

Çevik Uygulama Kılavuzu

1

1

- Ne zaman çevik uygulanmalı
- Çevik manifesto ve düşünce yapısı
- 12 temel prensip
- Bütünsel takım yaklaşımı
- Erken ve sık geribildirim
- Günlük toplantı
- Retrospektifler
- Sürüm ve yineleme planlama
- İşbirlikçi kullanıcı hikayesi oluşturma
- Gösteriler ve gözden geçirmeler
- Sürekli tümleştirme
- Liderlik
- SCRUM
- KANBAN

2

2

Yaşam döngüsü seçimi

- Öngörülse (predictive) yaşam döngüsü
 - Geleneksel «çağlayan» yaklaşımı
 - Başlangıçta çok yoğun planlama aktiviteleri
 - Sonrasında tek ve sıralı bir süreç takibi
- Tekrarlayan (iterative) yaşam döngüsü
 - Bitmeden işi iyileştirmek için geribildirim olarak değiştirmeye dayalı yaklaşım
- Artırmalı (incremental) yaşam döngüsü
 - Müşterinin hemen kullanabileceği tamamlanmış teslimatlar sunan yaklaşım
- Çevik (agile) yaşam döngüsü
 - Hem tekrarlayan hem artırmalı
 - İş ürünlerini iyileştirip sık teslimat sağlayan yaklaşım

3

3

Yaşam döngüsü özellikleri

Yaklaşım	Gereksinim	Aktiviteler	Teslimat	Amaç
Öngörülse (predictive)	Sabit	Tüm proje için tek sefer gerçekleştirilir	Tek teslimat	Maliyet yönetimi
Tekrarlayan (iterative)	Dinamik	Doğruya ulaşına dek tekrar edilir	Tek teslimat	Çözümün doğruluğu
Artırmalı (incremental)	Dinamik	Her bir artım için tek sefer gerçekleştirilir	Sık ve küçük teslimatlar	Hız
Çevik (agile)	Dinamik	Doğruya ulaşına dek tekrar edilir	Sık ve küçük teslimatlar	Sık teslimat ve geri bildirimler ile müşteri değeri

4

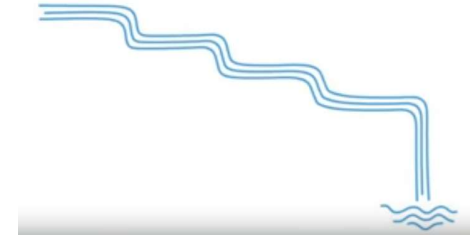
4

Yaşam döngüsü özellikleri

Yaklaşım	Gereksinim	Aktiviteler	Teslimat	Amaç
Öngörüsöl (predictive)	Sabit	Tüm proje için tek sefer gerçekleştirilir	Tek teslimat	Maliyet yönetimi
Tekrarlayan (iterative)	Dinamik	Doğruya ulaşına dek tekrar edilir	Tek teslimat	Çözümün doğruluğu
Artırmalı (incremental)	Dinamik	Her bir artım için tek sefer gerçekleştirilir	Sık ve küçük teslimatlar	Hız
Çevik (agile)	Dinamik	Doğruya ulaşına dek tekrar edilir	Sık ve küçük teslimatlar	Sık teslimat ve geri bildirimler ile müşteri değeri

