

Mikroservis Mimarisinde Analiz ve Tasarım: Türkiye’den Bir İnceleme

Analysis and Design of Microservices: Results from Turkey

Burak Bilgin
Bilgisayar Mühendisliği Bölümü
İzmir Yüksek Teknoloji Enstitüsü
İzmir, Türkiye
burakbilgin@iyte.edu.tr

Hüseyin Ünlü
Bilgisayar Mühendisliği Bölümü
İzmir Yüksek Teknoloji Enstitüsü
İzmir, Türkiye
huseyinunlu@iyte.edu.tr

Onur Demirörs
Bilgisayar Mühendisliği Bölümü
İzmir Yüksek Teknoloji Enstitüsü
İzmir, Türkiye
onurdemirors@iyte.edu.tr

Özet—Günümüzde esnek, güvenilir ve duyarlı (responsive) yazılımlara olan talebin artması ve bu talebe karşılık verebilen internet altyapısının olması yeni nesil bulut uygulamalarının gelişmesine öncülük etmiştir. Bunun sonucunda, bulut tabanlı dağıtık web uygulamalarının birlikte çalışarak oluşturduğu mikroservis tabanlı mimari popülerlik kazanmıştır. Mikroservis tabanlı mimari oldukça yeni olup bu mimariyi kullanarak yazılım geliştiren firmaların kültürlerini değiştirme gerekliliği doğmuştur. Ancak, literatürde mikroservis tabanlı mimaride analiz ve tasarım konusunda uygulanan yöntemleri ele alan çalışma sayısı çok azdır. Bu çalışmada, Türkiye’deki yazılım organizasyonlarının mikroservis tabanlı proje geliştirirken başvurdukları analiz ve tasarım yöntemlerini ele alan bir anket düzenlenmiştir. Anket sonuçları, mikroservis tabanlı proje konusunda tecrübesi olan yazılım organizasyonlarının analiz ve tasarım konusundaki bakış açılarını ortaya çıkarmaktadır. Elde edilen sonuçlar mikroservis kullanan organizasyonlar ile ilgili genel tabloyu göstermekte ve araştırmacılar için çalışma yapılabilecek konuları önermektedir.

Anahtar Kelimeler—mikroservis, yazılım analizi, yazılım tasarımı, anket, bulut platformu

Abstract— The demand for more flexible, responsive and reliable software applications and the availability of internet infrastructure to respond to this demand, led to a new generation of cloud-based web applications. As a result, cloud-based distributed web applications working together in a microservice-based architecture has gained popularity. The concept of microservice-based architecture is quite new and software organizations need to transform their culture to develop applications in this fashion. However, there is a lack of research studies in the literature that explores the common practices for the analysis and design of microservices. Thus, we performed a survey to explore the organizational choices on software analysis and design when working with microservices. In this paper, we present the results from Turkey. The results provide a snapshot of the software industry that utilizes microservices and suggests a set of challenges researchers can focus on in the area.

Keywords—microservices, software analysis, software design, practitioner survey, cloud applications

I. GİRİŞ

Günümüz yazılımları, ölçeklenebilir ve neredeyse sınırları olmayan kaynaklardan oluşan uygulamalar yönünde değişmektedir. Bulut platformu bu ise bu özellikleri sağlamaya yöneliktir. Bulut platformu sayesinde sağlanan bu özellikler kullanılarak iş mantığı farklı platformlara dağıtılabilir [1]. Mikroservis tabanlı mimari bu amaç için bir çözüm olarak yer alır. Mikroservis mimarisinde gevşek bağlı (loosely-coupled) servisler uyum içerisinde (cohesion) çalışır. Bu servisler birbirinden bağımsız olarak dağıtılabilir, ölçeklendirilebilir ve test edilebilir; aralarındaki iletişim ise REST benzeri arayüzler kullanarak ağ aracılığı ile gerçekleştirilir [1], [2].

Mikroservis mimarisinin yazılım endüstrisindeki popülerliği geçtiğimiz yıllarda daha da artmaya başlamıştır. Mikroservislerin birbirinden bağımsız olarak geliştirilebilmesi ve dağıtılabilmesi, önemli özellikleri olan kullanıma devam ettirilebilir ve ölçeklendirilebilir olmasını sağlar. Mikroservislerin aslında birer servis olmasına rağmen Servis Odaklı Mimariden temel karakteristik farklılıkları vardır. Bu farklılıklar boyut, sınırlı bağlam (bounded context) ve bağımsızlık olarak sıralanabilir [3]. İlk olarak, mikroservis boyutu bilinen servislere göre daha küçüktür; her bir servis genel olarak sadece bir iş mantığını sağlar. İkinci olarak, bir mikroservis ilgili bütün fonksiyonluları tek bir iş mantığı çerçevesinde toplar. Son olarak ise, her bir mikroservis bağımsız birer servistir. Her bir mikroservis tek sorumluluğu olduğu için değiştirilmesi için sadece bir neden olmalıdır ve diğer mikroservisler bu değişimden etkilenmeyecek şekilde dizayn edilmelidir.

