... **iii**
TEŞEKKÜR ..... **iv**
KISALTMALAR DİZİNİ ..... **vii**
ŞEKİLLER DİZİNİ ..... **viii**
ÇİZELGELER DİZİNİ ..... **ix**
1. GİRİŞ **1**
2. LİTERATÜR TARAMASI **4**
2.1 Tanımlar ve Kavramlar **4**
2.1.1 Risk **4**
2.1.2 Yazlım riski **4**
2.1.3 Risk yönetimi **5**
2.2 İlgili Çalışmalar **5**
2.2.1 Yazılımın operasyonel risk değerlendirmesi için bulanık uzman sistem uygulamaları **6**
2.2.2 Nedensellik kısıtlamaları ile Bayes ağlarını kullanarak yazılım proje risk analizi **8**
2.2.3 Yazılım risk idaresi için risk tanıma, azaltma ve kaçınma modeli **9**
2.2.4 Bulanık risk matrisi **10**
2.2.5 Yazılım proje risk listelerinin sıralamaları **10**
2.2.6 Diğer çalışmalar **11**
3. MATERYAL VE YÖNTEM **14**
3.1 Risk Faktörlerinin Benzer Çalışmalardan Toplanması **14**
3.2 Risk faktörlerinin Ayarlanması ve Uygun Olmayan veya Tekrarlanan Risk Faktörlerin Elimine Edilmesi. **15**
3.3 Risk Anketinin Tasarlanması ve Düzenlenlenmesi **18**
3.4 Uzman Görüşü Verilerinin Toplanması **22**
3.5 Verilerin Ön İşlemeden Geçirilmesi ve ilk Sonuçların Elde Edilmesi **23**
3.5.1 Risk matrisi **23**
3.5.2 İstatistik **25**

3.5.3 Risk faktörlerinde boyut indirgeme 31
3.6 Tahmin ve Destek Aracının Geliştirilmesi 33
3.6.1 İlk aşama 34
3.6.2 İkinci aşama 35
4. ÖRNEK VAKA 40
4.1 Örneğin Bilgileri ve Ayarları 40
4.2 Tahmin ve Destek Aracının Geliştirilmesi 41
4.3 Bulgular ve Sonuçlar 42
5. SONUÇ 48
KAYNAKLAR 50
EKLER 54
EK 1 Risk Listesi 55
EK 2 Yazılımın ikinci aşamasından örnek bir resim 60
ÖZGEÇMİŞ 61

KISALTMALAR DİZİNİ

BN	Bayesian Networks
BNCC	Bayesian Networks with Causality Constraints
CTAS	Canadian Triage and Acuity System
DBMS	Data Base Management System
LoC	Line of Code
TFN	Triangular Fuzzy Number
TRM	Traditional Risk Matrix

ŞEKİLLER DİZİNİ

Şekil 1.1 Yıllara göre yazılım projesi başarı oranları 1
Şekil 1.2 2015 yılı yazılım projeleri başarı oranları 2
Şekil 2.1 SEI Risk Yönetim Modeli 5
Şekil 2.2 Crisp ve Fuzzy (joint) probability farkı 7
Şekil 3.1 Yöntem ve araç şeması 14
Şekil 3.2 Risk matris genel şeması 23
Şekil 3.3 Risk puan dağılımı (ortalamalar) 26
Şekil 3.4 Risk puan dağılımı (Tüm sonuçlarda) 27
Şekil 3.5 Başarısızlık çizelgesinde en çok görülen riskler 29
Şekil 3.6 Üçgensel bulanık numara 36
Şekil 3.7 Üyelik fonksiyonları 38
Şekil 4.1 Yazılımın risk ve risk oranı öneri sayfası 43
Şekil 4.2 Başarısızlık tipleri 45
Şekil 4.3 Overschedule Başarısızlık tipi 46
Şekil 4.4 Verilen değerler 47

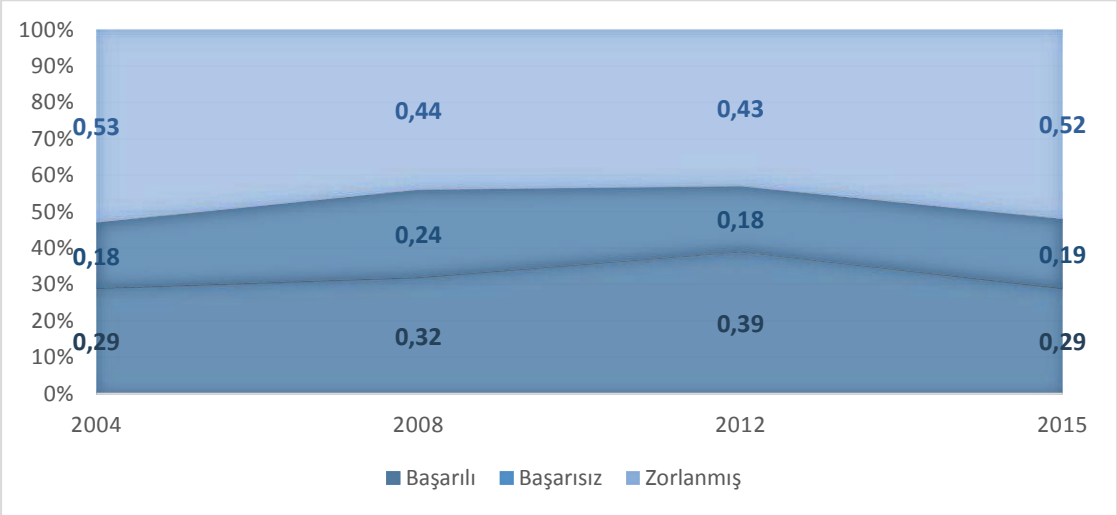
ÇİZELGELER DİZİNİ

Çizelge 2.1 BNCC sonuçları 8
Çizelge 2.2 Shahzad ve AL-Mudimigh'in risk listesi 9
Çizelge 3.1 Bazı değiştirilen riskler ve değiştirme gerekçeleri 16
Çizelge 3.2 Düzenlenen risklerden bazı örnekler 16
Çizelge 3.3 Önerilen kategoriler 17
Çizelge 3.4 Kategorilendirme örneği 18
Çizelge 3.5 Olasılık kavramları 19
Çizelge 3.6 Etki kavramları 20
Çizelge 3.7 Bölüm I soruları 20
Çizelge 3.8 Bölüm II soruları 21
Çizelge 3.9 Risk matris sonuçları (Sadece önemli riskler) 24
Çizelge 3.10 Risk puan dağılımı 25
Çizelge 3.11 Başarısızlık tiplerinde tehlikeli olan riskler 28
Çizelge 3.12 Önemli riskler karşılaştırması 30
Çizelge 3.13 Yüksek korelasyonlu risk çiftleri 32
Çizelge 4.1 Örnek vaka risk matrisi sonuçları (sadece çok önemli -kırmızı bölge- riskleri) 44

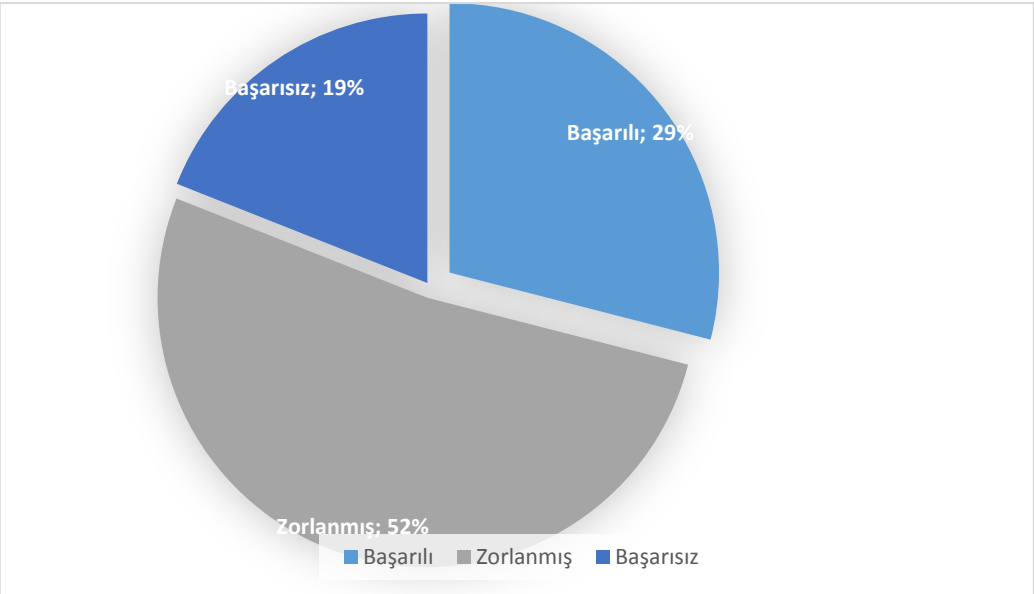
1. GİRİŞ

Yazılım projelerinde risk yönetimi büyük bir çaba gerektiren önemli bir aşamadır. Risk yönetimi bir yazılımın her aşamasında yapılmalıdır ve sadece özel bir aşamaya ait değildir. Maalesef risk yönetimi birçok proje yönetimi tarafından ve geliştirme takımlarında ciddiye alınmayan bir konu haline gelmiştir. Ayrıca, bazıları tarafından risk yönetimi, sadece olası risklerle ilgili küçük bir konuşmadan ibarettir. Risk yönetimi söz konusu takımlarda ve şirketlerde, özellikle küçük takımlarda ve şirketlerde, lüks, pahalı, çok çaba gerektiren ve işe yaramaz bir çalışma olarak görülmektedir. Hâlbuki risk yönetimi, yazılım yaşam döngüsünün her adımında özenle yapılmalıdır; yoksa projenin başarısız (failed) olma ihtimali yüksektir.

Araştırmalara göre (Hu vd. 2013, Anonymous 2013, Anonymous 2015), 2014 yılında, yazılım piyasasının 330 milyar dolarlık bir değere sahip olduğu tahmin ediliyor. Bu değer 2009 yılında 242 milyar dolardı. Rakamlarda göründüğü gibi 5 yılda bu piyasada %36'lık değer artış olmuştur. Gelecekteki yıllarda da değer artışı bekleniyor. Diğer taraftan CHAOS raporuna göre dünyada, Avrupa ve ABD başta olarak, yazılım proje başarı oranı sadece %29'dur. Yani projelerin %71'i başarısız veya zorlanmış (challenged) olarak değerlendirilmiştir; ki bu iyi bir sonuç değildir.



Şekil 1.1 Yıllara göre yazılım projesi başarı oranları



Şekil 1.2 2015 yılı yazılım projeleri başarı oranları

Şekil 1.1'de gösterildiği gibi, geçen on yılda yazılım projelerin başarı oranlarında önemli bir farklılık yaşanmamıştır.

CHAOS raporunda geçen rakamlar umut vaatedici değildir ve yazılım mühendisliğinin doğru yolda olmadığını göstermektedir. Bu rakamlara göre, yazılım projelerde, hep ortalama %30'dan daha düşük bir başarı oranı beklenir ve sorunların çözülmemiş olduğu aşikardır. Araştırmalara göre (Alberts ve Dorofee 2010) projelerin başarısız olmasının önemli bir nedeni, yanlış risk yönetimidir. Bu sebepten dolayı, konuyla ilgili 20 yılı aşkındır sayısız çalışmalar yapıldı ve yapılıyor.

Bu çalışmada, diğer çalışmalardan yola çıkarak, yazılım risk yönetimine bir yol bulmaya çalışıldı. Çok sayıda yöntem incelendi ve bazı yazılım geliştirme takımlarıyla, özellikle küçük takımlarla temasa geçildi. Ama sonuç ümit vaatedici değildi. Çünkü söz konusu takımlarda risk yönetimi konusunda organize bir çalışma yapılmıyor ve diğer taraftan var olan yöntemler ve araçlar zaman ve maddi sebeplerden dolayı kullanılmıyor. Bu yüzden bu çalışmada gerçek hayat yazılım proje risklerini

inceleyerek, yazılım geliştiren takımlar için, gerçektende kullanılabilir, pratik ve düşük maliyetli bir risk belirleme, izleme ve sonuç öngörü sistemi geliştirilmeye çalışıldı.

Çalışmanın adımları bir sonraki adımlarda kısaca özetlenmiştir.

1. Risk faktörlerinin literatürdeki çalışmalardan belirlenmesi.
2. Risk faktörlerinin ayarlanması ve uygunsuz veya tekrarlanan risk faktörlerin elimine edilmesi.
3. Anketin tasarlanması.
4. Verilerin toplanması.
5. Verilerin ön işlemeden geçirilmesi ve ilk sonuçların elde edinmesi
6. Tahmin ve destek aracının geliştirilmesi.
7. Yöntem ve aracın örnek vaka değerlendirilmesi ve doğrulanması.

2. LİTERATÜR TARAMASI

2.1 Tanımlar ve Kavramlar

Doğru bir risk yönetimi için, risk yönetimi ve risk kavramlarını anlamak gerek. Literatürde bu kavramlar için önemli tanımlar sunulmuştur ve bu kavramlar devamda açıklanmıştır.

2.1.1 Risk

Risk için benzer tanımlar mevuttur, ama risk ile ilgili (bu çalışmaya göre) en doğru tespit Pressman kitabında yapılmıştır ki devamda Türkçe çevirisi mevcuttur. " Risk potansiyel bir sorundur ki meydana gelip veya gelmeyebilir de. Ama sonuç ne olursa olsun, en doğru karar riskleri tanımlamak, olasılıklarını belirlemek, etkiyi tahmin etmek ve riskin gerçekleşmesine karşı bir plan oluşturmaktır." (Pressman 2010).Başka tanımlarda, bir tehdidin her hangi bir zayıflığı yakalama olasılığına risk denir.

2.1.2 Yazlım riski

Yazılım riski, yazılım projelerinde olumsuz etki yaratan, belirsiz bir olay veya koşuldur. (Xiaosong vd. 2009)Her risk genelde 2 bölümden oluşur:

- Olasılık (Probability)
- Etki (Impact)

Ama yazılım risklerini daha iyi incelemek için, Gelecekteki kök nedeni (Henüz gerçekleşmemiş) ayrıca dikkate alınmalıdır.Tanımlarda görüldüğü gibi, risk tanımı belirsizlikle gelir. Risklerdeki 2 önemli faktör, olasılık ve etkidir. Bu çalışmanın devamında söz konusu faktörler ve ilgili konulara sıkça rastlanacaktır.

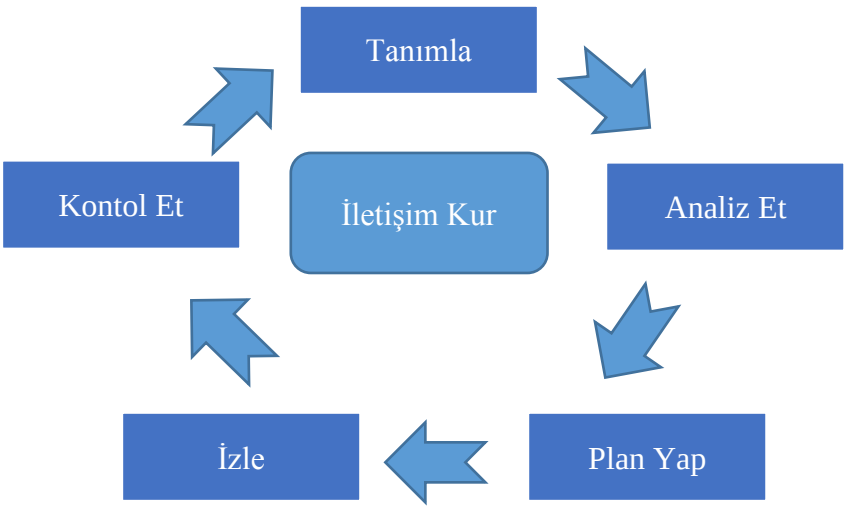
2.1.3 Risk yönetimi

Risk analizi ve yönetimi, yazılım geliştirme takımların belirsizlikleri anlamalarına ve

yönetimlerine yardım eden bir eylemdir (Pressman 2010). Yazılım riskleri 4 adımda ele alınır:

- Riski belirlemek
- Risk Projeksiyonu (tahmin etmek)
- Riski filtrelemek
- Riski azaltmak, izlemek ve yönetmek

Software Engineering Institute´te yayınlanan rapora göre, Risk Yönetim Modeli bu şekilde açıklanmıştır. (şekil 2.1) (Anonymous 2013)



Şekil 2.1 SEI Risk yönetim modeli

2.2 İlgili Çalışmalar

Risk konusuyla ilgili, şimdiye kadar yüzlerce çalışma yapılmıştır. Bu araştırmada söz

konusu çakışmaların önemli bir kısmı incelenmiş ve önerilen yöntemler ve sonuçlar gözden geçirilmiş. Bu konuda yapılan çalışmalar genelde 3 önemli konuyu kapsamaktadır:

- Genel risk ve risk faktörlerini bulmak.
- Riskler arasındaki ilişkiler bulmak için yöntemler.
- Risk faktörü birleştirme ve toplama yöntemleri.

Devamda en önemli çalışmalardan bazılarını özet olarak açıklıyoruz.

2.2.1 Yazılımın operasyonel risk değerlendirmesi için bulanık uzman sistem uygulamaları

"Application of fuzzy expert systems in assessing operational risk of software" (Xu vd. 2003) adlı çalışma, "NASA IV&V Facility" tarafından gerçekleşen ve 2003'te yayınlanan, yazılım operasyon riskleriyle ilgili bir çalışmadır. Bu çalışmanın temel yöntemi, makalenin isminde göründüğü gibi, bulanık mantıktır.

