



**NETAS**

**Agile  
Test Otomasyon  
DevOPS**

**Didem Çolak Arslan**

**Şubat 2019**

# Ajanda

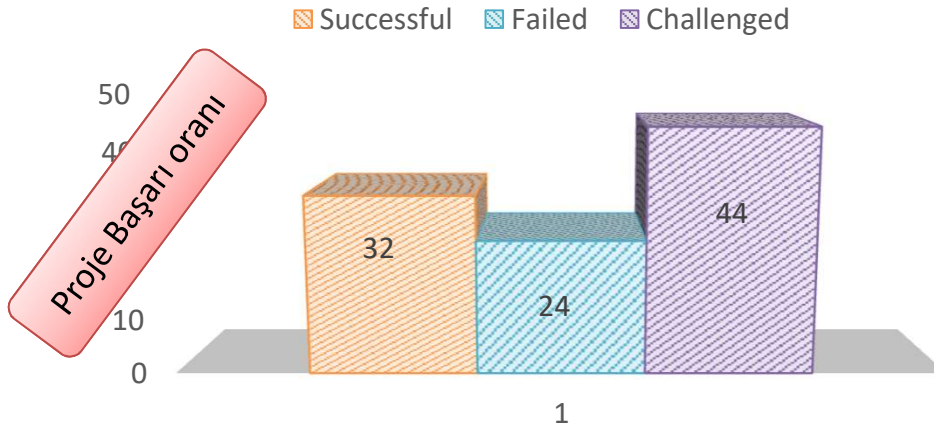
- ❖ Tanışma
- ❖ Waterfall Model
- ❖ Agile Yaklaşım
- ❖ Test ve Test Otomasyon
- ❖ CI & CD
- ❖ DevOPS

## Waterfall Model

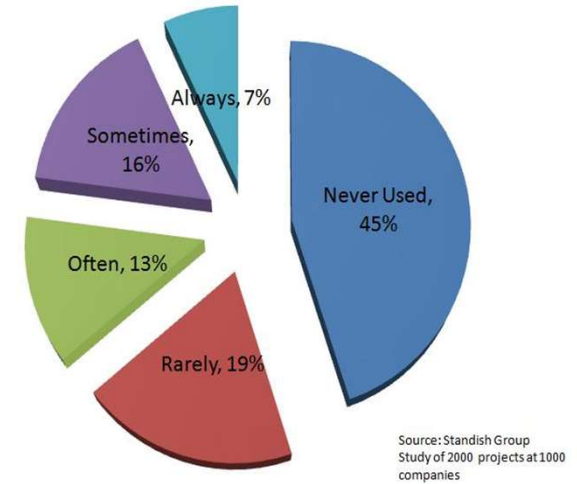
- Ürünün son halinin nasıl olacağı kesin belliyse
- Müşteriler kapsamı, geliştirme başladıktan sonra değiştir(e)meyecekse
- Hız, başarı için kritik değilse



# The Standish Group Reports



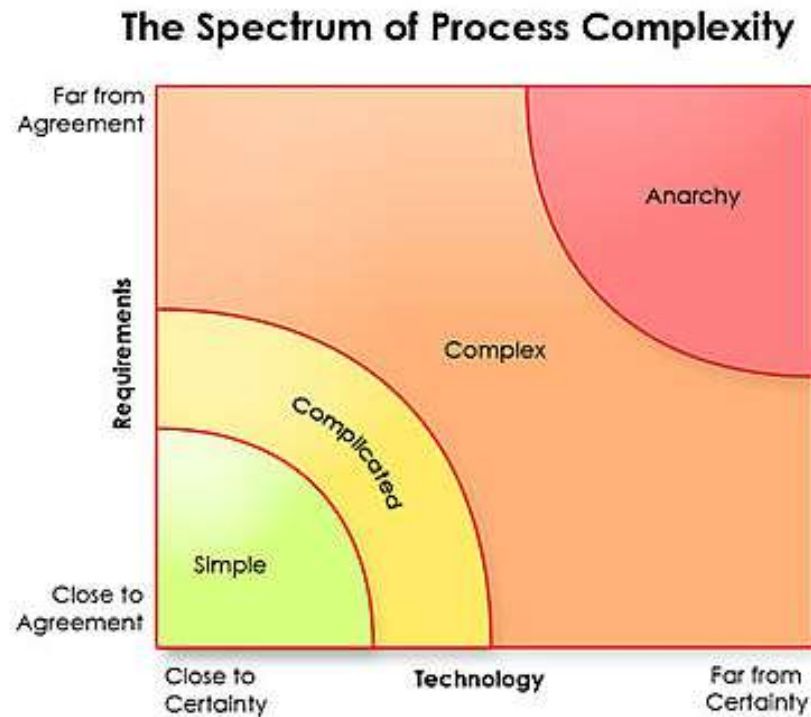
Usage of Features and Functions in Typical System



-Başarılı ve başarısız IT projeleri. Başarılı Proje = Proje planına uygun biten proje olarak tanımlanıyor. Planlanan zaman ve planlanan bütçe ve planlanan kapsam  
- Standish Group'un 1000 şirkette 2000 proje ile yaptığı bu araştırmada gösteriyor ki; bir ürünün mevcuttaki özelliklerinin sadece %7'si sıklıkla kullanılırken %45'i bir kere bile kullanılmıyor. Yani özelliklerin %64'ü ya hiç kullanılmıyor ya da çok nadir kullanılıyor.

# Stacey Matrix'i

Problems; Simple, Complicated, Complex





# Empiricism (Deneycilik)

Deneycilik, bilginin **deneyimden** ve bilinen şeylere dayanarak alınan **kararlardan** meydana geldiğini ileri sürer.

Üç ayakla desteklenir:

- Şeffaflık : Şeffaflığı arttır
- Gözlem : Problemleri görünür kıl
- Adaptasyon : İyileşme ve adaptasyonu arttır



# AGILE



## Agile Manifesto



Süreçler ve Araçlardan	Bireyler ve Onların etkileşimine
Detaylı dokümantasyondan	Çalışan yazılıma
Sözleşme pazarlığı yapmaktan	Müşteri ile işbirliği yapmaya
Planı izlemektense	Değişime yanıt vermeye

Daha çok değer verir.

Soldakilerin de değerli olduğunu biliyoruz. Ancak bizler sağdakileri daha değerli buluyoruz ve daha fazla önemsiyoruz...

