Agile Overview

**Agile Project Management**

Software projects face the risk of failure during the life cycle due to misuse of resources, wrong prioritization and inaccurate understanding of customer demands. This has led companies to experiment with a variety of new methods to produce more productive projects. Agile is a new approach to project management methods resulting from these experiments. This method is based on an incremental and iterative approach that helps teams deliver a product to their customers faster and more desirable. In this respect, the framework of agile promises **significant advantages over traditional methods**. In recent years, most companies have adopted the agile approach, which they find more successful than traditional methods. In this context, we can say that almost every software developer has used the agile method in some form.

Yazılım projeleri, kaynakların yanlış kullanımı, yanlış önceliklendirme ve müşteri taleplerinin yanlış anlaşılması nedeniyle yaşam döngüsü boyunca arıza riskiyle karşı karşıyadır. Bu, şirketlerin daha verimli projeler üretmek için çeşitli yeni yöntemler denemelerine yol açtı. Agile bu deneylerden kaynaklanan proje yönetimi yöntemlerine yeni bir yaklaşımdır. Bu yöntem, ekiplerin müşterilerine daha hızlı ve daha arzulanan bir ürün teslim etmesine yardımcı olan artımlı ve yinelemeli bir yaklaşıma dayanmaktadır. Bu bağlamda, çevik çerçeve geleneksel yöntemlere göre önemli avantajlar vaat etmektedir. Son yıllarda, çoğu şirket geleneksel yöntemlerden daha başarılı buldukları agile yaklaşımı benimsemiştir. Bu bağlamda, neredeyse her yazılım geliştiricisinin agile yöntemi bir şekilde kullandığını söyleyebiliriz.

**Dividing Product Development Life Cycle into Small Components**

Agile provides a light framework for development teams. That helps them focus on fast delivery while maintaining functionality. In Agile, the product development life cycle is divided into small components (called iterations), so the product can be easily and rapidly developed and tested. In this way, changes can be made without having to wait for the final product.

**Ürün Geliştirme Yaşam Döngüsünü Küçük Bileşenlere Bölme**

Agile geliştirme ekipleri için hafif bir çerçeve sunar. Bu, işlevselliği korurken hızlı teslimata odaklanmalarına yardımcı olur. Agile'de, ürün geliştirme yaşam döngüsü küçük bileşenlere (yineleme adı verilir) ayrılır, böylece ürün kolayca ve hızlı bir şekilde geliştirilebilir ve test edilebilir. Bu şekilde, nihai ürünü beklemek zorunda kalmadan değişiklik yapılabilir.

### Customer's Desired Requirements

The agile method provides that the product is optimized throughout the development process.Using **iterative planning** and getting **feedback** ensures that a delivered product is tailored to the customer's requirements. The status of the project is measured and evaluated throughout the process to easily adapt to changing needs. Measurement and evaluation provide accurate and early visibility into the development of each project.

**Müşterinin İstenilen Gereksinimleri**

Agile yöntem, ürünün geliştirme süreci boyunca optimize edilmesini sağlar. Yinelemeli planlama ve geri bildirim almak, teslim edilen bir ürünün müşterinin gereksinimlerine göre uyarlanmasını sağlar. Projenin durumu değişen ihtiyaçlara kolayca uyum sağlamak için süreç boyunca ölçülür ve değerlendirilir. Ölçme ve değerlendirme, her bir projenin geliştirilmesinde doğru ve erken görünürlük sağlar.

### Building the Right Product

It can be said that the agile method helps companies build the **right product**. Rather than attempting to sell the product before it is released, the agile method encourages teams to improve the product during its development. This helps the company to be in the marketplace as profitable as possible. That retains the vital business value and guarantees the desirable product. This is the reason why the agile method is an appealing option for both partners and developers.

**Doğru Ürünü İnşa Etmek**

Agile yöntemin şirketlerin doğru ürünü oluşturmalarına yardımcı olduğu söylenebilir. Agile yöntem, ürünü piyasaya sürülmeden önce satmaya çalışmak yerine, ekipleri geliştirme sırasında ürünü geliştirmeye teşvik eder. Bu, şirketin pazarda mümkün olduğunca kârlı olmasına yardımcı olur. Bu, hayati ticari değeri korur ve istenen ürünü garanti eder. Agile yöntemin hem ortaklar hem de geliştiriciler için çekici bir seçenek olmasının nedeni budur.

## History of Agile

### The Agile Manifesto

In 2001, 17 independent Software Leaders met in America for brainstorming. They aimed to find out how to develop software better by using different knowledge and approaches. After two days of brainstorming, they released the Agile Manifesto.

The Agile Manifesto was a powerful statement, carefully crafted using only 68 words. Everyone agreed that the Agile Manifesto was both short and authoritative. While traditional methods advocated a stable plan and avoided changes, the manifesto focused on people, communication, the product, and flexibility.

The Agile Manifesto basically mentions the following four values:

**1. Individuals and interactions over processes and tools:**

Valuing people more highly than processes or tools is easy to understand. Because it is the human who responds to business needs and drives the development process.

**2. Working software over comprehensive documentation:**

Historically, great amounts of time have been spent on documenting the product for development and ultimate delivery. Technical specifications and requirements, interface design documents, test plans, documentation plans, and approvals required for each. The Agile Manifesto values documentation, but it values working software much more.

**3. Customer collaboration over contract negotiation:**

With traditional development models such as Waterfall, customers negotiate the requirements for the product, usually in great detail, prior to any work starts. This means the customer is involved in the process of development before development began and after it was completed, but not during the process. However, the Agile Manifesto describes a customer who is engaged and collaborates throughout the development process.

**4. Responding to change over following a plan:**

Traditional software development methods regard change as an expense, so it is to be avoided. With Agile, the shortness of an iteration cycle means priorities can be shifted from iteration to iteration and new features can be inserted into the next iteration. Agile’s view is that changes always improve a project and provide additional values.

Agile Tarihi

**Agile Manifesto**

2001 yılında 17 bağımsız Yazılım Lideri beyin fırtınası için Amerika'da bir araya geldi. Farklı bilgi ve yaklaşımları kullanarak yazılımın nasıl daha iyi geliştirileceğini bulmayı hedeflediler. İki günlük beyin fırtınasından sonra Agile Manifestosunu serbest bıraktılar.

Agile Manifesto, sadece 68 kelime kullanılarak özenle hazırlanmış güçlü bir ifadeydi. Herkes Agile Manifesto'nun hem kısa hem de yetkili olduğunu kabul etti. Geleneksel yöntemler istikrarlı bir planı savunurken ve değişikliklerden kaçınırken, manifesto insanlar, iletişim, ürün ve esnekliğe odaklandı.

Agile Manifesto temel olarak aşağıdaki dört değerden bahseder:

1. Bireyler ve süreçler ve araçlar üzerindeki etkileşimler:

İnsanlara süreçlerden veya araçlardan daha yüksek değer vermek, anlaşılması kolaydır. Çünkü iş ihtiyaçlarına cevap veren ve gelişim sürecini yönlendiren insandır.

2. Kapsamlı belgeler üzerinde çalışan yazılımlar:

Tarihsel olarak, ürünün geliştirilmesi ve nihai teslimat için belgelenmesi için büyük zaman harcanmıştır. Her biri için gerekli teknik özellikler ve gereksinimler, arayüz tasarım belgeleri, test planları, dokümantasyon planları ve onaylar. Agile Manifesto belgelere değer veriyor, ancak çalışan yazılımlara çok daha fazla değer veriyor.

