Çevik Yöntemler
(Agile Methodology)

Prepared By: İsmail Yarar

Çevik Yöntem Nedir?

Çevik Yöntem (Agile Methodologies), yazılım geliştirme süreçlerinde değişen gereksinimlere hızlı bir şekilde yanıt vermeyi ve iterasyonlar aracılığıyla sürekli değer üretmeyi hedefleyen esnek yöntemlerdir.

Çevik Yöntemin resmi doğuşu, 2001 yılında Amerika'nın Utah eyaletinde bir grup yazılım uzmanının bir araya gelip 12 prensipten oluşan **Çevik Manifesto'**yu yayınlamasıyla gerçekleşti.

Çevik Manifestonun Temel İlkeleri

İnsanlar Arasındaki İletişim ve İşbirliği, kullanılan süreçler ve araçlardan daha önemlidir

Çalışan Yazılım, kapsamlı dokümantasyondan daha önemlidir.

Müşteri İşbirliği, sözleşme müzakeresinden daha önceliklidir.

Değişikliklere Hızlıca Uyum Sağlamak, önceden belirlenen bir planı takip etmekten daha önemlidir.

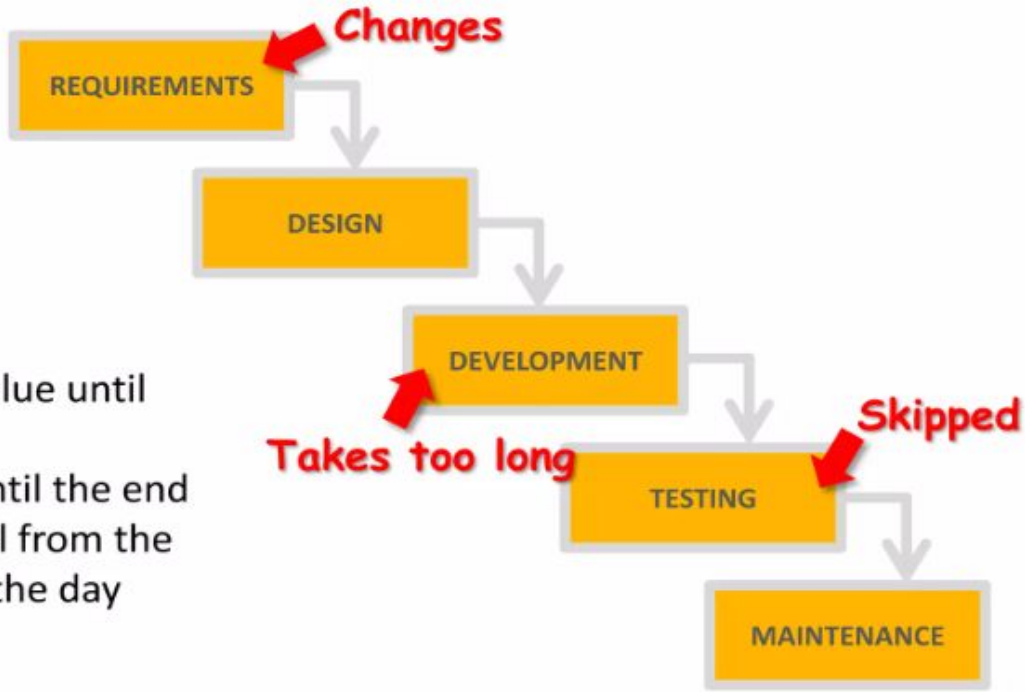
12 AGILE PRINCIPLES BEHIND THE AGILE MANIFESTO

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.	5 Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	6 Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
7 Working software is the primary measure of progress.	8 The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	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.

Neden Çevik Yöntemler?

Çevik yöntemler, geleneksel "**Waterfall**" (şelale) modelinden farklı olarak, proje aşamalarını adım adım ve sırayla değil, iteratif ve döngüsel bir şekilde yürütür. Waterfall modelinde, yazılım geliştirme süreci analiz, tasarım, geliştirme, test ve dağıtım aşamalarından geçer ve her aşama tamamlanmadan bir sonrakine geçilmez.