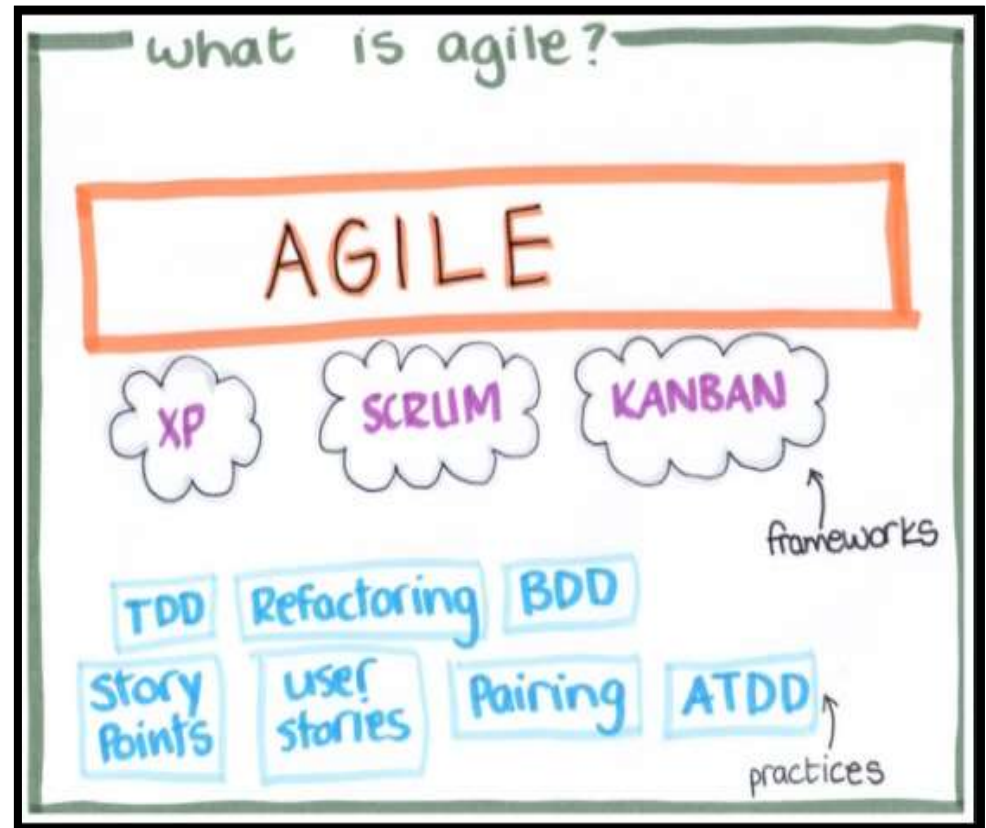
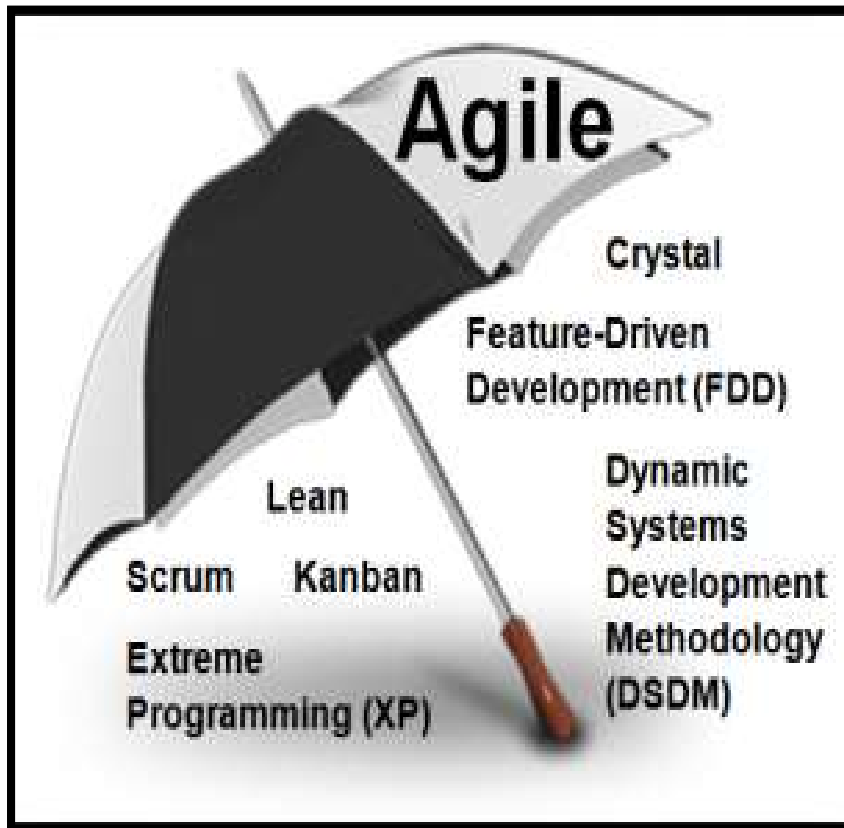


# Agile'in 12 Prensibi



- ✓ En önemli önceliğimiz değerli yazılımın erken ve devamlı teslimini sağlayarak müşterileri memnun etmektir.
- ✓ Değişen gereksinimler yazılım sürecinin son aşamalarında bile kabul edilmelidir. Çevik süreçler değişimi müşterinin rekabet avantajı için kullanır.
- ✓ Çalışan yazılım, tercihen kısa zaman aralıkları belirlenerek birkaç haftada ya da birkaç ayda bir düzenli olarak müşteriye sunulmalıdır.
- ✓ İş süreçlerinin sahipleri ve yazılımcılar proje boyunca her gün birlikte çalışmalıdırlar.
- ✓ Projelerin temelinde motive olmuş bireyler yer almalıdır. Onlara ihtiyaçları olan ortam ve destek sağlanmalı, işi başaracakları konusunda güven duyulmalıdır.
- ✓ Bir yazılım takımında bilgi alışverişinin en verimli ve etkin yöntemi yüzyüze iletişimdir.
- ✓ Çalışan yazılım ilerlemenin birincil ölçüsüdür.
- ✓ Çevik süreçler sürdürülebilir geliştirmeyi teşvik etmektedir. Sponsorlar, yazılımcılar ve kullanıcılar sabit çalışma temposunu sürekli devam ettirebilmelidir.
- ✓ Teknik mükemmeliyet ve iyi tasarım konusundaki sürekli özen çevikliği artırır.
- ✓ Sadelik, işin özü olmayan işlerin yapılmamasını en üst seviye tutmak elzemdir.
- ✓ En iyi mimariler, gereksinimler ve tasarımlar kendi kendini örgütleyen takımlardan ortaya çıkar.
- ✓ Takım, düzenli aralıklarla nasıl daha etkili ve verimli olabileceğinin üzerinde düşünür ve davranışlarını buna göre ayarlar ve düzenler.

