



# Bir Ejderhadan Yazılım Test Dersi

Kari Kakkonen

5.2.2021

Version 1.0



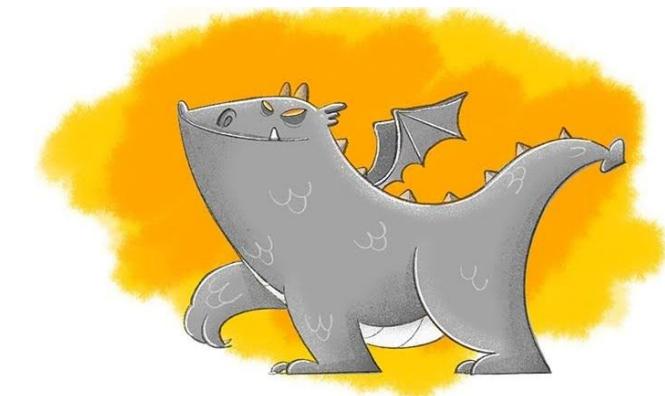
# Sunum hakkında

# Sunum hakkında

- Sunumda, bilgi teknolojileri, kodlama ve yazılım testiyle ilgili konular, fantastik hikayeler kullanılarak anlatılmaktadır.
- Sunum, okullarda 1 veya 2 derste bilgi teknolojilerinin öğretimini desteklemek için kullanılabilir.
- Sunum Creative Commons lisanslıdır, yani ticari olmayan amaçlar için ücretsiz olarak kullanabilir.
- Sunumun en güncel İngilizce haline aşağıdaki adresten erişebilirsiniz:
  - <https://www.dragonsout.com/p/presentation-for-teachers.html>
- Geribildirimlerinizi aşağıdaki adrese gönderebilirsiniz:
  - [feedback@dragonsout.com](mailto:feedback@dragonsout.com)
- Haydi hep birlikte yazılım testini yeni kuşaklara tanıtalım!



Ejderhalar Dışarı - Bir Ejderhadan Yazılım Test Dersi Kari Kakkonen tarafından hazırlanan PowerPoint Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License lisanslıdır.



# “Ejderhalar Dışarı!” Kitabı Hakkında

- Bu sunum, Ejderhalar Dışarı! kitabındaki dünyaya ve karakterlere dayanmaktadır.
- Kitap, sunumda değinilen konularla ilgili öğrenme deneyimini tamamlayıcı niteliktedir.
- Yazar: Kari Kakkonen
- İllüstratör: Adrienn Széll
- Türkçe'ye kazandıran: Turkish Testing Board
- [www.turkishtestingboard.org](http://www.turkishtestingboard.org)
- Metin ve çizim hakları: Dragons Out Oy
- Daha fazla bilgi için: [www.dragonsout.com](http://www.dragonsout.com)



# Turkish Testing Board Hakkında

- Turkish Testing Board, 2006 yılında ISTQB®'a bağlı olarak kurulduğundan bu yana, Türkiye'deki bilişim profesyonellerinin yazılım testi alanında bilgilendirilmesi ve sertifikalandırılması için kar amacı gütmenden çalışmalarını sürdürmektedir.
- ISTQB® sınavlarının düzenlenip başarılı olan profesyonellerin sertifikalandırılmasının yanında, her sene TestIstanbul konferansını düzenleyerek sektör profesyonellerinin bir araya gelmesine öncülük etmektedir.
- Bunun yanında paneller ve seminerler düzenleyerek, uluslararası raporlar yayinallyarak ve çeviri çalışmalarıyla yazılım test terminolojisini Türkçe'ye kazandırarak yazılım test uzmanlarının bilgilendirilmesi görevini üstlenmektedir.
- Pek çok üniversite öğrencisine burs imkanı sağlamaktadır.



# Alıştırmalar hakkında

- Çizim alıştırmaları
  - Gerekli malzemeler: kalem ve kağıt.
  - Çizimlerin fotoğrafını çekebilir ve bir uygulamaya yükleyebilirsiniz,örneğin:  
<https://padlet.com>
    - Öğretmenin, ders için Padlet panosu oluşturması gereklidir.
  - Fotoğrafları çekmek için bir cep telefonuna ihtiyacınız olacak.
- Test etme alıştırmaları
  - Kendi cep telefonunuza veya tabletinize, ve test etmek istediğiniz herhangi bir uygulamaya ihtiyacınız olacak.
    - Telefonunuzu bir arkadaşınızla paylaşabilir ve alıştırmayı birlikte yapabilirsiniz.
- Alıştırmaları 5 dakikada hızlıca yapabilir veya istediğiniz kadar vakit ayırabilirsiniz.



# Ejderhalar / Hatalar

# Yazılım nedir?

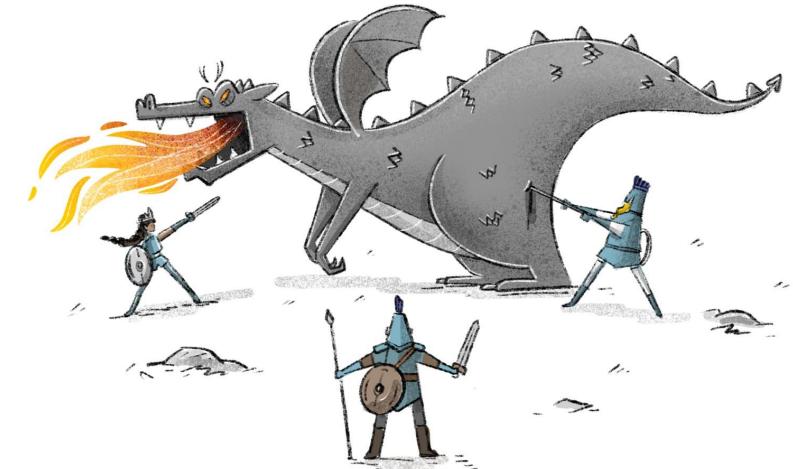
- Her yerde yazılım var, oyunlarda, online alışveriş sitelerinde, arabalarda, tipki fantastik dünyalarda her yerde şatolar ve köyler olduğu gibi.
- Yazılımlar aynı zamanda cihazları çalıştırabilirler.
- Yazılımlar birden fazla programdan oluşurlar.

Yazılım = kullanıcının bir şey yapmasına, mesela oyun oynamasına imkan veren bilgisayar kodudur.