3. Sözleşme görüşmesi üzerinden müşteri işbirliği:

Şelale gibi geleneksel geliştirme modelleriyle müşteriler, herhangi bir çalışma başlamadan önce ürünle ilgili gereksinimleri genellikle ayrıntılı olarak tartışırlar. Bu, müşterinin geliştirme başlamadan önce ve tamamlandıktan sonra geliştirme sürecine dahil olduğu, ancak işlem sırasında olmadığı anlamına gelir. Bununla birlikte, Agile Manifesto, geliştirme süreci boyunca meşgul olan ve işbirliği yapan bir müşteriyi tanımlar.

4. Bir planı takip ederek değişime cevap vermek:

Geleneksel yazılım geliştirme yöntemleri değişimi bir masraf olarak görür, bu nedenle bundan kaçınılmalıdır. Agile ile, bir yineleme döngüsünün kısalığı, önceliklerin yinelemeden yinelemeye kaydırılabileceği ve yeni özelliklerin bir sonraki yinelemeye eklenebileceği anlamına gelir. Agile'nin görüşü, değişikliklerin her zaman bir projeyi geliştirmesi ve ek değerler sağlamasıdır.

**Agile Principles**

In the months following the publication of the Agile Manifesto, the original signatories continued to communicate. They augmented the four values of the manifesto with the following 12 principles.

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 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. 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.
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.

**Agile İlkeleri**

Agile Manifesto'nun yayımlanmasını takip eden aylarda, orijinal imzalayanlar iletişim kurmaya devam ettiler. Manifesto'nun dört değerini aşağıdaki 12 ilke ile artırdılar:

1) En yüksek önceliğimiz, değerli yazılımların erken ve sürekli teslimi ile müşteriyi memnun etmektir.

2) Değişen gereksinimlere hoş geldiniz, geliştirme aşamasında bile. Agile süreçler, müşterinin rekabet avantajı için değişimi kullanır.

3) Çalışan yazılımı, birkaç haftadan birkaç aya kadar, daha kısa zaman ölçeğine tercih ederek sık sık verin.

4) İş adamları ve geliştiriciler proje boyunca günlük olarak birlikte çalışmalıdır.

5) Motive olmuş bireyler etrafında projeler 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) Agile süreçler sürdürülebilir kalkınmayı teşvik eder. Sponsorlar, geliştiriciler ve kullanıcılar süresiz olarak sabit bir hızda kalmalıdır.

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 organize eden ekiplerden ortaya çıkar.

12) Düzenli aralıklarla ekip, nasıl daha etkili olabileceğini düşünür, ardından davranışını buna göre ayarlar ve ayarlar.

Scrum Overview

Scrum Framework and Sprint Concept

Scrum is the most common methodology to implement the Agile. It is an iterative development model used in complex software development processes. In scrum, larger projects are divided into smaller parts that can be managed with **sprints** . Sprints are the periods from one to four weeks. It can be even a few days when needed. Steady sprint length reduces variability; a scrum team can safely predict what they can do on each sprint based on what they have done in previous sprints. The implementation of sprints allows scrum teams to make arrangements for instant improvement, rather than at the end of the project. At the end of each sprint, something remarkable is revealed. For gaining feedback from users or investors, the product produced during each iteration should be demonstrated. The scrum framework defines specific roles, artifacts, and activities for projects. The following figure shows all of these components of the scrum framework that we will discuss later one by one.

Scrum'a Genel Bakış

Scrum Çerçevesi ve Sprint Kavramı

Scrum, Agile'ı uygulamak için en yaygın yöntemdir. Karmaşık yazılım geliştirme süreçlerinde kullanılan yinelemeli bir geliştirme modelidir. Scrum'da, daha büyük projeler sprintlerle yönetilebilen daha küçük parçalara ayrılır. Sprintler bir ila dört hafta arasındaki sürelerdir. Gerektiğinde birkaç gün bile olabilir. Sabit sprint uzunluğu değişkenliği azaltır; bir scrum ekibi, önceki sprintlerde yaptıklarına dayanarak her sprint'te neler yapabileceklerini güvenli bir şekilde tahmin edebilir. Sprintlerin uygulanması, scrum ekiplerinin projenin sonunda değil, anında iyileştirme için düzenlemeler yapmalarına izin verir. Her sprint sonunda dikkat çekici bir şey ortaya çıkıyor. Kullanıcılardan veya yatırımcılardan geri bildirim almak için her bir yineleme sırasında üretilen ürün gösterilmelidir. Scrum çerçevesi projeler için belirli roller, eserler ve aktiviteler tanımlar. Aşağıdaki şekil, daha sonra tek tek tartışacağımız scrum çerçevesinin tüm bu bileşenlerini göstermektedir.

Scrum Roles

There are three main roles in scrum projects. These are the **Product Owner**, **Scrum Master**, and **Development Team**.

**Product Owner** (PO) is the business representative in the team and speaks for the needs of the project for maximizing the value delivered in each sprint. The product owner represents stakeholders and is the voice of the customer. Therefore, the product owner works together with stakeholders and prioritizes the product requirements.

**Scrum Master** coaches the team, protects the team from organizational distraction, clears any obstacles encountered and helps team members focus on what they do. Scrum Master ensures that scrum is understood well by the team members and it is working properly. Scrum Master constantly improves the team's environment. While the product owner has a directing role, Scrum Master has an enabling role in a scrum team.

A **Development team** usually consists of 3-9 people and performs daily tasks. The team is project-oriented and dedicated to the success of the project. Each team member is very talented that is, the team members are skilled in certain subjects. Each member can do more than one job on the project. Discipline and integrity are the key terms for a successful team.

Scrum Rolleri

Scrum projelerinde üç ana rol vardır. Bunlar Ürün Sahibi, Scrum Master ve Geliştirme Ekibi.

**Ürün Sahibi** (PO), ekipteki iş temsilcisidir ve her sprintte verilen değeri en üst düzeye çıkarmak için projenin ihtiyaçları için konuşur. Ürün sahibi paydaşları temsil eder ve müşterinin sesidir. Bu nedenle, ürün sahibi paydaşlarla birlikte çalışır ve ürün gereksinimlerine öncelik verir.

**Scrum Master** takıma koçluk yapar, takımı organizasyonel dikkat dağınıklığından korur, karşılaşılan engelleri kaldırır ve ekip üyelerinin yaptıklarına odaklanmasına yardımcı olur. Scrum Master, scrum'un ekip üyeleri tarafından iyi anlaşılmasını ve düzgün çalışmasını sağlar. Scrum Master ekibin çevresini sürekli iyileştirir. Ürün sahibinin yönetmenlik rolü olsa da Scrum Master'ın scrum ekibinde etkinleştirme rolü vardır.

Bir **Geliştirme ekibi** genellikle 3-9 kişiden oluşur ve günlük görevleri yerine getirir. Ekip proje odaklı ve projenin başarısına adanmış. Her takım üyesi çok yeteneklidir, yani takım üyeleri belirli konularda yeteneklidir. Her üye projede birden fazla iş yapabilir. Disiplin ve dürüstlük, başarılı bir ekip için anahtar terimlerdir.