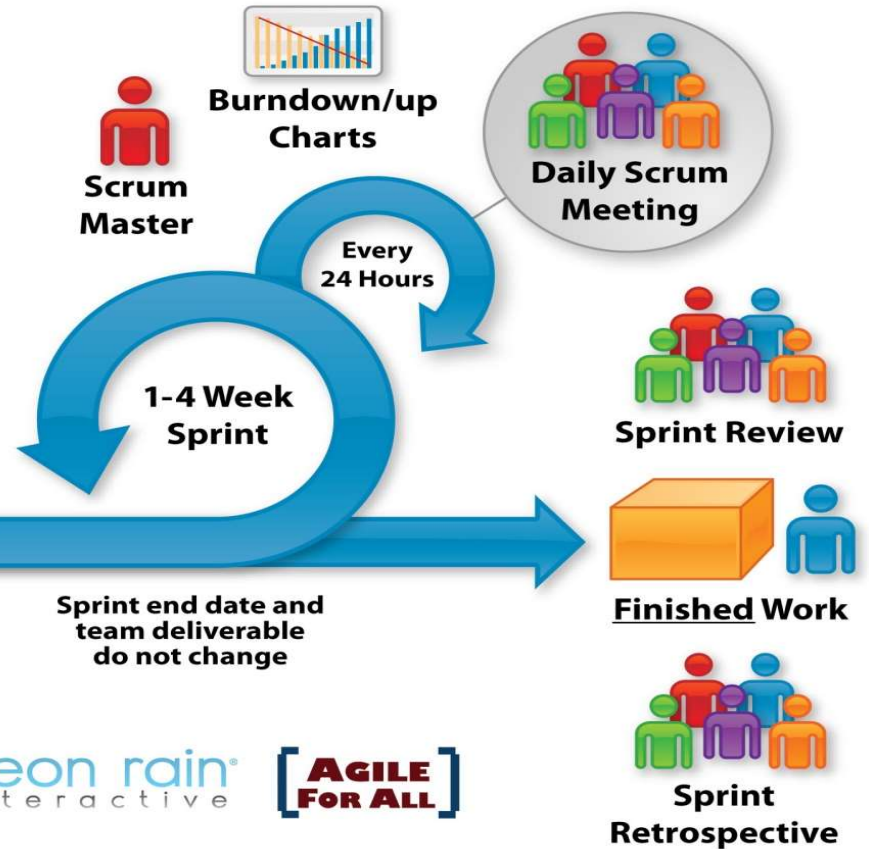
## Agile



# Agile Yaklaşımında Scrum Framework

## The Agile: Scrum Framework at a glance

Inputs from Executives,  
Team, Stakeholders,  
Customers, Users



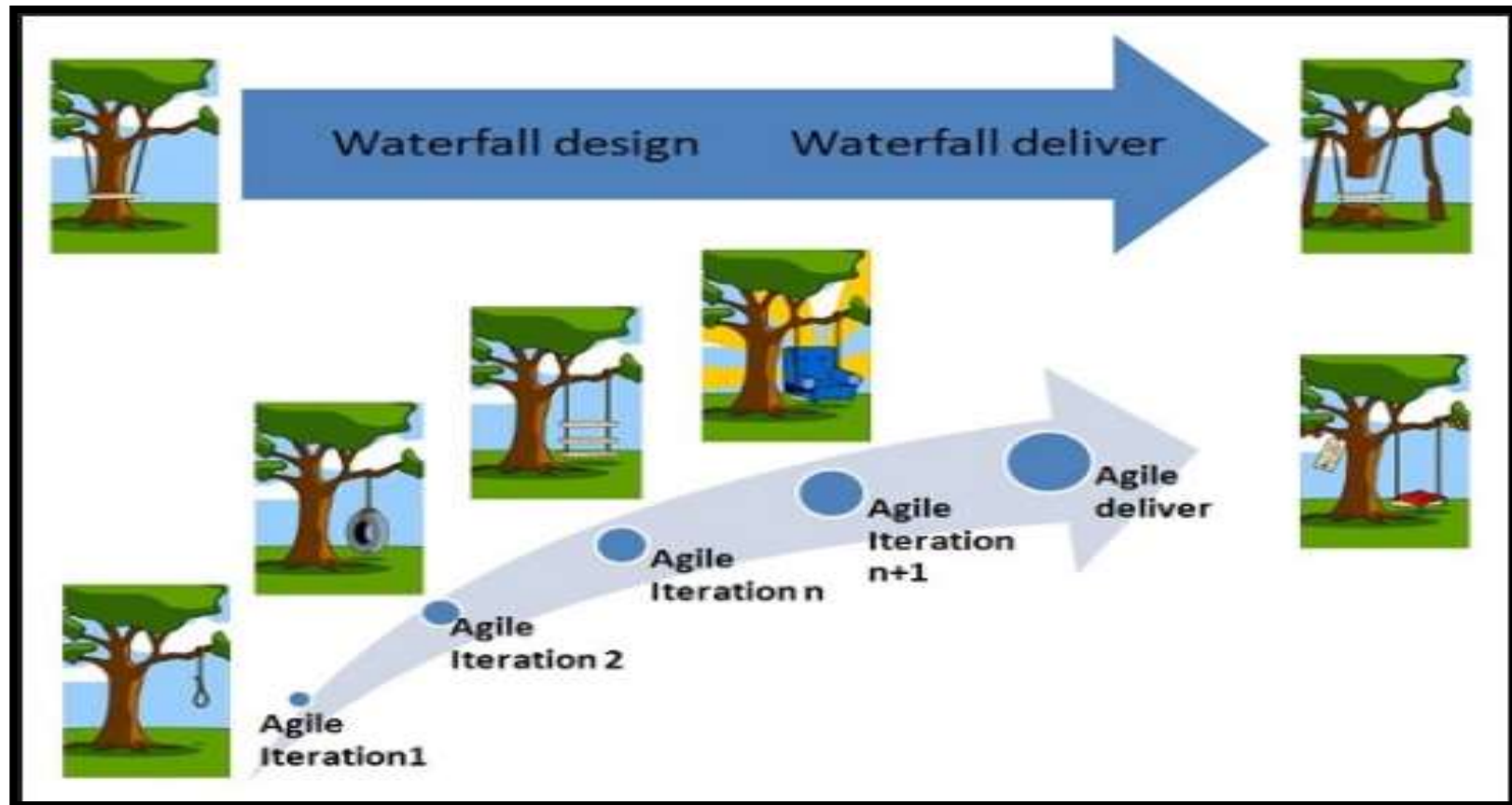
neon rain<sup>®</sup>  
interactive [AGILE FOR ALL]