Bilindiği gibi, bir projeyi yapmadan ve yapım aşamasına gelmeden, gelecekteki risklerle ilgili yeterli bilgi edinmek mümkün değildir, o yüzden bir çok belirsizlikle karşı karşıya gelmek söz konusudur. Normal risk matris yöntemini kullanmak için, kesin bilgilere ihtiyaç olduğu için, risk matrisi yöntemini kullanmak mümkün olamaz.

Bu sebeplerden dolayı bu makalede operasyon risklerini belirlemek için, risk matrisi kullanmaktansa, "Bulanık Olasılık" (Fuzzy probability) kullanarak bir "Bulanık Uzam Sistemi" (Fuzzy expert system) önerilmiştir. Bu çalışmadaki kullanılan kavramlar aşağıdaki gelmiştir:

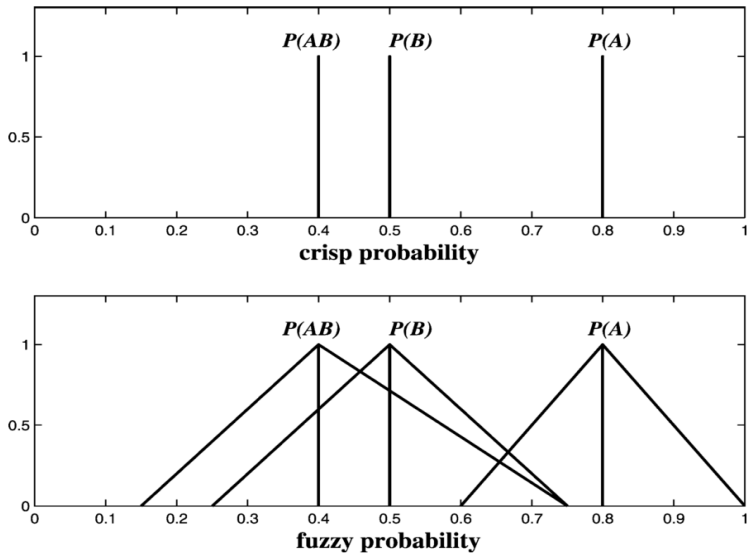
- Bulanık Olasılık: Mevcut veriler "crisp" olmadığı için, bulanık olasılık kullanılacaktır.
- Birleşik Bulanık Olasılık (Joint fuzzy probability): Eğer 2 risk faktörü arasında bir

ilişki varsa, yani 2 faktör aynı zamanda tetiklenince risk büyüme oranı farklı olursa, "Birleşik Olasılık" (Joint probability) kullanılmalıdır. Bu ilişkiler uzmanlara sorulabilir veya "Bayes Theorem" yardımıyla çözulur. Bu yöntemin ismi "Fuzzy Extended Bayes Theorem".

Joint Probability: $P(A.B) = P(AB) \quad => \quad P(AB) = P(A|B)(.)P(B)$ (2.1)

Örnek olarak bir "crisp" ve bir "fuzzy" olasılık modeli şekil 2.1 (Xu vd. 2003) de gösterilmektedir.

Devamda bir nedensellik modeli (Causal model) yapılmıştır ve bulanık kurallar (Fuzzy rules), uzmanlar ve "Fuzzy Extended Bayes Theorem" yardımıyla çıkartılmıştır. Bu çalışmada bahsedilen "Bulanık Uzman Sistem" uygulanmış ve riskler için (tam da gerçek riskler için kalibre olmayan) modellemeler yapılmıştır.



Şekil 2.2 Crisp ve Fuzzy (joint) probability farkı

2.2.2 Nedensellik kısıtlamaları ile Bayes ağlarını kullanarak yazılım proje risk analizi

"Software project risk analysis using Bayesian networks with causality constraints" (Hu vd. 2013) 2013 yılında yapılan ve "Decision Support Systems"de yayınlanan bir makaledir ki daha iyi bir risk kontrolü sağlamak için, risk planlama aşamasının nedenselliğe göre yapılması öneriliyor. Önerilen yöntem "BNs with causality constraints" (BNCC)'dir.

Daha iyi bir risk kontrolüne ulaşmak için, uzman görüşü, V-structure keşif algoritması ve otomatik ağ öğrenimi yöntemlerini kullanarak bir Bayesian Network (BN) üretilmiştir.

Bu araştırmada BNCC gerçek veri üzerine yapılmıştır. Ana fikir böyledir, ilk V-yapılar bulunduktan sonra birbirine bağlanıp bir nedensellik ağı oluşturuyorlar. En sonda uzmanlar tarafından ve nedensellik ağlarıyla tanımlanan "yapı kısıtlamaları"(Sructure constraints)'na dayalı otomatik BN öğrenim uygulanıyor.

Bu yöntemi uygulamak için 500 kişiden veri toplamaya çalışılmış ve bu verilerden 302 örnek geçerli olarak sayılmıştır. Bu yöntem diğer mevcut veri işleme algoritmalarıyla karşılaştırılmıştır (Çizelge 2.1) (Hu vd. 2013).

Çizelge 2.1 BNCC sonuçları

Algoritma	**Ürün performansı**	**İşlem performansı**	**Parametreler**
Önerilem yöntem	75,15%	74,42%	
Weka_Logistic	68,87%	70,19%	default
Weka_J48	70,86%	72,19%	default
Weka_naiveBayes	72,85%	71,17%	default
Weka_BayesNetwork	73,18%	66,23%	Simple estimator –A0.5 K2 –P 1000 –S BAYES
Weka_BayesNetwork	74,17%	68,54%	Simple estimator –A0.5 TAN –S BAYES

2.2.3 Yazılım risk idaresi için risk tanımlama, azaltma ve kaçınma modeli

"Risk Identification, Mitigation and avoidance Model for Handling Software Risk" (Shahzad ve Al-Mudimigh 2010), sadece 14 risk üzerinde yapılan bir araştırmadır. Bu 12 risk, araştırmaya göre en önemli ve en yaygın risklerler ve amaç bu riskleri kontrol altına almaktır.

Bu çalışmada "Risk Identification, Management and Avoidance Model" (RIMAM) adlı bir model tanıtılmıştır. RIMAM kısaca riskleri tanımlamak, yönetmek ve risklerden kaçınmak için yapılan hareketleri göstermektedir. Bu çalışmada riskleri bulmak için yapılan matematiksel bir yöntemdense, risklerle baş etmek için yöntemler sunulmuştur. Bu yöntemler bir karar destek sistemin çözüm üreten aşamasında yardımcı olabilir, ama bu yöntem sadece kısıtlı risk sayısında geçerlidir (Çizelge 2.2) (Shahzad ve Al-Mudimigh 2010).

Çizelge 2.2 Shahzad ve AL-Mudimigh'in risk listesi

Immature requirements -olgunlaşmamış gereksinimler	Staff turnover -personel dönüşü
Divergent estimation of cost & time -maliyetin ve zamanın farklı tahminleri	Staff ınexperience -personel deneyimsizliği
Massive user stress -muazzam kullanıcı stresi	Backup issues -yedekleme sorunları
Less reusability -daha az yeniden kullanılabilirlik	Natural disasters - doğal afetler
Project delivery milestones -proje teslim kilometre taşları	Excessive error detection -aşırı hata tespiti
Funding un-certainty - finansman belirsizlikleri	Developer's faithfulness -geliştiricinin sadakati
Over-optimistic technology perceives -aşırı iyimser teknoloji algısı	Preservation of intellectuals -entelektüellerin korunması

2.2.4 Bulanık risk matrisi

Sayısal yöntemlerden kaynaklanan ve risk faktörlerine sayısal değerlerinin belirlenmesinin zorlukları yüzünden, Markowski ve Mannan "Fuzzy risk matrix" (Markowski ve Mannan 2008) çalışmasında risk matrislerinde bulanık mantık kullanmayı önerdiler. Sözü geçen çalışmada, sıradan risk matrisi kullanmak yerine farklı ve 3 boyutlu bir risk matrisi kullanılmıştır ki risklerin 3 boyutunun (Risk değeri, Risk olasılığı, Risk etkisi) ilişkilerini tek bir şekilde göstermek mümkündür.

Markowski´nin çalışmasından yola çıkarak, bu çalışmada bulanık mantık kullanmaya karar verildi. Tabi ki bu çakışmada önerilen yöntem sadece risk matrisi değildir ve tahmin ve karar matris ve yöntemlerini içeriyor. Bu tezde önerilen yöntemde risk matrisi *crisp* olarak yapılmıştır, ama karar destek yönteminde bulanık mantık kullanıldı. Bunu başarmak için tüm risk faktörleri değerleri için bulanık rakamlar atandı. Detaylı bilgi gelecek bölümlerde açıklanmıştır.

2.2.5 Yazılım proje risk listelerinin sıralamaları

Arnuphaptrairong "Top Ten Lists of Software Project Risks: Evidence from the Literature Survey" (Arnuphaptrairong 2011) çalışmasında 12 risk tabanı üzerinden ve yapılan 8 anketten gelen bilgilerle, yazılım risklerinin en önemli ilk 10 listelerini 8 tabloda oluşturmuştur. Arnuphaptrairong´un çalışması 2011'de yapılan bir literatür anketidir ve o yıla kadar yapılan önemli risk listelerini göz önünde bulundurarak ve diğer araştırmacıların yaptığı anketlerden yola çıkarak bu listeleri oluşturmuş ve incelemiştir.

Elde edilen listelerin başlıkları bu şekilde:

1. Bohem'in en önemli ilk 10 listesi
2. Schmidt vb. 'ın en önemli ilk 10 listesi (USA)

3. Schmidt vb. ’ın en önemli ilk 10 listesi (HONG KONG)
4. Schmidt vb. ’ın en önemli ilk 10 listesi (FINLAND)
5. Addison ve Vallabh’ın en önemli ilk 10 listesi
6. Addison’un en önemli ilk 10 listesi
7. Han ve Huang’ın en önemli ilk 10 listesi
8. Pare vb. ’ın en önemli ilk 10 listesi

Ayrıca Arnuphaptrairong´un çalışmasınada riskler ve risk listeleri tip olarak incelenmişler ve davranışlar analiz edilmiştir. Bu çalışma, Arnuphaptrairong´un çalışmasıyla çok benzer yönlere sahipler ve aslında bu çalışmanın birinci kısmı Arnuphaptrairong´un çalışmasına bir güncelleme sayılır. Aynen Arnuphaptrairong´un çalışması gibi bu çalışmada benzer yayınlardan ve kitaplardan risk faktörleri alındı ve bir anket yapılarak anketin sonuçları Arnuphaptrairong´un topladığı sonuçlarla karşılaştırıldı.

2.2.6 Diğer çalışmalar

Çalışmaların büyük bir kısmı risk değerlerini toplamak için iyi yöntemler sunmak çabasındalar. Mesela, Büyüközkan’ın çalışması (Büyüközkan ve Ruan 2010) Choquet Integral a dayalı bir toplama yöntemi önermektedir. Bu çalışmada çok nitelikli toplama yöntemi (Multi attribute aggregation method), karar vermek için önerildi. Ayrıca bu çalışmada küçük risk seti ve kategoriler ilavetan önerilmiş ve kullanılmıştır.

Diğer taraftan risk faktörlerini toplamak, tanıtmak ve inceleyen çalışmalar da mevcuttur. Mesela (Xiaosong vd. 2009, Hizazi vd. 2014, Verner vd. 2014) çalışmalarda risk faktörleri toplanmış, açıklanmış ve anlatılmıştır. Ayrıca Verner’in çalışmasında (Verner vd. 2014)77 risk azaltma tavsiyesi bulunmaktadır.

Bazı diğer çalışmalarda, mesela 2013'te yapılan “RisCal - A Risk Estimation Tool for Software Engineering Purposes” (Haisjackl vd. 2013) adlı bir çalışmada risk yönetimi

için bir araç önerilmştir. Bu çalışmada klasik risk modeli (yani "Olasılık*Etki") daha farklı bir model kullanılmıştır:

$$R(a) = (P(a).E(a)) \circ I(a) \quad (2.2)$$

E(a) bu model de evrim faktörü (Evolution factor) ve bu faktör 0 ve 1 arasında, bir sürü kriter ile, mesela değişim oranıyla değişen bir değerdir. Özet olarak bu yöntem 3 aşamada yapılmaktadır:

- Risk Tanımlama
- Risk Analizi
- Risk Önceliklendirme

3. adım karar vermenin temelinni oluşturuyor. Başka bir çalışmada, "esrcTool : A Tool to Estimate the Software Risk and Cost" ta (Sadiq vd. 2010) başka bir araç önerilmiş ki mimari olarak çalışmada açıklanmıştır ve bu tez çalışmasında tam detaylı açıklamaya ihtiyaç duyulmadı. Risk yönetimi için bazı diğer ticari uygulamalar mevcuttur. Mesela "STREAM" adlı bir yazılımın "Entery Edition"ı 295$ ve "Ultimate edition" 2.495$ fiyatındadır.

Tüm çalışmalarda risk sorununu çözmek için büyük çaba sarf edilmiş ve başarılı sonuçlar elde edilmiştir. Ama bizim çözmeye çalıştığımız sorunlar çözülmemiş. Bu sorunlar bu şekildeler:

- Gerçek hayat proje risklerini bulmak.
- Çok sayıda riski yönetmek ve risk sayısını kısıtlamamak.
- Risk yönetim maliyetini düşürmek.
- Risk yönetim zamanını azaltmak.

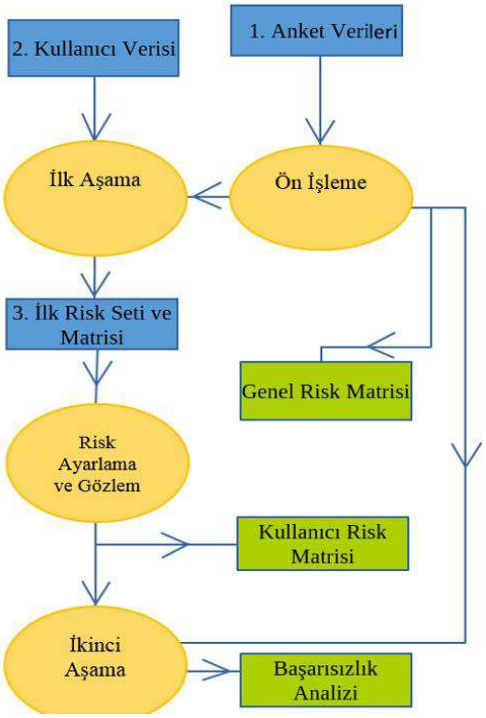
Göründüğü gibi, çeşitli çalışmalarda, risk bulma ve risk yönetimi için yöntemler önerilmiştir. Önerilen yöntemler arasında risk listeleri yapılmıştır ve çeşitli risk toplama metodları önerilmiştir. Öncelikle bu çalışmada hedeflenen sonuç, yazılım risk analizi için bir testek aracıdır. Literatür taramasında bazı yazılımlar bu konuda önerilmiştir. Ama var olan çalışmalarda ve yazılımlarda aşağadaki eksiklikler görünmektedir:

1. Risk listeleri öneren çalışmalar, risk önerisinde bulunmuyorlar.
2. Risk tahmininde bulunan çalışmalar, kısıtlı sayıda riskin tahmininde bulunuyorlar.
3. Yazılımlar başarısızlık analizinde bulunmuorlar.

Kısaca, bu çalışmada tasarlanan yazlılım prototipi, öncelikle bir risk listesi önermelidir ki yazılımcılar risk bulma aşamasında zaman kaybetmesinler. Daha sonra riskler için, proje ayarlarına göre mühtemel bir değerlendirme yapmaktır. Bu sayede yazılımcılar riskler hakkında fikir sahibi olurlar. Üçüncü aşama projenin toplam risk gözlemi için bir risk matrisi ve toplam sayısal risk üretimidir. Ek olarak yazılım kapsamında bu risklerinhangi başarısızlık tipine yol açacağı tahmin veya espit edilmelidir. Tüm bunları yaparak, yazılımcılar riskler konusunda ve risklerin etkileri hakkında fikir sahibi olurlar ve risk izleme çok daha kolay olabilir.

3. MATERYAL VE YÖNTEM

Bu çalışmada kullanımı kolay ve akıllı bir risk yönetim uygulaması önerildi ve yapıldı. Bu amaca giriş bölümünde bahsedilen 7 adımda ulaşıldı. Bu adımlarla ilgili bilgiler bir sonraki bölümlerde detaylı olarak açıklanmıştır. Devamda açıklanan yöntem ve amaç bir proaktif yöntem ve araçtır ve sadece proje yapılmadan risk yönetiminde yardımcı olur. Şekil 3.1 bu yazılımın şemasını özet olarak anlatmaktadır.



Şekil 3.1 Yöntem ve araç şeması

3.1 Risk Faktörlerinin Benzer Çalışmalardan Toplanması

Bu çalışmaya başlamadan, konuyla ilgili 40´ten fazla çalışma ve kitap gözden geçirildi. Ama hiç bir çalışmada kapsamlı bir "risk-set"ine rastlanmadı. Gözlemlenen çalışmalarda genellikle yazılım geliştirme aşamasındaki belirli ve sınırlı bir alanda riskler incelenmişler. Bu nedenle yapılan ilk şey, risk olarak hesaplana bilecek

faktörleri, ilgili çalışmalardan bulmaktır(Xiaosong vd. 2009, Zardari 2009, Appari ve Benaroch 2010, Shahzad vd. 2009, Shahzad ve Al-Mudimigh 2010, Song vd. 2010, Wu vd. 2010). Söz konusu çalışmaların bir kısmında Boehm, COCOMO ve benzeri ünlü risk modelleri referans alınarak risk kümeleri yapılmış veya bu modelden direk alınmıştır. Diğer çalışmalar genelde kendi risk kümelerini tanımlamış ve kullanmışlar.