5

WATERFALL



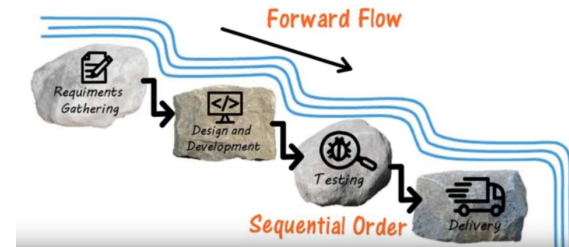
6

WATERFALL APPROACH

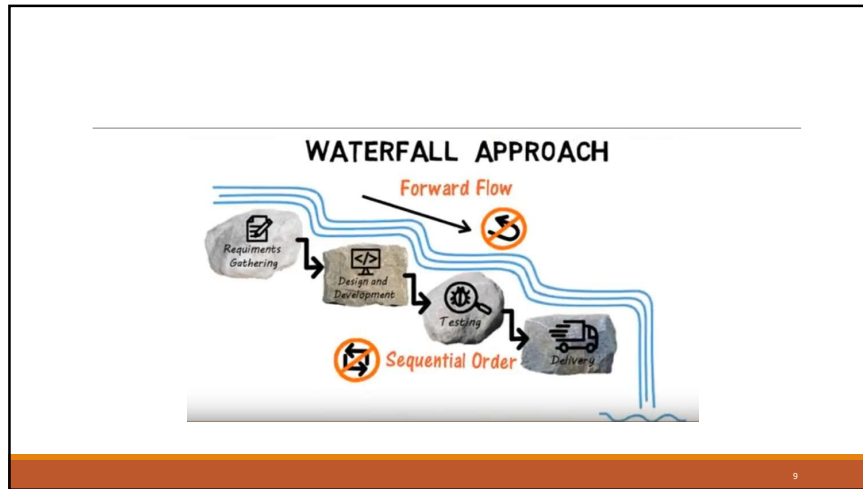


7

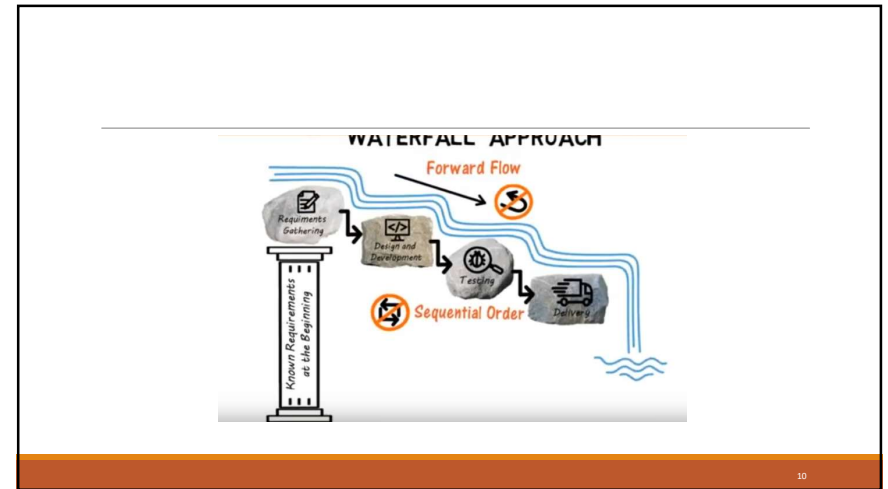
WATERFALL APPROACH



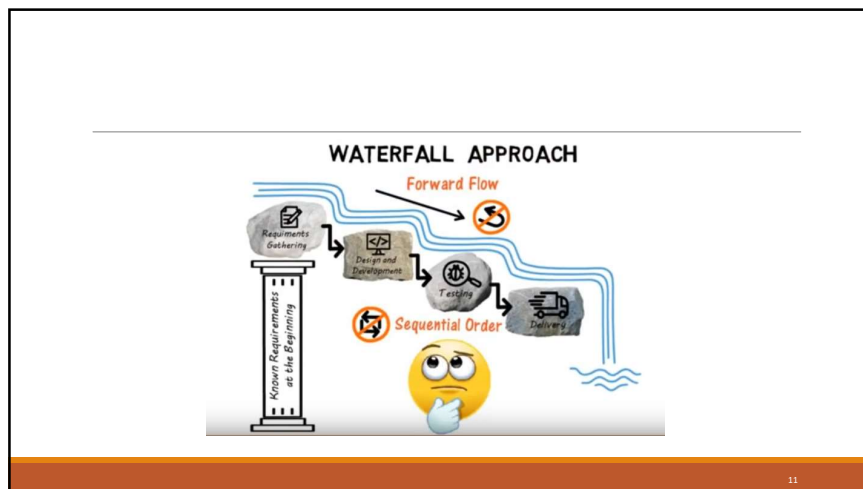
8



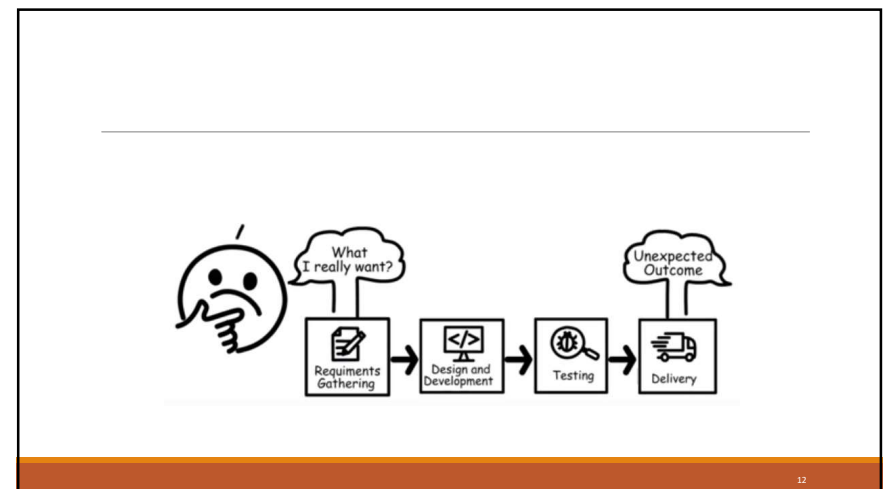
9



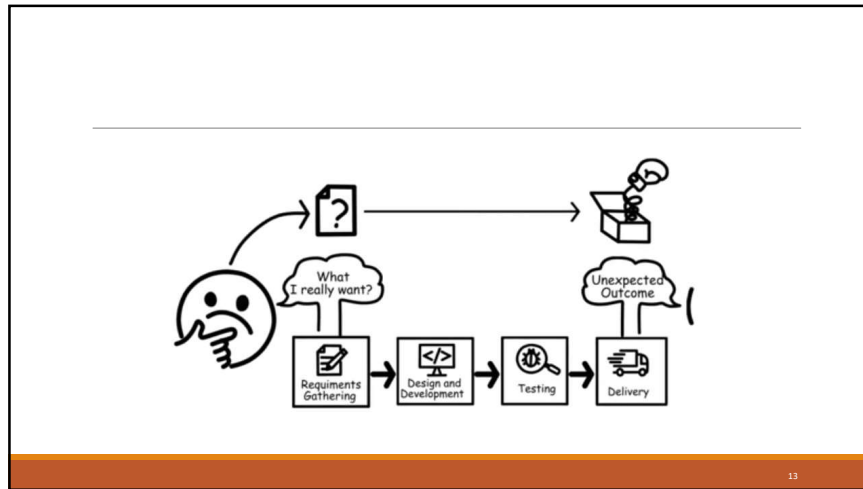
10



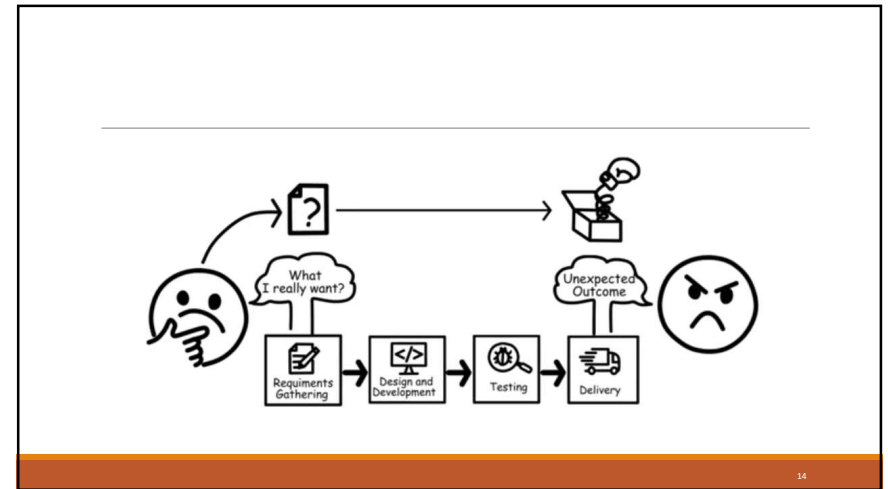
11



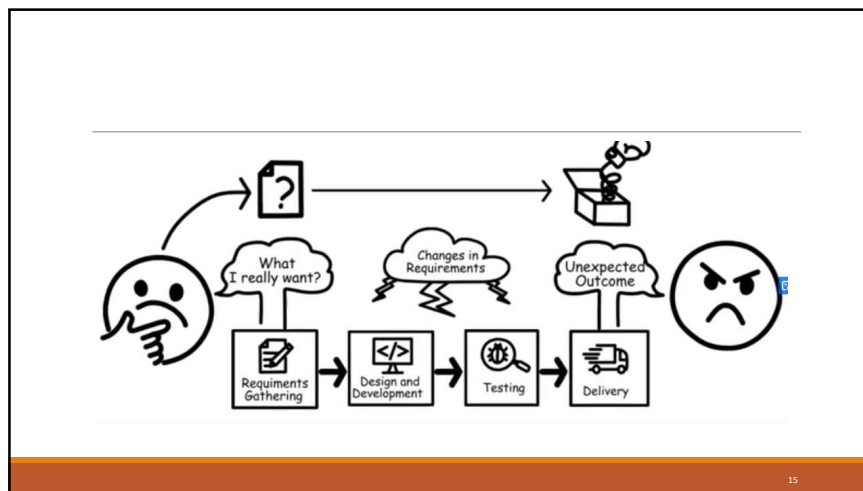
12



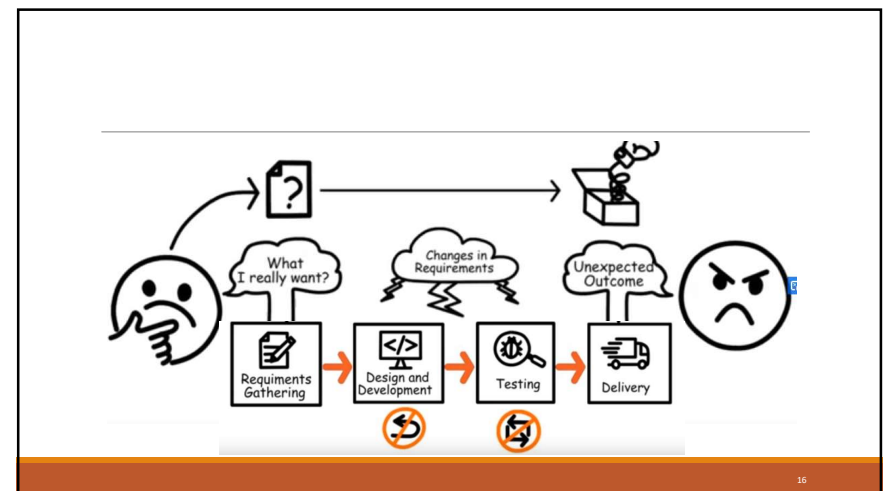
13



14



15



16



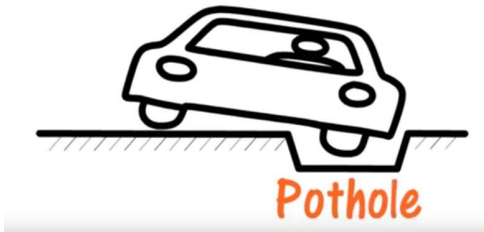
17

Öngörüsöl (predictive) yaşam döngüsü

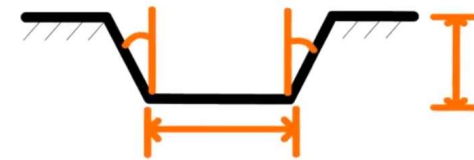
- Geleneksel «çağlayan» yaklaşımları
 - Gereksinimler
 - Sabit takım
 - Düşük risk
- } gibi konularda kesinlik
- Bunun sonucunda proje aktiviteleri doğrudan seri bir şekilde uygulanır
 - Takım detaylı bir plan doğrultusunda ilerler, kapsam, zaman ve bütçeyi etkileyebilecek değişiklikleri izler ve kontrol eder
 - Projeler, projenin sonunda kadar iş değeri ortaya koyamaz