Servis odaklı mimari ile yazılım geliştiren organizasyonlar mikroservisler ile benzer konseptleri kullandıklarını, DevOps uygulayan organizasyonlar ise otomatikleştirilmiş dağıtım başladıklarını gözlemleyebilirler. Benzer olarak, çevik yazılım geliştiren organizasyonlar aslında kültürlerini mikroservis dünyasına uydurmaya başladıklarını gösterir [3]. Ancak, bu değişim görüldüğü kadar kolay olmamakla birlikte görülmemiş zorlukları da beraberinde getirir. Mikroservis tabanlı mimari ile yazılım geliştirmeye başlayan organizasyonlar bir yandan yazılımda sorumluluğu dağıtım ve özerklik ile diğer yandan da yazılım takımlarının çıktılarını

etkili bir şekilde yönetme ve entegre etmek ile uğraşmaktadırlar. Bunun yanı sıra, geleneksel analiz ve tasarım tekniklerinin mikroservisler ile çalışırken etkili olmadığını görmekteyiz. En temel soyutlama kavramı olan “nesne” bile değişmek zorunda kalabilir. Literatürde ise geleneksel tekniklerin yanı sıra birçok farklı modelleme notasyonu bulunmaktadır ancak, kullanılan modelleme tekniklerinin anlaşılabilir olması önemlidir [4]. Bu anket ile amacımız Türkiye’deki organizasyonların bu zorluklar karşısında uyguladıkları yöntemleri göstermektir.

Bu çalışmanın başlarında, mikroservis tabanlı mimaride analiz ve tasarım için başvurulan yöntemler ile ilgili sistematik literatür taraması yapılmıştır. Ancak, bu konuda literatürde çalışmalarda eksiklik olduğu görülmüştür ve metodoloji organizasyonların mikroservis tabanlı yazılım geliştirirken izlediği analiz ve tasarım tekniklerini ele alan bir anket çalışması olarak değiştirilmiştir. Bu çalışmada Türkiye’deki yazılım organizasyonların mikroservis tabanlı projelerde uyguladığı analiz ve tasarım teknikleri ile ilgili sonuçları ele almak amaçlanmıştır.

Bu çalışmanın geri kalan kısımları şu şekilde organize edilmiştir. Bölüm 2 ilgili çalışmaları, Bölüm 3 araştırma yöntemini vermektedir. Anket sonuçları 4. Bölümde verilmekte ve 5. Bölümde bulgular tartışılmaktadır. Son olarak, 6. Bölümde sonuçlar ve gelecekte planlanan çalışmalar ele alınmaktadır.

II. İLGİLİ ÇALIŞMALAR

Literatürdeki çalışmalar incelendiğinde sektördeki organizasyonların mikroservis tabanlı projelerin analiz ve tasarım aşamalarını nasıl gerçekleştirdiğini ele alan bir araştırma olmadığı görülmektedir. Ancak, mikroservis alanında farklı konuları ele alan sistematik literatür taramaları bulunmaktadır. Pahl ve Jamshidi [5] tarafından yapılan çalışmada mikroservis ile ilgili 33 çalışma metodoloji, mimari ve platform desteği başlıklarında kategorize edilmiştir. Bir diğer çalışmada, Francesco ve diğerleri [6] mikroservis mimarisindeki çalışmaları yayın trendi, araştırma odağı ve endüstriyel adaptasyon potansiyeli başlıkları ile ele almıştır. Bu çalışma, mikroservisler ile ilgili yayınların 2014 yılında başladığını ve 2015 yılında yayın sayısının en fazla olduğunu göstermiştir. Alshuqayran ve diğerleri [7] tarafından yapılan çalışmada ise mikroservis ile ilgili çalışmalar mimari zorluklar, kalite nitelikleri ve mimari diyagramlar yönünden ele alınmıştır. Çalışma sonuçları mikroservis mimarisi için en çok kullanılan diyagramların bileşen (component), sıralama (sequence), süreç (process), dağılım (deployment), sınıf (class) ve kullanım durumu (use case) diyagramları olduğunu göstermektedir. Son olarak, Taibi ve diğerleri [8] tarafından yapılan çalışmada ise mikroservis tabanlı mimari prensip ve kalıpları ele alınmış ve bu kalıplar avantaj ve dezavantajları ile birlikte tartışılmıştır.

III. ARAŞTIRMA YÖNTEMİ

Bu çalışmada, organizasyonların mikroservis tabanlı projelerde uyguladıkları analiz ve tasarım yöntemleri hakkında bilgi edinebilmek amacıyla anket metodolojisine başvurulmuştur. Anket, belli kişilerin ya da grupların bir konu üzerindeki deneyimlerini kısa sürede anlayabilmek için sıklıkla başvurulan bir yöntemdir. Birçok farklı alanda başvurulduğu gibi, yazılım mühendisliği alanında da çeşitli konularda örnekleri bulunmaktadır [9]–[11]. Bu amaçla, anonim çevrimiçi anket yapılmasına karar verilmiştir. Bu seçiminin arkasında 3 ana neden bulunmaktadır. İlk olarak,

anketin çevrimiçi olarak uygulanmasının nedeni bu sayede daha fazla katılımcıya ulaşabilmektir. İkinci olarak, anketin anonim olması ile katılımcıların şirket bilgilerini paylaşmadan tecrübelerini paylaşmaları sağlanmıştır. Son olarak ise elde edilen verinin daha kolay kategorize ve analiz edilmesini sağlanmıştır. Anket Google Forms uygulaması aracılığı ile uygulanmıştır.