# Agile yaklaşım ne zaman tercih edilmeli?

- Ürünün son hali net değilse
- Bilinmeyenlerin olduğu bir ortamda geliştirme yapılıyorsa
- Başlangıçta yeteri kadar bilgi yoksa
- Müşteriler ne istediklerini net bilmiyor, ifade edemiyorsa
- Müşteriler kapsamı sıklıkla değiştiriyorsa
- Standartların sıklıkla değiştiği bir ürün geliştiriliyorsa
- Hız önemliyse

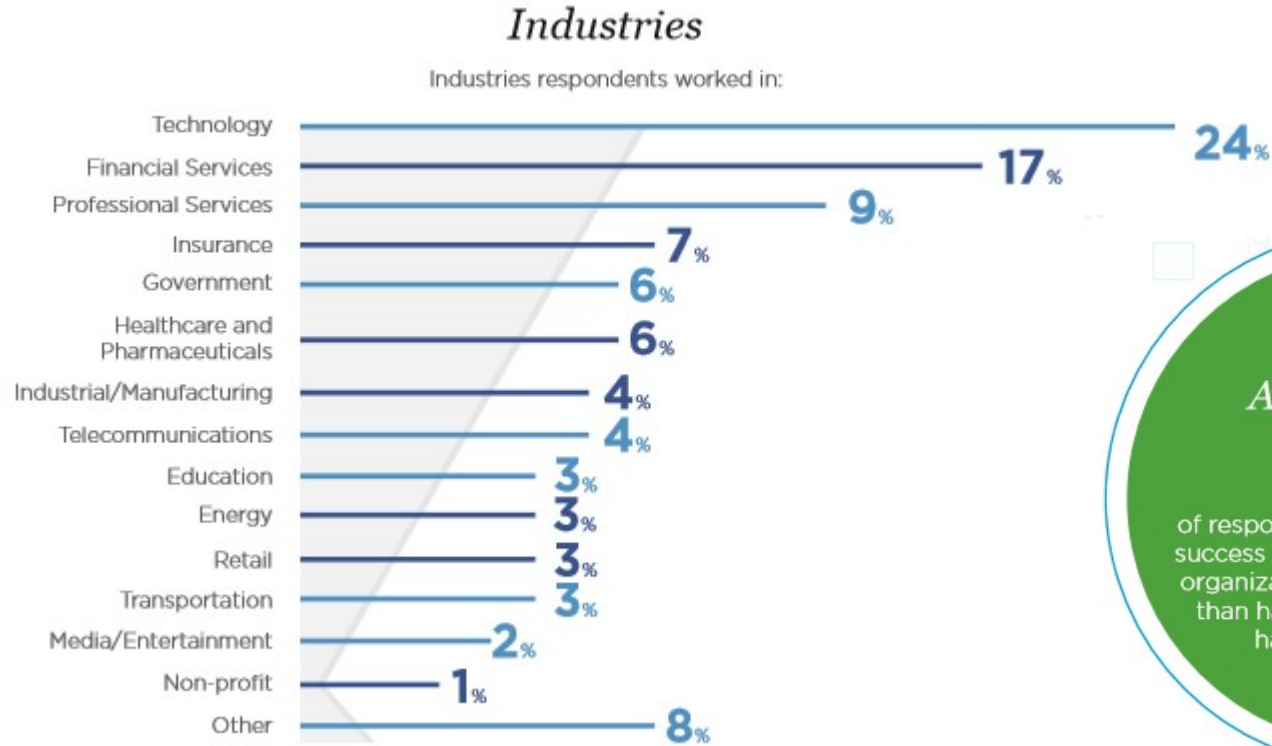


## Agile vs Waterfall





## Agile Hangi Sektörlerde Kullanılıyor



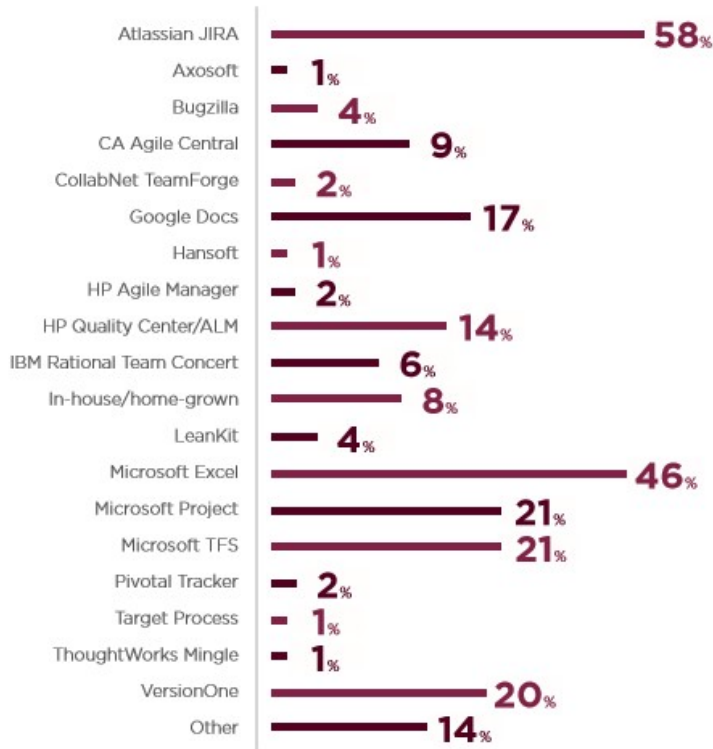
### *Success of Agile Projects*

**98%**

of respondents said that they had success with agile projects in their organization. 74% indicated more than half of their agile projects had been successful.