3.2 Risk faktörlerinin Ayarlanması ve Uygun Olmayan veya Tekrarlanan Risk Faktörlerin Elimine Edilmesi

Risk toplamada ikinci adım, risk kümesini küçültmektir. Bu işin nedeni ve çözümleri aşağıda yer almıştır:

- Bu aşamada 200 e aşkın risk faktörü vardır. Bu sayıda risk yapılmak üzre olan çalışma için çok fazladır ve ayrıca risklerin bir kısmı kavramsal olarak ve bazıları tamamen tekrar ediliyor.
- Benzer riskleri bulmak için, ilk adımda anahtar kelime bazında arama kullanılmıştır. Bu sayede bir anahtar kelimeye dayalı risk faktör sıralaması oluşturuldu ve çok benzer olan risklerin büyük bir kısmı, sırada yan yana yer aldılar. Benzer riskler bu listede elimine edildi. Eliminasyon için, sıralamada yan yana yer alan riskler tek tek incelendi ve en kapsamlı ve anlamlı risk faktörü temel alındı ve diğerleri elimine edildi.
- Bazı riskler bu araştırmada risk faktörü değil ve aslında proje başarısızlık tipi (Failure mode) veya başarı faktörüdür. Başarısızlık kavramı daha sonra, bölüm 3.3´te açıklanmıştır.
- Daha sonra bir kez daha kavramsal olarak riskler incelendi ve benzer riskler arasından en kapsamlı riskler, risk listesinde yer aldı ve diğerleri elimine edildi. Bazı diğer çalışmalarda bunun tam tersi yapıldı. Bu tamamen duruma bağlı ve kavramsal bir iş olarak yapıldı. Asıl amaç tekrarları azaltmak ve risk kavramlarını daha anlaşılır hale getirmektir. Çizelge 3.1 elimine edilen bazı riskleri gerekçelerle beraber göstermektedir.

- Riskler araştırma boyunca İngilizce kullanıldı ki materyal olarak risklerde her hangi bir değişiklik olmasın ve doğruluk oranı düşmesin.

Çizelge 3.1 Bazı değiştirilen riskler ve değiştirme gerekçeleri

Poor Re-plan (Kötü Yeniden planlama)	**Development Over-Schedule (Geliştirme Zaman Aşımı)**	**Team (Takım)**	**Disaster Recovery (Felaket kurtarma)**
Proje planlama ve yönetimi hakkında bir sürü risk faktörü ilk risk setinde mevcut ve bu örtüşen riskleri azaltma ve düzenleme aşamasında bu risk elimine edildi.	Bu risk bu çalışmadaki risklerle kıyasen çok daha genel bir risk ve bazı çalışmalarda bu bir başarısızlık tipi olarak değerlendiriliyor. O yüzden bu çalışmada bu risk elimine edilip ve başarısızlık tipi olarak değerlendirildi.	Takım kavramı yalnız olunca bir risk ifade etmiyor ve anlamsız. Eğer Takımla ilgili riskleri kastediyorsa bu risk, fazlasıyla geneldir ve takımla ilgili diğer riskler önerilen referans model de mevcutlar.	Kötu bir "disaster recovery" aslında önemlidir, ama bu çalışmada "Disaster / Catastrophe" riskin Kullanılması tercih edildi. Çünkü yeni risk daha kapsamlı ve fazla genel değil ve ankette yanlış anlaşılmaya yol açmaz.

Bu aşamada elde toplam 128 risk vardır. Ama bu risklerin çoğunun olumlu veya olumsuz oldukları belirsiz ve risk ölçümü mümkün değildir. Bu sebepten dolayı tüm 128 risk olumsuz cümleler şekline getiriliyor. (çizelge 3.2)

3.2 Düzenlenen risklerden bazı örnekler

Önce	**Sonra**
database size (veri taban boyutu)	Large Database Size (Büyük Veri Taban)
Management support and involvement (Yönetimin desteği ve iştirakı)	Lack of Management Support and Involvement (Yönetimin Dsteğinin ve İştirakının yetersizliği)
Design complexity (Tasarım karmaşıklığı)	High Design Complexity (Tasarımın Yüksek Karmaşıklığı)
Required development schedule (talep edilen geliştirme programı)	Bad Development Schedule (Kötü geliştirme programı)

Bu sayede tüm risk faktörleri negatif ve olumsuz olarak değerlenir ve risk puanlaması daha sağlıklı yapılır. Ayrıca, bu negatif cümleler ankette her hangi bir kafa karışıklığına yol açılmasına engel olur. Çalışmalardan alınan ve düzenlenen risk faktörleri EK 1'de mevcuttur.

Riskleri ayarlamak ve hazırladıktan sonra riskleri daha iyi anlamak ve kök sebeplerini bulmak ve anlamak için çok işlevsel olabilir. Literatürde kategorilendirme için bazı çalışmalar (Arnuphaptrairong 2011, Wang ve Fu 2011)yapılmıştır, ama genellikle toplanan risk modelini tam kapsamakta yetersizler. O yüzden yeni bir kategorilendirme bu çalışma kapsamında sunulmuştur. Yeni kategoriler daha kapsamlı ve benzer çalışmalardan esinlenerek eklenmişler.

Çizelge 3.3 Önerilen kategoriler

Risk Kategorileri		
1	Planning	Planlama
2	Requirement	Gereksinim
3	Design	Tasarım
4	Development	Geliştirme
5	Testing	Deneme
6	Implementation	Uygulama
7	Operational Risks	Operasyonel Riskler
8	Organizational Risks	Kurumsal Riskler

Önerilen kategori kümesi, sonraki adımlarda ve özellikle riskler arasında ve diğer veriler arasındaki ilişkileri bulmakta faydalı olabilirler. Bu kategori kümesi yazılım mühendisliği geliştirme döngüsü (Software engineering life-cycle) artı "Operational Risks" (operasyon riskleri) ve "Organizational Risks" (kurumsal riskler) ten oluşan bir kategori kümesidir ki tüm riskleri kapsamaktadır. Operasyon riskleri diğer çalışmalarda, mesela (Xu vd. 2003), tartışılan ve üzerine gidilen bir risk kategorisi olduğu için bu kategori kümesine eklenmiştir. Kurumsal riskler bu kümeye eklenen diğer kategoridir ki

aslında geliştirme riskleri alt kümesinde yer alabilir. Ama daha iyi ve doğru tespit için, bu kategoriyi ayırmak tercih edildi. Ayrıca kurumsal riskler geliştirmenin farklı aşamalarında tetiklenebilir ve karışıklığa sebep verebilir. Bu yüzden bu kategoriyi ayırmak mantıklı olabilir.

Kategorilendirilmiş risklerden bazı örnekler çizelge 3.4'te gösterilmiştir. Bu çalışma kapsamında kategorilendirme sadece önerildi ve açıklandı, ama risk yönetim yöntemimde kullanılmadı ve gelecekteki çalışmalarda kullanılması bekleniyor. Yeni önerilen kategoriler daha kapsamlı ve kolayca anlaşılırdır ve genelde kabul edilen bileşenlerden oluşmuştur.

Çizelge 3.4 Kategorilendirme örneği

Risk		**Category**
Required software reliability	Gerekli yazılım güvenilirliği	7
Big database size	Büyük veritabanı boyutu	6 , 7
Project resource conflict	Proje kaynak çatışması	2
High political influences	Yüksek siyasi etkiler	1
Low usage of software support tools	Yazılım destek araçlarının düşük kullanımı	4 , 5

3.3 Risk Anketinin Tasarlanması ve Düzenlenlenmesi

Bu çalışmada bir sonraki adım risk anketini düzenlemektir. Bu aşamada 128 risk faktörü kaydedilmiştir. Anket için gerekli olan diğer bilgiler bu şekildedir:

- Riskler için değerlendirme metrikleri
- Uzmanlardan alınacak proje bilgileri
- Projenin başarı kriterleri

Diğer çalışmalara bakarak ve özellikle risk matrisi kullanılan ve benzeri konular içeren çalışmalarda, risk kriterleri için üçlü değerleri kullanmak daha çok tavsiye edilmiştir. Bu çalışmada kullanılan değerler 1 (düşük) 2 (orta) ve 3 (yüksek) tir.

Riskler 3 temel parçadan oluşuyorlar. Bu parçalar "olasılık", "etki" ve "risk değeri" olarak bilinir (Haisjackl vd. 2013). Risk değeri; etki ve olasılığın çarpımından elde edilen bir değerdir (Olid ve Mannan 2008). Olasılık ve etki değerleri için üçlü değerlendirmede çizelge 3.5 ve 3.6 (Zardari 2009) gibi açıklanabilirlerler. Bu üçlü değerlendirme kavramları, yapılan çalışmaya daha uyumlu olsun diye çok az değiştirildi ve orijinalinden farklıdır.

$$\text{Risk değeri} = \text{Etki} * \text{Olasılık} \tag{3.1}$$

Çizelge 3.5 Olasılık kavramları

Olasılık değerleri	
Yüksek / Muhtemel (High / Very Likely)	Yüksek meydana gelmek ihtimali ve sorunu yaşamak (> 70%)
Orta /Olası (Medium / Probable)	Bu risk bazen sorun olabilir (30% < P < 70%)
düşük (Low / Improbable)	Düşük sorun yaşama olasılığı (< 30%)

Görüşler bu ankette web ortamında toplanmıştır ve halen devam ediyor. Anket 3 bölümden oluşuyor.

Proje bilgileri (Bölüm I) - 12 soru

Proje başarı oranlarl (Bölüm II) - 10 soru

Proje risk profili (Bölüm III) - 1 soru - 128 risk - her risk için olasılık ve etki

Bu çalışmada risklerle ilişkili olasılık ve etki değerleri bulmayı planladığımız için, uzmanlardan bir zorlanmış veya başarısız projeyi düşünerek proje ile ilgili temel bilgiler alınmaktadır. İstenen bilgiler çizelgede 3.7'de gösterilmiştir.

Çizelge 3.6 Etki kavramları

Etki değerleri	
Yüksek **High / Catastrophic**	Sistem / proje kaybı; Sistem / proje işlemlerinde geri alınmaz çöküşler; Sisteme büyük hasar; Zamanlama kaymasından kaynaklanan piyasaya sürmek tarihini kaçırmak; Maliyetin 50% den daha fazla bütçeyi aşması
Orta **Medium / Critical**	Önemli ama operasyon kapasite bakımından geri alınabilir ve düşük sistem hasarı; Maliyetin 10% ve 50% arası bütçeyi aşması
Düşük **Low / Marginal**	Projeye düşük sistem hasarı; Geri alınabilir operasyon kapasite kaybı; Zamanlama kayması piyasaya sürme tarihini etkilemez; Bütçe aşımı 10% un altındadır

Çizelge 3.7 Bölüm I soruları

Sorular	
Primary programming language of the project – Projenin ilkin programlama dili	Secondary programming language of the project – Projenin ikinci programlama dili
Approximate project size in lines of code - Kod satırlarında yaklaşık proje boyutu	Main DBMS used in the project - Projede kullanılan ana veitabanı yönetim sistemi
Project Business type area - Proje işletme türü alanı	How many years have you been in software business - Yazılım işinde kaç yıl bulundunuz
How many software projects you have been involved as a development team member - Kaç yazılım projesinde geliştirme ekibi üyesi olarak dahil oldunuz?	What was the software type? - Yazılım türü neydi?
What was the software programming paradigm? - Yazılım programlama paradigması ne idi?	Development methodology of the software project - Yazılım projesinin geliştirme metodolojisi
Design pattern(s) used in the project - Projede kullanılan tasarım örüntü (leri)	Development type in the project - Projedeki geliştirme türü

Projeler için 10 farklı başarı kriteri seçildi. Bu kriterler genelde ilgili çalışmalardan alınmıştır (Xu vd. 2003, Shahzad vd. 2009, Xiaosong vd. 2009, Zardari 2009, Appari ve Benaroch 2010, Keshlaf ve Riddle 2010, Shahzad ve Al-Mudimigh 2010, Song vd. 2010, Wu vd. 2010, Hazir 2015). Bu kriterler temel olarak değiştirilmedi ama yanlış anlamaya yol açılmasın diye hafifçe değiştirildi ve bazı kriterler daha küçük kriterlere bölündü. Kriterlerde, eğer tüm değerler yüksek olursa, proje başarılıdır ve eğer düşük olursa başarısızdır. Aksi taktirde zorlanmış veya başarısız olarak belirlenir. Bu kriterlerde 3lü değerlendirme sistemle ölçülmektedir. Bu kriterler, değerlendirmek için çizelge 3.8'de gösterilen sorular şeklinde katılımcılara soruluyor.

Anketin son kısmı, riskleri barındıran bölümdür ki katılımcılara 128 yaşanmış veya yaşanmamış risklerle ilgili olasılık ve etki değerleri soruluyor. Anketin bu kısmını daha kolay ve cazip bir şekle getirmek için katılımcılara birleşik "slider" şeklinde ve renk tabanlı cevap alanı tasarlandı ve ayrıca anlaması zor olan bazı risklerin tanımı bazı referanslardan (Hizazi vd. 2014) ek bilgi olarak tooltip şeklinde verildi.

Çizelge 3.8 Bölüm II soruları

Sorular	
How much the users perceived that the system meets the intended functional requirements? - Kullanıcılar, sistemin tasarlanan işlevsel gerekliliklerinin karşılandıklarını algılamış mıdır?	How much system meets user expectations with respect to ease of use? - Kullanım kolaylığı açısından sistem kullanıcı beklentilerini ne kadar karşılıyor?
How much system meets user expectations with respect to response time? - Sistem cevap süresi bakımından kullanıcı beklentilerini ne kadar karşılıyor?	How much system meets user expectations with respect to reliability? - Güvenilirlik açısından sistem kullanıcının beklentilerini ne kadar karşılıyor?
How much the application developed is easy to maintain? - Geliştirilen uygulamanın devam etmesi ne kadar kolaydır?	How much the users are satisfied with the developed application? - Geliştirilen uygulamadan kullanıcıların memnuniyeti ne kadardır?
How much is the overall quality of the developed application? - Geliştirilen uygulamanın genel kalitesi ne kadardır?	How well the system was completed within budget? - Sistem bütçede içersinde ne kadar iyi tamamlandı?
How good the system was completed within schedule? - Sistem program içinde ne kadar iyi tamamlandı?	How do you rate software defects? - Yazılımı kusurlarını nasıl değerlendiriyorsunuz?

Ayrıca ankette her hangi bir yanlış anlaşılmaya yol açmamak için bir anket bitiriş tuşu anketin sonuna eklendi ve bu tuş tıklanmadan anket sonlanmayacaktır ve sonuçlar sonlanmış anket sonuçlarına eklenmeyecekler. Bu tuş anketteki tek bitirme kriteri değildir ve anketi bitirmek için, giriş zamanı ve bitiş zamanı arası en az 10 dakika olmalıdır. Birkaç denemede anlaşıldı ki soruları okumak ve sağlıklı cevap almak için en az 10 dakika zaman gerekmektedir.

3.4 Uzman Görüşü Verilerinin Toplanması

Tasarlanan anket web uygulaması olarak yapıldı ve katılımcıların web üzerinden erişimi sağlandı. Anket ve bazı sonuçların en güncel versiyonları *http://survey.labs.tips/* sitesinde mevcuttur ve güncellenme halen devam ediyor. Ayrıca EK 1'de bu tarihe kadar bu çalışmada yapılan en güncel referans risk model tablosu mevcuttur.

Anketin soru sayısı çok olduğu için, ortalama 30 dakikada yapılabilen bir ankettir. Bu yüzden katılımcı sayısı beklentilerin altında oldu ve katılımcıların önemli bir kısmı anketi yarım bırakmışlar ve sonlandırmışlar anketi. Katılımcılardan yaklaşık 150 kişi anketi sonlandırmış ve bu sayıda katılımcıdan sadece 40 anket sonucu, doğru cevap olarak hesaplandı. Bu 40 cevap çapraz kontrol ve akran incelemesiyapıldıktan sonra seçilmiştir. Çapraz kontolde bazı sorular 2 kez soruldu ve yanlış cevaplayan katılımcıların sonuçları alınmadı. Bazı diğer katılımcıların sonuçları, proje veri eksikliği için va diğer sonuçlar random ve değersiz oldukları için atıldı.

Katılımcıların büyük bir yüzdesi, yazılım tecrübesine sahip olan öğrencilerden oluşuyor ki istenen bir durum değil. Bunun sebebi sonraki bölümlerde sonuçlardan sonra açıklanacaktır. Söz konusu olan az katılımcı sayısı ve katılımcı kalitesi ve kitlesi, sonuçları açıkladıktan sonra bir kere daha tartışılacaktır.

3.5 Verilerin Ön İşlemeden Geçirilmesi ve ilk Sonuçların Elde Edilmesi

Anketten elimize geçen bilgilerle güncel bir risk kümesi oluşturuluyor. Bu kümeyi oluşturmak için, önce risk değerleri hesaplanmalı. Risk değerleri klasik etki çarpı olasılık yöntemiyle yapılabilir.