Anketlerde olan ana problemlerden biri de anketin belirsiz sorular içerebilmesidir [12]. Olası belirsizlikleri kaldırmak amacı ile öncelikle bir pilot anket çalışması uygulanmış ve tamamlama süresi ile birlikte geri dönüşler alınmıştır. Bu uygulamadan sonra sorularda bazı düzeltmeler yapılmış ve anket bu düzeltmelerden sonra İzmir Yüksek Teknoloji Enstitüsü Etik Komitesi’nden onay alınarak uygulamaya başlanmıştır.

Anketin hedef grubu mikroservis tabanlı mimaride tecrübesi olan yazılım alanında çalışan uzmanlar olup bu katılımcılara ulaşmak için çeşitli eposta grupları, LinkedIn ve araştırmacıların çevresindeki bu konuda deneyimli uzmanlar ile anket paylaşılmıştır.

A. Amaç ve Araştırma Soruları

Bu çalışmada, organizasyonların mikroservis tabanlı projelerde başvurduğu analiz ve tasarım yöntemlerini belirlemek amacı ile araştırma soruları aşağıdaki şekilde belirlenmiştir:

- *Organizasyonlar mikroservis tabanlı proje geliştirmede hangi analiz tekniklerini kullanmaktadır?*
- *Organizasyonlar mikroservis tabanlı proje geliştirmede hangi tasarım tekniklerini kullanmaktadır?*

B. Anket Tasarımı

TABLO I. ANALİZ VE TASARIM SORULARI

Bölüm	Soru	Soru Tipi	
		Tek Yanıt	Çok Yanıt
5	Yazılım analizi için standart bir süreç kullanılıyor mu?	✓	
5	Şu anki (eğer bittiyse en son) mikroservis tabanlı projenizde fonksiyonel gereksinimlerini nasıl gösterdiniz?		✓
5	Şu anki (eğer bittiyse en son) mikroservis tabanlı projenizde problemi nasıl analiz ettiniz?		✓
5	Şu anki (eğer bittiyse en son) mikroservis tabanlı projenizde analiz için hangi notasyonları kullandınız?		✓
5	Şu anki (eğer bittiyse en son) mikroservis tabanlı projenizde problemi nasıl mikroservislere ayırtırdınız?	✓	
6	Şu anki (eğer bittiyse en son) mikroservis tabanlı projenizde tasarımı göstermek için hangi notasyonları kullandınız?		✓
6	Şu anki (eğer bittiyse en son) mikroservis tabanlı projenizde hangi tasarım kalıplarını (pattern) kullandınız?		✓

Anket, mikroservis mimarisi ile ilgili 6 bölümden oluşmaktadır. İlk bölüm anket ile ilgili bilgi vermekte ve

katılımcının onayını almaktadır. İkinci bölüm katılımcılar ve çalıştıkları organizasyonlar hakkında bilgi alma amaçlı demografik sorular içermektedir. Bu bölümün son sorusu katılımcının mikroservis tecrübesi olup olmadığını sormaktadır. Katılımcı bu soruda tecrübesi olmadığını belirttiği taktirde anket sona ermektedir. Üçüncü bölümde katılımcının mikroservis tecrübesi ile ilgili genel sorular sorulmaktadır. Dördüncü bölüm yazılım ölçümü ve efor kestirimi ile ilgili sorular içermektedir. Beşinci bölümde yazılım analizi ve altıncı bölümde is yazılım tasarımı ile ilgili sorular yer almaktadır. Bu çalışmada mikroservis tabanlı projelerde yazılım analizi ve tasarımı bölümleri ile ilgili sonuçlar ele alınacak olup bu iki bölümde yer alan sorular Tablo 1’de gösterilmektedir.

C. Doğrulama Kriterleri

Anketin hedef grubu mikroservis tecrübesi olan katılımcılar olduğu için mikroservis tabanlı proje geliştirmemiş katılımcıların cevapları sonuçlara yansımamalıdır. Bu nedenle, ikinci bölümde yer alan yedinci soruda katılımcının mikroservis tecrübesi olup olmadığı sorulmuştur. Bu soruya olumsuz cevap veren katılımcılarda anket sonlandırılmış ve diğer bölümlerdeki soruları yanıtlaması engellenmiştir.

IV. ANKET SONUÇLARI

Bu anket çalışmasına yazılım sektöründe çalışan 46 kişi katılmıştır, ancak bu katılımcıların 29’unun mikroservis ile yazılım geliştirme deneyimi bulunmaktadır ve bu başlık altında sadece deneyimi olan 29 katılımcının sonuçları göz önünde bulundurulacaktır.