18



19



20

17

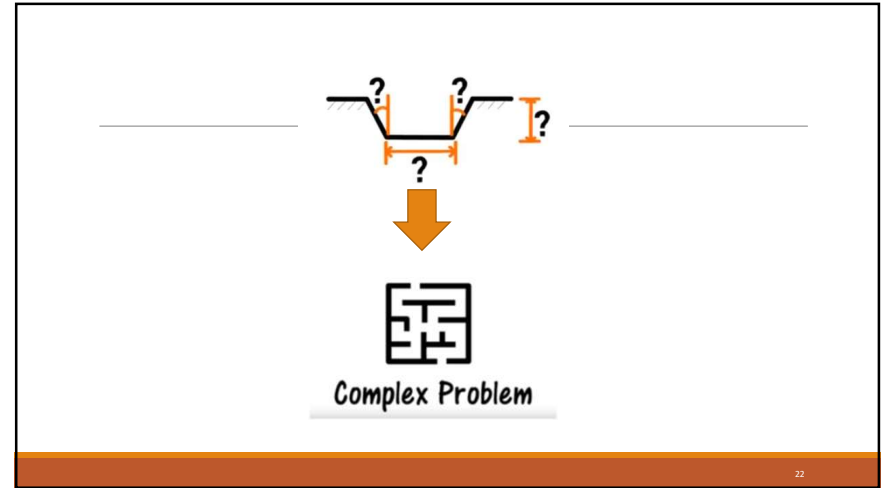
18

19

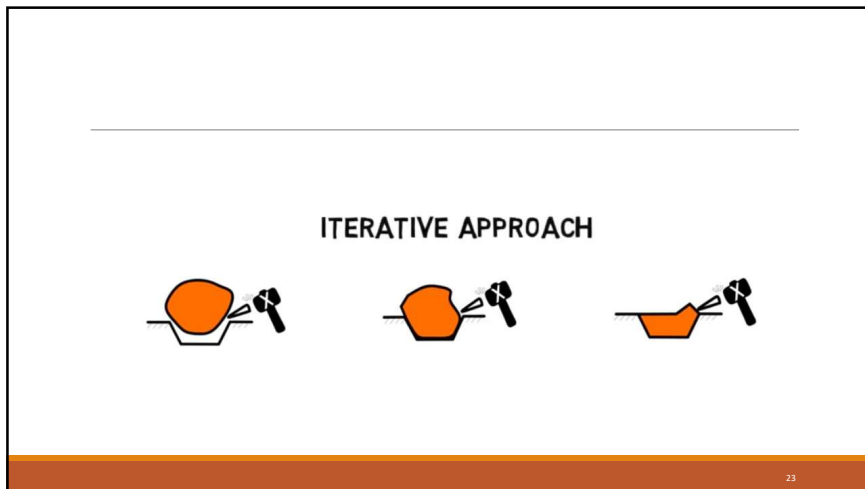
20



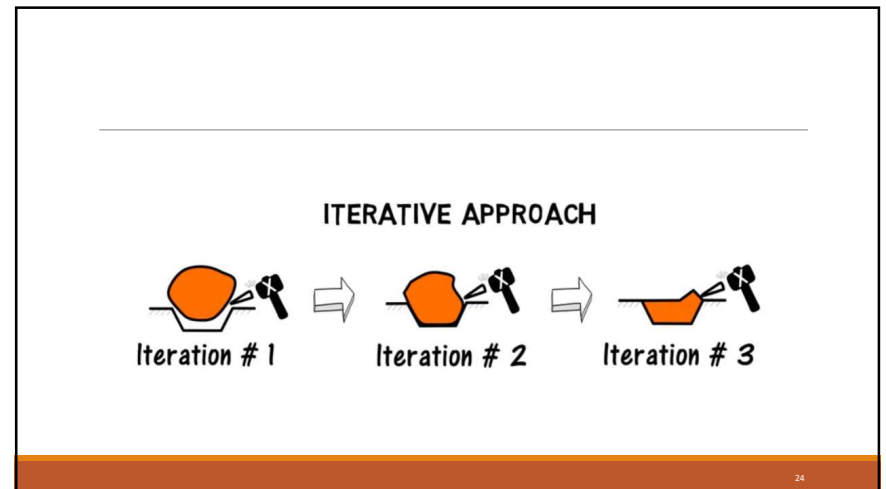
21



22



23



24

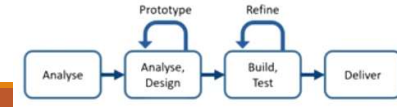
Yaşam döngüsü özellikleri

Yaklaşım	Gereksinim	Aktiviteler	Teslimat	Amaç
Öngörüsöl (predictive)	Sabit	Tüm proje için tek sefer gerçekleştirilir	Tek teslimat	Maliyet yönetimi
Tekrarlayan (iterative)	Dinamik	Doğruya ulaşana dek tekrar edilir	Tek teslimat	Çözümün doğruluğu
Artırmalı (incremental)	Dinamik	Her bir artım için tek sefer gerçekleştirilir	Sık ve küçük teslimatlar	Hız
Çevik (agile)	Dinamik	Doğruya ulaşana dek tekrar edilir	Sık ve küçük teslimatlar	Sık teslimat ve geri bildirimler ile müşteri değeri

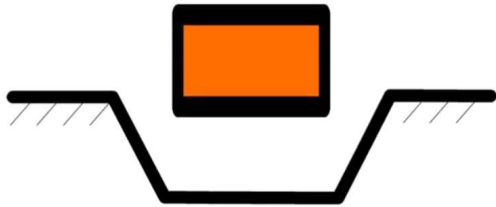
25

Tekrarlayan (iterative) yaşam döngüsü

- Birbirini izleyen **prototipler** ürün ya da sonucu iyileştirir
- **Kavram kanıtları (proof of concepts)** ile
- Her yeni prototip yeni bir paydaş geribildirimini ya da takım bilgilerine yol açar
- Takımlar bir tekrarlama için **zaman-sınırlı (time-boxed) 2-4 haftalık süreler** kullanır
- Prototip yapar, geri bildirim alır ve bunu bir sonraki tekrarlama da kullanır
- Özellikle **karmaşıklık çoksa** ve **kapsam değişime açık** ise kullanışlıdır
- Tekrarlayan yaşam döngüleri **hızlı teslimattan ziyade öğrenmeye optimize** edildiklerinden daha uzun sürer



26



27

27



28

28

INCREMENTAL APPROACH



29

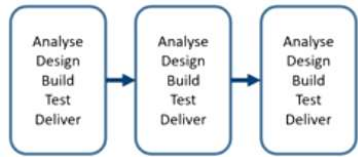
Yaşam döngüsü özellikleri

Yaklaşım	Gereksinim	Aktiviteler	Teslimat	Amaç
Öngörüsöl (predictive)	Sabit	Tüm proje için tek sefer gerçekleştirilir	Tek teslimat	Maliyet yönetimi
Tekrarlayan (iterative)	Dinamik	Doğruya ulaşana dek tekrar edilir	Tek teslimat	Çözümün doğruluğu
Artırımlı (incremental)	Dinamik	Her bir artım için tek sefer gerçekleştirilir	Sık ve küçük teslimatlar	Hız
Çevik (agile)	Dinamik	Doğruya ulaşana dek tekrar edilir	Sık ve küçük teslimatlar	Sık teslimat ve geri bildirimler ile müşteri değeri