# Hata nedir?

- Hata yapmak insana özgü olduğu için, her yazılımda hata bulunur.
- Fantastik hikayelerde ejderhaların şatolara ve şato sakinlerine zarar verdiği gibi, hatalar da yazılımların kullanımını zorlaştırmır, onlara zarar verirler.
- Hatalar, kazara yazılımlara yerleşirler ve tercihen bir kullanıcı onunla karşılaşmadan önce bulunup çıkarılmaları gereklidir.
- Hata genellikle “bug” (böcek) olarak adlandırılır.



Hata = Yazılımda arızaya neden olan yazılım kodundaki bir sorundur. İnsanlar buna bazen “error” derler, “error” en doğru tanımı kodda “hataya” neden olan insan eylemidir.

# 5-15 dakikalık alıştırma

## Kendi hatanızı tasarlayın – kendi ejderhanızı çizin

### // İhtiyacınız olan şeyler

Kalem ve kağıt



### // Görev

- 1 **Karşılaştığınız bir hatayı düşünün**
  - Hatanın adını yazın ve birkaç sözcükle tasvir edin.
- 2 **Ona eşdeğer bir ejderha düşünün.**
  - Ejderhanın özelliklerini yazın.
  - Hata ne kadar büyükse, ejderha da o kadar büyük olmalı, gibi.
- 3 **Ejderhayı çizin.**
  - Düşüncenizi tasvir ederken esasen ejderhanın hatayı nasıl temsil ettiğine odaklanın. Kusursuz bir çizim olmasına gerek yok.
- 4 **Çiziminizi arkadaşlarınıza gösterin.**
  - Örneğin Padlet kullanın.

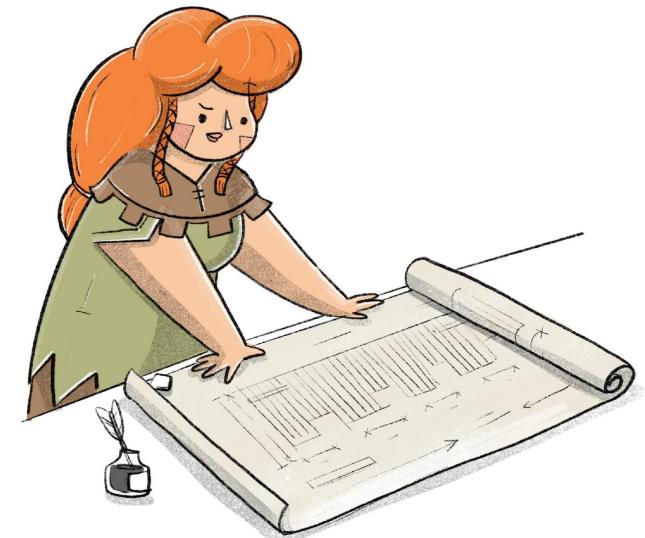


# **Yazılım testi nedir? Yazılım geliştirme nedir?**

# Yazılım geliştirme nedir?

- Öncelikle yazılımın ne yapmasını istediğinizize karar vermelisiniz.
- Kodlamalısınız.
- Yazılımın nasıl çalıştığını test etmelisiniz.
- Aynı bir fantastik hikayedeki gibi köyün etrafına çekteceğiniz duvarı planlamalı ve inşa etmelisiniz, aynı zamanda ona zarar veren ejderhayı da alt etmelisiniz.

Yazılım geliştirme = gereksinimlerin tanımlanması, kodlama ve test etme gibi çeşitli aktiviteleri içermektedir.



# Yazılım testi nedir?

Test, hataları aramak ve bulmaktır, tipki bir fantastik hikayede duvarlardan ejderhayı gözlemek gibi.

- Düzeltilebilmesi için hatayı öncelikle bulmanız gereklidir.
- Yazılımı test ederek hataları bulmaya çalışırsınız.



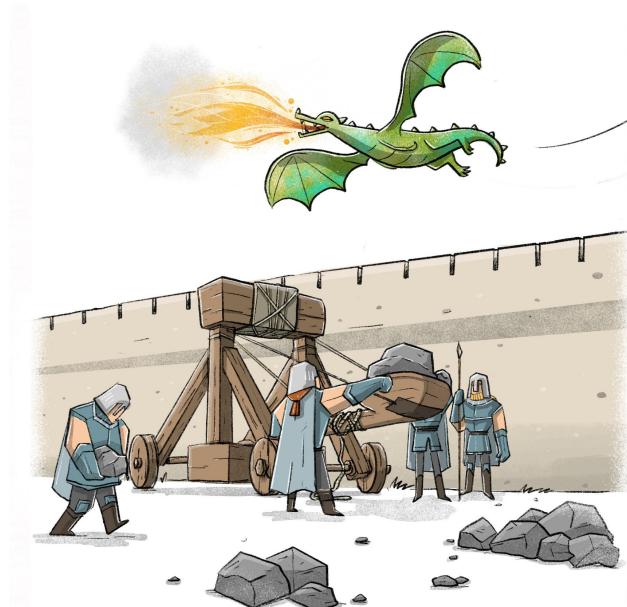
Test etmek = bir yazılımın istenildiği gibi çalışıp çalışmadığını farklı senaryo, değer ve yollarla denemektir.

# 5-15 dakikalık alıştırma

## Kendi yazılımınızı tasarlayın - kendi şatonuzu çizin

### // İhtiyacınız olan şeyler

Kalem ve kağıt



### // Görev

- 1 Kullandığınız bir yazılımı düşünün.**
  - Yazılımın adını yazın
  - Yazılımı birkaç kelimeyle tanımlayın.
  - Bu yazılımın hataları nasıl tolere ettiğini yazın.
- 2 Bir ejderha saldırısına dayanabilecek güchte bir şato düşünün.**
  - Şatonun özelliklerini yazın.
  - Örneğin küçük, büyük? Duvarlar? Silahlar?
- 3 Şatoyu çizin.**
  - Düşüncenizi gerçekleştirirken esasen şatonun yazılımı nasıl temsil ettiğine odaklanın.
  - Kusursuz bir çizim olmasına gerek yok.
- 4 Çiziminizi arkadaşlarınıza gösterin.**
  - Örneğin Padlet kullanın.

# Test teknikleri

- Yazılımı farklı yollarla test etmenin yöntemi çeşitli test teknikleri kullanmaktadır.
  - Yazılımı geçerli girdi ve senaryolarla kullanmayı deneyin.
  - Yazılımı geçersiz girdi ve senaryolarla kullanmayı deneyin.
  - Yazılımın nasıl çalıştığını keşfetmeye çalışın.
  - Başka biri kullanırken yazılımı gözlemleyin.
- Aynı anda pek çok test tekniğini kullanabilirsiniz, tıpkı fantastik bir hikayede şövalyenin hem harita kullanıp hem de insanlara ejderhayı en son nerede gördüklerini sorması gibi.