Epic, User Story and Task

Epic, User Story and Task Concepts

The three basic terms used in scrum projects are epic, user story and task. In order to fully comprehend a scrum project, it is very important to understand what these concepts are. The hierarchy between these concepts is explained as follows.

Scrum projelerinde kullanılan üç temel terim epic, user story ve task concepts dir. Bir scrum projesini tam olarak anlamak için, bu kavramların ne olduğunu anlamak çok önemlidir. Bu kavramlar arasındaki hiyerarşi aşağıda açıklanmaktadır.

Epic

An epic refers to a set of jobs that cannot be easily achieved in a single sprint. It usually takes months to perform an epic. Normally, it refers to a series of requirements that have not yet been split into user stories. We can also consider the epic as one of the major goals for your agile team to work on, yet not simplified and divided into various tasks.

Epics are generally large-scaled and do not contain details. They must be divided into multiple user stories before being worked on. Therefore, they are located at the top of the hierarchy. An epic could be a feature of the product, customer demand or business requirement.

***Examples of Epics:***

* As a bank, we want a facial recognition system in our branches.
* As the marketing department, we want a mobile application and a website to reach more customers.

Epic

Epic, tek bir sprintte kolayca gerçekleştirilemeyen bir dizi işi ifade eder. Epic yapmak genellikle aylar alır. Normalde, henüz kullanıcı öykülerine bölünmemiş bir dizi gereksinimi ifade eder. Epic, çevik ekibinizin üzerinde çalışabileceği, ancak basitleştirilmemiş ve çeşitli görevlere bölünmemiş ana hedeflerden biri olarak düşünebiliriz.

Epic’ler genellikle büyük ölçeklidir ve ayrıntı içermez. Üzerinde çalışılmadan önce birden çok kullanıcı hikayesine bölünmelidir. Bu nedenle, hiyerarşinin en üstünde bulunurlar. Epic bir ürün, müşteri talebi veya iş gereksinimi olabilir.

Epic Örnekleri:

Banka olarak şubelerimizde yüz tanıma sistemi istiyoruz.

Pazarlama departmanı olarak, daha fazla müşteriye ulaşmak için bir mobil uygulama ve bir web sitesi istiyoruz.

User Story

At the hierarchy, the user story is between the epic and the task. It describes a software feature from the customer’s perspective and includes the **type of user**, **what they want**, and **why they want it**. Therefore, it answers the ‘who’, ‘what’ and ‘why’ in a simple language. The product owner has the responsibility of user stories.

Leaving out the technical aspect, it should describe the behavior from a user’s perspective.

**💡Tips:**

* Template of a User Story:  
  As a < type of user >, I want < some goal > so that < some reason>

***Examples of User Stories:***

* As a registered user, I want to add items to the cart so that I can purchase multiple items at once.
* As a student, I want to apply for the exam online so that I can save time.

Kullanıcı hikayesi

Hiyerarşide, kullanıcı hikayesi destansı ve görev arasındadır. Müşterinin bakış açısından bir yazılım özelliğini tanımlar ve kullanıcının türünü, ne istediklerini ve neden istediklerini içerir. Bu nedenle, basit bir dilde "kim", "neyi" ve "neden" i yanıtlar. Ürün sahibinin kullanıcı hikayeleri sorumluluğu vardır.

Teknik yönü terk ederek, davranışı kullanıcının bakış açısından tanımlamalıdır.

Tips:

Bir Kullanıcı Hikayesi Şablonu:

<Kullanıcı türü> olarak, <bazı hedefler> istiyorum, böylece <bazı nedenler>

Kullanıcı Hikayeleri Örnekleri:

Kayıtlı bir kullanıcı olarak, aynı anda birden fazla ürün satın alabilmem için sepete ürün eklemek istiyorum.

Öğrenci olarak, zaman kazanabilmem için online sınava başvurmak istiyorum.

Task

Tasks are detailed pieces of work that are necessary to realize a user story. In other words, they define the solutions for customer needs. The time for tasks can range from a couple of hours to several hours and assigned to a team member. Tasks are displayed on a scrum board for easy tracking, as shown in the figure below.

Python Usage

The scrum board helps to visualize the tasks and usually contains cards or post-it notes on a whiteboard. The board is usually divided into three categories: *to do*, *in progress*, and *done*. The team members update the board by moving task cards through the columns on the board.

Task

Task ler, bir kullanıcı hikayesini gerçekleştirmek için gerekli olan ayrıntılı çalışma parçalarıdır. Başka bir deyişle, müşteri ihtiyaçlarına yönelik çözümleri tanımlarlar. Görev süresi birkaç saatten birkaç saate kadar değişebilir ve bir ekip üyesine atanabilir. Görevler, aşağıdaki şekilde gösterildiği gibi, kolay izleme için bir scrum panosunda görüntülenir.

Python Kullanımı

Scrum panosu görevleri görselleştirmeye yardımcı olur ve genellikle beyaz tahtada kartlar veya post-it notlar içerir. Kurul genellikle üç kategoriye ayrılır: yapılacak, devam eden ve yapılan. Ekip üyeleri, görev kartlarını tahtadaki sütunlar arasında hareket ettirerek kartı günceller.

Scrum Artifacts

Introduction

In scrum, artifacts serve to capture the common understanding of the team. The Scrum framework defines three major artifacts. These concepts are **product backlog**, **sprint backlog**, and **product increment**.

Scrum Artefaktları

Giriş

Scrum'da, eserler ekibin ortak anlayışını yakalamaya hizmet eder. Scrum çerçevesi üç ana eser tanımlar. Bu kavramlar **product backlog**, **sprint backlog**, and **product increment**.

Product Backlog

The product backlog refers to the **list of everything that needs to be done to complete the project**. Beside all user stories, it also includes technical tasks. The product backlog is the responsibility of the product owner. The product owner fulfills this responsibility by creating the product backlog, **prioritizing** the requirements in the product backlog list and constantly updating this list. The product owner updates the product backlog because once a story is completed, it should be removed from the list. Sometimes, however, new stories are added, as the project grows.

Ürün İş Listesi

Ürün biriktirme listesi, projeyi tamamlamak için yapılması gereken her şeyin listesini ifade eder. Tüm kullanıcı hikayelerinin yanı sıra teknik görevler de içerir. Ürün biriktirme listesi, ürün sahibinin sorumluluğundadır. Ürün sahibi bu sorumluluğu ürün biriktirme listesi oluşturarak, ürün biriktirme listesi listesindeki gereksinimlere öncelik vererek ve bu listeyi sürekli güncelleyerek yerine getirir. Bir ürün tamamlandıktan sonra listeden kaldırılması gerektiğinden ürün sahibi ürün biriktirme listesini günceller. Ancak bazen proje büyüdükçe yeni hikayeler eklenir.

Sprint Backlog

The sprint backlog can be defined as a subset of the product backlog. The sprint backlog is generated from the product backlog during the sprint planning meeting at the beginning of each sprint. The user stories selected from the product backlog, which will be completed during the sprint constitute the sprint backlog.

The sprint backlog is **not a flexible list** like a product backlog. That means the sprint backlog is unchanged during the sprint period. Once agreed upon in the sprint planning meeting, the stories, and steps to complete them remain stable during the sprint length. If there are stories left still unfinished by the end of the sprint, they will be added back to the product backlog and addressed during the next sprint.