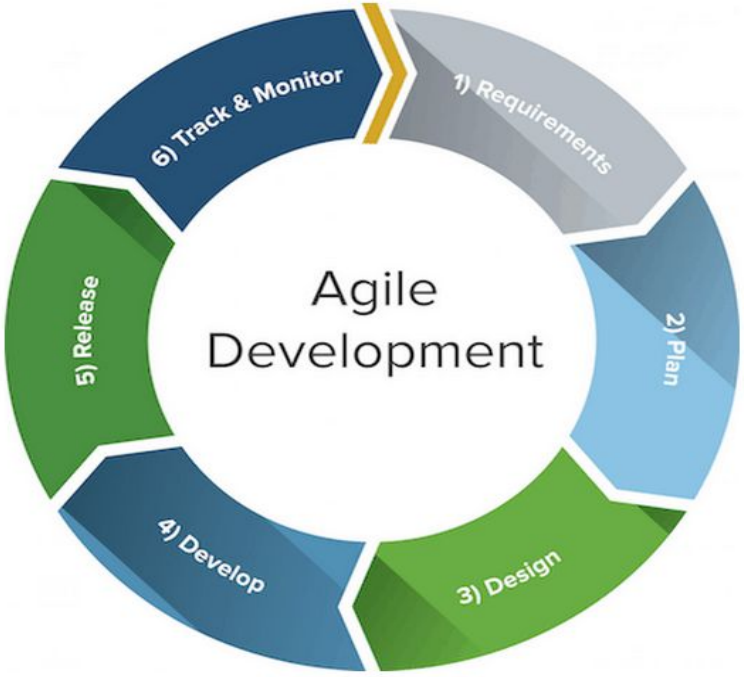
Çevik yöntemlerde ise sürekli geri bildirim alarak esnek bir şekilde ilerlenir ve her döngüde güncellemeler yapılabilir.



- You don't realize any value until the end of the project
- You leave the testing until the end
- You don't seek approval from the stakeholders until late in the day

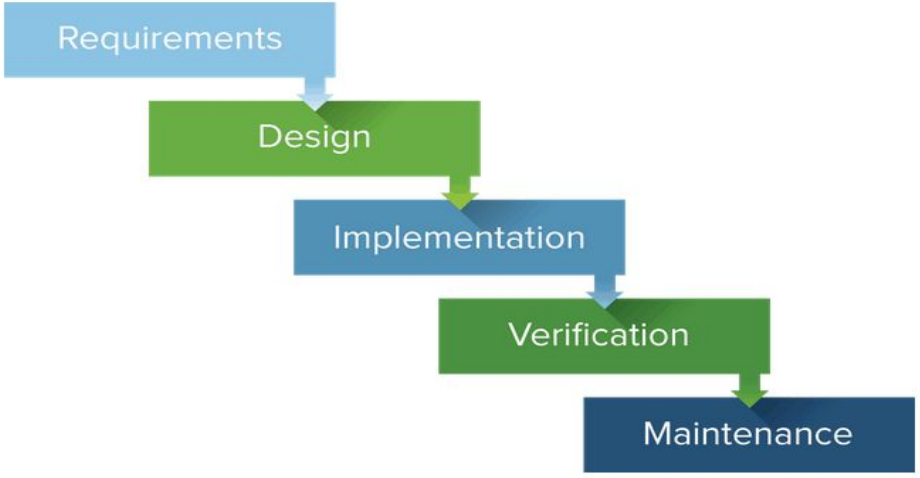
This approach is **highly risky**, often more **costly** and generally **less efficient** than **Agile** approaches

Agile



- Continuous cycles
- Small, high-functioning, collaborative teams
- Multiple methodologies
- Flexible/continuous evolution
- Customer involvement

Waterfall



- Sequential/linear stages
- Upfront planning and in-depth documentation
- Contract negotiation
- Best for simple, unchanging projects
- Close project manager involvement

Çevik Yöntemler ve Metodolojiler

Scrum: Kısa sprintler ile iteratif geliştirme süreci.