# Agile Yaklaşımlarda Kullanılan Araçlar ve Mühendislik Pratikleri

## Use of Agile Management Tools



## Engineering Practices Employed

This year's survey demonstrated an increased use of coding standards (64% compared to 56% last year) and the use of continuous integration and refactoring were cited less as practices used.



\*Respondents were able to make multiple selections.

## Agile Uygulayan Bazı Firmalar

Google

amazon

Spotify®

hp

IBM

YAHOO!

Apple™

SAP®

NASA

LOCKHEED  
MARTIN

Microsoft

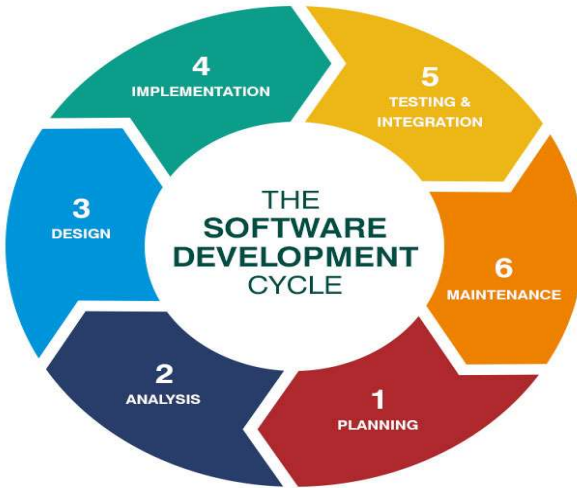
LinkedIn

Etsy

NETFLIX

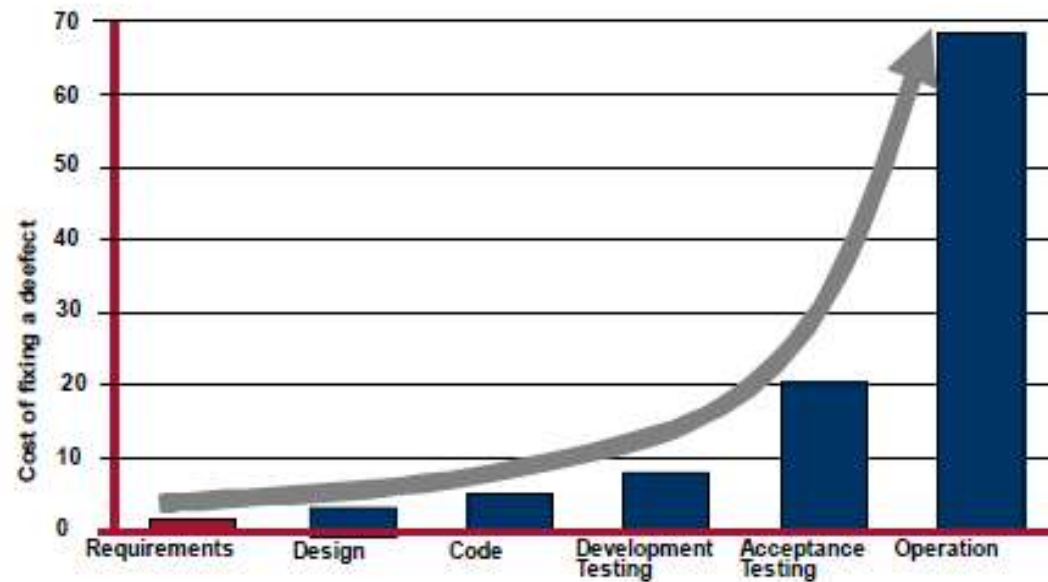
## Test Nedir?

- Test, bir sistemi veya bileşenlerini belirli şartları yerine getirip getirmediğini bulma niyetiyle değerlendirme sürecidir.
- Gerçek gereksinimlerin yanında, herhangi bir gap, hata ya da eksik yapılmış gereksinimi belirlemek için test yürütülmektedir.



# Hata Maliyetleri

## Effects of Defects



Source: "Boisland: Best Practices for Requirements Development & Management"



# Hata Maliyetleri

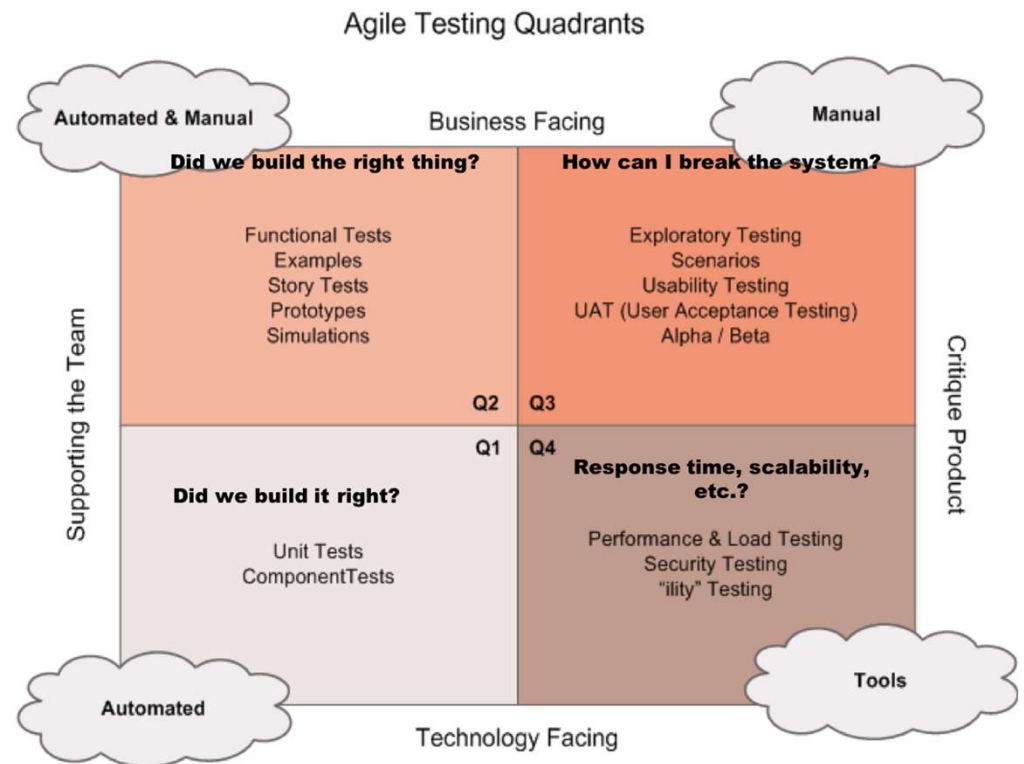
## TCO, Quality

- Production (%10-%50) => What ?? Ne yaptığımız (PO & Business)
- Maintenance & Operation (%90-%50) => How ?? Nasıl yaptığımız (Dev Team)