Sprint İş Listesi

Sprint biriktirme listesi, ürün biriktirme listesinin bir alt kümesi olarak tanımlanabilir. Sprint iş listesi, her sprint başlangıcındaki sprint planlama toplantısında ürün biriktirme listesinden oluşturulur. Sprint sırasında tamamlanacak olan ürün biriktirme listesinden seçilen kullanıcı hikayeleri, sprint biriktirme listesini oluşturur.

Sprint biriktirme listesi, ürün biriktirme listesi gibi esnek bir liste değildir. Bu, sprint birikiminin sprint döneminde değişmediği anlamına gelir. Sprint planlama toplantısında kararlaştırıldıktan sonra, hikayeler ve bunları tamamlama adımları sprint uzunluğu boyunca sabit kalır. Sprint sonunda hala tamamlanmamış hikayeler varsa, bunlar ürün biriktirme listesine geri eklenecek ve bir sonraki sprint sırasında ele alınacaktır.

Product Increment

The Product Increment is the sum of all the product backlog items finished during the sprint. In other words, by the end of each sprint, the development team creates a new software that gets built into the main product and this new software is called product increment. The product increment aims to invest in **small amounts** in the new features of the main product. This helps to shorten the time before receiving feedback.

As the name implies, product increment continues to increase within the subsequent sprints. That means each product increment includes all the previous sprint increment values as it is cumulative. It should be a **working product** at the end of each sprint, which means that the product should be ready for shipping. But that doesn’t mean it can be released in the market. It should go through a detailed testing phase and be of high quality before releasing it in the market.

Ürün Artışı

Ürün Artışı, sprint sırasında biten tüm ürün birikmiş iş kalemlerinin toplamıdır. Diğer bir deyişle, her sprint sonunda, geliştirme ekibi ana ürüne yerleştirilen yeni bir yazılım oluşturur ve bu yeni yazılıma ürün artışı denir. Ürün artışı, ana ürünün yeni özelliklerine küçük miktarlarda yatırım yapmayı amaçlamaktadır. Bu, geri bildirim almadan önce süreyi kısaltmaya yardımcı olur.

Adından da anlaşılacağı gibi, sonraki artışlarda ürün artışı artmaya devam ediyor. Bu, her bir ürün artışı, birikimli olduğu için önceki tüm sprint artış değerlerini içerdiği anlamına gelir. Her sprint sonunda çalışan bir ürün olmalı, yani ürünün nakliyeye hazır olması gerekir. Ancak bu, piyasaya sürülebileceği anlamına gelmez. Ayrıntılı bir test aşamasından geçmeli ve piyasaya sürülmeden önce yüksek kalitede olmalıdır.

Scrum Ceremonies

List of Ceremonies in Scrum

In the scrum process, a series of meetings called ceremonies are held regularly. These ceremonies aim to minimize the need for meetings that are not defined in the scrum and to ensure regularity. These ceremonies are:

1. Sprint Planning Meeting,
2. Grooming Meeting,
3. Daily Stand Up Meeting or Daily Scrum,
4. Sprint Review Meeting,
5. Sprint Retrospective Meeting.

Scrum Törenleri

Scrum'da Törenler Listesi

Scrum sürecinde düzenli olarak tören adı verilen bir dizi toplantı düzenlenir. Bu törenler, scrum'da tanımlanmayan toplantılara olan ihtiyacı en aza indirmeyi ve düzenliliği sağlamayı amaçlamaktadır. Bu törenler:

Sprint Planlama Toplantısı,

Bakım Toplantısı,

Günlük Stand Up Toplantısı veya Günlük Scrum,

Sprint İnceleme Toplantısı,

Sprint Retrospektif Toplantısı.

1- Sprint Planning Meeting

This activity takes place at the start of each sprint and all the scrum roles take part in this meeting. Normally, it lasts for one or two hours. In sprint planning meetings, the scrum team decides the scope and the goals of the sprint.

In this respect, the product owner presents to the team the prioritized list of the user stories from the product backlog. The team members discuss the stories and predict how much work the team can perform in the coming sprint. Then they select some of the stories to complete during the sprint. These selected stories constitute the sprint backlog.

1-Sprint Planlama Toplantısı

Bu aktivite her sprint başlangıcında gerçekleşir ve bu toplantıda tüm scrum rolleri yer alır. Normalde bir veya iki saat sürer. Sprint planlama toplantılarında, scrum ekibi sprint'in kapsamını ve hedeflerini belirler.

Bu bağlamda, ürün sahibi, ürün biriktirme listesinden kullanıcı hikayelerinin öncelikli listesini ekibe sunar. Ekip üyeleri hikayeleri tartışır ve ekibin önümüzdeki sprint'te ne kadar iş yapabileceğini tahmin eder. Sonra sprint sırasında tamamlamak için bazı hikayeler seçin. Seçilen bu hikayeler sprint biriktirme listesini oluşturur.

2-Grooming Meeting

The backlog grooming meeting is held to **review the backlog** and to ensure the backlog contains the appropriate items. The product owner, scrum master and some or all of the development team participate in the meeting.

Reviewing the backlog items involves; rewriting some items to be more expressive, splitting big ones into smaller items, deleting old or no more need items. That helps the team keep the product backlog tidy and getting backlog items ready for upcoming sprints. At the end of the grooming meeting, the team has a fresh prioritized product backlog.

**Tips:**

* Apart from the grooming meeting, the other four meetings are held at specific times. For example, daily stand-up meetings are held every day, and others are held at the beginning or end of each sprint. However, no specific date has been allocated for the grooming meeting. It is usually arranged in the middle of the sprint period.

2-Bakım Toplantısı

Biriktirme listesi toplantısı biriktirme listesini gözden geçirmek ve biriktirme işleminin uygun öğeleri içerdiğinden emin olmak için yapılır. Toplantıya ürün sahibi, scrum master ve geliştirme ekibinin bir kısmı veya tamamı katılır.

Biriktirme listesi öğelerinin gözden geçirilmesi; bazı öğeleri daha etkileyici olacak şekilde yeniden yazarak, büyük olanları daha küçük öğelere bölerek, eski öğeleri silebilir veya daha fazla ürüne gerek kalmaz. Bu, ekibin ürün birikmiş işlerini düzenli tutmasına ve birikmiş işlerini yaklaşan sprintler için hazırlamasına yardımcı olur. Bakım toplantısının sonunda, ekip yeni bir öncelikli ürün birikimine sahiptir.

Tips:

Bakım toplantısının yanı sıra, diğer dört toplantı belirli zamanlarda yapılır. Örneğin, günlük stand-up toplantıları her gün yapılır ve diğerleri her sprint başında veya sonunda yapılır. Ancak, bakım toplantısı için belirli bir tarih tahsis edilmemiştir. Genellikle sprint döneminin ortasında düzenlenir.

3-Daily Stand Up Meeting

The daily stand up meeting is an internal meeting for the development team. The scrum master ensures that the team has the meeting, but the development team itself is responsible for conducting the daily stand up meeting.  
  
Regardless of the length of the sprint, the maximum time allocated for this event is 15 minutes. The members of the development team give three statements about  
  
- What they completed yesterday,  
  
- What they will focus on today,  
  
- What are the things impeding them?  
  