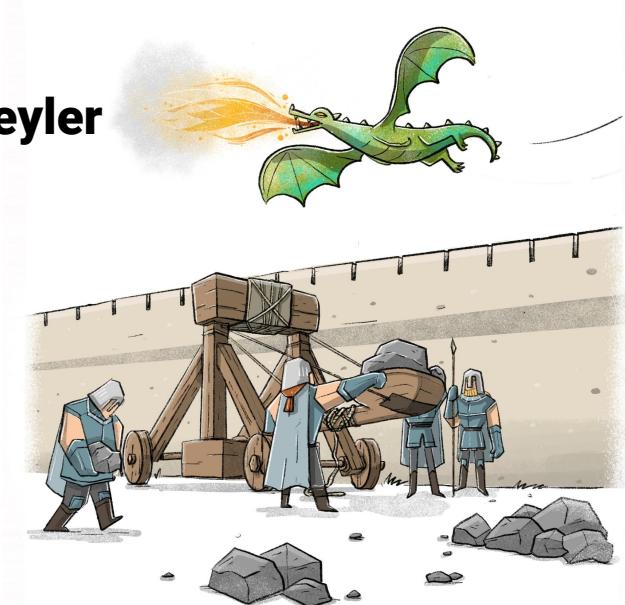
Test teknigi = yazılımın farklı yönlerini test etmenizi sağlayacak yol ve yöntemler. Farklı ihtiyaçlar için çok çeşitli test teknikleri bulunur.

# 5-30 dakikalık alıştırma

Cep telefonunuzda bir uygulamayı test edin

## // İhtiyacınız olan şeyler

Cep telefonu,  
tablet  
veya bilgisayar



## // Görev

- 1 Bir yazılım seçin.
  - Örneğin; oyun.
- 2 Yazılımı farklı açılardan düşünün.
  - Ne doğru çalışıyor?
  - Ne yanlış çalışıyor?
  - Ne yavaş?
  - Ne garip?
- 3 Bu açıları göz önünde bulundurarak yazılımı kullanın.
  - Bulduklarınızı yazın.
- 4 Bulduklarınızı arkadaşlarınızla paylaşın.
  - Tartışın!



# **Şövalye - Yazılımcı ve test uzmani**

# Kim daha çok test yapar?

- Genellikle yazılımcı yazılımı kodlar ve kodlarken test de eder.
- Aynı zamanda test etmekte uzmanlaşmış test uzmanları da vardır. Onlar ise daha çok hata bulma üzerine uzmanlaşmışlardır.
- Yazılımcılar, iş analistleri ve test uzmanları birlikte yazılım geliştirme ekibini oluşturur, fantastik bir hikayede pek çok şövalye olduğu gibi.



Yazılım geliştirme ekibi = birlikte iş ihtiyacını analiz edip yazılımı geliştiren ve test eden bir grup profesyonel.

# Ejderha geldiğinde, ihtiyacın olacak...

## Hikaye

- Kuğugölü atını çevirdi ve hızla çitlere doğru sürdü. Şövalyelere ve taş ustası Aidan'a ejderhanın geldiğini söyledi. "Tüm kazıkların acilen çitlerdeki geçide taşınması gerekiyor. Kimde mızrak ve kılıç varsa, getirmeli. Mevcut tüm su kovalara dökülmeli." dedi. Sonra da Şatoya Sarışakal'ı bulmaya gitti.



## Açıklama

- Hikayede ejderha, çitlerin onarımı esnasında köye gelir. Aynı şekilde, yazılımdaki çoğu hata yazılım geliştirme esnasında, canlıya alınmadan önce bulunur. Bu zaman aralığında, hata bulmaya çalışan profesyoneller (test uzmanları) ve hatayı düzeltten profesyoneller (yazılımcılar) bulunmaktadır. Genellikle bir test uzmanı hatayı bulur, böylece bir kullanıcının daha sonra bu hatayı bulması önlenmiş olur. Bu hikayede Kuğugölü, hatayı yani ejderhayı bulup tanımlayan bir test uzmanıdır. Bir test uzmanı olarak, hatayı düzeltmedi ama yazılımcılara haber verdi.*

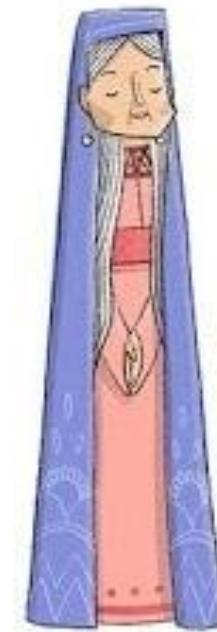


## **Diğer test uzmanları**

# Kullanıcılar

- Herkes test yapabilir ama bekleni, yazılımcıların ve test uzmanlarının en çok test etmesi yönündedir.
- Yazılımı kullanacak olan kullanıcılar da teste katılabilirler, fantastik hikayelerdeki köylüler, çocuklar ve yetişkinler gibi.
- Bu tür testlere kabul testi denir.
- Kullanıcılar aynı zamanda yazılım geliştirmeye de yardım edebilir.

Test uzmanı: test eden kişi. Test etme yetkinliği olan kişi. Testi yapan kişi tam zamanlı bir test uzmanı, yazılımcı, yönetici veya kullanıcı olabilir.



# Destek/Operasyon ekibi

- Destek ekibi, yazılımı izler ve yazılımın çalışır durumda kalmasını sağlar. Kisaca onu desteklerler.
- Bunu yapmak için hem test eder hem de hataları düzeltirler.
- Aynı zamanda kullanıcılara yardım da ederler.
- Destek ekibi, işlerini genellikle kendi başına halletmeye çalışır, ama ihtiyaç duyduklarında yazılımcılardan da yardım isterler, fantastik hikayelerde avcının ejderhayı öldürmek için şövalyeden yardım istemesi gibi.
- Bazen, destek personeli, yazılımcılardan da oluşabilir. (yazılım geliştirme ekibi DevOps ekibi haline gelir.)

Destek/Operasyon ekibi = canlıda kullanıcılar yazılımı kullanırken yazılıma göz kulak olan kişiler.



# DevOps – Sürekli Teslimat

- Bazen, bir operasyon personeli de yazılım geliştirme ekibinde yer alır. Bu durumda, takım DevOps ekibi olur, fantastik hikayelerde avcının şövalyelerle birlikte çalıştığı gibi.
- Bu tür bir ekip, yazılımı her zaman geliştirebilir ve test edebilir, kullanıcılarla yeni işlevler sunar ve aynı zamanda yazılım kullanılırken kullanıcıları destekler.