# Test Levels...

- Functional Testing
  - Unit Testing
  - Integration Testing
  - System Testing
  - Regression Testing
  - Acceptance Testing
  - Alpha Testing
  - Beta Testing
- Non-functional Testing
  - Performance Testing
  - Load Testing
  - Stress Testing
  - Usability Testing
  - UI vs Usability Testing
  - Security Testing
  - Portability Testing



## Agile Süreçlerde Test Senaryo Yazım Örneği

- Hedefler:
  - Müşterilerin bankaya uğramadan bankacılık işlemleri yapabilmesi
  - Şubede operasyon yoğunluğunun azaltılması, maliyetlerin düşürülmesi
  - Müşterinin bankaya olan erişiminin artırılması
- Business Requirement:
  - Yapılması gereken ne? (high-level what) => ATM projesi
- User Requirement:
  - Para çekme
  - Bakiye Sorgulama
  - Para transferi
- User Story:
  - PBI-1: Müşteri olarak ATM'den para çekebilmeyi ve bu sayede bankaya uğramamayı isterim.
  - Acceptance Kriteria:
    - Müşteri tanımlı tutardan birini seçerek para çekebilir.(100/250/500/1000TL)
    - Müşteri kendi belirlediği tutarı girerek para çekebilir.Günlük limit 2000TL'dir.
    - Müşteri tutarı seçtikten sonra on saniye sonra parayı ATM'nin haznesinden alabilmelidir.

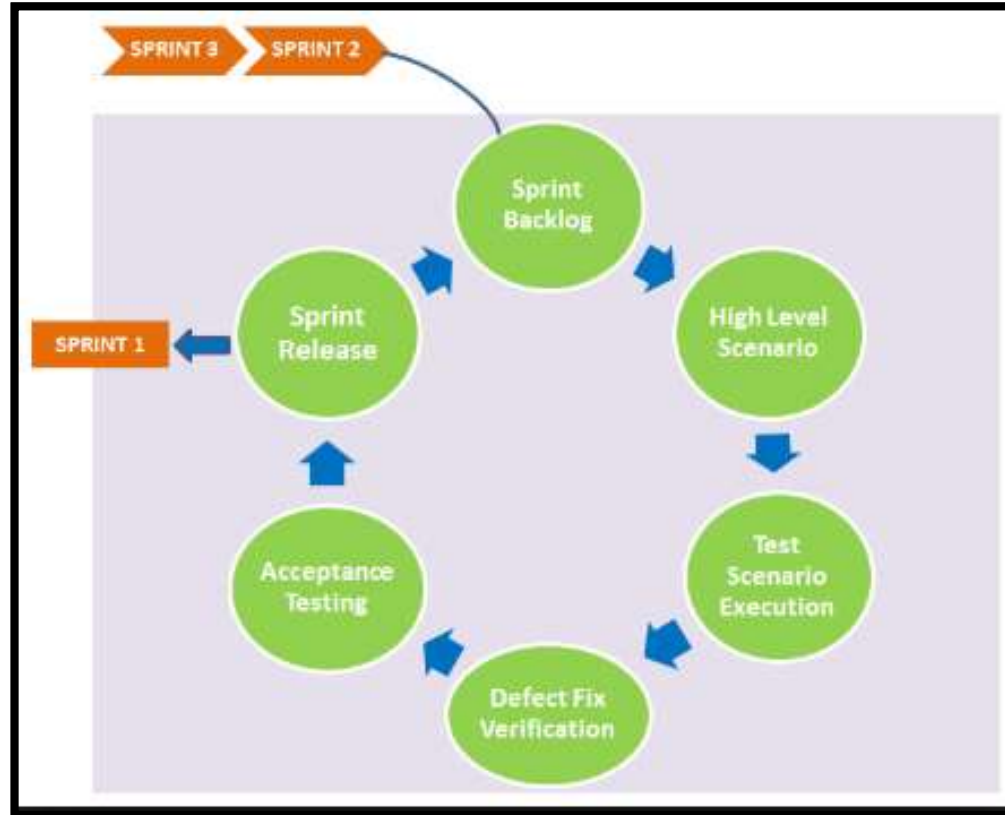


## Agile Süreçlerde Test Senaryo Yazım Örneği

- Ön Koşul:
  - Müşterinin geçerli bir kart ve şifre ile ATM'ye giriş yapması
  - Yeterli limitinin olması
- Senaryo-1:
  - Müşterileri ATM menüsünden 'para çekme' opsiyonunu seçer
  - Müşteri seçenekleri görüntüler
  - Müşteri tanımlı tutarlardan XXX tl yi seçer
  - Müşteri XXX TL'yi alır.
- Senaryo-2:
  - Müşterileri ATM menüsünden 'para çekme' opsiyonunu seçer
  - Müşteri seçenekleri görüntüler
  - Müşteri kendi belirlediği XXX tl yi seçer
  - Müşteri XXX TL'yi alır.
- Senaryo-3:
  - Müşterileri ATM menüsünden 'para çekme' opsiyonunu seçer
  - Müşteri seçenekleri görüntüler
  - Müşteri 2000 tl den fazla para girişi yapar.
  - Müşteri 2000 tl fazla para çekemeyeceğini MESAJ ile ekranda görür.
- Senaryo-4:
  - Müşterileri ATM menüsünden 'para çekme' opsiyonunu seçer
  - Müşteri seçenekleri görüntüler
  - Müşteri iptal seçeneğine basarak para çekme işi sonlandırır.