The scrum master deals with the removal of blocking issues. Some blocking issues may also need to be removed by the product owner.

3-Günlük Stand Up Toplantısı

Günlük stand up toplantısı, geliştirme ekibi için dahili bir toplantıdır. Scrum ustası ekibin toplantıya katılmasını sağlar, ancak geliştirme ekibinin kendisi günlük stand-up toplantısının yürütülmesinden sorumludur.

Sprint uzunluğundan bağımsız olarak, bu etkinlik için ayrılan maksimum süre 15 dakikadır. Geliştirme ekibi üyeleri hakkında üç açıklama yaparlar:

- Dün tamamladıkları,

- Bugün neye odaklanacaklar,

- Onları engelleyen şeyler neler?

Scrum ustası engelleme sorunlarının giderilmesi ile ilgilenir. Bazı engelleme sorunlarının ürün sahibi tarafından da kaldırılması gerekebilir.

4-Sprint Review Meeting

Sprint review meeting takes place after each sprint. The participants of this meeting are typically the product owner, the development team, the scrum master, customers and developers from other projects. In this meeting, the development team demonstrates to the whole organization **what they accomplished** during the sprint and **receives feedback**. It is important to receive feedback because that helps the product owner to update the product backlog and consider the scope of the next sprint.

The sprint review meeting is usually held informally, typically with rules that prohibit the use of PowerPoint slides and not allowing more than a couple of hours of preparation time for the meeting.

4-Sprint İnceleme Toplantısı

Sprint inceleme toplantısı her sprint'ten sonra yapılır. Bu toplantının katılımcıları genellikle ürün sahibi, geliştirme ekibi, scrum master, diğer projelerden müşteriler ve geliştiricilerdir. Bu toplantıda, geliştirme ekibi tüm organizasyona sprint sırasında neler başardıklarını gösterir ve geri bildirim alır. Geri bildirim almak önemlidir çünkü bu, ürün sahibinin ürün biriktirme listesini güncellemesine ve bir sonraki sprint kapsamını dikkate almasına yardımcı olur.

Sprint inceleme toplantısı genellikle gayri resmi olarak, tipik olarak PowerPoint slaytlarının kullanımını yasaklayan ve toplantı için birkaç saatten fazla hazırlık süresine izin vermeyen kurallarla yapılır.

5-Sprint Retrospective Meeting

Like sprint review meetings this meeting is also held at the end of each sprint. The attendees of this meeting are the development team, the scrum master and the product owner. In this meeting, all the participants discuss:

* What went well in the previous sprint?
* What didn’t work well?
* What are the improvement areas to increase team performance?

A two-hour retrospective meeting is enough for a two-week sprint. The aim of the retrospectives is not to listen to complaints but to find **effective solutions to problems** and to develop action plans.

5-Sprint Retrospektif Toplantısı

Sprint inceleme toplantıları gibi bu toplantı da her sprint'in sonunda yapılır. Bu toplantının katılımcıları geliştirme ekibi, scrum master ve ürün sahibidir. Bu toplantıda tüm katılımcılar:

Önceki sprint'te iyi giden nedir?

Ne işe yaramadı?

Takım performansını artıracak iyileştirme alanları nelerdir?

İki haftalık bir sprint için iki saatlik retrospektif toplantı yeterlidir. Retrospektiflerin amacı şikayetleri dinlemek değil, sorunlara etkili çözümler bulmak ve eylem planları geliştirmektir.

Other Artifacts in Scrum

Burndown Chart

In addition to the before-mentioned main artifacts, scrum projects also include some other artifacts, such as a burndown chart, timeboxing, story point or velocity.

A burndown chart is a graphical demonstration of work left to do versus time. In the chart, the vertical axis usually represents the remaining work while the horizontal axis represents the time or days. It is useful for forecasting when all of the work will be completed. A simple example of a burndown chart is given above.

Scrum içindeki diğer Eserler

Açılış tablosu

Daha önce sözü edilen ana eserlere ek olarak, scrum projeleri, bir işten çıkarma çizelgesi, zaman çizelgesi, hikaye noktası veya hız gibi diğer bazı yapay nesneleri de içerir.

Bir burndown grafiği, zamana karşı yapılması gereken işin grafiksel bir gösterimidir. Grafikte, dikey eksen genellikle kalan işi, yatay eksen ise zamanı veya günleri temsil eder. Tüm işlerin ne zaman tamamlanacağını tahmin etmek için kullanışlıdır. Bir burndown grafiğinin basit bir örneği yukarıda verilmiştir.

Timeboxing

A timebox is a period of time in which a team works to achieve a goal. Instead of waiting until the goal is reached, the timeboxing approach stops when the time limit is reached. The timeboxing aims **to define and limit the amount of time dedicated to a certain event**. Scrum framework uses timeboxing as a tool for all of the scrum activities for concretely defining ambiguous tasks.

Timeboxing

Zaman çizelgesi, bir ekibin bir hedefe ulaşmak için çalıştığı bir dönemdir. Hedefe ulaşılmayı beklemek yerine, zaman sınırına ulaşıldığında zamanlama yaklaşımı durur. Zaman kutusu, belirli bir etkinliğe ayrılan süreyi tanımlamayı ve sınırlamayı amaçlamaktadır. Scrum çerçevesi, zaman kümelemesini belirsiz görevlerin somut olarak tanımlanması için tüm scrum aktiviteleri için bir araç olarak kullanır.

Story Point

Story Points are decided upon and used by individual scrum teams. A Story Point is a relative unit of measure to provide relative predictions of effort for completing tasks or user stories.

A Story Point provides an easier estimation to the team. Rather than assessing a product backlog item and estimating it in hours, teams consider only how much effort this item will require, relative to other product backlog items.

In other words, a story point is a numeric value that indicates the difficulty level of the user story. Before the development team makes an estimation, story points are assigned to each user story using the Fibonacci numbers (1, 2, 3, 5, 8, 13, 21, 34...). A story that is assigned 2 story points should be twice as difficult as a story that is assigned 1 story point.

Hikaye Noktası

Hikaye Puanları bireysel scrum takımları tarafından belirlenir ve kullanılır. Öykü Noktası, görevleri veya kullanıcı öykülerini tamamlamak için göreceli çaba tahminleri sağlamak için göreli bir ölçü birimidir.

Bir Hikaye Noktası takıma daha kolay bir tahmin sağlar. Ekipler, bir ürün biriktirme listesi öğesini değerlendirmek ve saatler içinde tahmin etmek yerine, diğer ürün biriktirme öğeleriyle karşılaştırıldığında bu öğenin ne kadar çaba gerektireceğini düşünür.

Başka bir deyişle, bir hikaye noktası, kullanıcı hikayesinin zorluk seviyesini gösteren sayısal bir değerdir. Geliştirme ekibi bir tahmin yapmadan önce, Fibonacci sayıları (1, 2, 3, 5, 8, 13, 21, 34 ...) kullanılarak her bir kullanıcı hikayesine hikaye noktaları atanır. 2 öykü puanı atanan bir öykü, 1 öykü noktası atanan bir hikayenin iki katı kadar zor olmalıdır.

Velocity

Velocity is an indication of the average amount of items from the product backlog turned into a product increment during each sprint. In other words, it is a measure of the amount of work the development team can handle during a single sprint. In scrum, velocity is the key metric.