İlk yapılan işlem her risk faktörü için etki ve olasılık değerlerin ortalama katılımcı değerini hesaplamaktır. Bir sonraki adım bu değerleri çarparak ortalama risk değerlerini hesaplamaktır. Bu sayede her risk faktörü için tek bir etki, bir olasılık ve bir risk değerimiz olacaktır. Bu değerler basitçe bir sıralamada kullanılabilir, ama çok işlevsel olamaz. O yüzden bu adımda risk matrisi kullanmak daha tercih edilmektedir.

3.5.1 Risk matrisi

Risk matrisi, risk değerlendirmelerinde sıkça kullanılan yöntemlerdendir. Risk matrisinin bir boyutu etki ve diğeri olasılıktır. Bu iki boyutun karışımı risk matrisini yapılandırır ki değerlendirmeyi kolaylaştırır. Sıkça kullanılan risk matrisleri arasında 5x5, 7x5, 7x4 risk matrisleri mevcutlar, ama kolaylık nedeniyle tercih edilen risk matrisi 3x3 matrisidir (Markowski ven Mannan 2008). Kullanılan matriste her eksen üç düzeye bölünüyor ve 9 hücreli bir Niteliksel (Qualitaive) matris oluşturuluyor (Stoneburner vd. 2002). Bu matris 3 bölüme sahiptir:

		ETKİ		
		DÜŞÜK	ORTA	YÜKSEK
OLASILIK	2/3-1 YÜKSEK	I1P3	I2P3	I3P3
	1/3-2/3 ORTA	I1P2	I2P2	I3P2
	0-1/3 DÜŞÜK	I1P1	I2P1	I3P1

Şekil 3.2 Risk matris genel şeması

- High / Major Concern - Yüksek (kırmızı): Bu alanlarda risk yüksektir ve bu konuda bir şeyler yapılmalıdır.
- Medium / Concern - Orta (sarı): Bu alanlarda orta seviyede risk mevcuttur ve bu alandaki riskler projeyi etkileme şansları var.
- Low / No Concern - Düşük (yeşil): Bu alanlarda risk düşük ve kabul edilir bir seviyededir. Bu riskler göz ardı edilebilir.

Devamda eldeki verileri kullanarak ve yukarıdaki şemayı temel alarak bir risk matrisi oluşturuldu. Bu matrise göre, çizelge 3.9 da, riskler göz ardı edilemezler ve sarı ve yada kırmızı bölgelerde yer almaktadırlar. (Bazı sarı bölge riskleri.)

Çizelge 3.9 Risk matris sonuçları (Sadece önemli riskler)

Sıra	**Risk faktörü (EK 1 ID Referans)**	Risk Matris Bölgesi
1	Proje Yönetimi Deneyimindeki Yetersizlik (52)	I3P3
2	Zayıf Proje Planlaması (111)	I3P2
4	Kötü Proje Yönetimi Yaklaşımı / Yöntemi (51)	I2P2
5	Proje Ekibinin Geliştirme Teknolojisi Deneyimi Eksikliği (61)	I2P2
6	Proje- Kaynak Çakışması (12)	I2P2
7	Yazılım Gereksinimlerinin Genişletilmesi (91)	I2P2
8	Gereksinimlerle İlgili Müşterilerin Yetersiz Bilgisi ve Anlayışı (93)	I2P2
9	Kötü Geliştirme Programı (4)	I2P2
10	Analist Yeteneğinin Yetersizliği (5)	I2P2
11	Platform Deneyimi Eksikliği (6)	I2P2
12	Kısa Vadeli Çözüm (Uzun Vadeli Çözüm Yokluğu) (18)	I2P2
13	Test edilebilirlik eksikliği (34)	I2P2
14	Uygulama Zorluğu (35)	I2P2
15	Hataların Geç Belirlenmesi (45)	I2P2
16	Hatalı Kusur Takibi (46)	I2P2
17	Yazılım Mühendisliği Süreci ile Deneyim Eksikliği (57)	I2P2
18	Takım Yetiştirme Yetersizliği (58)	I2P2
19	Birkaç Önemli Kişiye Bağımlılık (101)	I2P2
20	Diğer Sistemlerle Entegre Olma Gerekliği (114)	I2P2
21	Proje liderinin deneyim eksikliği (107)	I2P2

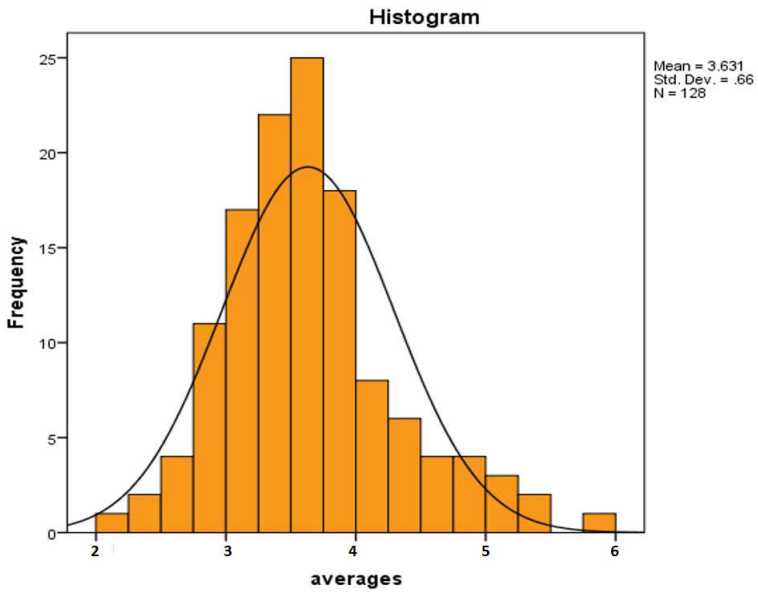
3.5.2 İstatistik

128 riskli modelin tüm riskleri ve elde olan 40 veri istatistiksel olarak incelendi. Eldeki verileri daha iyi anlamak için, önce descriptive istatistik ve frekanslar bu veriler için hesaplandı.

Ortalama risk değerlerin dağınıklığı ve frekanslarını hesaplamak için tüm veriler için etki ve olasılık değerlerin çarpımı hesaplandı ve bu en son bu değerlerin ortalamaları alındı ki her riskin ortalama değerini temsil etmektedir. Daha farklı bir yaklaşımda etki ve olasılık değerlerin çarpımı alınabilir, ama bu yaklaşım veri kaybına yol açar. Bu yüzden ilk yöntem kullanılmalıdır.

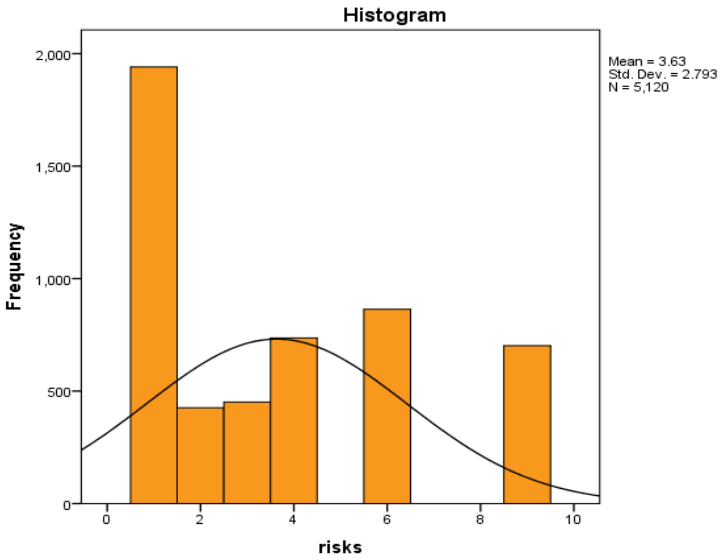
Çizelge 3.10 Risk puan dağılımı

N	Valid	128
	Missing	0
Mean	3.63	
Median	3.55	
Mode	3.3	
Std. Deviation	0.66	
Range	3.70	
Minimum	2.22	
Maximum	5.92	



Şekil 3.3 Risk puan dağılımı (ortalamalar)

Çizelge 3.10 ve şekil 3.3, 128 risk ve 40 veri için istatistiksel olarak frekans şeklindeki verilerin detaylı bilgisidir. Göründüğü gibi, risklerin büyük kısmı ortalama olarak 3.63 civarındadırlar. İstatiksel ve crisp veri hesaplamalarında, benzer çalışmalarda bahsedilenler dışında başka sorunlarla da karşı karşıya gelindi. En önemli sorun, Etki ve Olasılık oranların çarpımında ortaya çıkıyor. Şekil 3.3'te göründüğü gibi, 5,7 ve 8 değerleri 0 olarak görünüyor. Bunun sebebi tam sayı kullanmaktır ve ortada boşlukların meydana gelmesi beklenilir. Bu boşluklar ortalamanın sola doğru kaymasına yol açabilir. Bu yüzden, bulanık mantığa geçmeden alternatif olarak, proje kapsamında başka bir yöntem de denendi ve sonuçlar karşılaştırıldı.



Şekil 3.4 Risk puan dağılımı (Tüm sonuçlarda)

Denemek amacıyla ortaya konılan yönetm çerçevesinde, frekanslar incelendiğinde, ortalama değerleri karşılaştırmaktan ise, bu yöntemde sadece yüksek risk puanları hesaplamak öneriliyor. Yani 4, 6 veya 9 olan risk değerleri hesaplansın ve diğer değerler hiç hesaplamasın. Bu şekilde tehlikeli olarak değerlendirilen riskleri bulmak mümkündür. Başka bir deyişle 4, 6 ve 9 puanlı riskler, risk matrisinde sarı ve kırmız bölgelerinde yer alan riskler. Yani önerilen yöntemde sarı ve kırmızı bölgelerde, 40 veri arasında en çok yer alan riskler, daha tehlikeli olarak alınır ve sıralamaya sokulur.

Be hesaplamayı açıklamak amacıyla, devamda ekstrem bir örnek verildi:

Durum A => 30 veride sadece 1 puan alınmıştır ve 10 veride 9 veya 6 alınmıştır;

Toplam puan bu şekilde hesaplanabilir: 30*1 + 5*(9+6) = 105

Durum B => 36 veride sadece 2 veya 3 puan alınmıştır ve 4 veride 9 veya 6 alınmıştır;

Toplam puan bu şekilde hesaplanabilir: 18*(2+3) + 2*(9+6) = 120

Sonuç olarak, ortalama veri kullanılldığı için, *Durum B* daha yüksek ortalamaya sahip olacaktır. Ama göründüğü gibi *Durum A* yı 10 kişi tehlikeli olarak tespit etmiştir. Bu

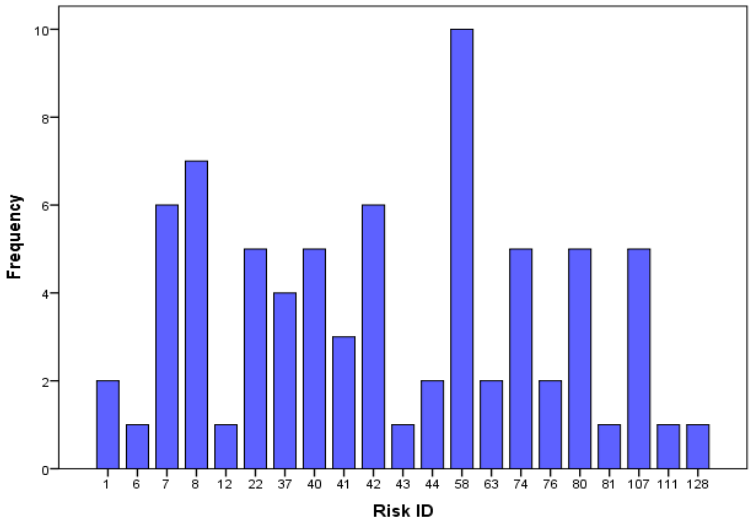
çok aşırı bir durumdur ve belki de en doğru cevap daha yüksek ortalamaya sahip olandır. Ama bu çalışmada bu konun üzerine gidildi ve bu sonuçlar alındı. Sonuçların doğruluğu gelecekteki anketler ve uzman görüşleri tarafından onaylanabilir veya reddedilebilir. Yeni sıralamada yer alan tüm riskler, ilk risk matrisi sıralamasında tekrarlanmıştır. Yani yeni sonuçların en az %90 ıyla, hatta %100 örtüşmektedir. Bu sonuçlar yeni yöntemin kullanılabilir olduğunu göstermektedir. Ankette, risklere ilaveten başarısızlık tipleri de soruldu. Az önceki yöntemle yapılan hesaplamalar yardımıyla, her başarısızlık tipi için en tehlikeli riskler sıralandırıldı ve çizelge 3.11'de gösterilmektedir.

Çizelge 3.11 Başarısızlık tiplerinde tehlikeli olan riskler

Yüksek toplam değeri olan riskler - Öncelik soldan sağa (sadece EK 1 referans ID)							
Project Over Budget							
58	74	22	40	37	44		
Software Defected							
7	58	42	74	40	81	6	128
1	111	12	8	44	41	43	
Software Not Easy to Use							
58	8						
Software Difficult to Maintain							
7	58	8	74	40	80	42	107
Software Difficult to Maintain							
7	58	8	74	40	80		
42	107						
Bad Software Quality							
7	58	8	74	40	80	42	107
Unreliable Software							
107	8	7	58	22	76	42	80
Software Not Meeting Functional Requirements							
58	76	42	37	41	22		
Low Response Time							
58	63	37	41	1			
Customer Not Satisfied							
107	7	8	58	22	80		
Project Over Schedule							
58	63	37	22				

Bu tablodan anlaşılıyor ki bazı riskler, birden fazla başarısızlık tipinde önemliler.

Şekil 3.5, her riskin kaç başarısızlık tipinde etkili ve önemli olduğunu göstermektedir. Bu şekilde göründüğü gibi, özellikle 58. risk (Lack of Training of Team) tüm başarısızlık tiplerinde yüksek puan almıştır. Başka bir değişle, bu riskin yüksek olduğu çalışmalarda başarısızlık tipleri daha fazla görünebilir.



Şekil 3.5 Başarısızlık çizelgesinde en çok görülen riskler

Arnuphaptrairong, bir literatür taramasında (Arnuphaptrairong 2011), toplanan risk faktörlerini karşılaştırmış ve risk faktör listelerin anketlere göre en önemli 10 risklerini[1] çıkartmıştır. Bu çalışmada Arnuphaptrairong'un listeleri, çalışmanın önerdiği 20'li liste ile karşılaştırıldı (Çizelge 3.12). Tekrar edilen riskler * ile işaretlendiler. Daha önce 3.1. ve 3.2. bölümlerinde sözü geçtiği gibi, bu çalışmadaki risklerin bir kısmında kavramsal olarak (bazen tam ve bazen kısmen) örtüşme yaşanmaktaydı. O yüzden riskler indirgendi ve düzenlendi. Bu sebepten dolayı Arnuphaptrairong'un listelerindeki tüm riskler bu çalışmada aynen mevcut değiller. Bu sorunu çözmek için ve doğru bir karşılaştırma yapa bilmek için, listelerdeki risklerin kavramsal olarak bu çalışmadaki risklere yakın olan riskler tam tekrar edilmediği için ***** işareti kullanmak yerine **+** işareti kullanıldı.

[1] Top 10 risks

Çizelge 3.12'de göründüğü gibi, bazı riskler bu çalışmadaki önemli riskler listesinde tekrarlanmışlar. Ama bir çok risk bu çalışmanın risk faktörleri ile uyumsuzdur ve tekrarlanmamış. Bu farklılığın bir sebebi, yapılan çalışmalar ve anketlerin zamanıyla bağlantılı olduğu düşünülüyor. Arnuphaptrairong çalışmasında aynı mantıkla Bohemin sonuçlarıyla ilgili yaşanan farklılığı açıklamıştır. Bu çalışmalar 1991 ve 2008 arasına yapılan anket sonuçlarıyla yapılmışlar ki bu tezi güçlendiriyor. Geçen 2 onyıl da teknolojik ilerlemeler riskler ve etkilerinde büyük değişimler yaratmış. Mesela "Developing wrong user interface" şimdi eskisi kadar tehlikeli bir risk olmayabilir. (Bu sadece bir tahmindir).