Katılımcılarımızın lisans eğitimleri; Bilgisayar Mühendisliği, Yazılım Mühendisliği, Bilgisayar Bilimi, Matematik Mühendisliği, Elektrik ve Elektronik Mühendisliği ve Deniz Ulaştırma Mühendisliği, bölümlerinde iken, yüksek lisans eğitimleri; Bilgisayar Mühendisliği, Bilgisayar Bilimi, Bilgi Sistemleri, Elektrik ve Elektronik Mühendisliği ve Endüstri Mühendisliği bölümlerindedir. Katılımcılarımızın organizasyondaki rolleri ise; Yazılım Geliştirici (Developer), Kıdemli Yazılım Geliştirici (Senior Developer), Yazılım Geliştirme Yöneticisi, Yazılım Mimarı, Proje Yöneticisi ve Teknolojiden Sorumlu Yönetici (CTO)’dur. Katılımcılarımızın mevcut rollerindeki deneyimleri 1 yıldan daha az süre ile 15 yıl arasında değişmektedir ve ortalama 4 yıldır. Yazılım sektöründeki deneyimleri ise 1 yıl ile 30 yıl arasında değişmektedir ve ortalaması 8 yıldır. Katılımcıların mikroservisler ile olan deneyimleri ise 1 yıl ile 10 yıl arasında değişmektedir ve ortalamaları 2,5 yıldır.

Katılımcıların çalıştıkları son mikroservis tabanlı projelerin bağlı olduğu sektörler; Otomasyon, Müşteri İlişkileri Yönetimi, E-ticaret, İnternet, Lojistik, Finans, Milli Yazılım, Mobil Yazılım, Telekom ve Kamu Hizmetleri’dir. Çalıştıkları son mikroservis projelerinin türlerine bakıldığında ise 13 katılımcı için (%44,8) monolitik mimari ile geliştirilmiş bir yazılımın mikroservis mimarisinde yeniden yazılması, 12 katılımcı için (%41,4) yeni bir mikroservis tabanlı proje yazılması, 3 katılımcı için (%10,3) bir sistemin bazı parçalarını monolitik mimaride bir proje ile beraber çalışmak amacıyla mikroservis olarak yazılması, 1 katılımcı için (%3,4) ise var olan bir mikroservis tabanlı sistemin düzenlenmesi olduğu görülmüştür. Mikroservisler ile çalışılırken başvuru yazılım geliştirme modellerinde

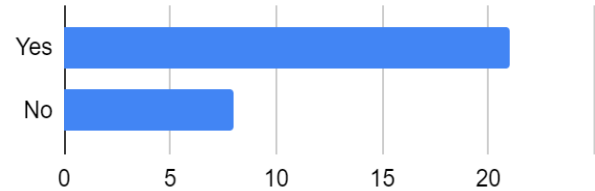
ise; 15 katılımcı (%51,7) sadece Scrum, 4 katılımcı (%13,8) sadece Kanban, 3 katılımcı (%10,3) Kanban ile Scrum, 2 katılımcı (%6,9) Kanban ile Ekstrem Programlama (EP), 2 katılımcı (%6,9) Kanban, Scrum ve EP, 2 katılımcı (%6,9) genel olarak Çevik (Agile) yazılım geliştirme, 1 katılımcı (%3,4) ise Şelale (Waterfall) modelini takip etmiştir. Katılımcıların çalıştıkları son mikroservis tabanlı projedeki personel sayısı minimum 2, maksimum 50, ve ortalama 9~kişidir, organizasyonlarında paralel olarak devam eden mikroservis projesi sayısı ise; minimum 1, maksimum 75, ve ortalama 13’tür. Katılımcılarımızın çoğunluğu (58.6%) aynı anda birden fazla mikroservis projesinde çalışmaktadır.

Katılımcıların organizasyonlarının çoğunluğunda (%82,8) DevOps uygulanmaktadır ve Jenkins en çok tercih edilen (%47,9) entegrasyon aracıdır. Mikroservis projelerinin test otomasyonuna bakıldığında ise, katılımcıların çoğunluğu (%37,9) 1-5 arası Likert ölçeğinde 4 olarak oylamıştır. Mikroservis tabanlı mimariler karşılaşılan en kritik problem sorulduğunda ise; katılımcıların çoğunluğu (%41,4) geçiş süreçlerin yönetimini, peşinden ise monolitik tabanlı bir yazılımı parçalamanın zorluğunu (29.3%), asenkron iletişimin zorluğunu (%18,9), veri tabanı ayrımlarının zorluğunu (%6,9), ve sürüm çıkılırken karşılaşılan kompleks adımların zorluğunu (%3,4) belirtmiştir.

A. Yazılım Analizi

Bu başlık altında anketin yazılım analizi ile ilgili olan sonuçları verilmektedir. Verilen cevapları gösteren grafikler o seçeneğe cevap veren katılımcı sayısını göstermektedir.

Bu bölümde katılımcılarımızın cevapladığı ilk soru organizasyonlarında yazılım analizi için standart bir yazılım sürecinin olup olmadığıdır. Sonuçlar organizasyonların çoğunluğunun (%72,4) standart bir yazılım analiz sürecini takip ettiğini göstermektedir (Şekil. 1).



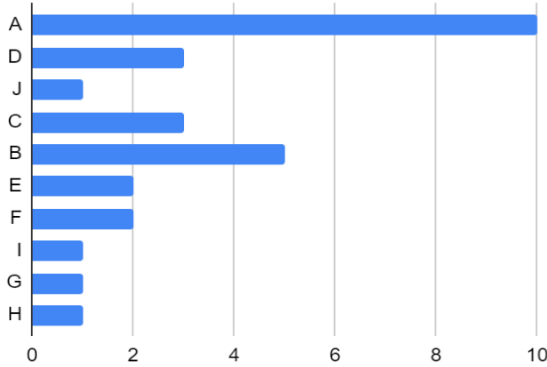
Şekil. 1. Yazılım analizi için standart bir süreç kullanılıyor mu?