At the end of each sprint, velocity is calculated by summing up the story points for completed user stories. Points from not completed or partially completed user stories should not be considered in calculating velocity. After calculating the velocity at the end of the first sprint, the team can make a new estimation of how long the project will take to complete.

Suppose the total value of story points for the remaining user stories is 80, and the total value of story points for the completed user stories in the first sprint is 20. That means the team needs 4 more sprints to complete the whole project.

Hız

Hız, her sprint sırasında ürün biriktirme listesinden bir ürün artışına dönüştürülen ortalama öğe miktarının bir göstergesidir. Başka bir deyişle, geliştirme ekibinin tek bir sprint sırasında yapabileceği iş miktarının bir ölçüsüdür. Scrum'da hız anahtar metriktir.

Her sprint sonunda, hız tamamlanmış kullanıcı hikayeleri için hikaye noktaları toplanarak hesaplanır. Tamamlanmamış veya kısmen tamamlanmış kullanıcı hikayelerinden alınan puanlar, hızın hesaplanmasında dikkate alınmamalıdır. İlk sprint sonunda hızı hesapladıktan sonra ekip, projenin ne kadar sürede tamamlanacağına dair yeni bir tahmin yapabilir.

Kalan kullanıcı öyküleri için öykü puanlarının toplam değerinin 80 olduğunu ve ilk sprint'te tamamlanan kullanıcı öyküleri için öykü puanlarının toplam değerinin 20 olduğunu varsayalım. Bu, ekibin tüm projeyi tamamlamak için 4 sprint'e daha ihtiyacı olduğu anlamına geliyor.

Kanban Overview

Introduction to Kanban

Scrum is the most common way to implement the agile, however, kanban is another popular methodology for implementing the agile system in the business.

The word "kanban" is in Japanese and can be translated as "the card you can see" or "signboard". As the name implies, it is a visual framework used to implement agile and shows **what to produce**, **when to produce it**, and **how much to produce**.

Kanban focuses on **visualizing the workflow process**. The primary goal of kanban is to identify potential flaws in the process and to correct them to ensure that works progress at the desired pace.

Kanban Genel Bakış

Kanban'a Giriş

Scrum, agile uygulamanın en yaygın yoludur, ancak kanban, agile sistemi işte uygulamak için bir başka popüler yöntemdir.

"Kanban" kelimesi Japonca'dır ve "görebileceğiniz kart" veya "tabela" olarak tercüme edilebilir. Adından da anlaşılacağı gibi, çevik uygulamak için kullanılan görsel bir çerçevedir ve ne üreteceğini, ne zaman üreteceğini ve ne kadar üretileceğini gösterir.

Kanban, iş akışı sürecini görselleştirmeye odaklanıyor. Kanban'ın birincil amacı, süreçteki potansiyel kusurları tanımlamak ve işlerin istenen hızda ilerlemesini sağlamak için düzeltmektir.

A Brief History of Kanban

Kanban's story dates back to the 1940s. During these years, Toyota updated its production method based on the model that supermarkets use to manage stocks on shelves. Supermarkets stock enough products to meet consumer demand. This is a method that optimizes the flow between the supermarket and the consumer. Since inventory levels match the consumption rate, the supermarket stores the optimum quantity of products at any given time.

Toyota adopted the system used by the supermarket. To monitor capacity levels in real-time, the company started using a card called "kanban" between different production teams. When a bin of material used on the production line was unloaded, a kanban was sent to the warehouse, explaining which material was required. With the new materials provided by the warehouse, production continued without interruption, and the process monitored continuously.

The kanban method controls the entire chain from production to the end consumer. In this way, it prevents both the failure of supply and the over-stocking of the goods at various stages of the production process. This requires continuous monitoring of the process.

In the following lesson, you will watch an interactive video about the kanban process.

Kanban'ın Kısa Tarihi

Kanban'ın hikayesi 1940'lı yıllara dayanır. Bu yıllar boyunca Toyota, üretim yöntemini süpermarketlerin raflardaki stokları yönetmek için kullandıkları modele göre güncelledi. Süpermarketler tüketici talebini karşılayacak kadar ürün stoklar. Bu, süpermarket ve tüketici arasındaki akışı optimize eden bir yöntemdir. Envanter seviyeleri tüketim oranına uyduğundan, süpermarket herhangi bir zamanda optimum miktarda ürünü depolar.

Toyota, süpermarket tarafından kullanılan sistemi benimsedi. Kapasite seviyelerini gerçek zamanlı olarak izlemek için şirket, farklı üretim ekipleri arasında "kanban" adlı bir kart kullanmaya başladı. Üretim hattında kullanılan bir malzeme kutusu boşaltıldığında, depoya hangi malzemenin gerekli olduğunu açıklayan bir kanban gönderildi. Deponun sağladığı yeni malzemelerle üretim kesintisiz olarak devam etti ve süreç sürekli izlendi.

Kanban yöntemi üretimden son tüketiciye kadar tüm zinciri kontrol eder. Bu şekilde, üretim sürecinin çeşitli aşamalarında mal arzının ve stokların fazla stoklanmasını önler. Bu, sürecin sürekli izlenmesini gerektirir.

Kanban in Software Development

Introduction

Although its story dates back to the 1940s, the implementation of kanban in the world of computer technologies is a relatively new topic. In 2004, David J. Anderson introduced the idea of using the kanban concept for software development, and in 2010 he wrote a book named "Kanban: Successfully Evolutionary Change for your Technology Business ”. Kanban's use in software development begins after these attempts by David J. Anderson.

Yazılım Geliştirmede Kanban

Giriş

Hikayesi 1940'lara kadar uzanmasına rağmen, kanban'ın bilgisayar teknolojileri dünyasında uygulanması nispeten yeni bir konudur. 2004 yılında, David J. Anderson yazılım geliştirme için kanban kavramını kullanma fikrini tanıttı ve 2010 yılında "Kanban: Teknoloji İşiniz İçin Başarılı Evrimsel Değişim" adlı bir kitap yazdı.

\*\* **Kanban**, işinizin mevcut durumu ve yazılım geliştirme sürecinin her aşamasında ürün üzerinde yapılması gereken iş hakkındaki tüm bilgileri taşıyan bir flash kart gibidir. (A Kanban is like a flash card carrying all the information about the current status of your work and the work required to be done on the product at each stage of the software development process.)

Comparison of Kanban and Scrum

It is not easy to completely separate the kanban and scrum methods. You remember the scrum board. For example, most people believe that scrum teams use a kanban board in the scrum process. On the other hand, there are some fundamental differences between the two methods.

* There are no certain time limits in kanban while sprints in scrum have a start and end dates,
* While there are rigid deadlines for the tasks on the scrum board, kanban is more flexible in this regard,
* In contrast to designated roles in a scrum project like product owner, scrum master, and development team, there are no certain roles in kanban,
* The team rearranges the scrum board after each sprint while the same kanban board is used throughout the project.