DevOps = Yazılım geliştirmeyi ve operasyonunu birleştirmektir.



# Ürün sahibi

- Ürün sahibi, yazılım geliştirici ekibe ihtiyaçlarını iletir.
- Bu bir kişi veya bir şirket de olabilir.
- Yazılımin ne yapması gerektiğini tanımlarlar ama yazılım geliştirici ekibi de dinlerler, fantastik hikayelerdeki lord ve leydilerin şövalyelere şato sipariş edip birlikte şatoyu planlamaları gibi.



Ürün sahibi = bir yazılım geliştirilmesini isteyen kişi. Ürün sahibi aynı zamanda bir kişi tarafından temsil edilen bir firma da olabilir.

# Uzman yardımcı

- Yazılım geliştirme ekipleri her şeyi bilmeyebilirler ve farklı alanlardaki uzmanların yardımına ihtiyaç duyar, fantastik hikayelerde bilgelerin köylülere ve şövalyelere yardım etmesi gibi.
- Tipik uzmanlar arasında kullanılabilirlik, güvenlik ve performans uzmanları bulunur. Bu uzmanlar gerektiğinde yazılım geliştirme ekibine yardımcı olurlar.
- Örneğin, kullanılabilirlik uzmanı, sık sık kullanılabilirlik testi gerçekleştirir veya bu konuda diğer ekip üyelerini yönlendirir.

Kullanılabilirlik/Kullanıcı deneyimi uzmanı = yazılımın kullanıcı deneyimini tasarlamakta uzmanlaşmış kişi.

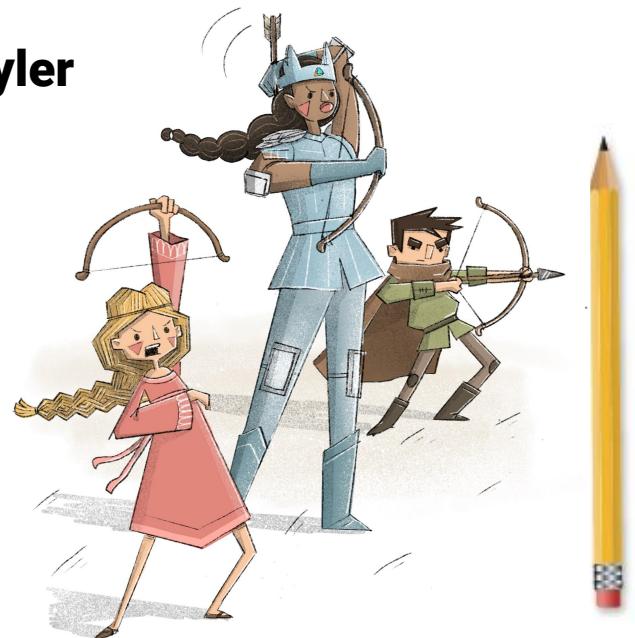


# 5-15 dakikalık alıştırma

Kendi test uzmanınızı  
tasarlayın - Kendi  
şövalyenizi çizin

## // İhtiyacınız olan şeyler

Kalem ve kağıt



## // Görev

- 1 İyi bir test uzmanının nasıl olacağını düşünün.**
  - Meraklı? Yakışıklı? Hızlı? Sabırlı?
  - Test uzmanının özelliklerini yazın.
- 2 Ejderhaları yakalayabilecek bir şövalye düşünün.**
  - Şövalyenin özelliklerini yazın.
  - Zırhlı? Silahlı? Dikkatli?
- 3 Şövalyeyi çizin.**
  - Tasarımınızı yaparken şövalyenin test uzmanını nasıl yansıtğına odaklanın.
  - Kusursuz bir çizim olmasına gerek yok.
- 4 Tasarımınızı arkadaşlarınıza gösterin.**
  - Örneğin Padlet kullanın.



## Farklı türde hatalar

# Farklı türde hatalar vardır

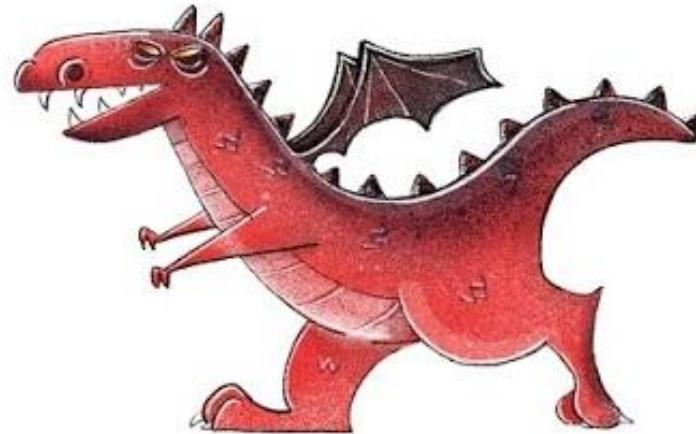
- Tüm hataları bulamazsınız ama hata bulmak için uğraşmaya devam etmelisiniz.
- Bazı hatalar önemlidir, bazıları küçüktür, bazılarını kolaylıkla düzeltbilirsiniz, tipki fantastik bir hikayede farklı türde ejderhalar olduğu gibi.
- Önemli bir hata hemen düzeltilmelidir.

Önem = hataları sıklıkla önemine göre sınıflandırabilirsiniz. Bir hata kritik olabilir, bu da masraflı veya çok zarar vereceği anlamına gelir.



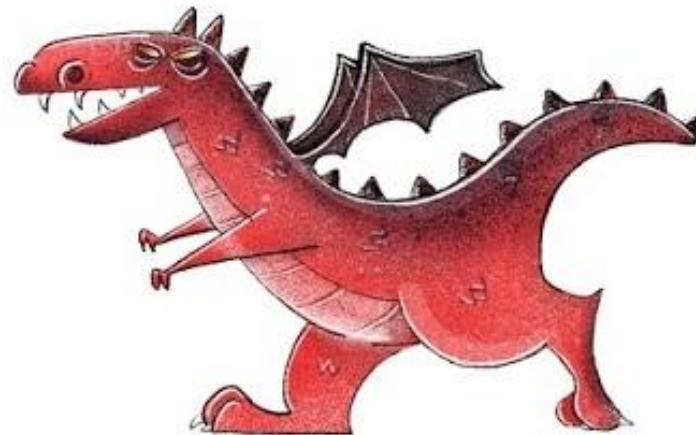
# Gıcık ejderha

- **Renk:** Kırmızı
- **Boyut:** Orta
- **Bulunması:** Zor
- **Kurtulunması:** Kolay
- **Uçabilir mi?:** Hayır
- **Kanatları nasıl?:** Küçük
- **Alev üfler mi?:** Evet
- **En sevdiği şey:** Kuzuları yemek