İkinci soruda mikroservis tabanlı projelerde fonksiyonel gereksinimlerin nasıl gösterildiği sorulmuştur. Sonuçlar %34,5’nin sadece kullanıcı hikayesi (user story) (A), %17,2’nin sadece doğal dili (B), %10,3’ün kullanıcı hikayesi ve doğal dili beraber kullandığını (C), %10,3’ün kullanıcı hikayesi ve kullanım durumu (use case) senaryolarını beraber kullandığını (D), %6,9’un sadece kullanım durumu senaryolarını (E), %6,9’un kullanıcı hikayesi ve kullanım durumu diyagramlarını beraber kullandığını (F), %3,4’ün kullanıcı hikayesi, kullanım durumu senaryoları ve kullanım durumu diyagramlarını beraber kullandığını (G), %3,4’ün kullanıcı hikayesi, kullanım durumu senaryoları, kullanım durumu diyagramları ve docstring türü kod dokümantasyonlarını beraber kullandığını (H), %3,4’ü kullanıcı hikayesi ve kullanım durumu senaryolarını beraber kullandığını (I), %3,4’ün ise kullanıcı hikayesi ve Event Process Chain (EPC) kullandığını (J) göstermiştir (Şekil. 2).

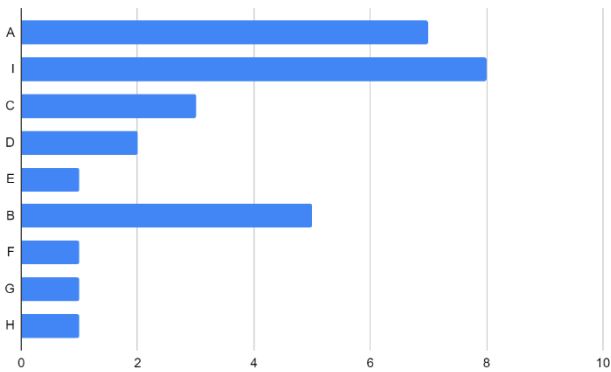
Sonuçlara göre, başka yöntemler ile beraber kullanılsa bile kullanıcı hikayesi yönteminin fonksiyonel

gereksinimlerin belirtilmesinde organizasyonlar arasında en tercih edilen yöntem olduğu söylenebilir.

Üçüncü soruda katılımcılara mikro servis tabanlı projelerde problemi nasıl analiz ettikleri sorulmuştur. Sonuçların %24,1'i olay fırtınası (event storming) (A), %17,2'si Ad-hoc (B), %10,3'ü olay tabanlı (event-based) modellemeyi (C), %6,9'u olay fırtınası ve olay tabanlı modellemeyi (D), %3,4'ü Ad-hoc, olay fırtınası, ve olay tabanlı modellemeyi (E), %3,4'ü Ad-hoc ve olay fırtınası (F), %3,4'ü Ad-hoc ve olay tabanlı modellemeyi (G), %3,4'ü olay fırtınası, olay tabanlı model ve İş Süreci Model ve Gösterim (Business Process Model and Notation (BPMN)) (H), %27,6'sı ise hiçbirini (I) seçmiştir (Şekil. 3). Bu soruda hiçbiri seçeneğini seçen katılımcıların bu cevap ile problem analizi yapmadıkları mı yoksa kendileri için başka takımların mı problem analizi yaptığını bilmesek de en yüksek oranı bu cevabın almış olması dikkat çekicidir.



Şekil. 2. Şu anki (eğer bittiye en son) mikro servis tabanlı projenizde fonksiyonel gereksinimlerini nasıl gösterdiniz?

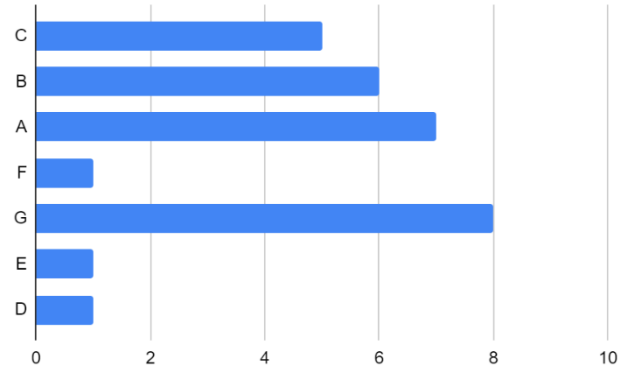


Şekil. 3. Şu anki (eğer bittiye en son) mikro servis tabanlı projenizde problemi nasıl analiz ettiniz?