Kanban ve Scrum Karşılaştırması

Kanban ve scrum yöntemlerini tamamen ayırmak kolay değildir. Scrum tahtasını hatırlıyorsun. Örneğin, çoğu insan scrum takımlarının scrum sürecinde kanban tahtası kullandığına inanır. Öte yandan, iki yöntem arasında bazı temel farklılıklar vardır:

\*\* Scrum'daki sprintlerin başlangıç ve bitiş tarihleri varken kanban'da belirli bir zaman sınırı yoktur,

\*\* Scrum tahtasındaki görevler için katı son teslim tarihleri olsa da, kanban bu konuda daha esnektir,

\*\* Ürün sahibi, scrum ustası ve geliştirme ekibi gibi bir scrum projesinde belirlenen rollerin aksine, kanban'da belirli roller yoktur,

\*\* Takım her sprintten sonra scrum tahtasını yeniden düzenlerken, proje boyunca aynı kanban panosu kullanılır.

The Kanban Board

A kanban board is a tool designed to visualize the work and increase efficiency using cards and columns. As you remember the word "kanban" means "the card you can see", therefore the board on which cards are attached is the main item of a kanban project.

There are three main components of a kanban board: **visual signals**, **columns** and **work-in-progress limits (WIP)**.

**Visual Signals:** Project and work items are written onto cards by kanban teams. For an agile project, each card can correspond to a different user story or a task. Those cards help the team quickly understand **what they working** on and **what is the current situation** of the project.

**Columns:** Each **stage of a project** is demonstrated with a column. Those stages together compose a **workflow**. Visual signals or cards go through the workflow until they are completed.

**WIP Limits:** The maximum number of cards that can be in a column at any given time is called WIP limits. WIP limits are written **on the top of each column** on the board.

Kanban Kurulu

Kanban kartı, kartları ve sütunları kullanarak işi görselleştirmek ve verimliliği artırmak için tasarlanmış bir araçtır. Hatırladığınız gibi, "kanban" kelimesi "görebileceğiniz kart" anlamına gelir, bu nedenle kartların takıldığı tahta bir kanban projesinin ana maddesidir.

Bir kanban panosunun üç ana bileşeni vardır: görsel sinyaller, sütunlar ve devam eden çalışma sınırları (WIP).

**Visual Signals(Görsel Sinyaller):** Proje ve iş öğeleri kanban ekipleri tarafından kartlara yazılır. Agile bir proje için, her kart farklı bir kullanıcı hikayesine veya bir göreve karşılık gelebilir. Bu kartlar, ekibin ne üzerinde çalıştıklarını ve projenin mevcut durumunun ne olduğunu hızlı bir şekilde anlamasına yardımcı olur.

**Columns(Sütunlar):** Bir projenin her aşaması bir sütun ile gösterilir. Bu aşamalar birlikte bir iş akışı oluşturur. Görsel sinyaller veya kartlar tamamlanıncaya kadar iş akışından geçer.

**WIP Limits(WIP Limitleri):** Belirli bir zamanda bir sütunda olabilecek maksimum kart sayısına WIP limitleri denir. Devam Eden Çalışma sınırları tahtadaki her bir sütunun üzerine yazılır.

Types of Kanban Boards

A kanban board can be a physical whiteboard or a digital board. Because of its size, a whiteboard can easily give you an overview of the current status. A physical kanban board is flexible; it is quite convenient to draw new lines and add new columns if required. It also allows the whole team to stand around it, understand the process of the project. If the whiteboard has wheels, you can move it to the meeting rooms. On the other hand, if your team is not working in the same office they can use kanban digital boards remotely.

Kanban Board Çeşitleri

Bir kanban panosu fiziksel bir beyaz tahta veya dijital bir tahta olabilir. Büyüklüğü nedeniyle, bir beyaz tahta size mevcut durum hakkında kolayca genel bir fikir verebilir. Fiziksel bir kanban tahtası esnektir; yeni çizgiler çizmek ve gerekirse yeni sütunlar eklemek oldukça uygundur. Ayrıca tüm ekibin etrafında durmasına, projenin sürecini anlamasına izin verir. Beyaz tahtanın tekerlekleri varsa, toplantı odalarına taşıyabilirsiniz. Öte yandan, ekibiniz aynı ofiste çalışmıyorsa, kanban dijital kartlarını uzaktan kullanabilirler.

\*\* Kanban uses digital or physical boards to demonstrate the team’s workflow. The tasks demonstrated by cards move from left to right representing the progress. So at any given time, the organization can see progress, capacity, productivity, and efficiency.

\*\* Kanban, ekibin iş akışını göstermek için dijital veya fiziksel panoları kullanır. Kartların gösterdiği görevler ilerlemeyi temsil eden soldan sağa hareket eder. Böylece, herhangi bir zamanda organizasyon ilerleme, kapasite, verimlilik ve verimliliği görebilir.

Principles and Practices of Kanban

Principles of Kanban

Kanban has adopted Four Foundational Principles and Six Core Practices to manage the workflow and increase productivity. The four principles of kanban are:

**Start with what you are doing now**: Do not make instant changes to the existing setup or process. Kanban must be directly applied to the existing workflow. Necessary changes should be made slowly and gradually, so, the team feels comfortable.

**Agree to pursue incremental, evolutionary change**: Make minor incremental changes rather than major changes that might lead to resistance within the team.

**Initially, respect current roles, responsibilities, and job-titles**: You do not need to modify your existing roles and functions that perform well. The team will coordinate and implement the necessary changes to the roles and titles. These three principles help managers overcome the expected emotional resistance and fear of change.

**Encourage acts of leadership at all levels**: Constant improvement at all levels of the organization is encouraged. It is desirable that all team members produce ideas, show leadership and contribute to continuous improvement.



Kanban İlke ve Uygulamaları

Kanban İlkeleri

Kanban, iş akışını yönetmek ve verimliliği artırmak için Dört Temel İlke ve Altı Temel Uygulama benimsemiştir. Kanban'ın dört ilkesi:

**Şimdi ne yaptığınızla başlayın**: Mevcut kurulum veya işlem üzerinde anında değişiklik yapmayın. Kanban, mevcut iş akışına doğrudan uygulanmalıdır. Gerekli değişiklikler yavaş ve kademeli olarak yapılmalıdır, bu nedenle takım kendini rahat hisseder.

**Artımlı, evrimsel değişimi sürdürmeyi kabul edin**: Takım içinde direnişe yol açabilecek büyük değişiklikler yerine küçük artımlı değişiklikler yapın.

**Başlangıçta, mevcut rollere, sorumluluklara ve iş unvanlarına saygı gösterin**: İyi performans gösteren mevcut rollerinizi ve işlevlerinizi değiştirmeniz gerekmez. Ekip, roller ve unvanlarda gerekli değişiklikleri koordine edecek ve uygulayacaktır. Bu üç ilke yöneticilerin beklenen duygusal direnci ve değişim korkusunu aşmalarına yardımcı olur.

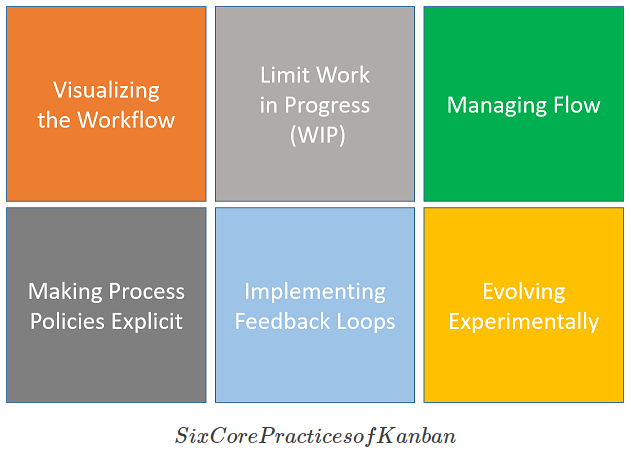
**Her düzeyde liderlik eylemlerini teşvik edin**: Kuruluşun tüm düzeylerinde sürekli iyileştirme teşvik edilir. Tüm ekip üyelerinin fikir üretmesi, liderlik göstermesi ve sürekli gelişime katkıda bulunması arzu edilir.