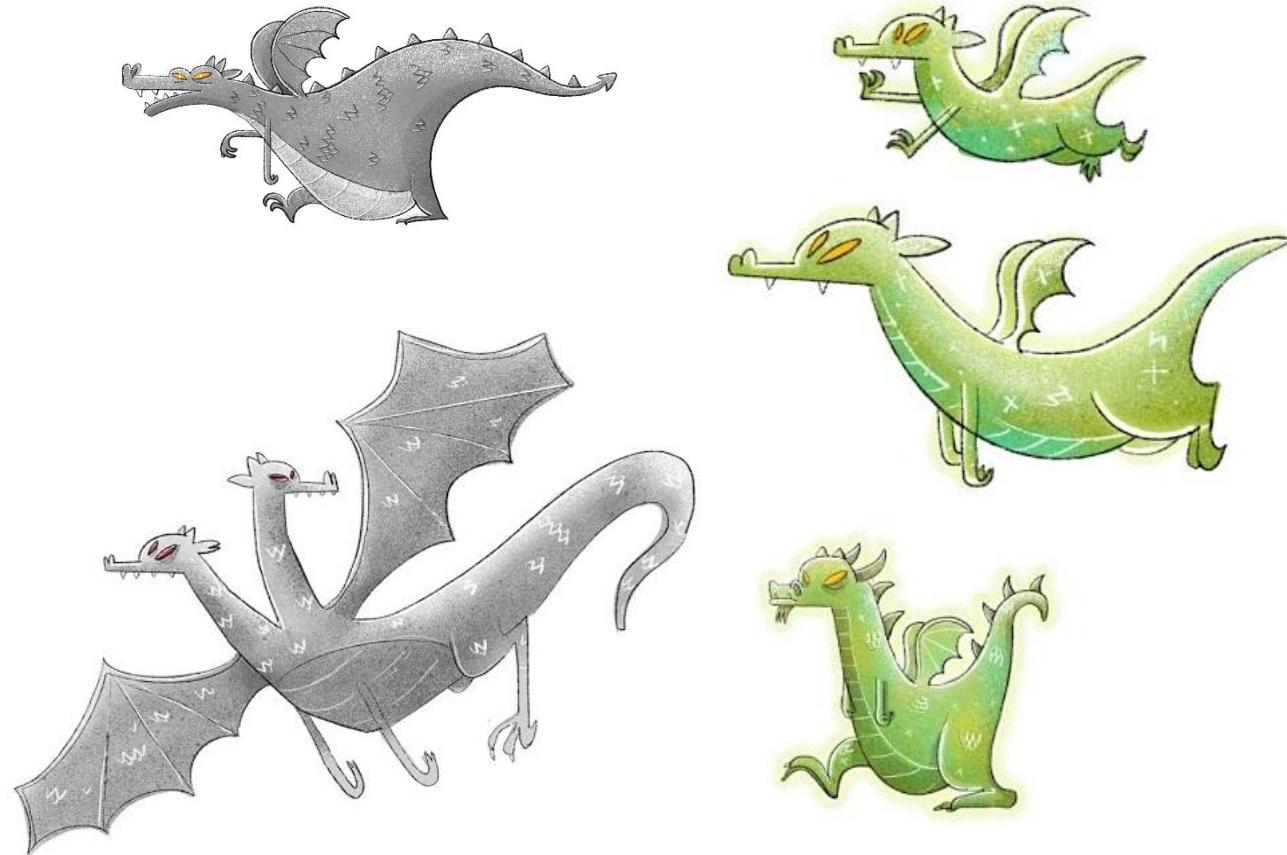
# Gıcık ejderha

- **Hata ismi:** Bellek sizıntısı
- **Önem:** Orta
- **Hatanın belirtileri:** Bilgisayar hiç çalışmaz duruma gelene kadar yavaşlar ve kapanır.
- **Hatanın nedenleri:** Bellek alanı yazılım için ayrıılır ama kullanıldıktan sonra serbest bırakılmaz.
- **Temel nedenleri:** Yazılımcı belleği serbest bırakma konusunda dikkatli değildir. Nasıl yapıldığını bilmiyor veya unutmuş olabilir.
- **Test etme:** Yazılımı kullanırken kullanılan belleği ölçersin. Kullanılan bellek sürekli artıyorsa, muhtemelen bellek sizıntısı problemi vardır.
- **Çözüm:** Düzeltilmesi gereken noktayı bulana kadar yazılımın her seferinde bir kod satırını çalıştırırsınız.



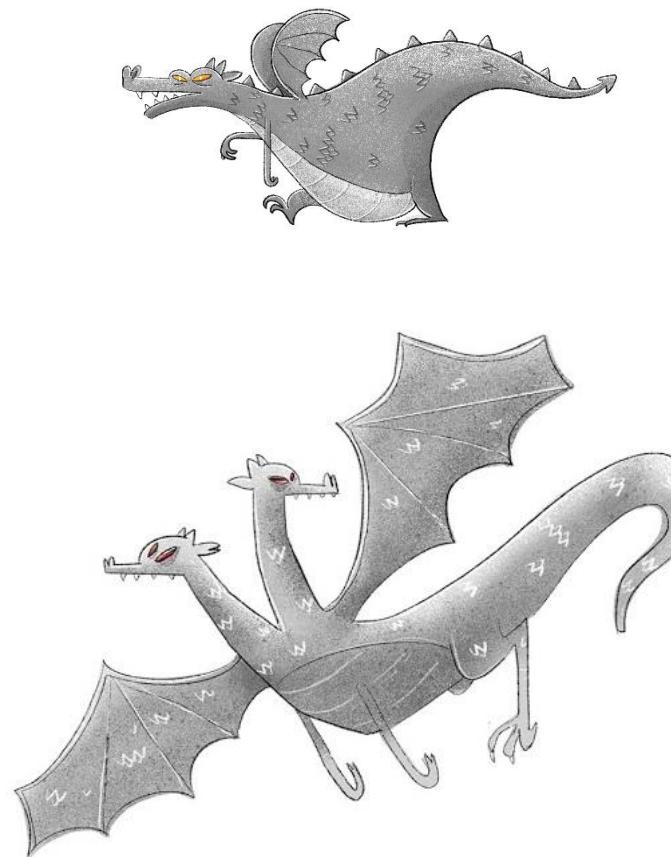
# Hırsız ejderha

- **Renk:** Parlak yeşil veya gri
- **Boyut:** Küçükten büyüğe çeşitli boyutlarda
- **Bulunması:** Kolaydan zora çeşitli derecelerde
- **Kurtulunması:** Kolaydan zora çeşitli derecelerde
- **Uçabilir mi?:** Bazları uçar, bazıları uçamaz
- **Kanatları nasıl?** Küçükten büyüğe çeşitli boyutlarda
- **Alev üfler mi?:** Evet
- **En sevdiği şey:** Yemek yemek ve hazineleri çalmak