Kanban: Görsel iş akışını yönlendiren bir metodoloji .

Extreme Programming (XP): Kaliteyi artırmak ve değişen gereksinimlere hızla yanıt vermek için yoğun geliştirme teknikleri.

1.SCRUM

Scrum, kısa ve sabit süreli çalışmalardan (sprintler) oluşan, iteratif bir geliştirme sürecidir. Her sprint, genellikle 1-4 hafta arasında değişir ve belirlenen hedeflerin tamamlanması için yoğun bir iş temposunda çalışılır.

Product Owner: Mimaride kritik kararlar alınırken, iş gereksinimlerini ve öncelikleri belirleyen kişidir.

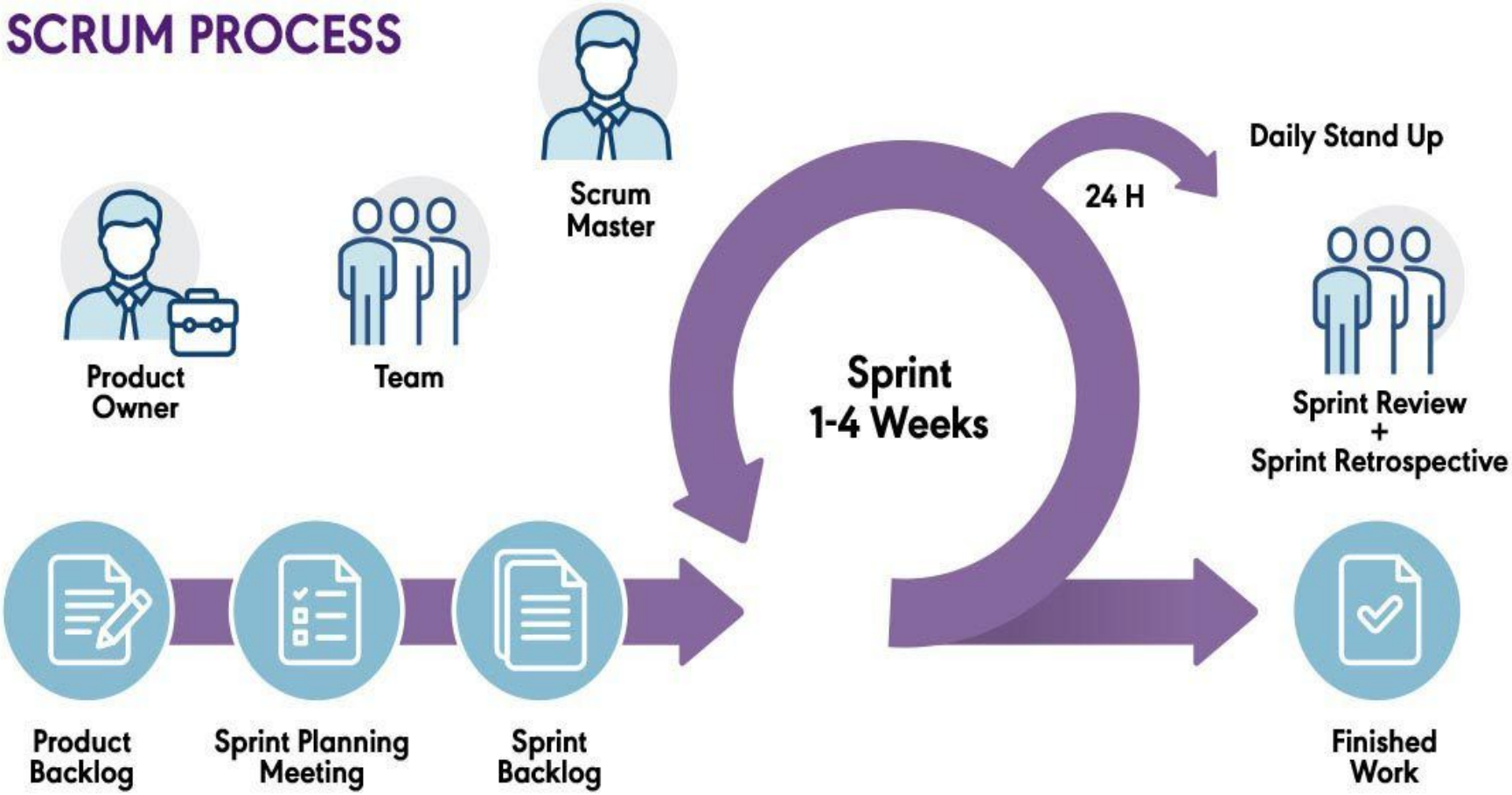
Scrum Master: Scrum süreçlerinin düzgün işleyişini sağlar, ekibin mimariyi etkili bir şekilde geliştirebilmesi için engelleri ortadan kaldırır ve ekibi motive eder.