30

Artırımlı (incremental) yaşam döngüsü

- Her şeyin **tamamlanmasını beklemeyi karşılayamayacak** projeler
- Çözümün tamamı yerine **küçük bir kümesine** karşılık gelen **sık ve küçük sürümler**
- Her bir seferinde tek bir özellik sunma ya da ekleme
- Bu yöntemde de artımlar yine **zaman sınırlıdır (time-boxed)**



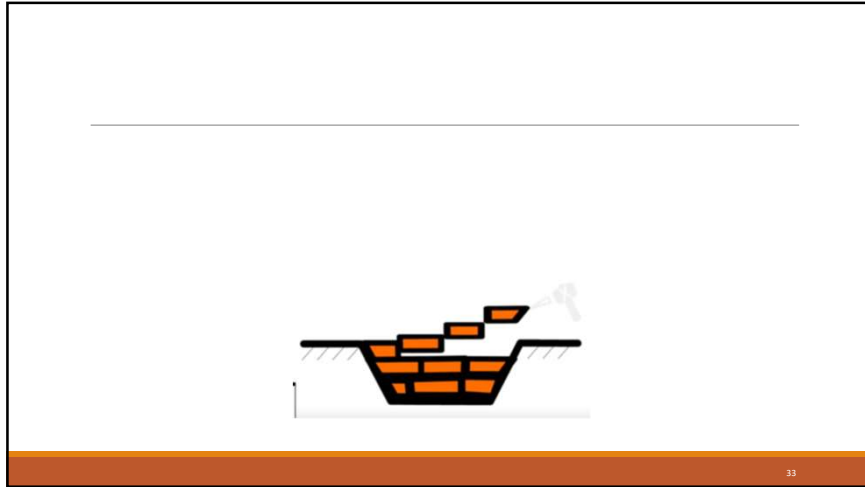
31



32

31

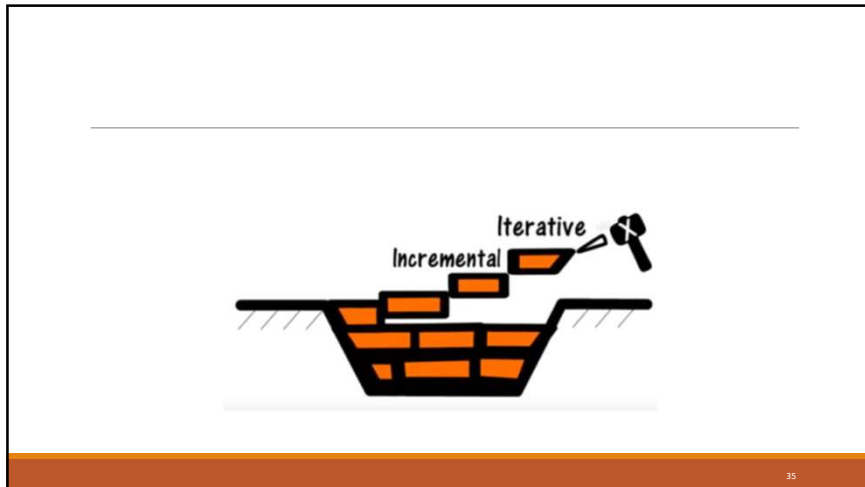
32



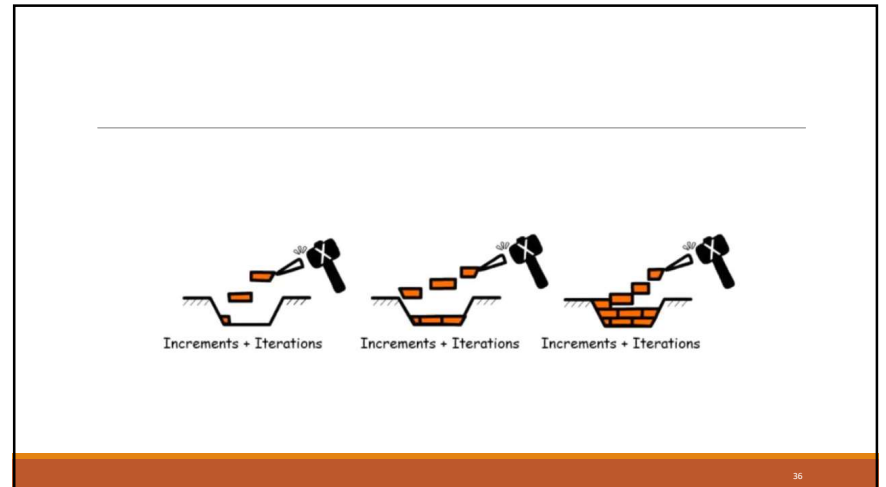
33



34

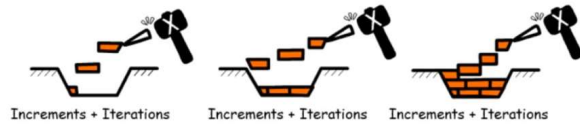


35



36

AGILE APPROACH



37

Yaşam döngüsü özellikleri

Yaklaşım	Gereksinim	Aktiviteler	Teslimat	Amaç
Öngörüsöl (predictive)	Sabit	Tüm proje için tek sefer gerçekleştirilir	Tek teslimat	Maliyet yönetimi
Tekrarlayan (iterative)	Dinamik	Doğruya ulaşana dek tekrar edilir	Tek teslimat	Çözümün doğruluğu
Artırmalı (incremental)	Dinamik	Her bir artım için tek sefer gerçekleştirilir	Sık ve küçük teslimatlar	Hız
Çevik (agile)	Dinamik	Doğruya ulaşana dek tekrar edilir	Sık ve küçük teslimatlar	Sık teslimat ve geri bildirimler ile müşteri değeri