# Hırsız ejderha

- **Hata ismi:** İşlevsellik hatası
- **Önem:** Düşük-orta-yüksek
- **Hatanın belirtileri:** Yazılım yapması gerekeni yapamaz. Hesaplamalar yanlış sonuç verir. Kullanıcı bilgiyi yanlış yerde görür.
- **Hatanın nedenleri:** İşlevsellik yanlış kodlanmıştır.
- **Temel nedenleri:** Yazılımcı, kullanıcının neden bahsettiğini tam anlayamamıştır. Dikkatsizlik veya acele sonucu hata ortaya çıkar.
- **Test etme:** Yazılımı normal bir şekilde, test uzmanı deneyimlerine veya gereksinim tanımlarına göre kullanırsınız.
- **Çözüm:** İlgili kod satırı yeniden yazılır.



# Kaba ejderha

- **Renk:** Siyah
- **Boyut:** Küçük
- **Bulunması:** Zor
- **Kurtulunması:** Orta
- **Uçabilir mi?:** Evet
- **Kanatları nasıl?:** Orta
- **Alev üfler mi?:** Çok
- **En sevdiği şey:** Yakalanmadan yiyecek ve hazine çalmak



# Kaba ejderha

- **Hata adı:** Güvenlik hatası
- **Önem:** Yüksek
- **Hata belirtileri:** Kullanıcıya ait özel bir bilginin sistem dışına çıkması. (örneğin banka kartı bilgileri)
- **Hatanın nedenleri:** Bir suçlu güvenlik hmasını sisteme girmek için kullanır, sonra da bir şeyler çalar veya zarar verir.
- **Temel nedenleri:** Yazılımcı en son güvenli kodlama ilkelerine uymamıştır. Belki bu konuda bilgisi yoktur.
- **Test etme:** Bir güvenlik testi yazılımı kullanarak yazılımdaki bilinen güvenlik açıklarını ararsınız. Aynı zamanda kodları gözden geçirebilirsiniz. Bilinen hata listelerini kullanarak yazılımı kontrol etmek faydalı olabilir.
- **Çözüm:** Bilinen bir açığın aynı zamanda bilinen bir çözümü de vardır. Kodlamada veya sistem ayarlarında çözülür.



# Hızlı ejderha

- **Renk:** Mor
- **Boyut:** Küçük
- **Bulunması:** Zor
- **Kurtulunması:** Zor
- **Uçabilir mi?:** Evet
- **Kanatları nasıl?:** Geniş
- **Alev üfler mi?:** Biraz
- **En sevdiği şey:** Gizlice altın çalmak



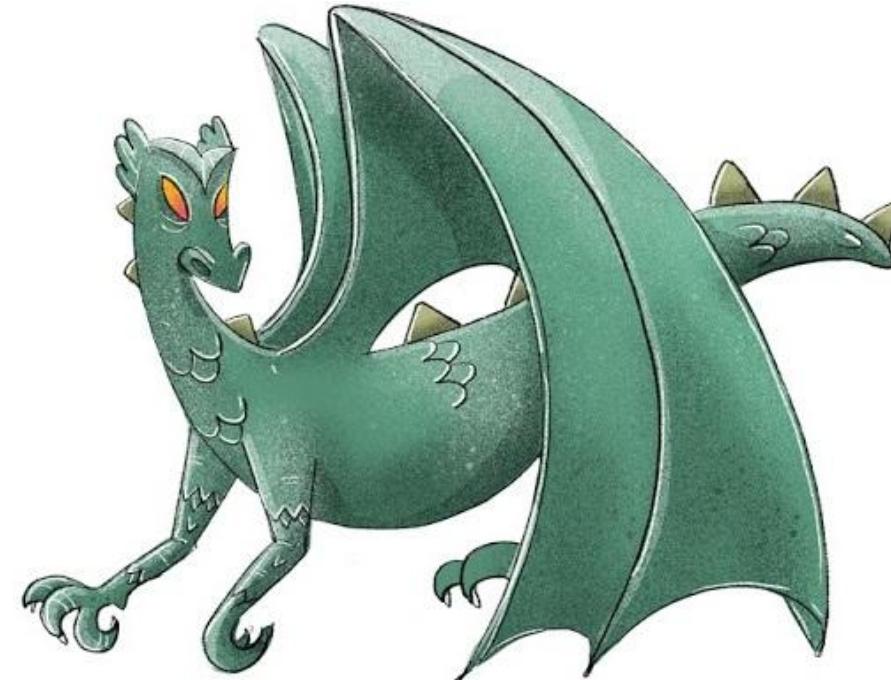
# Hızlı ejderha

- **Hata adı:** Performans hatası
- **Önem:** Orta
- **Hata belirtileri:** Yazılım olması gerektiğinden daha yavaş çalışır.
- **Hatanın nedenleri:** Kodun bir kısmı verimsiz veya yanlış çalışır. Ayarlar da yanlış yapılmış olabilir. Bu durumda yazılım gereksiz bir şeyler yapmayı dener. Bu da zaman alır.
- **Temel nedenleri:** Performans kodlama esnasında hesaba katılmaz. Yazılımcı kodlama ortamının tüm olasılıklarını bilmeyebilir. Yazılımcı yazılımın geneline hakim olmayabilir.
- **Test etme:** Performans test aracıyla yazılımı birden fazla kullanıcıyla aynı anda kullanmaya çalışırsınız. Hızını ve yanıt sürelerini ölçersiniz.
- **Çözüm:** Kodun yavaş çalışan kısımlarını değiştirirsiniz. Hızı tekrar ölçersiniz.