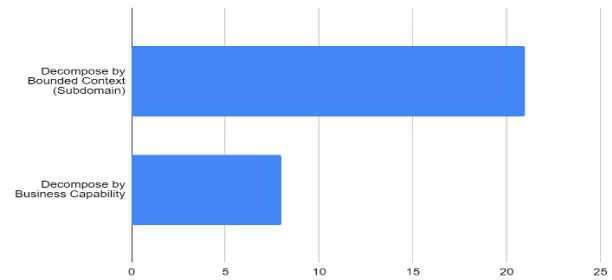
Dördüncü soruda katılımcıların mikro servis tabanlı projelerin analizinde kullandıkları notasyonlar sorulmuştur. Katılımcıların %24,1'i sadece akış şeması (flow chart) (A), %20,7'si akış şeması ve aktivite diyagramı (B), %17,2'si sadece aktivite diyagramı (C), %3,4'ü akış şeması ve zihin haritası (mindmap) (D), %3,4'ü akış şeması, Aktivite diyagramı, değer akışı (value stream) haritaları ve BPMN (E), %3,4'ü eEPC (F), %27,6'sı ise hiçbir (G) notasyon kullanmadıklarını belirtmiştir (Şekil. 4). Önceki soruda olduğu gibi hiçbiri seçeneğini seçen katılımcıların bir notasyon kullanılmadığını mı yoksa kendilerinin kullanmadıklarını belirttiklerini bilmesek de mikro servis tabanlı projelerin analizleri sırasında bir notasyona ihtiyaç

duyulduğunda akış şemasının popüler bir tercih olduğu söylenebilir.

Son olarak katılımcılara problemi mikro servislere nasıl ayrıştırdıkları sorulmuştur. Sonuçlar bize organizasyonların çoğunlukla mikro servislere sınırlı bağlama (bounded context) göre ayrıştırdıklarını (%72,4) ve bazı organizasyonların ise ayrıştırmayı iş yeteneklerine (business capability) göre (%27,6) gerçekleştirdiklerini göstermiştir (Şekil. 5).



Şekil. 4. Şu anki (eğer bittiye en son) mikro servis tabanlı projenizde analiz için hangi notasyonları kullandınız?



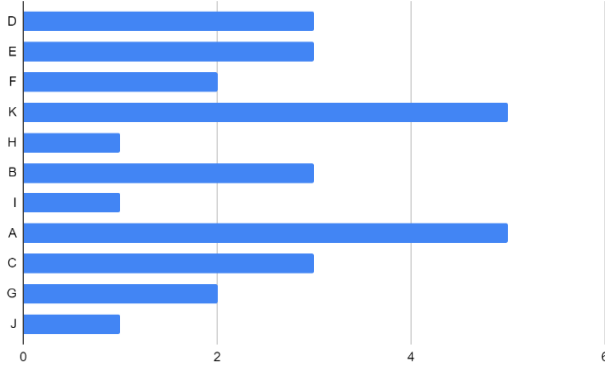
Şekil. 5. Şu anki (eğer bittiye en son) mikro servis tabanlı projenizde problemi nasıl mikro servislere ayrıştırdınız?

B. Yazılım Tasarımı

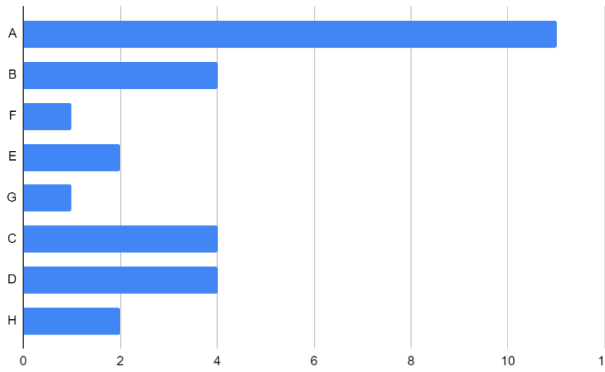
Yazılım tasarımı ile ilgili olarak katılımcılara ilk soruda mikro servis tabanlı projelerinde tasarımı göstermek için hangi notasyonları kullandıkları sorulmuştur. Sonuçların %17,2'si sıralama (sequence) diyagramı (A), %10,3'ü sınıf diyagramı (B), %10,3'ü sıralama ve sınıf (class) diyagramı (C), %10,3'ü sınıf, sıralama ve aktivite (activity) diyagramı (D), %10,3'ü sınıf, sıralama, aktivite ve varlık ilişki modeli (Entity Relationship Diagram (ERD)) (E), %6,9'u sıralama ve aktivite diyagramı (F), %6,9'u sınıf diyagramı ve ERD (G), %3,4'ü aktivite diyagramı (H), %3,4'ü ERD (I), %3,4'ü sınıf, sıralama diyagramları ve ERD (J), %17,2'si ise hiçbirinin (K) kullanılmadığını göstermiştir (Şekil. 6). Bu cevaplardan organizasyonlar arasında yazılım tasarımı için bir diyagrama ihtiyaç duyulduğunda sıralama ve sınıf diyagramlarının tercih edilen bir yansıtma sistemi olduğu çıkarılabilir.

İkinci soruda ise, katılımcılara son mikro servis tabanlı projelerinde kullandıkları tasarım kalıpları (design pattern) sorulmuştur. Sonuçların %27,6'sı devre kesici (circuit breaker) (A), %24,1'i olay kaynağını belirleme düzeni (event sourcing) (B), %13,8'i komut ve sorgu sorumluluğu ayrımı (Command Query Responsibility Segregation (CQRS)) (C), %13,8'i devre kesici, olay kaynağını belirleme düzeni ve CQRS (D), %6,9'u olay kaynağını belirleme düzeni ve CQRS

(E), %3,4'ü devre kesici ve bölme perdesi düzeni (bulkhead) (F), %3,4'ü API ağ geçidi (gateway), servis paylaşımı veri tabanı (shared database per service), log birleştirme (aggregation), performans metrikleri, dağıtılmış izleme (distributed tracing) (G), %6,9'u ise hiçbirini (H) kullanmadıklarını göstermiştir (Şekil. 7). Bu sonuçlar devre kesici modelinin mikroservis tabanlı mimariler sıkça tercih edilen bir model olduğunu, peşinden de olay kaynağını belirleme düzeni modelinin geldiğini göstermiştir.