38

Çevik (agile) yaşam döngüsü

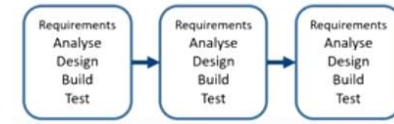
- Takım gereksinimlerin değişmesini bekler
- Tekrarlanan ve artırmalı yaklaşımlar beraber kullanılır -
 - Bir sonraki tekrarlamayı planlamak için **geri bildirim** sağlanır (Yine 2-4 haftalık tekrarlamalar)
 - Artımlar kullanarak da **gizli ve yanlış anlaşılan gereksinimler** ortaya çıkarılır
- Kanban ya da akış tabanlı çevik yöntemlerde ilerleyen iş (Work In Progress) ya da tahtadaki özellik sayısı tamamlanana kadar tekrarlama kullanılır
 - Takım tahtadaki backlog (ürün iş listesi) sütunundan kendi kapasitesine göre bir özellik seçer



39

Çevik (agile) yaşam döngüsü

- Erken ve sürekli **değer teslimi** ile **müşteri memnuniyeti artar**
- İlerlemenin ana ölçümü değer yaratan artırmalı ve **fonksiyonel teslimatlar**dır



40

39

40

Hibrit yaşam döngüsü

➤ Öngörülse, tekrarlanan , artımlı ya da çevik yaklaşımları bir arada kullanan yaklaşımlardır.

➤ Örnek

➤ Bir miktar öngörülse yoğunlukla çevik

➤ Farklı bir tedarikçi tarafından geliştirilen bir bileşeni entegre etme durumunda ➔ teslim edilen bileşen tek bir tekrarlama ile gelebilir

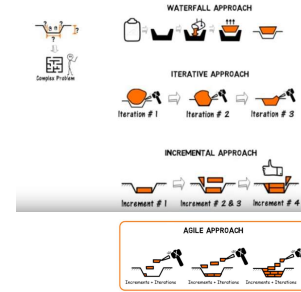
➤ Bir miktar çevik yoğunlukla öngörülse

➤ Basit bir proje geliştirirken yeni bir bileşenin olması durumu

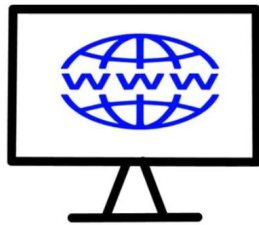
➤ Birleşik öngörülse ve çevik yaklaşım

➤ En sonunda bir proje ürünü teslim edilecek olsa bile görevlerin Kanban ile takip edildiği veya güncellemeler için günlük Scrum toplantılarının kullanıldığı lineer bir proje

41

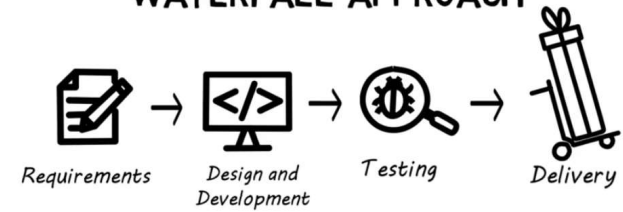


42



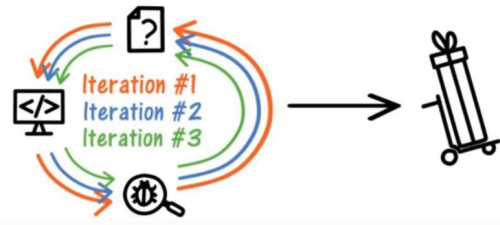
43

WATERFALL APPROACH



44

ITERATIVE APPROACH



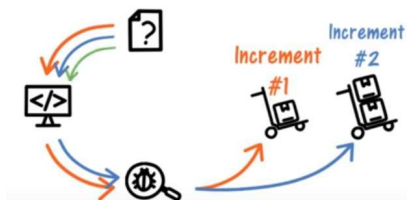
45

Increment



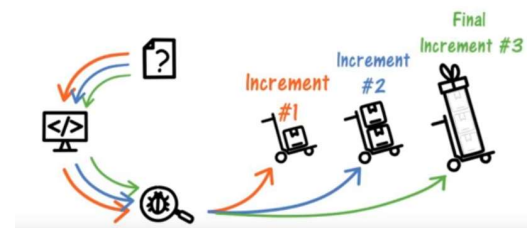
46

INCREMENTAL APPROACH

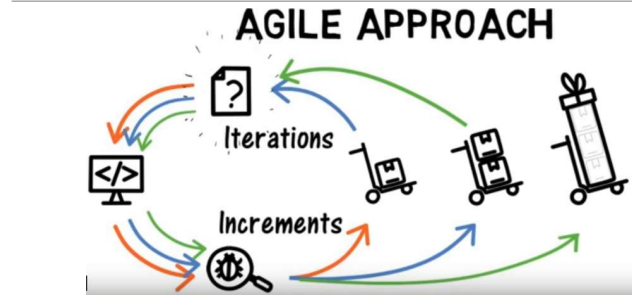


47

INCREMENTAL APPROACH



48



49

Çevik Manifesto ve düşünce yapısı

50

Agile Manifesto

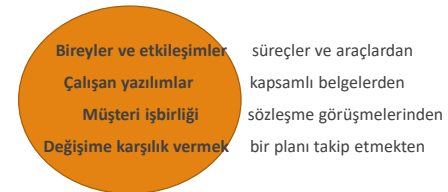
2001 yılında bir grup bağımsız yazılım geliştirici hantal olmayan yazılım geliştirme metodolojilerini bir araya getirerek günümüzde Agile Manifesto olarak bilinen ortak değer ve prensipler kümesini ortaya koymuşlardır

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

51

Agile Manifesto

Yazılımı geliştirerek ve başkalarının bunu yapmasına yardımcı olarak yazılım geliştirmenin daha iyi yollarını keşfediyoruz. Bu çalışma sayesinde:



değerlidir.

Yani, soldaki öğeler sağdaki öğelerden daha fazla değer verdiğimiz konulardır.