## Test Süreci Yaşam Döngüsü-Agile





# Test Automation



Yazılım test süreçlerinin verimliliğini artırmak ve test uzmanlarının manuel test yükünü azaltmak için otomasyon çözümlerini öneriyoruz.

## Test Otomasyonun Faydaları

- ✓ Sürekli tekrarlanan testlerin otomatize edilmesi test maliyetini azaltır.
- ✓ Altyapısal değişiklikte Regresyon Testinin (Regression Testing) hızlı bir şekilde tamamlanmasında önemli bir rol oynar.
- ✓ Testlerin yeniden kullanımını kolaylaştırır.
- ✓ Testlerle kapsanan kod yüzdesini artırır (Code Coverage).
- ✓ Testler 7×24 çalışabilir.
- ✓ Testlerin raporlama kalitesinin arttırılmasını sağlar.
- ✓ Geliştirilen ürünün kalitesini arttırır.

# Test Automation

Otomasyonun ROI'ni artırmak için aşağıdaki kriterler kullanılarak, otomatikleştirilecek test senaryoları seçilebilir:

- Yüksek Risk – Business ile ilgili Kritik Test Örnekleri
- Tekrar tekrar yürütülen test senaryoları
- Manuel olarak gerçekleştirmesi çok zor olan Test senaryoları
- Zaman alan Test senaryoları

Otomasyona uygun olmayan durumlar:

- Yeni tasarlanmış ve en az bir kez manuel olarak çalıştırılmayan Test senaryoları
- Gereksinimlerin sıklıkla değiştiği Test senaryoları
- Geçici olarak geliştirilmiş Test senaryoları

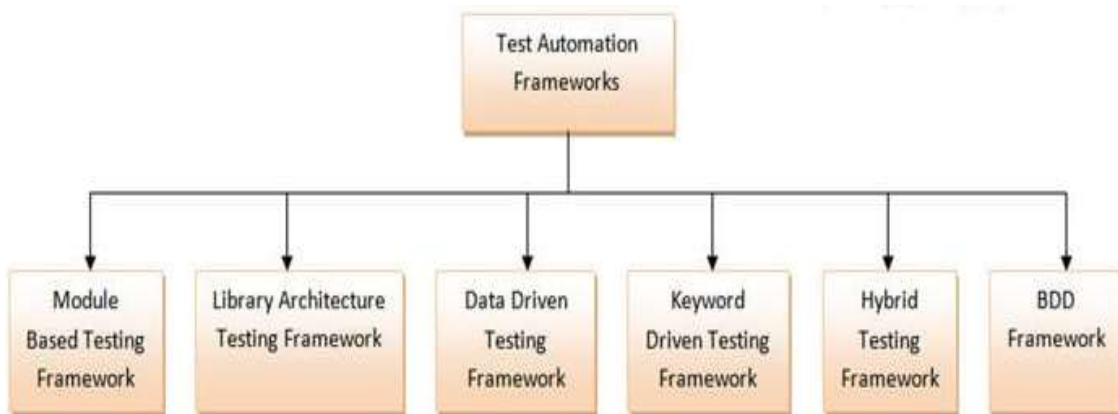
# Test Automation Framework

## Framework (Çerçeve) :

- Kullanıcıya otomasyon test scriptlerini verimli bir şekilde geliştirmelerine, yürütmelerine ve raporlamalarına yardımcı olmayı sağlar.

## Test Otomasyonu frameworklerin avantajları:

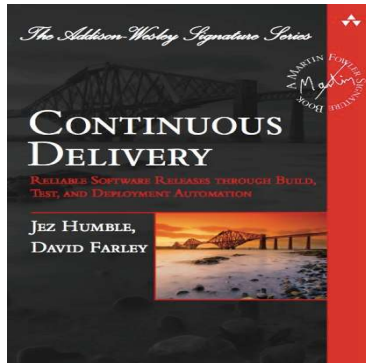
- Kodun tekrar kullanılabilirliği (Reusability of code)
- Maximum coverage
- Recovery scenario
- Düşük maliyetli bakım
- Minimum manuel müdahale
- Kolay Raporlama