Şekil. 6. Şu anki (eğer bittiyse en son) mikroservis tabanlı projenizde tasarımı göstermek için hangi notasyonları kullandınız?



Şekil. 7. Şu anki (eğer bittiyse en son) mikroservis tabanlı projenizde hangi tasarım kalıplarını (pattern) kullandınız?

V. TARTIŞMA

Bu çalışmada anketimizin Türkiye sonuçları verilmiştir. Anket sonuçlarına göre, katılımcıların mikroservis tabanlı projelerde analiz ve tasarım için genel olarak başvurdukları bir yaklaşım olmadığını göstermektedir.

Katılımcıların çoğunun çalıştığı organizasyonlarda yazılım analizinde standart bir süreç kullandıklarını belirtmişlerdir. Ancak, bu süreçte başvurdukları yol değişkenlik göstermektedir. Kullanıcı hikayelerinin bu konuda en çok başvurulan yöntem olduğu ve bunun yanında doğal dil ve kullanım durumu (use case) senaryolarının da sıklıkla kullanıldığı görülmektedir. Bu sonuç, nesne tabanlı analiz yöntemlerinin mikroservis tabanlı projelerde de fonksiyonel gereksinimlerin belirlenmesinde kullanıldığını göstermektedir. Katılımcıların mikroservis tabanlı projelerde başvurdukları problem analiz yöntemine bakıldığında ise olay fırtınası (event storming) yönteminin en çok ve ad-hoc path ile olay tabanlı (event-based) modellemenin de sıklıkla kullanıldığı görülmektedir. Problem analizinde olay tabanlı yöntemlerin sıklıkla kullanılmasına rağmen fonksiyonel gereksinimlerin çıkarılmasında nesne tabanlı yöntemlere

başvurulması dikkat çekmektedir. Analiz sürecinde bu organizasyonların nesne tabanlı yöntemler ile başlayıp sonrasında ise olay tabanlı modelleme ile devam etmeleri ve bu geçişi nasıl sağladıkları incelenmesi gereken bir konudur. Analiz için kullanılan notasyonlarda ise yine nesne tabanlı yöntemler (akış şeması ve aktivite diyagramı) kullanıldığı göze çarpmaktadır. Sadece bir katılımcı olay tabanlı modelleme şekli olan eEPC kullandığını belirtmiştir. Analiz konusunda son olarak katılımcıların problemi mikroservisler nasıl ayrıştırdıkları sorulduğunda büyük çoğunluğun sınırlı bağlama (bounded context) göre gerçekleştirdikleri görülmektedir. Bu katılımcıların problemi olay bazlı yöntemlere başvurarak analiz ederken geleneksel olan nesne tabanlı notasyonlar kullanarak göstermesi sonucunda sınırlı bağlamı nasıl sağladıkları ele alınabilecek bir başka konudur.

Mikroservis tabanlı projelerde tasarım nesne tabanlı projelere göre farklılık göstermektedir. Bunun en önemli nedeni, mikroservislerde sınırlı bağlamın nesnelere ile sağlanmamasıdır. Sonuçlara bakıldığında, nesne tabanlı tasarımın en önemli gösterim şekillerinden biri olan sınıf diyagramlarının mikroservis tabanlı projelerde de sıklıkla kullanıldığı görülmektedir. Ancak, mikroservis mimarisinde sınıf kavramı ve bu sınıfların arasındaki ilişki artık önemini yitirmekte ve farklı gösterim şekillerine ihtiyaç duyulmaktadır. Kullanılan tasarım kalıplarına bakıldığında ise yine olay tabanlı bir kalıp olan olay kaynağını belirleme düzeni (event sourcing) ikinci sırada görülmektedir. Bunun yanında kapalı devre (circuit breaker) en çok kullanılan tasarım kalıbıdır. Analiz ile paralel olarak, tasarım sürecinde de yine nesne tabanlı ve olay tabanlı yöntemlerin birlikte kullanıldığı görülmektedir. Ancak, nesne tabanlı gösterimden olay tabanlı tasarım kalıplarına nasıl geçiş yapıldığı yine incelenmesi gereken bir konudur.

Sonuç olarak, anketin Türkiye sonuçları organizasyonların, mikroservis tabanlı projelerde bir süreç izlediği ancak bu sürecin organizasyonlara göre farklılaştığı görülmektedir. Ancak, nesne tabanlı mimaride kullanılan yöntemler gibi kalıplaşmış bir yöntem bulunmamaktadır. Nesne tabanlı ve olay tabanlı yöntemler ile farklı kombinasyonlar kullanıldığı görülmektedir.