Çizelge 3.12 Önemli riskler karşılaştırması

Sıra	**İLK 20 RİSK (ID)**	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**
1.	Lack Of Project Management Experience (52)	*		*		*	*			*
2.	Lack Of Experience With Software Engineering Process (57)	*		+			+		+	
3.	Lack Of Platform Experience (6)	*	+				*			
4.	Bad Project Management Approach / Method (51)	*		*			*		*	
5.	Late Identification Of Defects (45)	*								
6.	Bad Defect Tracking (46)	*								
7.	Lack Of Development Technology Experience Of Project Team (61)	*							+	
8.	Poor Project Planning (111)	*				+			*	
9.	Low Knowledge And Understanding Of Clients Regarding The Requirements (93)	*						*		
10.	Lack Of Training Of Team (58)	*								
11.	Lack Of Analyst Capability (5)	*								
12.	Lack Of Testability (34)	*								
13.	Inaccurate Estimation Of Software Effort (92)	*							+	
14.	Dependency On A Few Key People (101)	*								
15.	Project- Resource Conflict (12)	*								
16.	Bad Or Low Mix Of Team Skills (56)	*				+	+			
17.	Expansion Of Software Requirements (91)	*	+				+		+	

Çizelge 3.12 Önemli riskler karşılaştırması (devam)

18.	**Bad Development Schedule (4)**	*****	**+**				**+**			
19.	Large Project Size (92)	*								
20.	High Extent Of Changes In The Project (108)	*		+	+		+			

A. İlk risk matrisi sıralamasında tekrarlanmıştır.
B. Bohem'in ilk 10 risk listesinde (Arnuphaptrairong 2011) tekrarlanmıştır.
C. Schmidt et al. (USA) in ilk 10 risk listesinde (Arnuphaptrairong 2011) tekrarlanmıştır.
D. Schmidt et al. (HONG KONG) in ilk 10 risk listesinde (Arnuphaptrairong 2011) tekrarlanmıştır.
E. Schmidt et al. (FINLAND) in ilk 10 risk listesinde (Arnuphaptrairong 2011) tekrarlanmıştır.
F. Addison and Vallabh'ın ilk 10 risk listesinde (Arnuphaptrairong 2011) tekrarlanmıştır.
G. Addison'un ilk 10 risk listesinde (Arnuphaptrairong 2011) tekrarlanmıştır.
H. Han and Huang'un ilk 10 risk listesinde (Arnuphaptrairong 2011) tekrarlanmıştır.
İ. Pare et al. in ilk 10 risk listesinde (Arnuphaptrairong 2011) tekrarlanmıştır.

Tablolardaki farklılığın ikinci sebebi bu çalışmadaki anket katılımcıların kalite, tecrübe ve piyasadaki rolleri ile ilgili olabilir. Anketin katılımcıları genelde development aşamasındaki riskleri önemsemişler ve en büyük korkuları, proje yönetimindeki genel yetersizliklerdir. Yani katılımcı, yönetim ve benzeri rollerde olmamıştır. Diğer taraftan katılımcıların bilgileri incelendiğinde bu tezin doğru olduğu anlaşılıyor. Çünkü katılımcıların tümüne yakın bir yüzde, genç bilgisayar ve yazılım mühendisleri, geliştiricileri ve öğrencilerinden oluştuğu biliniyor.

Üçüncü sebep, risk kapsamıdır. Tez projesindeki riskler çok daha kapsamlılar, ama diğerçalışmalar kısıtlı kategorileri kapsamaktadırlar. Mesela bu çalışmada geliştirme sonrası riskleride kayda alınmışlar. Ama Arnuphaptrairong'un çalışmasındaki risk listeleri öyle değildir.

3.5.3 Risk faktörlerinde boyut indirgeme

Bu bölümde istatistiksel olarak, yüksek sayıdaki risk faktörlerini sayı olarak azaltmaya çalışıldı. Bu analizdeki ilk amaç risk faktörlerin arasında her hangi bir bağın olup veya olmadığını tespit etmektir. Eğer risk faktörleri arasında bağ varsa ve risk faktörleri sonuç olarak başka faktörlere tabi tutuluyorlarsa, yüksek bağlı risk faktörler elimine edilebilirler. Bu yüzden 128 risk faktörün arasındaki korelasyon katsayıları (Correlation

Coefficient) hesaplandı. Risk verilerin normal bir dağıtıma sahip oldukları için, Pearson korelasyon katsayıları riskler arası bağlılık gücünü ölçmek için hesaplandı. Bir örnekte korelasyon değeri *r* ile gösterilmektedir ve sınırlamaları bu şekildedir:

$$-1 \leq r \leq +1 \tag{3.2}$$

Weinberg in kitabına göre (Weinberg ve Abramowitz 2008), $r = \pm 0.5$ veya daha güçlü Pearson korelasyon katsayılar, kuvvetli bir bağ sayılır ve ± 1 e yakın değerler en kuvvetli bağları gösterir. Evans'a göre, ± 0.6 ve ± 0.79 aralığındaki korelasyon katsayıları, kuvvetli bir korelasyonu gösterir (Evans 1996) . Bu çalışmadaki durumun özelliği için, düşük katsayı aralığını seçmek riskli olabilir, bu yüzden çalışmada ± 0.6 ve ± 1 aralığındaki korelasyon katsayıları kuvvetli katsayılar olarak alınmıştır. Bir sonraki çizelge kuvvetli korelasyon katsayılı özgün risk çiftlerini göstermektedir. Kuvvetli korelasyon katsayılı veriler istatistik bakımdan tekrar edilen veriler gibi negatif etki yaratırlar. Bu sebepten dolayı çizelge 3.13'te kalın font ila belirlenen riskleri elimine etmek mantıklı olabilir.

Çizelge 3.13 Yüksek korelasyonlu risk çiftleri

ID	**ID**	**korelasyon**	**ID**	**ID**	**korelasyon**
3	126	0.73161	**52**	103	0.66168
5	17	0.62796	**55**	111	0.84347
9	54	0.70722	**56**	74	0.78317
12	66	0.65666	**57**	88	0.74041
18	125	0.83277	**58**	117	0.70219
19	33	0.68681	**60**	96	0.66049
21	30	0.80628	**63**	76	0.63441
23	106	0.62268	**64**	67	0.69514
24	50	0.85583	**65**	120	0.76273
25	74	0.84403	**68**	74	0.6806
26	11	0.71208	**70**	74	0.74817
27	48	0.71146	**71**	93	0.77567
28	95	0.72975	**79**	73	0.69163
29	31	0.77262	**84**	98	0.70398
32	75	0.65482	**85**	120	0.74108

Çizelge 3.13 Yüksek korelasyonlu risk çiftleri (devam)

34	109	0.70646	**86**	120	0.6732
35	48	0.63496	**87**	93	0.85617
36	126	0.74008	**90**	109	0.80085
37	51	0.82162	**92**	123	0.63385
38	126	0.84679	**97**	115	0.75964
39	49	0.82401	**101**	111	0.80295
41	108	0.62427	**105**	124	0.70634
43	104	0.71679	**122**	115	0.76228
44	95	0.71029	**47**	126	0.71017

Çizelge 3.13'te toplamda 48 özgün risk çifti bulunmaktadır. Söylendiği gibi, bu 48 risk elimine edilebilir, ama istatistik olarak veri sayısı düşük olduğu için, korelasyon güveni düşük olabilir ve şimdilik bu çalışma kapsamında eliminasyon yapılmayacaktır.

3.6 Tahmin ve Destek Aracının Geliştirilmesi

Bu aşamanın amacı, bir çok amaçlı araç tasarlamak ve yapmaktır. Bu araçta iki veri girişi ve üç veri çıkışı vardır. Aracın birinci girdisi uzman bilgi girişidir ki aracın anket mekanizmasıyla toplandı ve toplanmaktadır. Bu veri önce bir ön işlemeden geçmelidir ve yanlış, hatalı ve eksik veriler aradan çıkartmalılar. Bu bilgi yalnız bir genel risk matrisi üretmeye yeterlidir ve bu aşamada bu matris oluşturuluyor. İkinci veri girdisi, kullanıcı proje bilgileridir ki ön işlemeden geçen birinci veri ile aracın birinci aşamasını besliyorlar. İkinci aşamanın girişi birinci aşamadan gelen veriler ve birinci ön işlemeden geçen uzman verileridir. Aracın çıkışları genel bir risk matrisi, kullanıcı risk matrisi ve başarısızlık analizleridir.

Bu araç hep ankete açıktır ve bilgiler hep iyileştirme halindeler. Bu yüzden zamanla aracın çıkışları daha iyi olacaktır. Araç PHP programlama dili kullanarak yazılmıştır ve JavaScript, HTML 5 ve CSS3 yardımıyla kullanıcı dostu bir web uygulaması olarak sunuldu. Yapılan aracın girdiler ve çıktılarını daha iyi anlamak için ve ayrıca bu aracın içindeki adımları kısaca izlemek için şekil 3.1 yardımcı olabilir.

3.6.1 İlk aşama

Aracın ilk fazının ilk amacı, genel risk matrise ilaveten, kullanıcı projesine özel bir risk matrisi oluşturmak ve bu risk matrisi için gerekli bilgileri toplamaktır. Bu aşama aslında veriler arasında bir veri tabanı sorgusudur (database query) ve bu sorgunun kriterleri "proje büyüklüğü (LoC)" (proje büyüklüğü kod satır syısı olarak ölçüldü sadece), "proje metodolojisi", "proje kipi" (Paradigm) ve "geliştirme tipi"dir. Bu sorgunun sonucu filtrelenmiş uzman veri listesidir ki işlenmeden geçtikten sonra bir ortalama risk matrisi önerisi ve risk model önerisi kullanıcıya sunar. Sonuçlar her risk ortalama etki ve ortalama olasılık şeklinde kullanıcıya gösterilir ve önceki bölümlerde açıklandığı gibi, kullanıcıya 3x3 risk matrisi sunuluyor.

Bu önerilen risk model ve matrisi kullanıcıya çok değerli olabilir ve kullanıcıyı yapılmakta olan proje riskleri konusunda fikir sahibi yapar. Bu önerilen model ve matris kullanıcın projesi için geçerli bir matris değildir ve kullanıcıya özel şartlardan ve bu çalışmadaki veri yetersizliğinden dolayı kullanıcı sadece bu veriyi bir fikir ve başlangıç noktası olarak kullanmalıdır. Bu başlangıç noktası gerçek durumdan çok uzak olmayacaktır ve bu konu daha sonra bir örnek vaka da incelenecektir.

Buna ilaveten, kullanıcıya ikinci bir risk modeli ve listesi sunulmaktadır ki daha önce ve girilen 4 proje bilgi üzerinden çıkartılmıştır, ama bu risk modeli kullanıcı tarafından değiştirile bilen bir listedir ve kullanıcı her riske ait etki ve olasılık değerlerini manuel olarak değiştirebilir. Bu sayede kullanıcı projenin devamında riskleri düzenleyip ve 2nci bir risk matrisi oluşturabilir ve bu risk matrisi ilk matris ile karşılaştırabilir. Bu sayede kullanıcı projenin risklerini bir noktaya kadar takip edebilir.

Kullanıcı projenin sonunda bu risk değerleri uzman görüşü olarak aracın veri tabanına ekleyip ve aracın veri tabanını güçlendirip ve sonuçları daha sağlıklı yapabilir.

3.6.2 İkinci aşama

İkinci aşamada, başarısızlık tiplerin tahmini için Mamdani'nin sonuç çıkarma modeli (Inference model) (Markowski ve Mannan 2008) kullanılmıştır. Bu bir karar matrisidir ki her hangi bir projenin ilk aşamadan gelen risk modeli verilerini kullanarak başarısızlık puanlarını hesaplıyor ve sıralamada yardımcı oluyor.

Bulanık karar matrisi aslen TFN'ler için kullanılmıştır (Triantaphyllou ve Chi-Tun, 1996). Bulanık karar matrisi, bulanık sayıları sıralamak için çok uygun ve başarılıdır ve diğer taraftan benzeri yöntemlere göre da az komplekstir. Mamdani'in sonuç çıkarma yönteminde $\boldsymbol{\mu_i(m)}$ üyelik fonksiyonu ve $\boldsymbol{\tilde{n}_i}$ bulanık sayısı için denklem (3.3) kullanılır.

$$e_{ij} = max_{x \geq y} \left\{ min \left(\mu_i(x), \mu_j(y) \right) \right\} \qquad (3.3)$$

$$T\ddot{u}m \;\; i, j = 1,2, \dots, m$$

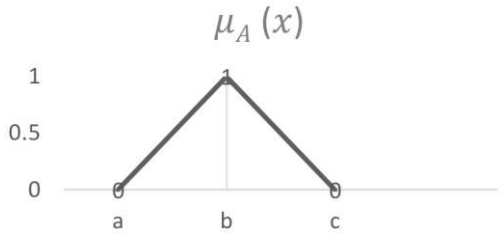
Şimdi $\boldsymbol{\tilde{n}_i}$ sadece ve sadece $\boldsymbol{e_{ij} = 1}$ ve $\boldsymbol{e_{ji} < Q}$ olduğu zaman $\boldsymbol{\tilde{n}_j}$ den büyük sayılır. Bu denklemde $\boldsymbol{Q}$ pozitif bir kesirdir ki 1 den küçüktür (mesela 0.7 ve 0.8). Bu büyüklük $\boldsymbol{\tilde{n}_i > \tilde{n}_j}$ şeklinde gösterilebilir.

Başarısızlık tipleri analizi için, her başarısızlık tipine özel uzman verilerden başarısızlık tipinde tam başarılı olamayan verilerin tümü alındı. Yani çalışmanın ölçeklerinde 3'ten 3 almayan tüm veriler alındı.

Bu verileri denkleme sokmak için önce tüm verilerdeki risk değerleri bulanık sayılar şekline getirilmelidir. Bunu yapmak için önce tüm alınmış ve filtrelenmiş başarısızlık tiplerine ait veriler (tüm 128 risk için) etki ve olasılık olarak işlenmeye başlıyor. Tüm etki ve olasılık puanları bulanık üyelik fonksiyonu yaratan bir işlemden geçiyorlar.

Daha önce tartışıldığı gibi, olasılık ve etki oranları 1 ve 3 arasında alınmaktadır ve bu

oranları bulanık sayıya çevirmek için, bu aşamada üçgen sayılar tercih edildi. Bu sayıları üretmek için denklem 3.4 kullanılabilir. Bu denklemde ***A*** bir üçgen bulanık sayısıdır.



Şekil 3.6 Üçgensel bulanık numara

$$\mu_A(x) = \begin{cases} 0 & if x < a \\ \dfrac{x-a}{b-a} & if a < x \leq b \\ \dfrac{c-x}{c-b} & if b < x \leq c \\ 0 & if x > c \end{cases}$$

$$A = (a, b, c) \tag{3.4}$$

Bu denklemi daha iyi anlamak için şekil 3.7, ***A*** sayısının üyeliğini göstermektedir.

Tüm olasılık ve etki puanlarını bulanıklaştırmadan sonra, tüm bu sayılar birleştirilmeli ler ki tek bir uzman görüşü rakamı haline gelsinler. Normal rakamlarda en basit yöntem ortalama hesaplamadır. Ama şimdi bulanık rakamlar söz konusu olduğu için daha farklı yöntemler kullanılmalıdır. Bu çalışmada TFN kullanıldığı için, sadece bu sayılarda sadık olan bir yöntem kullanıldı (Pandey vd. 2012). Bu yöntem, TFN lerde zirvenin sağ ve sol açılara dayalı toplama operasyonunu yapıyor. (3.5),(3.6) ve (3.7) numaralı denklemler bu yöntemin aşamalarını göstermektedir.

$$\acute{b} = \frac{1}{n}\sum_1^n b_i \tag{3.5}$$

$$\acute{a} = \frac{1}{n}\sum_1^n b_i - tan\left[\frac{1}{n}\sum_1^n tan^{-1}(b_i - a_i)\right] \tag{3.6}$$

$$\acute{c} = \frac{1}{n}\sum_1^n b_i + tan\left[\frac{1}{n}\sum_1^n tan^{-1}(c_i - b_i)\right] \quad (3.7)$$

Daha önce söylendiği gibi, risk puanı etki ve olasılık puanlarının çarpımıyla hesaplanır, ama bu aşamada sayılar TFN sayıları şeklindeler ve durum daha farklıdır. Bu yüzden etki ve olasılık çarpımını hesaplamak için, daha farklı bir yöntem (Gao vd. 2009), ufak değişikliklerle adapte olmak için kullanıldı. Bu yöntem de TFN ler ile uyumlu olduğu için tercih edildi. Benzer yöntemler diğer çalışmalarda önerilmiştir (Taleshian ve Rezvani 2011) ki biraz değişiklikle kullanılabilir. Şimdi $\mu_A(x)$ ve $\mu_B(x)$in çarpımı denklem (3.8) yardımıyla hesaplanabilir.

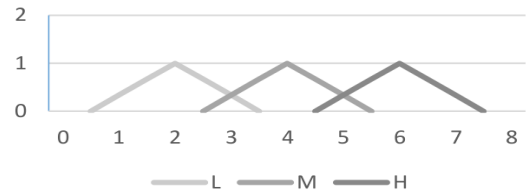
$$\mu_{\tilde{Q}}(Z) = \begin{cases} \frac{-(a_1b_2+a_2b_1-2a_1a_2)+\sqrt{(a_1b_2-a_2b_1)^2+4(b_1-c_1)(b_2-c_2)Z}}{2(b_1-a_1)(b_2-a_2)} & a_1a_2 \le Z \le b_1b_2 \\ \frac{-(c_1b_2+c_2b_1-2c_1c_2)-\sqrt{(a_1b_2-a_2b_1)^2+4(b_1-c_1)(b_2-c_2)Z}}{2(b_1-c_1)(b_2-c_2)} & b_1b_2 < Z \le C_1C_2 \\ 0 & otherwise \end{cases} \quad (3.8)$$

TNF leri her olasılık ve etki skoruna üretmek için, önce üyelik fonksiyonları tanımlanmalı. Daha önce bahsedildiği gibi, bulanık sayılar üçgen olmalılar ki denklemlerle uyumlu olsunlar.