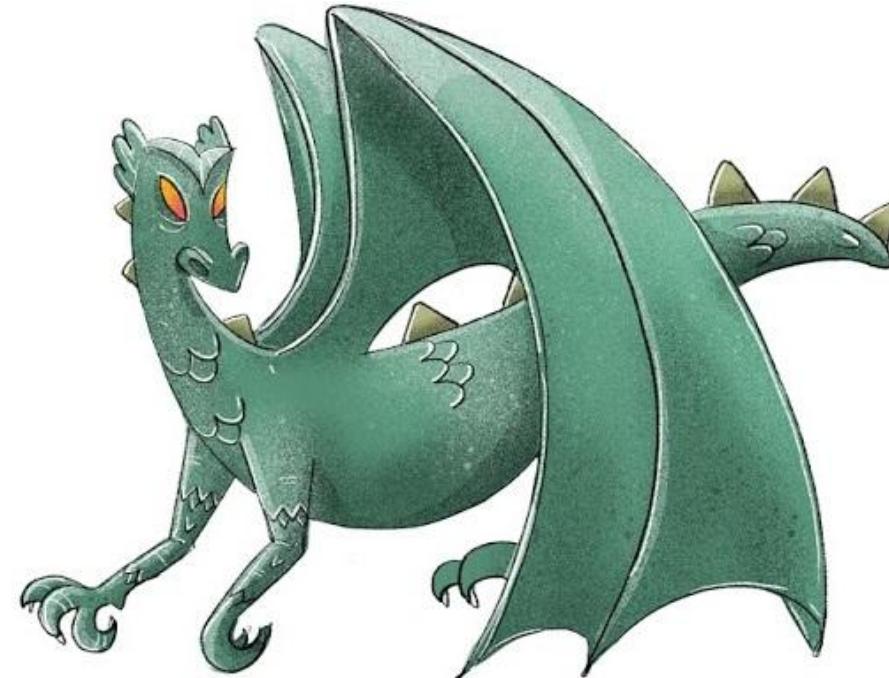
# Gösterişli, zor ejderha

- **Renk:** Yeşil
- **Boyut:** Büyük
- **Bulunması:** Kolay
- **Kurtulunması:** Zor
- **Uçabilir mi?:** Evet
- **Kanatları nasıl?:** Geniş
- **Alev üfler mi?:** Çok
- **En sevdiği şey:** Altını takip etmek ve kendi bölgesini korumak



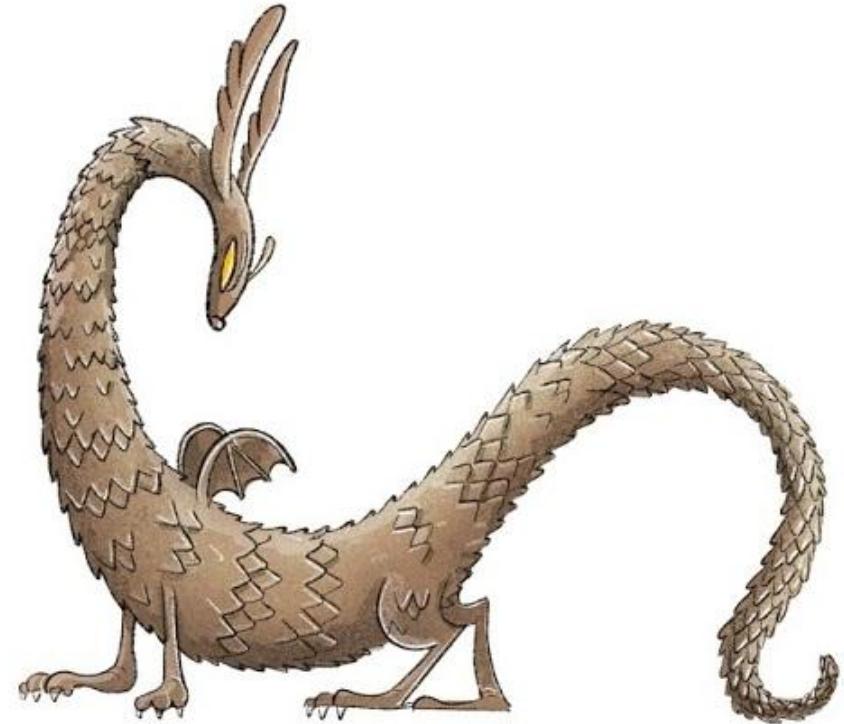
# Gösterişli, zor ejderha

- **Hata adı:** Kullanılabilirlik hatası
- **Önem:** Düşük
- **Hata belirtileri:** Kullanıcının yazılımı kullanması zorlaşırlar ama bir şekilde işini halleder.
- **Hatanın nedenleri:** Kodlama yalnızca işlevsellik göz önünde bulundurularak muhtemelen en hızlı şekilde yapılmıştır.
- **Temel nedenleri:** Kullanıcı ihtiyaçları tasarımda ve kodlamada hesaba katılmamıştır. Kullanılabilirliğin önemi anlaşılmamış olabilir.
- **Test etme:** Kullanılabilirlik testi yaparak neyin kullanılması kolay, neyin zor olduğu hakkında fikirler toplayabilirsiniz.
- **Çözüm:** Kullanılabilirlik testinin sonuçlarına göre kullanıcı deneyimini iyileştirmek için akışı, tasarımları, kodu değiştirebilirsiniz.



# Yeraltı ejderhası

- **Renk:** Kahverengi
- **Boyut:** Büyük
- **Bulunması:** Kolay
- **Kurtulunması:** Orta
- **Uçabilir mi?:** Hayır
- **Kanatları nasıl?:** Küçük
- **Alev üfler mi?:** Çok
- **En sevdiği şey:** Kolayca yiyecek bulmak ve yemek



# Yeraltı ejderhası

- **Hata adı:** Donanım hatası
- **Önem:** Yüksek
- **Hata belirtileri:** Bilgisayarın bir parçası veya tamamı çalışmaz.
- **Hatanın nedenleri:** Donanımının bir kısmı zamanla bozulur.
- **Temel nedenler:** Donanımının bir kısmı düşük kalitededir ve beklentiği kadar uzun süre dayanmaz. Muhtemelen parça, diğer parçalarla iyi çalışmaz ve bozulur.
- **Test etme:** Sistemi normal bir şekilde kullanabilirsiniz. Donanımı gözlemlersiniz. Test ortamı, kullanıcıların sahip olacağı donanıma benzer donanım kullanır.
- **Çözüm:** Bozuk parçayı yenisiyle değiştirirsiniz veya diğer parçalara daha iyi uyacak bir parçayla değiştirirsiniz.