52

12 Çevik Prensip

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
1. En yüksek önceliğimiz, değerli yazılımların **erken ve sürekli teslimi** ile müşteriyi memnun etmektir.
2. Geliştirmenin geç aşamasının bile değişen gereksinimlere hoş karşılanır. Çevik süreçler, müşterinin rekabetçi avantajı için **değişimi** kullanır.
3. Çalışan yazılımı sıklıkla ve tercihen birkaç haftadan birkaç aya kadarlık **kısa zaman aralıklarında** teslim edin.
4. İş insanları ve geliştiriciler proje boyunca **günlük olarak birlikte çalışmalıdır**.

53

53

12 Çevik Prensip

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
5. Projeleri motive olmuş bireyler etrafında oluşturun. Onlara ihtiyaç duydukları ortamı ve desteği verin ve işi yapmaları için **onlara güvenin**.
6. Bir geliştirme ekibine ve ekibin içinde bilgi aktarmanın en etkili ve etkili yöntemi **yüz yüze görüşmedir**.
7. **Çalışan yazılım** ilerlemenin birincil ölçüsüdür.
8. Çevik süreçler sürdürülebilir geliştirmeyi teşvik eder. Sponsorlar, geliştiriciler ve kullanıcılar süresiz olarak **sabit bir hızı** korumalıdır.

54

54

12 Çevik Prensip

9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity—the art of maximizing the amount of work not done—is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
9. Teknik mükemmelliğe ve iyi tasarıma **sürekli dikkat**, çevikliği artırır.
10. **Basitlik** - yapılmayan iş miktarını en üst düzeye çıkarma sanatı - esastır.
11. En iyi mimariler, gereksinimler ve tasarımlar, **kendi kendini düzenleyen ekiplerden** ortaya çıkar.
12. Düzenli aralıklarla ekip, nasıl daha etkili olabileceğini **düşünür**, ardından davranışını buna göre ayarlar ve uyarlar.

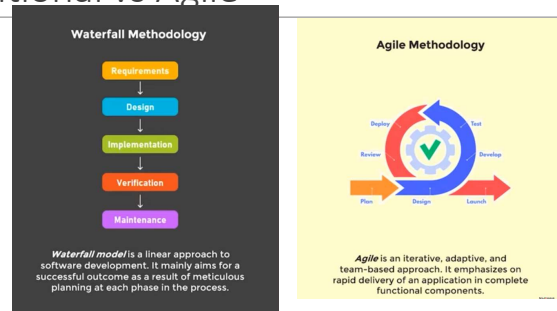
55

55

SCRUM FRAMEWORK

56

Traditional vs Agile



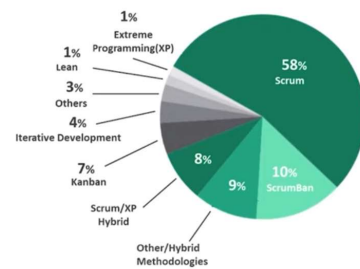
57

Agile Software Development



58

Agile Frameworks and Methodologies



Source: VersionOne's 14th Annual State of Agile Report

59

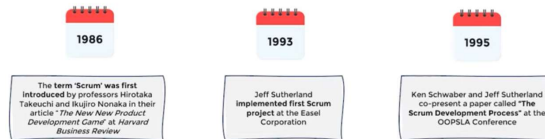
What's Scrum

Scrum Guide: **Scrum** is an Agile Framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value

- Lightweight
- Simple to understand
- Difficult to master

60

Scrum History

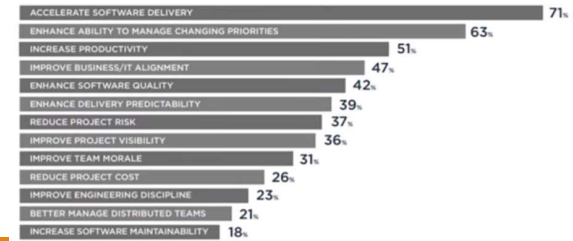


61

Benefits of Scrum

Scrum was initially developed for managing and developing products.

Currently, it is being used extensively worldwide across every industry in the world.



62

Scrum Roles

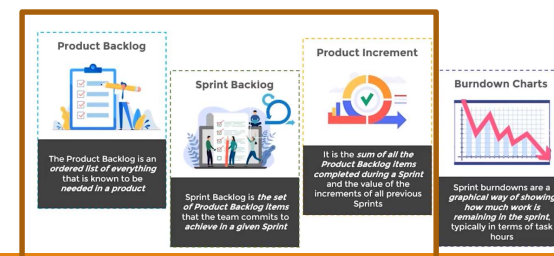
Scrum Teams are self-organizing and cross-functional



63

Scrum Artifacts

Scrum defines three artifacts: Product Backlog, Sprint Backlog, and a potentially releasable product increment to help manage work



64

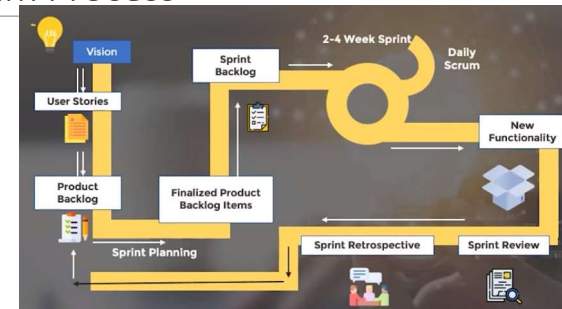
Scrum events (Scrum ceremonies)

Sprint is a time-box of one month or less during which a usable and potentially releasable product increment is created.



65

Scrum Process



66

Scrum Process



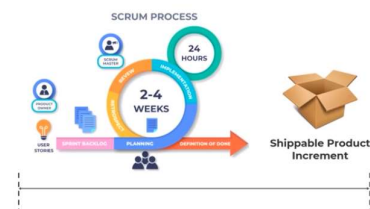
1. Product owner creates a product backlog
2. Scrum team attends Sprint Planning meeting to set the sprint goal
 - The subset of items from product backlog becomes sprint backlog
3. The team meets in the daily Scrum to access the progress of the sprint
 - Scrum master keeps the team focused on the goal
4. At the end of the sprint, sprint review is organized by product owner
5. After the sprint review, the scrum team gathers-up in Sprint Retrospective meeting
6. The team repeats steps 2-5 until the desired outcome of the product have been met

67

Scrum Process

In Scrum, you break down the phases of your project into smaller pieces that can be completed by a cross-functional team within a prescribed time period (called a **sprint**).