Test Automation Frameworks

# What is Continuous Delivery?

***“Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.” (Agile Manifesto)***

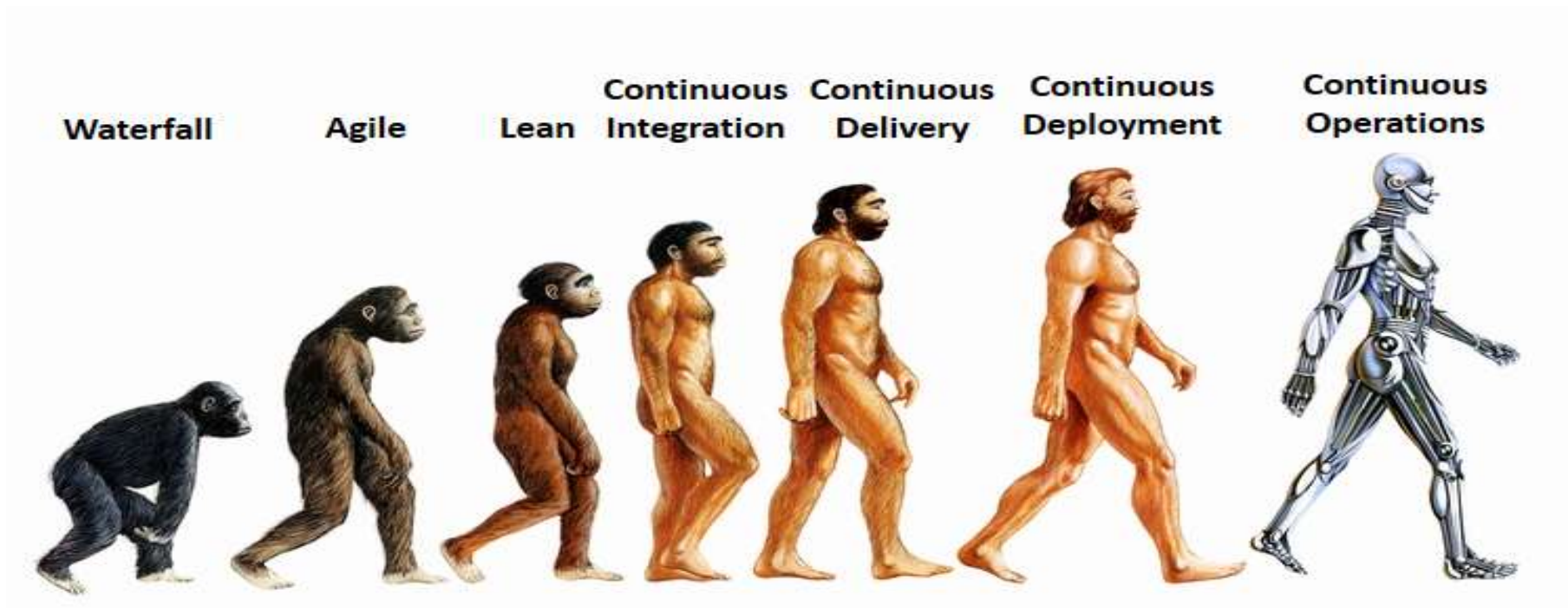


**“A set of **practices** and **principles** aimed at, *building, testing* and *releasing* software faster and more frequently”**

**A software development discipline where you build software in such a way that the **software can be released to production at any time.****

## Continuous Delivery

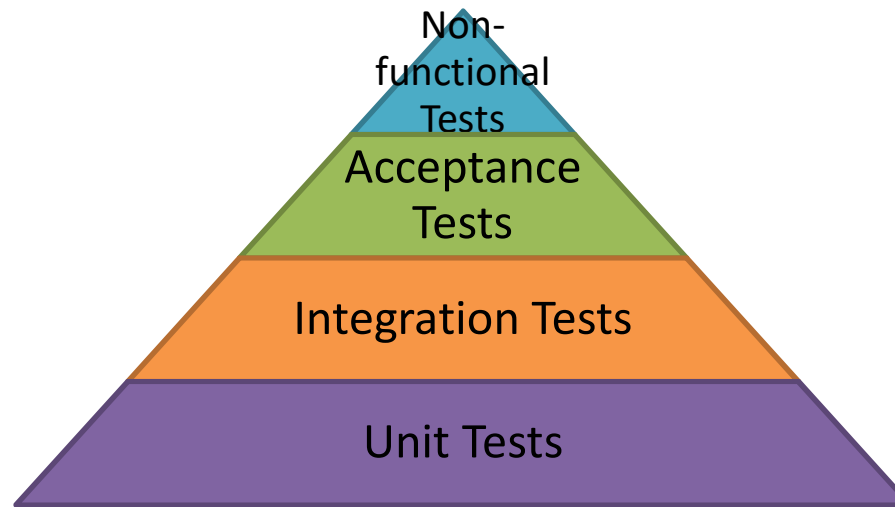
Agile first rule is that: Highest priority is to satisfy the customer through early and **continuous delivery** of valuable software





## Built-in Quality

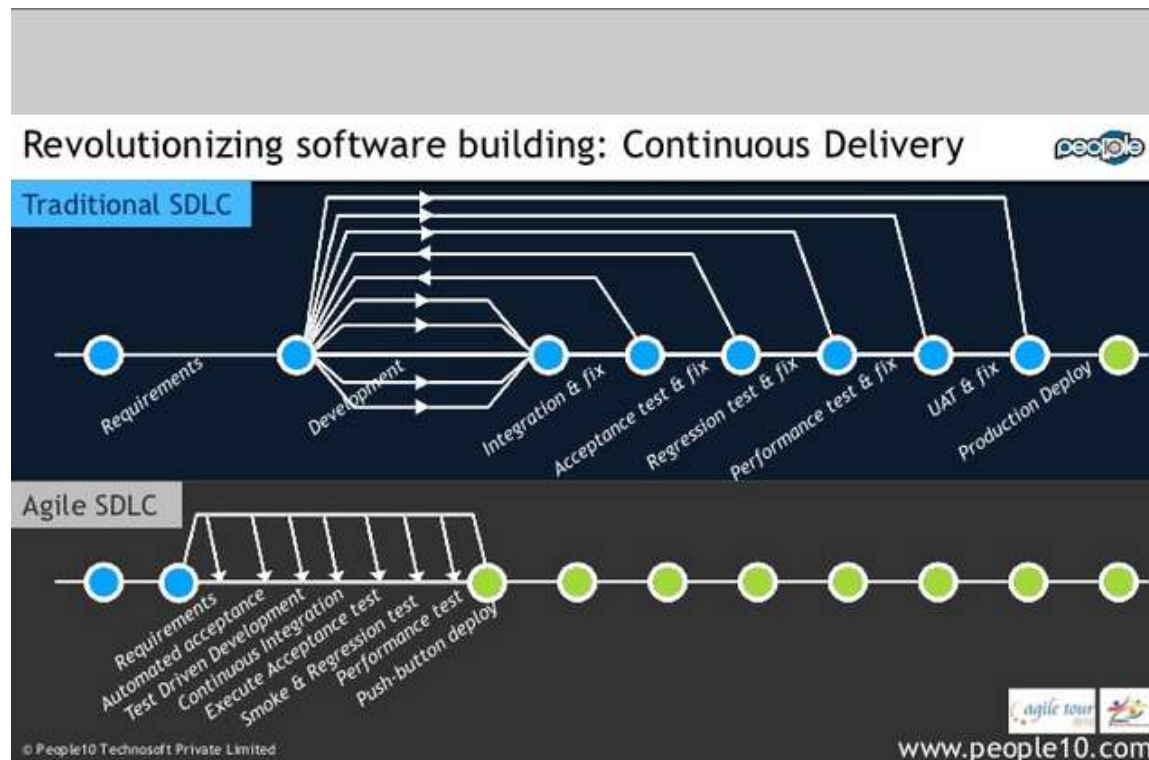
- Increased Cost
  - Better External Quality
  - Less/Slow Feedback
- 
- Better Internal Quality
  - More/Fast Feedback



Automate manual tests

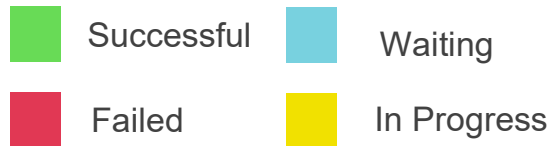
Write unit tests

## Continuous Delivery Life Cycle



<http://www.slideshare.net/IsecIndia/at2012-bengaluru-continuousdeliverynishas>

# Continuous Delivery Pipeline





## SORU & CEVAP