Development Team: Mimari tasarımın gerçekleştiren ekiptir.

SCRUM Terminolojileri

- Proje veya ürün, özelliklerin bir listesi olarak tanımlanır: **Backlog**.
- Özellikler, kullanıcı hikayeleri (user stories) cinsinden tanımlanır.
- Scrum ekibi, her hikaye ile ilişkili işi tahmin eder.
- Backlog'daki özellikler, önem sırasına göre sıralanır.
- **Sonuç**: Sıralı ve ağırlıklı bir özellik listesi, bir yol haritası (roadmap) elde edilir.
- **Günlük Scrum toplantısı**: Ekibin ne yaptığını, ne yapacağını ve karşılaştığı engelleri konuştuğu kısa bir toplantı.

SCRUM PROCESS

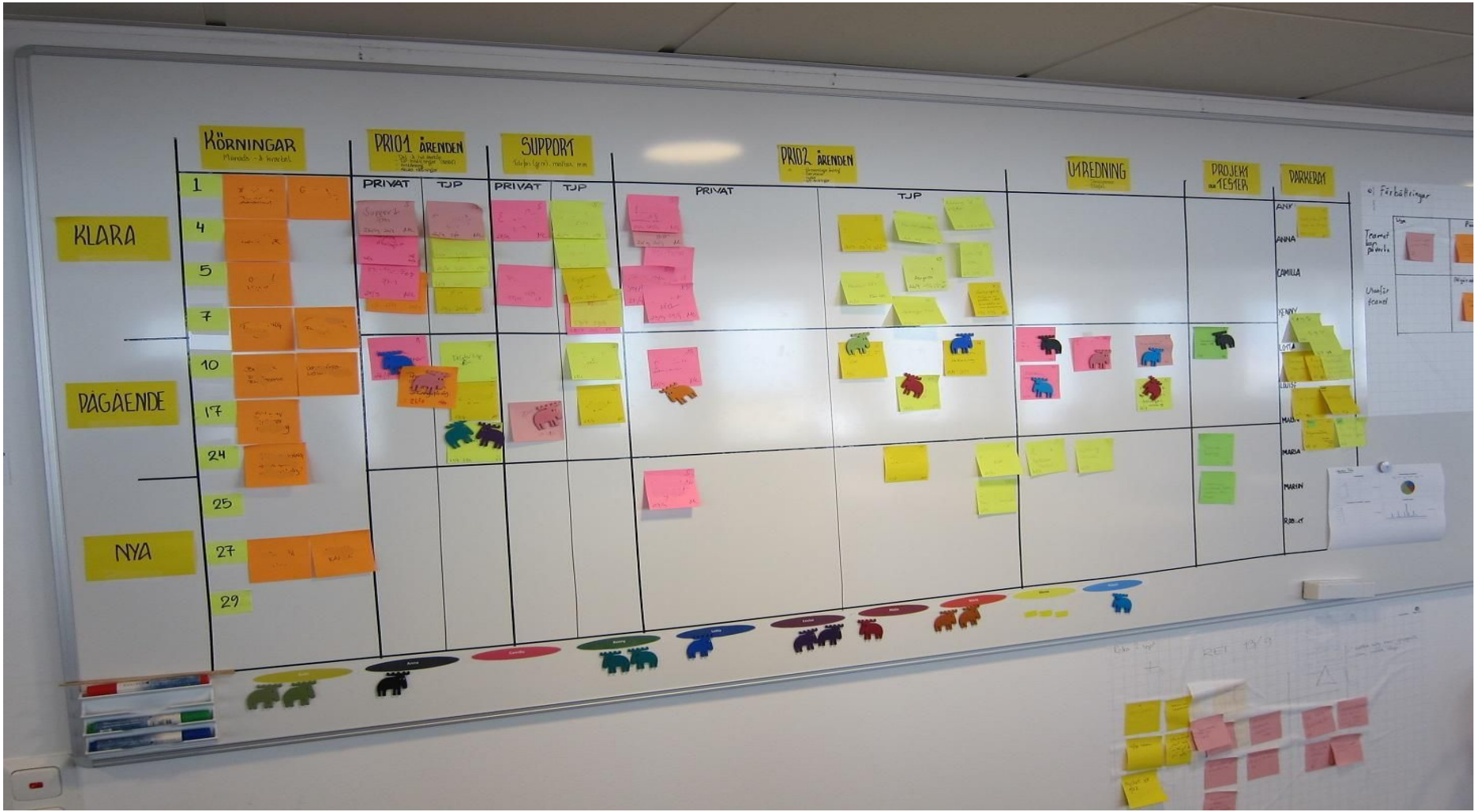


2.KANBAN

Kanban, görsel bir iş akışı yönetim sistemidir. Her bir görevin ilerleyişi bir Kanban panosu üzerinde gösterilir. Bu sistemde iş parçacıkları sürekli akış içerisinde olur ve tamamlanan işler hızla teslim edilir. Temel amaç, iş akışını optimize etmek ve darboğazları ortadan kaldırmaktır.


Stories
To do
In progress
Test
Done


KANBAN BOARD
Stories
To Do
In Progress
Testing
Done


KÖRNINGAR
PRIO1 ÄRENDEN
SUPPORT
PRIO2 ÄRENDEN
UTREDNING
PROJEKT och TESTER
PARKERAT
PRIVAT
TJP
KLARA
PÅGÅENDE
NYA
1
4
5
7
10
17
24
25
27
29
ANNA
CAMILLA
MARIA
MARIN

3.Extreme Programming (XP)

Extreme Programming (XP), yazılım geliştirmede kaliteyi artırmak ve değişen gereksinimlere hızla yanıt vermek amacıyla yoğun teknik uygulamalar kullanan bir çevik yöntemdir. XP nin temel felsefesi, hızlı değişikliklere esnek bir şekilde adapte olabilmek ve müşteri memnuniyetini artırmaktır.

Extreme Programing Terminolojileri

Continuous Integration: Developerlar kodlarını sürekli olarak ana koda entegre eder ve bu entegrasyonlar her gün birden fazla kez gerçekleşir. Bu da yazılım mimarisinin sürekli olarak test edilmesine ve hataların erken aşamada yakalanmasına olanak tanır.