Once the sprint begins, you aren't allowed to add any new requirements.



The duration of the sprint in scrum is anywhere from **2 to 4 weeks to almost a month**

68

Scrum Process

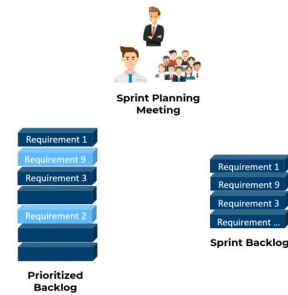


- Product owner
- creates the initial product backlog
 - prioritized each item



69

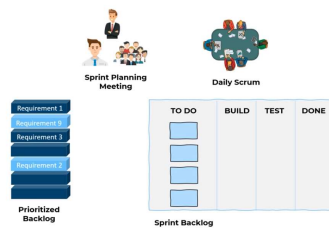
Scrum Process



- Product owner and Development Team
- creates the sprint backlog by pulling a chunk of requirements from prioritized product backlog.

70

Scrum Process

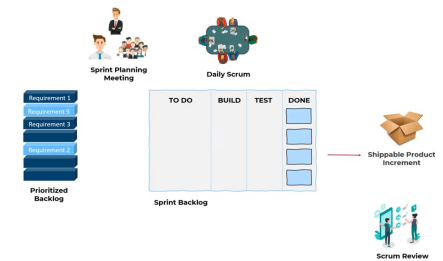


- Development Team
- Uses Daily Scrum meetings to assess its progress
 - Uses a Scrum board as a progress tracking tool
 - Visual representation of workflow

71

Scrum Process

- Scrum Team, at the end of each Sprint
- delivers a shippable product—functioning piece of product to show to customer
 - holds a Scrum Review meeting
 - Also conduct Sprint Retrospective meeting to analyze their own working progress to evolve



72

Scrum avantajları

- ✓ Hızlı ve Düzenli Teslimat
- ✓ Esneklik ve Adaptasyon
- ✓ Müşteri Memnuniyeti
- ✓ Daha İyi İletişim ve İşbirliği
- ✓ Risklerin Azaltılması
- ✓ Şeffaflık ve Sorumluluk

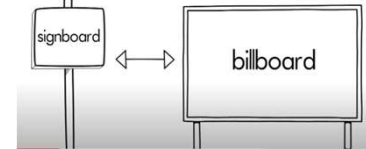
73

KANBAN

visual management
method

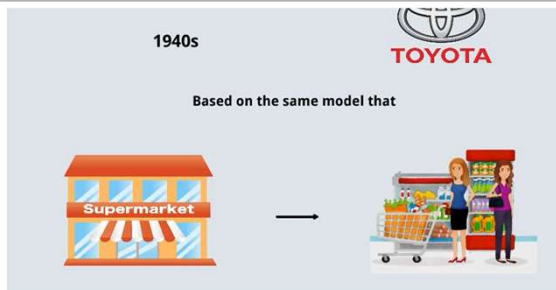


Kanban 看板



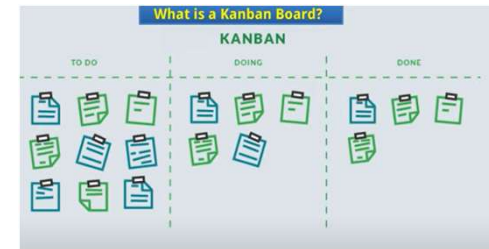
74

Tarihçe

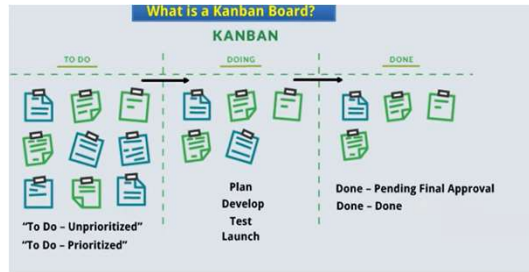


75

Kanban Tahtası

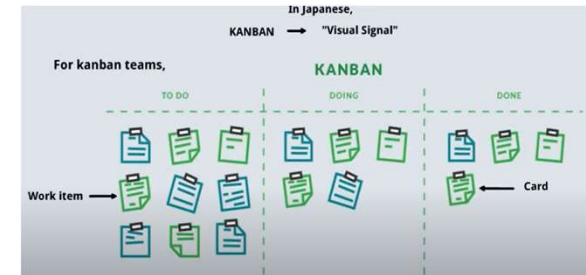


76



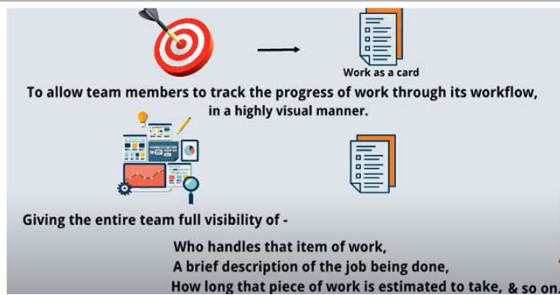
77

Kanban Kartları



78

Kanban Kartları



79

Kanban Pratikleri

1. İş akışını görselleştirin
2. İşlem sınırlarını belirleyin
3. Akışı yönetin
4. Süreç politikalarını açık hale getirin
5. Geribildirim alın
6. İyileştirme için işbirliği yapın ve deneysel olarak gelişin

80

Kanban faydaları

1. Sorunları Tanımlama
2. Esneklik
3. Zaman Tasarrufu Sağlayabili
4. Takımlara Güç Verir
5. Üretkenliği Dengeler
6. Artan Müşteri Memnuniyeti