# Gölge mi ejderha mı?

- **Renk:** Gri
- **Boyut:** Küçük
- **Bulunması:** Kolay
- **Kurtulunması:** Kolay
- **Uçabilir mi?:** Hayır
- **Kanatları nasıl?:** Küçük
- **Alev üfler mi?:** Evet
- **En sevdiği şey:** İnsanların kafasını karıştırmak



# Gölge mi ejderha mı?

- **Hata adı:** Test edilebilme hatası
- **Önem:** Düşük
- **Hata belirtileri:** İşlevsellik yanlış çalışıyormuş gibi görünür, örneğin bir hesaplamadan yanlış sonuç çıkması gibi.
- **Hatanın nedenleri:** Test uzmanı yanlış test verisine veya test ortamına sahip olabilir.
- **Temel nedenleri:** Test ortamı veya test verisi düzgünce planlanmamış olabilir.
- **Test etme:** Sistemi normal bir şekilde kullanırsınız ama ortamı ve veriyi gözlersiniz. Hataya testin yol açıp açmadığını daima göz önünde bulundurun.
- **Çözüm:** Test ortamını ve verisini daha iyi tanımlarsınız. Yanlış alarmlardan ders çıkarırsınız.



# İyi ejderha

- **Renk:** Parlak yeşil
- **Boyut:** Orta
- **Bulunması:** Kolay
- **Kurtulunması:** Kolay
- **Uçabilir mi?:** Evet
- **Kanatları nasıl?:** Orta
- **Alev üfler mi?:** Evet
- **En sevdiği şey:** Hayvanları yemek ve insanlara yardım etmek



# İyi ejderha

- **Hata adı:** Hata tohumlama, mutasyon testi - bilerek yaratılmış bir hata
- **Önem:** Düşük
- **Hata belirtileri:** İşlevsellik yanlış çalışıyordu gibi görünür, örneğin bir hesaplamadan yanlış sonuç çıkması gibi. Yani hata, işlevsellik hatası gibi görünür.
- **Hatanın nedenleri:** Test uzmanı veya yazılımcı kodlamada bilerek hata oluşturur.
- **Temel nedenleri:** Buradaki amaç, tohumlandırılan tüm hatalar bulunduğuunda, tüm hatalar bulunmuş olacaktır.
- **Test etme:** Sistemi normal bir şekilde kullanırsınız ve tohumlandırılan tüm hataları bulmaya çalışırsınız. Son tohumlandırılan hata da bulunduğuunda, test etmeyi bırakabilirsiniz.
- **Çözüm:** Tıpkı gerçek işlevsellik testindeki gibi tohumlandırılan hatalar için de kodu düzeltmeyi unutmayın.



# İnatçı ejderha

- **Renk:** Parlak yeşil
- **Boyut:** Büyük
- **Bulunması:** Kolay
- **Kurtulunması:** Zor
- **Uçabilir mi?:** Evet
- **Kanatları nasıl?:** Küçük
- **Alev üfler mi?:** Evet
- **En sevdiği şey:** İnsanlara rahat vermemek



# İnatçı ejderha

- **Hata adı:** Sürüm yönetimi hatası
- **Önem:** Orta
- **Hata belirtileri:** Önceden düzeltilmiş bir hata, tekrar ortaya çıkar. Bu işlevsellik hatası olabilir.
- **Hata nedenleri:** Sürüm yönetimi başarısız olur. Yazılımcı yenisi kullanılması gereken yerde yazılımın eski bir kod parçasını kullanır.
- **Temel nedenleri:** Acelecilik ve çok fazla sayıda paydaşın yazılım geliştirmeye plansızca dahil olması hatayı yaratır. Sürüm yönetimi tamamen eksik olabilir.
- **Test etme:** Sistemi normal bir şekilde test edersiniz. Bir hata bulunursa, daha önceden düzeltilmiş bir hata mı diye kontrol edersiniz. Öyleyse, sürüm yönetiminin nasıl çalıştığını araştırınız.
- **Çözüm:** Sürüm yönetimi yazılımı kullanarak sürüm yönetiminin sağlıklı çalışmasını sağlarsınız.



# Kafası karışmış ejderha

- **Renk:** Parlak yeşil
- **Boyut:** Küçük
- **Bulunması:** Kolay
- **Kurtulunması:** Kolay
- **Uçabilir mi?:** Evet
- **Kanatları nasıl?:** Orta
- **Alev üfler mi?:** Evet
- **En sevdiği şey:** Daha büyük ejderhalara zarar vermek



# Kafası karışmış ejderha

- **Hata adı:** Dokümantasyon hatası
- **Önem:** Düşük
- **Hata belirtileri:** İşlevsellik, dokümantasyonda belirtilenden farklı bir şekilde çalışır.
- **Hatanın belirtileri:** Planlar ve yönergeler yazılımla uyumlu değildir.
- **Temel nedenler:** Kodlama sırasında, planlanandan farklı bir yazılım oluşturan kararlar alınmış veya değişiklikler yapılmıştır. Dokümantasyon eski planlara göre oluşturulmuştur, yenilenmemiştir.
- **Test etme:** Sistemi normal şekilde kullanırsınız. Bir hata bulunursa, yazılımın veya belgelerin yanlış olup olmadığını değerlendirirsiniz.
- **Çözüm:** Eşleşmeleri için yazılımı veya belgeleri değiştirebilirsiniz.



# 5-15 dakikalık alıştırma

**Kendi iyi hatanızı tasarlayın -  
kendi iyi ejderhanızı çizin**

**// İhtiyacınız olan  
şeyler**

Kalem ve kağıt



## // Görev

**1 Yazılım hakkında bir şeyi daha iyi anlamanıza yardımcı olan bir hata düşünün.(veya hoşlandığınız herhangi bir hatayı düşünün.)**

- Hatanın adını yazın ve birkaç sözcükle anlatın.

**2 Bu yazdıklarınıza eşdeğer bir ejderha düşünün.**

- Ejderhanın özelliklerini yazın.
- Hata kötüyse, ejderha da daha büyük olacakmış gibi.

**3 Ejderhayı çizin.**

- Düşüncenizi gerçekleştirirken esasen ejderhanın hatayı nasıl yansittığına odaklanın.
- Kusursuz bir çizim olmasına gerek yok.

**4 Tasarımınızı arkadaşlarınıza gösterin.**

- Örneğin Padlet kullanın.



# Teşekkürler!

Dragons Out kitabıyla ilgileniyorsanız, buradan sipariş verebilirsiniz:

<https://www.dragonsout.com/p/order-dragons-out-book.html>

Kitap projesini takip edin ve paylaşın:

<https://www.dragonsout.com>

Soru sorun:

[kari.kakkonen@dragonsout.com](mailto:kari.kakkonen@dragonsout.com)



# Turkish Testing Board'a ulaşın!

Çalışmalarımıza göz atmak için web sitemizi ziyaret edebilirsiniz:

<https://www.turkishtestingboard.org/>

Sorularınız, görüş ve önerileriniz için bize ulaşabilirsiniz:

[info@turkishtestingboard.org](mailto:info@turkishtestingboard.org)