Çalışma aşamasında kullanılan üçlü oranlar kullanıcı dostu oranlar oldukları için çok uygunlar. Ama hesaplamalarda değişiklik oranları sadece kesirler şeklinde ortaya çıkabilir bu durumda. Ayrıca bu yanlış hesaplara yol açabilir. Çünkü ortada tam sayı çok azdır. Her hangi bir yanlış yorumlamaya yol açmamak için ve hesaplar ve sonuçlardaki değişiklikler daha iyi ortaya çıksın diye sonuçlardaki oranlar biraz değiştirildi. Oranlar daha geniş olsun diye, ikiye çarpıldı ve şimdi 1 ve 6 arasındadır. Bu değişiklik sadece oransal ve sonuçlar üzerinde her hangi bir etkisi yoktur ve her hangi

bir hataya yol açmaz. Tüm verilerde yapılan bu işlem sonuçları hiç bir şekilde değiştirmez ve sadece hatları düşürür ve sonuçları daha bariz yapar. Bu yeni ölçek ile, çarpının sonuçları 0 ve 64 aralığında olacaklar ki bu geniş menzilde bulanık rakamlar daha iyi yer alır. Şekil 3.7'de kullanılan üyelik fonksiyonunun bir gösterişidir. (L: düşük, M: orta, H: yüksek)

$$L = (0.5,2,3.5), M = (2.5,4,5.5), H = (4.5,6,7.5)$$



Şekil 3.7 Üyelik fonksiyonları

Yöntemin açıklamak hedefiyle bir örnek devamda gösterilmiştir ki açıklanan denklemler ve yöntem ile hesaplanmıştır. Bu örnekte tek bir risk için farz ediliyor ki etki oranı M ve olasılık oranı H dir. Üyelik fonksiyonları etki ve olasılık için bu şekilde olacaktır:

Probability L: [0/0,0.34/1,1/2,0.34/3,0/4,0/5,0/6,0/7,0/8] (3.9)

Impact M :[0/0,0/1,0/2,0.34/3,1/4,0.34/5,0/6,0/7,0/8] (3.10)

Şimdi denklem (3.8), (3.9) ve (3.10)´yi kullanarak toplam risk puanı, bulanık sayı şeklinde hesaplanabilir (denklem (3.11)).

R = [0/0, 0/1, 0.15/2, 0.33333333/3, 0.49/4, 0.63/5, 0.76/6, 0.88/7, 1/8, 0.89/9, 0.78/10, 0.69/11, 0.59/12, 0.50/13, 0.41/14, 0.33/15, 0.25/16, 0.17/17, 0.09/18, 0.01/19, 0/20, 0/21, 0/22, 0/23, 0/24, 0/25, 0/26, 0/27, 0/28, 0/29, 0/30, 0/31, 0/32, 0/33, 0/34, 0/35, 0/36, 0/37, 0/38, 0/39, 0/40, 0/41, 0/42, 0/43, 0/44, 0/45, 0/46, 0/47, 0/48, 0/49, 0/50, 0/51, 0/52, 0/53, 0/54, 0/55, 0/56, 0/57, 0/58, 0/59, 0/60, 0/61, 0/62, 0/63, 0/64] (3.11)

Bu aşamada, karar matrisi için gereken kullanıcı verisi ya alternatiflerdir. Bu veri aslında birinci aşamada kullanıcıdan alınıyor. Bu adımda işlenen bilgi kullanıcının değiştirdiği etki ve olasılık puanlarıdır. Eğer kullanıcı bu verileri değiştirmezse, otomatik önerilen risk modeli kullanılacaktır. Daha sonra bu veriler TFN ye çevrilip ve her risk için toplam bulanık risk sayısı hesaplanıyor.

Birinci denklem için uzman veriler ve kullanıcı verilerini (alternatifler) aldıktan sonra, denklem gereği önce her risk için uzman ve kullanıcı risklerin minimumları alınıyor. Bu adımda riskler kadar bulanık sayı vardır ve bu sayıların birinci denklem çerçevesinde maksimumunu bulmak bir sonraki adımdır. Bu maksimum değer, tek bir bulanık sayısı şeklindedir ve başarısızlık puanı olarak kullanılıyor. Bu puan bulanık sayı olduğu için anlaması zor olabilir, bu yüzden denklem (3.12)'ü kullanarak bu sayıyı normal bir sayı şekline getirilebilir.

$$a = \frac{\sum \mu_A(x_i) x_i}{\sum \mu_A(x_i)} \tag{3.12}$$

Eğer denklem (3.12)'ü kullanarak (3.11)'teki bulanık sayı normal sayıya çevrilirse, sonuç denklem (3.13) olacaktır.

$$a = \frac{69.5348}{9.0199} = 7.709 \tag{3.13}$$

Daha yüksek rakamlar, daha büyük risk/değer/başarısızlık olasılığı demektir. Ayrıca araç özelliklerine dahil, benzeri bir hesaplamada, toplam risk değeri bulanık ve normal sayı şeklinde, kullanıcıya verilir.

4. ÖRNEK VAKA

Yapılan araç ve yöntemi denemek amacıyla, örnek bir durumda araç kullanıldı ve gerçek dünya performansı test edildi. Bu vaka çalışmasında yazılım prototipinin ne kadar gerçek bir vakada işlevsel olduğu ölçülmeye çalışıldı. İşlevsellik bu tezin amaçlarını kapsayan bir kavramdır. Yani risk analiz ve takip hızı ve maliyeti ve genelde toplam gereken çabayı ölçmek ve değerlendirmektir. Ayrıca yazılımın verdiği sonuçların proje genelinde ne kadar istenen kaliteyi sağlamaktadır diye bir değerlendirme yapıldı. Bu yazılımın prototip aşamasında olduğunau bilerek, ve veri eksikliğini göz önünde bulundurarak vaka çalışması yapılmalıdır.

Vaka seçiminde bazı kriterler göz önünde bulunduruldu. Öncelikle vaka küçük bir takımda yapılması istenir ve bu yazılım ve yöntemin daha çok Agile odaklı olduğu için, Agile metoduyla yapılan bir yazılım projesi tercih edilmektedir. Ayrıca projenin daha ilk aşamalarda olması bu vaka denemesi için büyük bir öneme sahiptir. Çünkü sonuçları var olan bir yazılım, önyargıya sebep olabilir. Bu kriterlere uyan ve projesi hakkında bir mulakata zaman ayırmak isteyen bir proje bu örnek vakada incelenmiştir. Örnek vaka verileri, doğruluğu düşürmemek amacıyla bir mulakatta alındı ve kavramlarda ve veri alımında, proje sahibine yardım edildi. Vaka veri alımının tüm adımları izlendi ve her risk ve gerekçeleri tartışıldı.

4.1 Örneğin Bilgileri ve Ayarları

Bu yöntem daha çok, küçük, yeni ve risk yönetimi ve izlemede nispeten daha az tecrübeli kurumlar için tasarlandı. Benzer bir yapı ve özelliklere sahip olan bir vakada, geliştirlen prototip aracı denendi. Bu vaka olgunlaşmamış bir organizasyondur (Tanrıöver ve Demirörs 2015). Bu vakadaki projenin hedefi bir triaj karar destek yazılımı yapmaktır. Bu yazılım hastaları hastanelerin acil bölümüne vardıkları zaman ve triaj hemşiresi tarafından başvuruları değerlendiği zaman kategorize etmelidir. Hastalar, hemşirenin girdiği verilere dayanarak öncelikli kategoriye girebilirler. Bu

yazılım dünyada kabul edilen ve kanıtlanan CTAS (Iserson ve Moskop 2007) yöntemini kullanan, 5 aşamalı, 5 öncelik kategoriye sahip bir sistemdir. Bu yazılım sadece destek yazılımıdır ve sadece yardım amalı geliştirilmelidir. Yani insanların yerini tutamaz, tutmamalıdır ve bir profesyonel hüküm ve kararı olmalıdır.

Söz konusu sistem, SCRUM (Anonymous 2016) kullanarak, 2 kişilik bir takımda yapılması planlanıyor. Ayrıca, uyumsuzluk ve diğer problemleri indirgemek amacıyla Java programlama dilini kullanarak, bu projenin yapılması bekleniyor.

SCRUM da belirli bir risk yönetim yöntemi olmadığından dolayı, takımda risk incelemesi yapılmadı. Ama bu çalışma kapsamında 15 önemli requirement (mesela triaj edilmemiş hastaları görmek, triaj edilmiş hastaları görmek, hastanın medikal geçmişini görmek v.s.) belirtildi ve bu 15'ten üçü zor yapılanlar arasında. Ama yine de herhangi bir risk değerlendirme çıktısı yoktur. Bu sorun genelde Agile yöntemleriyle yapılan projelerde ortaktır. Bu projede, risk değerlendirmesi için İnternette bulunan bazı araçlar ve uygulamalar düşünüldü ve değerlendirildi. Bu araçlara Jira (Anonymous 2016), Risk Radar (Anonymous 2014) ve Risk Management Studio (Anonymous 2015)dahiller. Bu araçlar önce den belirlenmiş risk faktör listesi içermiyorlar ve risk modeli önermiyorlar. Bazı uygulamalarda, sadece bazı kısıtlı riskler için, önce den belirli riskler önerilmektedir ki yeterli değiller.

4.2 Örnekte Araç ve Metod Denemesi

Örnek kapsamında, 3 saatlik bir buluşmada, yapılan araç yarımıyla risk değerlendirmesi yapıldı. Uygulama prototip aşamasında olduğu için ve her hangi bir hataya yol açmaması için, kullanıcıya direk yardım edildi. Örnek proje gereksinimleri için resmi ve formal olmayan bazı belgeler yardımıyla ve müzakereler sonunda, sistemin proje için önerdiği risk oranları düzenlendi. Örnek proje daha ilk aşamalarında olduğu için ve diğer sebepler yüzünden (mesela metodoloji), projeyle ilgili belgeler ve veriler çok kısıtlıydı. Bu yüzden risk değerlendirme aşamasında bazı gözlemler yapıldı:

1. Başta, araç kapsamında önerilen 128 riske benzer bir işlevsellik diğer hiç bir araç ve yazılımda (mesela Jira, Radar, v.s.) yoktur.

2. Önerilen değerler, proje takımının her riskin etki ve olasılık değerlendirmesini yapımında önemli derecede yardımcı oldu. Bu önerilen değerler olmadan, riskleri değerlendirmek çok zaman alıcı olurdu ve bazı riskleri değerlendirmek mümkün olmazdı.

3. Uygulamada önerilen risk önceliklendirme ve risk matrisi, önemli riskleri bulup ve onlara daha çok zaman ayırmakta yardımcı oldu.

4. Kullanıcı ortamında, etki ve olasılık değerlerini değiştirmek için kullanılan oranlar, analog dereceler şeklinde değiştirilebilir ve üçlü oranlardan çok daha iyi ve kullanıcı dostudur. Ayrıca bu oranlar kullanıcıya değerlendirmede daha çok özgürlük vermekteler ve bulanık değerlendirmede da iyi sonuçlar alınır.

5. Uygulamanın önceki veriler üzerinden risk bilgileri önerdiğine rağmen, bazı risk değerleri projenin değerleriyle çok farklı olabilir ve mutlaka gözden geçirilmelidir. Devamdaki bölümde ve Ek 2'de bu farklılık gösterilmiştir.

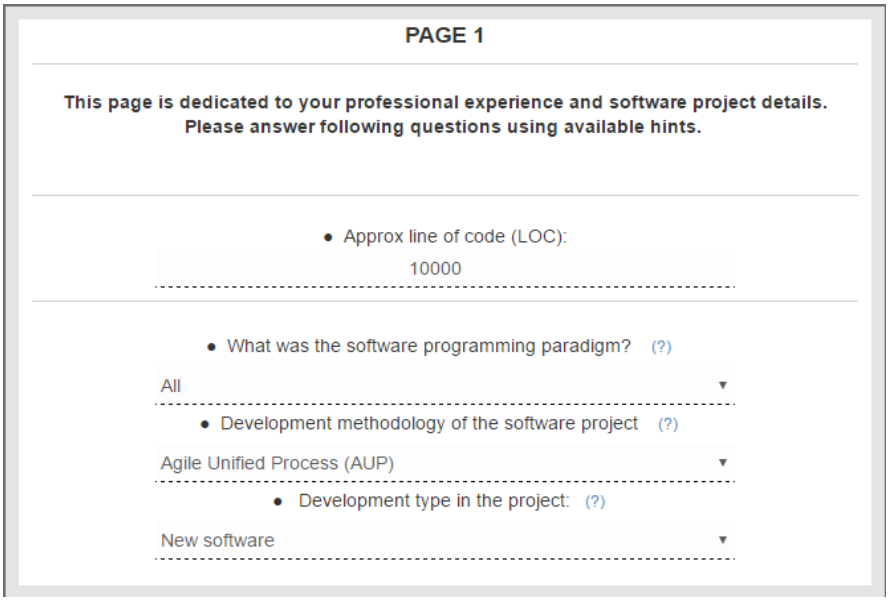
6. Projeden önce den düşünülmeyen bazı riskler, uygulama tarafından önerildi ki gerçekten önemsenmelidir. (Mesela "Backup Issues", "Potential Increase in database size", "Security Risks" riskleri.)

7. Uygulamada önerilen toplam risk değeri ve potansiyel başarısızlık tipi değerleri, projenin yönetiminde ve özellikle zaman ve maliyet olarak ayarlarında yardımcı olabilir.

4.3 Bulgular ve Sonuçlar

Uygulamanın kullanıcı ara yüzü 2 adımda çalışır. Birinci adım ilk proje bilgi girişi panelidir. Bu adımda (Data register) kullanıcıya projenin bilgileri soruluyor. Şekil 4.1'de bu adım görünüyor (Resimde girilen bilgiler temsili bilgiler ve örnek ile ilgisi yok).

İkinci adım değerlendirme adımıdır ki EK 2'de örnek bir resim bu adımdan gösterilmiştir. Bu adım kullanıcıya etki ve olasılık değerleri önerilir ve kullanıcı kendi değerlenirmesini bu önceden belirlenmiş verilere ilaveten, sisteme verebilir. Bu değerler analog oranlar ("visual analogue scales (Sáa vd. 2015)) şeklinde değiştirilebilir. Bu analog girişler kullanıcıya daha çok özgürlük ve kontrol verir. Değerler 0.3 mesafeyle ayrık alınmaktadır. Bunun sebebi değerleri daha kolay kontröl etmek ve esşitlemektir. Şekilde göründüğü gibi, sol tarafta yazılımın proje için önerdiği olasılık ve etki değerleri vardır ve sağ tarafta kullanıcının değerlendirmesi olasılık ve etki oranları şeklinde mevcuttur. Kırmızı noktalar toplam risk (olasılık * etki) oranını göstermektedir. Kırmızı önemli ve siyah önemsiz risk demektir. Bu olasılık ve etki değerlerini değiştirdikten sonra, araç hem önerilen hemde değiştirilen değerlere risk matrisleri oluşturur. Örnekteki risk matriste önemli olarak çıkan riskler çizelge 4.1 verilmiştir. Ayrıca örnek projenin toplam sayısal riski 50,30/64 dür ki sözel olarak yüksek bir risk olarak değerlendirilebilir.

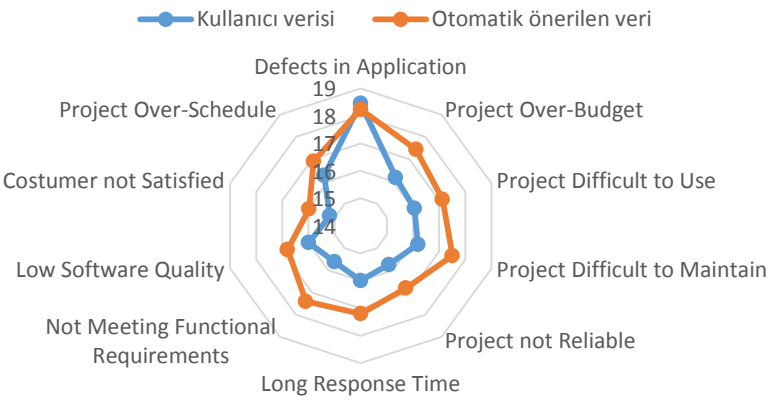


Şekil 4.1 Yazılımda değerlendirme için alınan bilgiler

Çizelge 4.1 Örnek vaka risk matrisi sonuçları (sadece çok önemli -kırmızı bölge- riskleri)

Low Knowledge and Understanding of Clients Regarding the Requirements (Gereksinimlerle İlgili Müşterilerin Yetersiz Bilgisi ve Anlayışı)	Instability and Lack of Continuity in Project Staffing (İstihbarat ve Proje Personelinde Süreksizlik)
Team Member Unavailability (Takım Üyesinin Kullanılamaması)	Lack of Expertise with Application Area (Domain) (Uygulama Alanıyla Uzmanın Yetersizliği (Etki Alanı))
Staff Turnover (Personel Devri)	Dependency On a Few Key People (Sayılı Önemli Kişiye Bağımlılık)
High Extent of Changes in The Project (Projede Çok Sayıda Değişiklikler)	Lack of Organizational Maturity
Lack of Requirements Stability (Gereksinimleri İstikrar eksikliği)	Need to Integrate with Other Systems (Diğer Sistemlerle Entegre Olunması Gerekliliği)
Lack of Frozen Requirements (Dondurulmuş Gereksinimlerin olmaması)	Excessive Reliance On a Single Development Environment (Tek Bir Geliştirme Ortamına Aşırı Güven ve Bağlılık)
Requirements NOT Complete and Clear (Gereksinimler Tam ve Temiz olmaması)	Misleading Estimation About Skills Of Workers (İşçilerin Becerileri Hakkında Yanıltıcı Tahmin)
Expansion of Software Requirements (- Yazılım Gereksinimlerinin Genişletilmesi)	Gold Plating (Altın Kaplama)
Lack of Software Developer Competence (Yazılım Geliştirici Yetkinliğinin Yetersizliği)	

Daha önce bahsedildiği gibi, araçta başarısızlık tipi analiz potansiyeli vardır ve bu örnekte başarısızlık tipleri incelendi. Başarısızlık tipi sonuçları önerilen ve sonradan girilen veriler için üretilmektedir ki gösteriş amaçlı, "*Overschedule*" başarısızlık tipi için sonuçlar şekil 4.3'te gösteriliyor. Başarısızlık tipi sonuçlarından tek bir incelemede anlam çıkartmak, bir kaç risk yapılandırma işleminden sonraki kıyaslama kadar başarılı olmayabilir. Diğer başarısızlık tipleri Şekil 4.2'de karşılaştırıldı. Sonuçlara göre bu projede "Defect" başarısızlık tipi en muhtemel başarısızlık tipidir. Ama bu başarısızlık tipleri rakamsal olarak değerlendirilmemeliler ve projenin farklı aşamalarında takip edilerek kıyaslanmalılar.