Literatür ile sektörü karşılaştırdığımızda; Alshuqayran ve diğerleri [7] tarafından yapılan çalışmada, mikroservis mimarisi için en çok kullanılan diyagramların bileşen (component), sıralama (sequence), süreç (process), dağılım (deployment), sınıf (class) ve kullanım durumu (use case) diyagramları olduğu belirtilmektedir. Anket sonuçları da sektörde bu gösterimlerin sıklıkla kullanıldığını, nesne tabanlı yöntemlere yine sıklıkla başvurulduğunu göstermekte ve literatür ile paralel bir sonuç göstermektedir.

VI. SONUÇ

Mikroservis tabanlı mimari, yazılım geliştirmede popüler ve etkili bir yol olarak değerlendirilmektedir. Ancak, mikroservis yeni bir konsept olup yazılım organizasyonlarının kültürlerini değiştirmelerini gerektirmektedir. Yazılım analizi ve tasarımı teknikleri mikroservis tabanlı projelerde farklılaşmaktadır. Literatürde, mikroservis tabanlı projelerde analiz ve tasarım süreçlerini konu alan çalışmalarda eksiklik görülmektedir.

Bu çalışma kapsamında, yazılım sektöründeki organizasyonların mikroservis tabanlı proje geliştirirken başvurduğu analiz ve tasarım tekniklerini analiz etmek amaçlı bir anket düzenlenmiş ve bu anketin sonuçları

organizasyonların bu konudaki tecrübelerini ortaya çıkarmak amacı ile sunulmuştur. Ankete Türkiye’den 46 kişi katılmıştır. Sonuçlar, ankete katılan organizasyonlarda mikroservis tabanlı projelerin analiz ve tasarımı için genel olarak benimsenen tutarlı bir yaklaşım olmadığını göstermektedir. Ankete katılan organizasyonlarda geleneksel nesne tabanlı analiz ve tasarım yöntemlerinin mikroservis tabanlı projelerin geliştirilmesinde de ağırlıklı olarak kullanıldığı görülmektedir. Ancak, geleneksel yöntemlerin mikroservis tabanlı projelerin analiz ve tasarım sürecinde ne derecede etkili olduğu tartışılması gereken önemli bir konudur.

Anket başka ülkelerden katılımcılar ile de devam etmekte olup, gelecek planlarında daha kapsamlı bir analiz planlanmaktadır. Ayrıca, bu makalede yer almayan yazılım ölçümü ve efor kestirimi sonuçlarının da ayrı bir çalışma kapsamında tartışılması planlanmıştır. Anket tamamlandığında, istatistiksel çalışmaların yapılması ve sonuçlar arasında korelasyon çıkarılması planlanmaktadır.

KAYNAKÇA

- [1] A. R. Sampaio *et al.*, “Supporting Microservice Evolution,” in *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, Sep. 2017, pp. 539–543, doi: 10.1109/ICSME.2017.63.
- [2] J. Thönes, “Microservices,” *IEEE Softw.*, vol. 32, no. 1, pp. 116–116, Jan. 2015, doi: 10.1109/MS.2015.11.
- [3] N. Dragoni *et al.*, “Microservices: Yesterday, Today, and Tomorrow,” in *Present and Ulterior Software Engineering*, M. Mazzara and B. Meyer, Eds. Cham: Springer International Publishing, 2017, pp. 195–216.
- [4] A. Dikici, O. Turetken, and O. Demirors, “Factors influencing the understandability of process models: A systematic literature review,” *Inf. Softw. Technol.*, vol. 93, pp. 112–129, Jan. 2018, doi: 10.1016/j.infsof.2017.09.001.
- [5] C. Pahl and P. Jamshidi, “Microservices: A Systematic Mapping Study,” in *CLOSER (1)*, 2016, pp. 137–146.
- [6] P. Di Francesco, P. Lago, and I. Malavolta, “Architecting with microservices: A systematic mapping study,” *J. Syst. Softw.*, vol. 150, pp. 77–97, Apr. 2019, doi: 10.1016/j.jss.2019.01.001.
- [7] N. Alshuqayran, N. Ali, and R. Evans, “A Systematic Mapping Study in Microservice Architecture,” in *2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA)*, Nov. 2016, pp. 44–51, doi: 10.1109/SOCA.2016.15.
- [8] D. Taibi, V. Lenarduzzi, and C. Pahl, “Architectural patterns for microservices: a systematic mapping study,” SCITEPRESS, 2018.
- [9] V. Garousi, A. Coşkunçay, and O. Demirörs, “A survey of software testing practices in Turkey,” in *2013 Turkish National Software Engineering Symposium (Ulusal Yazılım Mühendisliği Sempozyumu, UYMS)*, Sep. 2013.
- [10] V. Garousi, A. Coşkunçay, A. Betin-Can, and O. Demirörs, “A survey of software engineering practices in Turkey,” *J. Syst. Softw.*, vol. 108, pp. 148–177, Oct. 2015, doi: 10.1016/j.jss.2015.06.036.
- [11] D. Akdur, V. Garousi, and O. Demirörs, “A survey on modeling and model-driven engineering practices in the embedded software industry,” *J. Syst. Archit.*, vol. 91, pp. 62–82, Nov. 2018, doi: 10.1016/j.sysarc.2018.09.007.
- [12] F. Shull, J. Singer, and D. I. Sjøberg, *Guide to advanced empirical software engineering*. Springer, 2007.