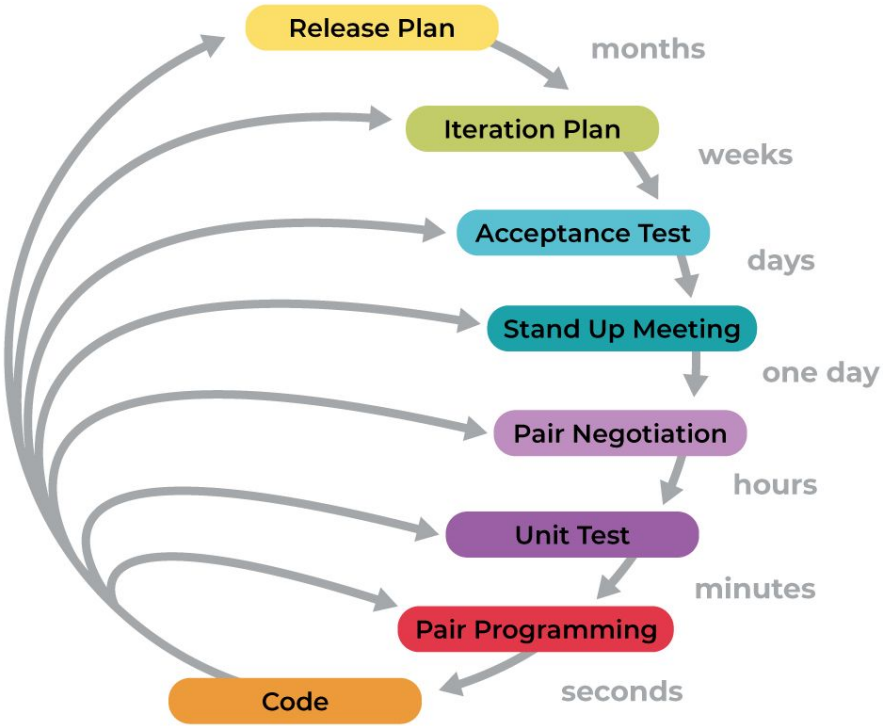
Pair Programming: İki developer'ın aynı kod üzerinde birlikte çalışmasıyla yapılan bu teknik, daha yüksek kod kalitesini sağlar. Pair Programming sırasında bir geliştirici kodu yazar, diğeri ise sürekli olarak denetler ve yönlendirir. Bu yöntem, yazılım mimarisinin daha dikkatli bir şekilde tasarlanmasına ve olası sorunların erken aşamada giderilmesine yardımcı olur.

Test-Driven Development - TDD: XP'de, developerlar önce testleri yazar ve sonra bu testleri geçecek şekilde kodu geliştirirler. Bu yaklaşım, mimari tasarımın baştan itibaren test edilebilir olmasını sağlar.

Continuous Feedback: Müşterilerden ve kullanıcıdan alınan geri bildirimler, mimaride düzenli olarak değişiklik yapılmasına ve mimarinin müşteri ihtiyaçlarına göre sürekli optimize edilmesine olanak tanır.

Small Releases: XP'de yazılım, küçük parçalara bölünerek hızlı bir şekilde teslim edilir. Bu da mimariyi aşamalı olarak geliştirir, her sürümde yeni öğeler eklenir ve test edilir. Yazılım mimarisi bu şekilde kademeli olarak büyütülür ve gereksinimlere göre uyarlanabilir hale gelir.

Extreme Programming
Planning / Feedback Loops



Extreme programming	Waterfall methodology
It is an iterative process that runs in the same lifecycle repeatedly until the final product is created.	It involves breaking down project activities into linear sequential phases, each of which is dependent on the previous one and categorized accordingly.
The planning begins at the start of the development cycle.	There is a dedicated planning phase that is independent of the development cycle.
The testing is performed before the actual code is implemented.	The testing is performed once the development requirements are completed.
This approach works when the project requirements change in every iteration.	This approach works when the project requirements and overall scope is fixed.

Günümüzde Çevik Yöntemler

Son yıllarda, çevik yöntemler **DevOps**, **Yapay Zeka** ve **Makine Öğrenimi** gibi teknolojiler ile daha da entegre hale geldi.

DevOps, geliştirme ve operasyon süreçlerini birleştirerek daha hızlı yazılım dağıtımı sağlarken, AI ve ML ise test otomasyonu, tahminsel analizler ve yazılım kalitesinin artırılmasında çevik sürecin hızını ve doğruluğunu iyileştiriyor.

Ayrıca, uzaktan çalışma kültürünün yaygınlaşması ile birlikte dijital işbirliği araçları (örneğin Jira, Trello, Slack) çevik ekiplerin daha verimli çalışmasına olanak tanıyor.

Dinlediğiniz için
Teşekkür Ederim.