Şekil 4.2 Başarısızlık tipleri

Yapılan örnek incelemesine göre, yazılımın ilk öğrenim ve kullanımı en fazla 3 saatlik bir süre gerektirir. Ama bu sürede projeyle ilgili tüm risk bilgileri veriilir ve sonuçlar tahmin edilir. Yani zaman olarak, benzer bir araca göre daha verimli ve az çaba ve zaman gerektiren bir risk analiz yazılımıdır. Bu 3 saatlik çalışmayı, projenin devamındaki risk analiz aşamalarında 1 saate indirgemek mümkündür. Yanı kısa bir hesaplamada eğer farz edelim ki bir projede toplam olarak (en az) 7 kez risk planlaması yapılacaksa , projenin hayatı boyunca yaklaşık 9 satlik bir çalışmada riskler takip edilir. Diğer araçlarda ilk risk tespiti çok daha uzun bir işlemdir. Kullanıcılar riskleri mevcut risk listelerinden aldıktan sonra, riskler ile ilgili hiç bir fikre sahip olmadan riskleri reğerlendirmeliler. Bu işlem kesinlikle saatlerce zaman gerektiren bir risk analiz sürecidir. Diğer taraftan risk analizi dışında, başarısızlık tipi öneren bir araçtır ve tez yapım aşamasında benzer bir araç bulunamadı.

Şekil 4.3 Overschedule Başarısızlık tipi



Şekil 4.4 Verilen değerler

5. SONUÇ

Bu çalışmada, yazılımcılar ve özellikle küçük ve orta büyüklükte yazılımcılar ve firmalar için bir proaktif yöntem önerildi ve bir araç tasarlandı. Aracın amacı yazılım projeleri için bir risk modeli ve değerlendirmesi sunmaktır. Ayrıca kalitatif ve belirsiz değerlere rağmen, bu araçta daha iyi bir değerlendirme yapmak amacıyla, bulanık mantık kullanarak riskler inceleniyor ve risk sonuçları, başarısızlık tipleri şeklinde değerlendiriliyor. Bu işler, yazılımcılar için büyük maliyet ve zaman kaybına yol açabilir ve bu aracı kullanarak bu kayıplar önlenebilir.

Bu konuda yapılan bazı benzer çalışmalar, frameworklar ve araçlar vardır. Ama diğer çalışmalarda benzeri risk önerisi ve analizler yapılmamaktadır. Mesela bazı çalışmalarda sadece kısıtlı sayıda riskleri baz alarak analizler yapılır ve bazı diğer araçlar ve çalışmalarda hiç bir risk önerilmemektedir. Bu mesele yazılımcıların işini daha da zor hale getirir. Çünkü risk tanımı ve ilk araştırma zor ve vakit gerektiren bir iştir. Bu yüzden bu eksiklikleri gidermeye çalışan bir araç tasarlandı ve yazılım yapıldı.

Çalışmanın devamında, yöntem ve aracın güvenirliğini ve performansını denemek ve doğrulamak amacıyla, örnek bir vakada araç test edildi. Örnek için önerilen riskler beklenildiği gibi orta ve yüksek seviyede çıktı. Ama programdaki az veri sayısından kaynaklanan hatalar yüzünden bazı risk değerleri beklenildiği gibi önerilmedi. Ama bunun tek sebebi veri eksikliği değildir ve her duruma özel farklı sonuçlar alınması mantıklıdır.

Sonuç olarak çalışma eksikliklere rağmen, örnek vakada çok pratik ve faydalı olarak değerlendi. Araç yazılım projelerinde risk yönetimi için çok yardımcı oldu ve yazılımcılara risklerle ilgili ve başarısızlık oranlarıyla ilgili önemli bilgiler verdi.

Gelecek çalışmalarda ve bu çalışmanın devamında, daha farklı bir uzman görüş toplanması planlanıyor. Bu sayede, sonuçların daha iyi ve gerçeğe yakın olması

beklenir. Ayrıca öngörü yönteminde değişiklikler yaratılabilir ve risklerin oluşumunu önlemek için, araç tavsiyede bulunabilir. İlaveten risklerin karakteristik özelliklerini Wang ve Fu çalışmasındaki (Wang ve Fu 2011)gibi incelemek ve hesaplarda kullanarak farklı sistemler ve hatta reactive bir sistem önerebilmek mümkündür.

KAYNAKLAR

Alberts, C.J. and Dorofee, A.J. 2010. Risk Management Framework.

Anonymous. 2013. A guide to the project management body of knowledge (PMBOK guide), Project Management Institute. doi:10.1002/pmj.20125

Anonymous. 2013. The Standish Group International, CHAOS MANIFESTO: Think Big, Act Small.

Anonymous. 2014. Pro Concepts LLC. Web Sitesi: http://www.proconceptsllc.com/risk-radar.html. Erişim Tarihi: 02.02.2016.

Anonymous. 2015. RM Studio Web Sitesi: http://www.riskmanagementstudio.com/. Erişim Tarihi: 02.02.2016.

Anonymous. 2015. The Standish Group International, CHAOS Report.

Anonymous. 2016. JIRA Software, Atlassian Found. Web Sitesi: https://www.atlassian.com/software/jira. Erişim Tarihi: 2.3.2016.

Anonymous. 2016. Scrum Web Sitesi: https://www.scrum.org/. Erişim Tarihi: 02.04.2016.

Appari, A. and Benaroch, M. 2010. Monetary pricing of software development risks: A method and empirical illustration. J. Syst. Softw. 83, 2098–2107. doi:10.1016/j.jss.2010.06.012

Arnuphaptrairong, T. 2011. Top Ten Lists of Software Project Risks: Evidence from the Literature Survey. Proc. Int. MultiConference Eng. Comput. Sci. I.

Büyüközkan, G. and Ruan, D. 2010. Choquet integral based aggregation approach to software development risk assessment. Inf. Sci. (Ny). 441–451. doi:10.1016/j.ins.2009.09.009

Evans, J.D. 1996. Straightforward Statistics for the Behavioral Sciences. Brooks/Cole Publishing Company.

Gao, S., Zhang, Z. and Cao, C. 2009. Multiplication operation on fuzzy numbers. J. Softw. 4, 331–338. doi:10.4304/jsw.4.4.331-338

Haisjackl, C., Felderer, M. and Breu, R. 2013. RisCal -- A Risk Estimation Tool for Software Engineering Purposes. 39th Euromicro Conf. Softw. Eng. Adv. Appl. 292–299. doi:10.1109/SEAA.2013.10

Hazir, Ö. 2015. A review of analytical models, approaches and decision support tools in project monitoring and control. Int. J. Proj. Manag. 33, 808–815. doi:10.1016/j.ijproman.2014.09.005

Hizazi, H., Arshad, N.H., Mohamed, A. and Nor, Z.M. 2014. Risk Factors in Software Development Phases. Eur. Sci. J. 10, 213–232.

Hu, Y., Zhang, X., Ngai, E.W.T., Cai, R. and Liu, M. 2013. Software project risk analysis using Bayesian networks with causality constraints. Decis. Support Syst. 56, 439–449. doi:10.1016/j.dss.2012.11.001

Iserson, K. V. and Moskop, J.C. 2007. Triage in Medicine, Part I: Concept, History, and Types. Ann. Emerg. Med. 49, 275–281. doi:10.1016/j.annemergmed.2006.05.019

Keshlaf, A.A. and Riddle, S. 2010. Risk Management for Web and Distributed Software Development Projects. Fifth Int. Conf. Internet Monit. Prot. 22–28. doi:10.1109/ICIMP.2010.12

Markowski, A.S. and Mannan, M.S. 2008. Fuzzy risk matrix. J. Hazard. Mater. 159, 152–157. doi:10.1016/j.jhazmat.2008.03.055

Olid, K. and Mannan, B. 2008. A Review of Software Risk Management for Selection of best Tools and Techniques 773–778. doi:10.1109/SNPD.2008.127

Pandey, M., Khare, N. and Shrivastava, S. 2012. New Aggregation Operator for Triangular Fuzzy Numbers based on the Arithmetic Means of the L-and R-Apex Angles. Submitt. Publ. 2, 990–992.

Pressman, R.S. 2010. Software Engineering a Practitioner's Approach, 7. ed, McGraw-Hill. doi:10.1109/6.476732

Sáa, S.R., Gil, M.Á., González-Rodríguez, G., López, M.T. and Lubiano, M.A. 2015. Fuzzy rating scale-based questionnaires and their statistical analysis. IEEE Trans. Fuzzy Syst. 23, 1–14. doi:10.1109/TFUZZ.2014.2307895

Sadiq, M., Rahman, A., Ahmad, S., Asim, M. and Ahmad, J. 2010. EsrcTool: A tool to estimate the software risk and cost, 2nd International Conference on Computer Research and Development, ICCRD 2010.; 886–890. doi:10.1109/ICCRD.2010.29

Shahzad, B. and Al-Mudimigh, A.S. 2010. Risk Identification, Mitigation and Avoidance Model for Handling Software Risk. 2nd Int. Conf. Comput. Intell. Commun. Syst. Networks 191–196. doi:10.1109/CICSyN.2010.82

Shahzad, B., Ullah, I. and Khan, N. 2009. Software Risk Identification and Mitigation in Incremental Model. 2009 Int. Conf. Inf. Multimed. Technol. 366–370. doi:10.1109/ICIMT.2009.104

Song, H., Wu, D., Li, M., Cai, C. and Li, J. 2010. An entropy based approach for software risk assessment: A perspective of trustworthiness enhancement. Softw. Eng., 575–578.

Stoneburner, G., Goguen, A. and Feringa, A. 2002. Risk Management Guide for Information Technology Systems : Recommendations of the National Institute of Standards and Technology. Natl. Inst. Stand. Technol., 1–25.

Taleshian, A. and Rezvani, S. 2011. Multiplication Operation on Trapezoidal Fuzzy Numbers. J. Phys. Sci. 15, 17–26.

Tanrıöver, Ö. and Demirörs, O. 2015. A process capability based assessment model for software workforce in emergent software organizations. Comput. Stand. Interfaces 37, 29–40. doi:10.1016/j.csi.2014.05.003

Triantaphyllou, E. and Chi-Tun, L. 1996. Development and evaluation of five fuzzy multiattribute decision-making methods. Int. J. Approx. Reason. 14, 281–310. doi:10.1016/0888-613X(95)00119-2

Verner, J.M., Brereton, O.P., Kitchenham, B. A., Turner, M. and Niazi, M. 2014. Risks and risk mitigation in global software development: A tertiary study. Inf. Softw. Technol. 56, 54–78. doi:10.1016/j.infsof.2013.06.005

Wang, Y. and Fu, S. 2011. A General Cognition to the Multi-characters of Software Risks. Int. Conf. Comput. Inf. Sci. ; 737–739. doi:10.1109/ICCIS.2011.33

Weinberg, S. and Abramowitz, S. 2008. Statistics using SPSS: An integrative approach.

Wu, D., Song, H., Li, M., Cai, C. and Li, J. 2010. Modeling Risk Factors Dependence Using Copula Method for Assessing Software Schedule Risk, içinde: Software Engineering and Data Mining (SEDM), 2010 2nd International Conference on.; 571–574.

Xiaosong, L., Shushi, L., Wenjun, C. and Songjiang, F. 2009. The Application of Risk Matrix to Software Project Risk Management. Int. Forum Inf. Technol. Appl. 480–483. doi:10.1109/IFITA.2009.542

Xu, Z., Khoshgoftaar, T.M. and Allen, E.B. 2003. Application of fuzzy expert systems in assessing operational risk of software. Inf. Softw. Technol. 45, 373–388. doi:10.1016/S0950-5849(03)00010-7

Zardari, S. 2009. Software Risk Management. 2009 Int. Conf. Inf. Manag. Eng. 375–379. doi:10.1109/ICIME.2009.138

EKLER

EK 1 Risk Listesi

Ek 2 Yazılımın ikinci aşamasından örnek bir resim

EK 1 Risk Listesi

ID	***Risk beyanı***	***Referans***	*ID*	***Risk beyanı***	***Referans***
1	Large Database Size - Büyük Veritabanı Boyutu	(Appari ve Benaroch 2010)	66	Developing Wrong User Interface - Yanlış Kullanıcı Arayüzünün Geliştirilmesi	(Zardari 2009, Keshlaf ve Riddle 2010)
2	Main Storage Constraint - Ana Depolama Kısıtlaması	(Appari ve Benaroch 2010)	67	Gold Plating - Altın Kaplama	(Keshlaf ve Riddle 2010)
3	High Platform Volatility - Yüksek Platform Volatilitesi	(Appari ve Benaroch 2010)	68	Shortfalls In Outsourced Components - Dışarıdan Kaynaklanan Bileşenlerdeki Düşüşler	(Keshlaf ve Riddle 2010)
4	Bad Development Schedule - Kötü Geliştirme Programı	(Appari ve Benaroch 2010)	69	Shortfalls In Externally Performed Tasks - Dışarıdan Uygulanan Görevlerdeki Düşüşler	(Keshlaf ve Riddle 2010)
5	Lack of Analyst Capability - Analist Yeteneğinin Yetersizliği	(Appari ve Benaroch 2010)	70	Real-time Performance Shortfalls - Gerçek Zamanlı Performans Açığı	(Keshlaf ve Riddle 2010)
6	Lack of Platform Experience - Platform Deneyimi Eksikliği	(Appari ve Benaroch 2010)	71	Bad Traceability - Kötü İzlenebilirlik	(Keshlaf ve Riddle 2010)
7	Lack of Use of Modern Programming Practices - Modern Programlama Uygulamalarının Kullanılmaması	(Appari ve Benaroch 2010)	72	Insufficient Verification and Validation - Yetersiz Onaylama ve Doğrulama	(Keshlaf ve Riddle 2010)
8	Low Usage of Software Support Tools - Yazılım Destek Araçlarının Düşük Kullanımı	(Appari ve Benaroch 2010)	73	Customer Unsatisfied At Project Delivery - Proje Tesliminde Müşteri Memnun Değil	(Keshlaf ve Riddle 2010)
9	Lack of Software Developer Competence - Yazılım Geliştirici Yetkinliğinin Yetersizliği	(Appari ve Benaroch 2010)	74	Risk Reducing Technique Producing New Risk - Risk Azaltma Tekniği Yeni Risk Üretiyor	(Keshlaf ve Riddle 2010)
10	Project NOT Fit To Customer Organization - Proje Müşteri Organizasyonuna Uygun Değil	(Xiaosong vd. 2009)	75	Catastrophe / Disaster - Felaket	(Keshlaf ve Riddle 2010)
11	Lack of Customer Perception - Müşteri Algılamasında Yetersizlik	(Xiaosong vd. 2009)	76	Incorrect Project Size Estimation - Yanlış Proje Boyutu Tahmini	(Shahzad vd. 2009)
12	Project- Resource Conflict - Proje- Kaynak Çakışması	(Xiaosong vd. 2009)	77	Project Funding Uncertainty - Proje Finansmanı Belirsizliği	(Shahzad vd. 2009)
13	Customer Conflict - Müşteri çatışması	(Xiaosong vd. 2009)	78	Rapid Change of Job - Hızlı İş Değişim	(Shahzad vd. 2009)

EK 1 Risk listesi (devam)

14	Lack of Leadership - Liderliğin eksikliği	(Xiaosong vd. 2009)	79	Change In Working Circumstances By Management - Çalışma Koşullarında Yönetim Tarafından Değiştirmek	(Shahzad vd. 2009)
15	Definition of The Program (ambiguity) - Programın Tanımı (belirsizlik)	(Xiaosong vd. 2009)	80	Hardware Default Changes - Donanım Varsayılan Değişiklikleri	(Shahzad vd. 2009)
16	High Political Influences - Yüksek Politik Etkiler	(Xiaosong vd. 2009)	81	Requirement Postponement - Gereksinimin ertelenmesi	(Shahzad vd. 2009)
17	Inconvenient Date - Uygunsuz Tarih	(Xiaosong vd. 2009)	82	Presence of High Bugs/errors Count - Yüksek Hata / Hatalar Sayısı	(Shahzad vd. 2009)
18	Short Term Solution (lack of Long Term Solution) - Kısa Vadeli Çözüm (Uzun Vadeli Çözüm Yokluğu)	(Xiaosong vd. 2009)	83	Technology Change - Teknoloji Değiştirme	(Shahzad vd. 2009)
19	Lack of Organization Stability - Organizasyonun Kararlılığı Yoksunluğu	(Xiaosong vd. 2009)	84	Underestimation of Data Increase Due To Software Success - Veri Başarısızlığının Yazılım Başarısı Nedeniyle Artması	(Shahzad vd. 2009)
20	Lack of Organization Roles and Responsibilities - Organizasyon Rolleri ve Sorumluluklarının Yetersizliği	(Xiaosong vd. 2009)	85	Lack of Design and Development Tool Independence - Tasarım ve Geliştirme Aracının Bağımsızlığı Yoksunluğu	(Shahzad vd. 2009)
21	Lack of Policies and Standards - Politikalar ve Standartların Eksikliği	(Xiaosong vd. 2009)	86	Risk of Intruders (hackers, Viruses, Trojan Horse) - Davetsiz misafirlerin tehlikesi (hackerlar, Virüsler, Truva Atları)	(Shahzad vd. 2009)
22	Lack of Management Support and Involvement - Yönetim Desteği ve Katılımı Eksikliği	(Xiaosong vd. 2009)	87	Misleading Estimation About Skills of Workers - İşçilerin Becerileri Hakkında Yanıltıcı Tahmin	(Shahzad vd. 2009)
23	Lack of Project Objectives - Proje Hedeflerinin Olmaması	(Xiaosong vd. 2009)	88	Lack of Technical Feedback - Teknik Geribildirim Yokluğu	(Shahzad vd. 2009)
24	Lack of User Involvement - Kullanıcı Katılımının olmaması	(Xiaosong vd. 2009)	89	Compromise On Profit To Save Name - Kâr Üzerinde Ödeyecek Tasarruf Verme	(Shahzad vd. 2009)
25	Lack of User Acceptance - Kullanıcı Kabulünün olmaması	(Xiaosong vd. 2009)	90	Risk of Economy Distortion - Ekonomi Bozulma Riski	(Shahzad vd. 2009)
26	High User Training Needs - Yüksek Kullanıcı Eğitim İhtiyaçları	(Xiaosong vd. 2009)	91	Expansion of Software Requirements - Yazılım Gereksinimlerinin Genişletilmesi	(Wu vd. 2010)
27	Large Project Size - Büyük Proje Boyutu	(Xiaosong vd. 2009)	92	Inaccurate Estimation of Software Effort - Yazılım Eforunun Hatalı Tahmini	(Wu vd. 2010)

EK 1 Risk listesi (devam)

28	Hardware Constraints - Donanım Kısıtlamaları	(Xiaosong vd. 2009)	93	Low Knowledge and Understanding of Clients Regarding The Requirements - Gereksinimlerle İlgili Müşterilerin Yetersiz Bilgisi ve Anlayışı	(Song vd. 2010)
29	Lack of Reusable Components - Yeniden Kullanılabilen Bileşenlerin olmaması	(Xiaosong vd. 2009)	94	Incorrect Requirements - Yanlış Gereksinimler	(Song vd. 2010)
30	Lack of Cost Controls - Masraf Kontrollerinin Olmaması	(Xiaosong vd. 2009)	95	Lack of Frozen Requirements - Dondurulmuş Gereksinimlerin olmaması	(Song vd. 2010)
31	Lack of Delivery Commitment - Teslimat Taahhüdü Eksikliği	(Xiaosong vd. 2009)	96	Undefined Project Success Criteria - Tanımsız Proje Başarı Kriterleri	(Song vd. 2010)
32	Lack of Requirements Stability - Gereksinimleri İstikrar eksikliği	(Xiaosong vd. 2009)	97	Conflicting System Requirements - Çakışan Sistem Gereksinimleri	(Song vd. 2010)
33	Requirements NOT Complete and Clear - Gereksinimler Tamamlanmıyor ve Temiz Değil	(Xiaosong vd. 2009)	98	Conflict Between User Departments - Kullanıcı Bölümleri Arasındaki Çatışma	(Song vd. 2010)
34	Lack of Testability - Test edilebilirlik eksikliği	(Xiaosong vd. 2009)	99	Low Number of Users In and Outside The Organization - Organizasyon İçinde ve Dışarda Düşük Kullanıcı Sayısı	(Song vd. 2010)
35	Implementation Difficulty - Uygulama Zorluğu	(Xiaosong vd. 2009)	100	Instability of The Client's Business Environment - Müşterinin İş Ortamının İstikrarsızlığı	(Song vd. 2010)
36	High System Dependencies - Yüksek Sistem Bağımlılıkları	(Xiaosong vd. 2009)	101	Dependency On A Few Key People - Birkaç Önemli Kişiye Bağımlılık	(Song vd. 2010)
37	Lack of Response or Other Performance Factors - Yanıt Yokluğu veya Diğer Performans Faktörleri	(Xiaosong vd. 2009)	102	Lack of Staff Commitment, Low Morale - Personel Taahhüdü, Düşük Moral	(Song vd. 2010)
38	High Customer Service Impact - Yüksek Müşteri Hizmet Etkisi	(Xiaosong vd. 2009)	103	Instability and Lack of Continuity In Project Staffing - İstihbarat ve Proje Personelinde Süreksizlik	(Song vd. 2010)
39	Data Migration Required - Veri Taşıma Gerekliliği	(Xiaosong vd. 2009)	104	High Number of People On Team - Takımda Çok sayıda insan	(Song vd. 2010)
40	Lack of Pilot Approach - Pilot Yaklaşım eksikliği	(Xiaosong vd. 2009)	105	Low Team Diversity - Düşük Takım Çeşitliliği	(Song vd. 2010)

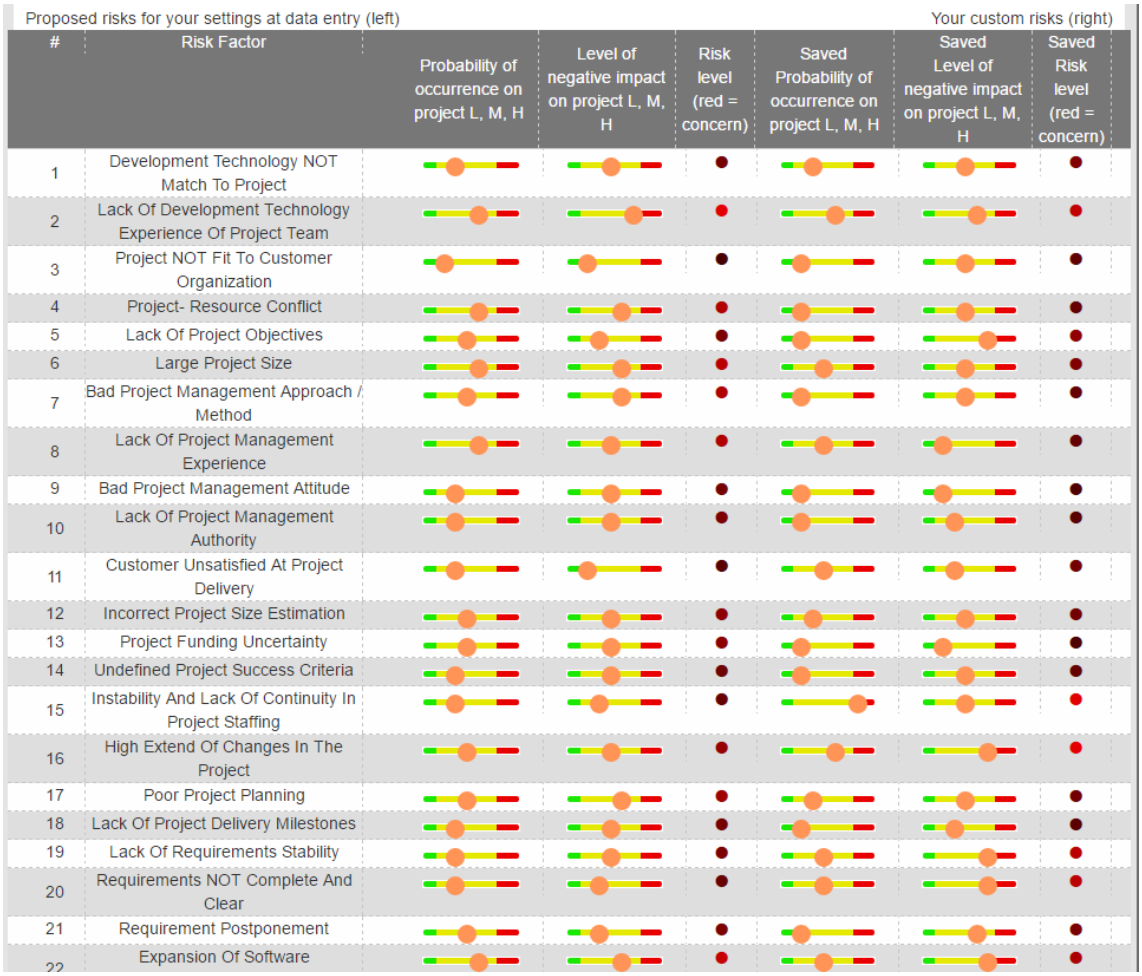
EK 1 Risk listesi (devam)

41	Lack of Alternatives Analysis - Alternatif Analiz eksikliği	(Xiaosong vd. 2009)	106	Lack of Organizational Maturity - Örgütsel Vade Olmaması	(Song vd. 2010)
42	Lack of Quality Assurance Approach - Kalite Güvencesi Yaklaşımı eksikliği	(Xiaosong vd. 2009)	107	Lack of Project leader's experience - Proje liderinin deneyim eksikliği	(Song vd. 2010)
43	Lack of Development Documentation - Geliştirme Dokümantasyonu eksikliği	(Xiaosong vd. 2009)	108	High Extent of Changes In The Project - Projede Yüksek Değişiklik Yapısı	(Song vd. 2010)
44	No Use of Defined Engineering Process - Tanımlanmış Mühendislik Sürecinin Kullanımı Olmaması	(Xiaosong vd. 2009)	109	Excessive Schedule Pressure - Aşırı Zamanlama Basıncı	(Song vd. 2010)
45	Late Identification of Defects - Hataların Geç Belirlenmesi	(Xiaosong vd. 2009)	110	Inadequate Cost Estimating - Yetersiz Maliyet Kestirimi	(Song vd. 2010)
46	Bad Defect Tracking - Hatalı Kusur Takibi	(Xiaosong vd. 2009)	111	Poor Project Planning - Zayıf Proje Planlaması	(Song vd. 2010)
47	Lack of or Bad Change Control For Work Products - İş Ürünlerinde Eksiklikler veya Kötü Değişim Kontrolü	(Xiaosong vd. 2009)	112	Ineffective Communication - Etkisiz İletişim	(Song vd. 2010)
48	Problem With Physical Facilities - Fiziksel Tesislerle İlgili Sorun	(Xiaosong vd. 2009)	113	Improper Definition of Roles and Responsibilities - Rollerin ve Sorumlulukların Uygunsuz Tanımı	(Song vd. 2010)
49	Problem With Hardware Platform - Donanım Platformu ile İlgili Sorun	(Xiaosong vd. 2009)	114	Need To Integrate With Other Systems - Diğer Sistemlerle Entegre Olunması Gerekiyor	(Song vd. 2010)
50	Tools Unavailability - Araçlar Kullanılamazlığı	(Xiaosong vd. 2009)	115	Inadequate Configuration Control - Yetersiz Yapılandırma Kontrolü	(Song vd. 2010)
51	Bad Project Management Approach / Method - Kötü Proje Yönetimi Yaklaşımı / Yöntemi	(Xiaosong vd. 2009)	116	Low Quality of Software and Hardware Supplier Support - Düşük Kalite Yazılım ve Donanım Tedarikçisi Desteği	(Song vd. 2010)
52	Lack of Project Management Experience - Proje Yönetimi Deneyimindeki Yetersizlik	(Xiaosong vd. 2009)	117	Excessive Reliance On A Single Development Environment - Tek Bir Geliştirme Ortamında Aşırı Güvence	(Song vd. 2010)
53	Bad Project Management Attitude - Kötü Proje Yönetimi Tutumu	(Xiaosong vd. 2009)	118	High Extent of Linkage To Other Organizations - Diğer Kuruluşlara Bağlantı Yüksekliği	(Song vd. 2010)

EK 1 Risk listesi (devam)

54	Lack of Project Management Authority - Proje Yönetim Otoritesi eksikliği	(Xiaosong vd. 2009)	119	Resource Insufficiency - Kaynak yetersizliği	(Song vd. 2010)
55	Team Member Unavailability - Takım Üyesinin Kullanılamaması	(Xiaosong vd. 2009)	120	Intensity of Conflicts - Çatışmaların Yoğunluğu	(Song vd. 2010)
56	Bad or Low Mix of Team Skills - Takım Becerilerinin Kötü veya Düşük Karması	(Xiaosong vd. 2009)	121	Lack of Control Over Consultants, Vendors ,sub-contractors - Danışmanlar, Satıcılar, Alt Yükleniciler Üzerindeki Kontrol Eksikliği	(Song vd. 2010)
57	Lack of Experience With Software Engineering Process - Yazılım Mühendisliği Süreci ile Deneyim Eksikliği	(Xiaosong vd. 2009)	122	Massive User Stress - Masif Kullanıcı Stresi	(Shahzad vd. 2009)
58	Lack of Training of Team - Takım Yetiştirme Yetersizliği	(Xiaosong vd. 2009)	123	Lack of Project Delivery Milestones - Proje Teslimat Kilometre taşlarının olmaması	(Shahzad vd. 2009)
59	Lack of Expertise With Application Area (Domain) - Uygulama Alanında Uzmanın Yetersizliği (Etki Alanı)	(Xiaosong vd. 2009)	124	Over-optimistic Technology Perceives - Aşırı iyimser Teknoloji Algısı	(Shahzad vd. 2009)
60	Development Technology NOT Match To Project - Kalkınma Teknolojisi Projeye Uygun Olmaması	(Xiaosong vd. 2009)	125	Staff Turnover – Personel Devri	(Shahzad vd. 2009)
61	Lack of Development Technology Experience of Project Team - Proje Ekibinin Kalkınma Teknolojisi Deneyimi Eksikliği	(Xiaosong vd. 2009)	126	Backup Issues - Yedekleme Konuları	(Shahzad vd. 2009)
62	Immaturity of Development Technology - Kalkınma Teknolojisinin Olgunlaşmamışlığı	(Xiaosong vd. 2009)	127	Bad Preservation of Intellectuals - Aydınların Kötü Korunması	(Shahzad vd. 2009)
63	High Design Complexity - Yüksek Tasarım Karmaşıklığı	(Xiaosong vd. 2009)	128	Inability To Secure Confidential Customer Data - Gizli Müşteri Verilerini Koruyamamak	
64	Lack of Support Personnel - Destek Personeli Yokluğu	(Xiaosong vd. 2009)			
65	Developing Wrong Software Functions - Yanlış Yazılım İşlevlerini Geliştirme	(Zardari 2009, Keshlaf ve Riddle 2010)			

Ek 2 Yazılımın ikinci aşamasından örnek bir resim



Proposed risks for your settings at data entry (left) — Your custom risks (right)

#	Risk Factor	Probability of occurrence on project L, M, H	Level of negative impact on project L, M, H	Risk level (red = concern)	Saved Probability of occurrence on project L, M, H	Saved Level of negative impact on project L, M, H	Saved Risk level (red = concern)
1	Development Technology NOT Match To Project						
2	Lack Of Development Technology Experience Of Project Team						
3	Project NOT Fit To Customer Organization						
4	Project- Resource Conflict						
5	Lack Of Project Objectives						
6	Large Project Size						
7	Bad Project Management Approach / Method						
8	Lack Of Project Management Experience						
9	Bad Project Management Attitude						
10	Lack Of Project Management Authority						
11	Customer Unsatisfied At Project Delivery						
12	Incorrect Project Size Estimation						
13	Project Funding Uncertainty						
14	Undefined Project Success Criteria						
15	Instability And Lack Of Continuity In Project Staffing						
16	High Extend Of Changes In The Project						
17	Poor Project Planning						
18	Lack Of Project Delivery Milestones						
19	Lack Of Requirements Stability						
20	Requirements NOT Complete And Clear						
21	Requirement Postponement						
22	Expansion Of Software						

ÖZGEÇMİŞ

Adı Soyadı :Sahand VAHIDNIA

Doğum Yeri :İRAN / URMİA

Doğum Tarihi :16-09-1990

Medeni Hali :Bekar

Yabancı Diller:İngilizce (IELTS 7.0, YDS 91.25), Türkçe (TÖMER C1)

Eğitim Durumu:

Lise : Alavi Lisesi, Matematik ve Fizik Alanı (2008)

Lisans : Azarbaijan Shahid Madani University, Bilişim Teknoloji ve Bilgisayar Fakültesi, Bilişim Teknoloji Mühendisliği (2013)

Yüksek Lisans :Ankara Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği, (Şubat 2014- Aralık 2016)

Araştırmalar:

Vahidnia, S., Tanrıöver, Ö., Askerzade, I.N., 2016. GENERAL RISK FACTOR BASED INITIAL SOFTWARE PROJECT RISK ASSESSMENT FOR EMERGENT SOFTWARE ORGANIZATIONS: A SUPPORT METHOD AND TOOL, Journal of Information and Software Technology (Submitted, In peer review)

Vahidnia, S., Tanrıöver, Ö., Askerzade, I.N., 2017. AN EVALUATION STUDY OF GENERAL SOFTWARE PROJECT RISK BASED ON SOFTWARE PRACTITIONERS EXPERIENCES, International Journal of Computer Science and Information Technology (IJCSIT)