Practices of Kanban

Besides the abovementioned four principles, there are six core practices in kanban. When you understand these six practices well, you will learn about the kanban process. Because these six practices explain the kanban process.

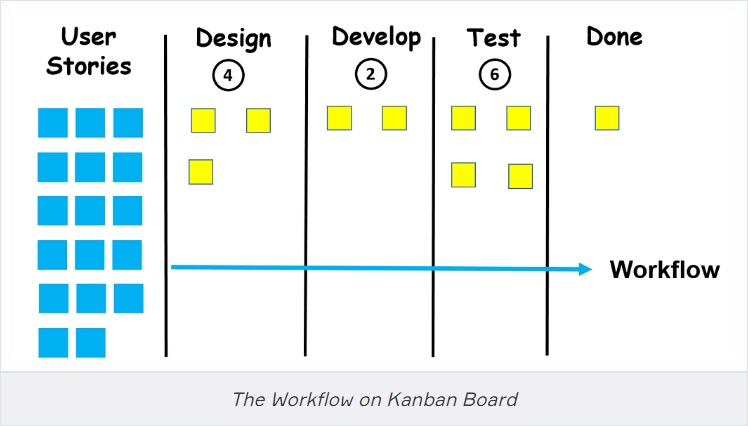
Kanban Uygulamaları

Yukarıda belirtilen dört ilkenin yanı sıra kanban'da altı temel uygulama vardır. Bu altı uygulamayı iyi anladığınızda, kanban süreci hakkında bilgi edineceksiniz. Çünkü bu altı uygulama kanban sürecini açıklıyor.



1-Visualizing the Workflow

Visualization is the first step to start with kanban. You need to visualize the current workflow on either a physical whiteboard or a digital Kanban board. This board can be simple or detailed depending on the complexity of the work. When you visualize the process, it can be seen what you and your team are currently doing.

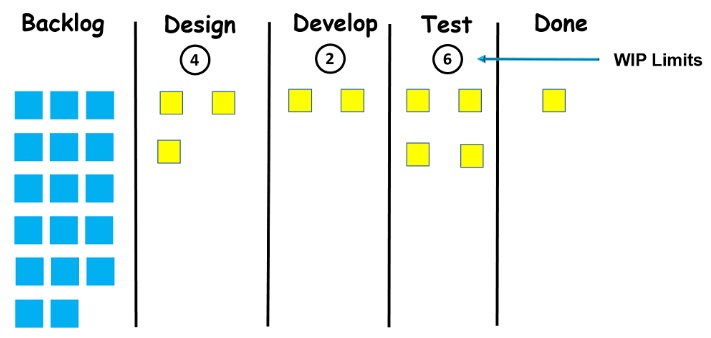


1-İş Akışını Görselleştirme

Görselleştirme kanban ile başlamak için ilk adımdır. Geçerli iş akışını fiziksel bir beyaz tahtada veya dijital bir Kanban panosunda görselleştirmeniz gerekir. Bu tahta, işin karmaşıklığına bağlı olarak basit veya ayrıntılı olabilir. Süreci görselleştirdiğinizde, sizin ve ekibinizin şu anda ne yaptığını görebilirsiniz.

2-Limit Work in Progress (WIP)

Limiting Work in Progress (WIP) is the second practice in the kanban. The maximum number of cards that can be in a column at any given time is called WIP limits. When you assign a limit to each column, your team doesn't work more than they can handle.



In other words, you expect your team **to complete the current work first** before taking up a new one. So, you create capacity in the system and new jobs can be received by the team. Initially, it may not be so easy to predict the WIP limits of your team. Therefore, to start with no WIP limits can be considered. After making observations and getting sufficient data about your team you can define WIP limits for each stage in the workflow. Or you can start with a WIP limit of 1 to 1.5 times the number of people taking part in each stage or each column.

2-Devam Eden Çalışmayı Sınırla (Devam Eden Çalışma)

Devam Eden Çalışmanın Sınırlandırılması (WIP) kanbanın ikinci uygulamasıdır. Herhangi bir zamanda bir sütunda olabilecek maksimum kart sayısına WIP limitleri denir. Her sütuna bir sınır atarsanız, ekibiniz üstesinden gelebileceğinden fazla çalışmaz.

Başka bir deyişle, yeni bir işe başlamadan önce ekibinizin mevcut işi tamamlamasını beklersiniz. Böylece sistemde kapasite yaratırsınız ve ekip tarafından yeni işler alınabilir. Başlangıçta, ekibinizin Devam Eden Çalışma sınırlarını tahmin etmek o kadar kolay olmayabilir. Bu nedenle, herhangi bir Devam Eden Çalışma sınırı olmadan başlamak düşünülebilir. Gözlem yaptıktan ve ekibiniz hakkında yeterli veri aldıktan sonra, iş akışındaki her aşama için Devam Eden Çalışma sınırları tanımlayabilirsiniz. Veya her aşamada veya her sütunda yer alan kişi sayısının 1 ila 1,5 katı olan Devam Eden Çalışma sınırı ile başlayabilirsiniz.

3-Managing Flow

After visualization and limiting each stage, the third practice relates to improving the flow of the process. Managing and improving flow is the key point of your kanban system after applying the first two practices. Because there are two options at this point. You will either come across a smooth flow within WIP limits or you will observe the works start piling up. This affects the delivery time of the products.

Related to the delivery time of the product a new concept called **lead time** comes into the picture. Lead time is the period between creating a task in your workflow and its final departure from the kanban board. Therefore, Kanban helps you monitor workflow, identify blockages, and make adjustments to improve the flow for reducing the lead time. As you manage the flow and improve it, your team's pace becomes more predictable. This makes it easy for you to make commitments to your customers. And it is very important to make reliable commitments in the kanban philosophy.

3-Akışı Yönetme

Görselleştirmeden ve her aşamayı sınırlandırdıktan sonra, üçüncü uygulama işlemin akışını iyileştirmekle ilgilidir. Akışı yönetmek ve geliştirmek, ilk iki uygulamayı uyguladıktan sonra kanban sisteminizin kilit noktasıdır. Çünkü bu noktada iki seçenek var. Devam Eden Çalışma sınırları dahilinde düzgün bir akışla karşılaşacaksınız ya da işlerin birikmeye başladığını gözlemleyeceksiniz. Bu ürünlerin teslimat süresini etkiler.

Ürünün teslim süresi ile ilgili olarak teslim süresi adı verilen yeni bir konsept resme girer. Teslim süresi, iş akışınızda bir görev oluşturma ile kanban panosundan son ayrılması arasındaki süredir. Bu nedenle Kanban, iş akışını izlemenize, tıkanıklıkları tanımlamanıza ve sağlama süresini azaltmak için akışı iyileştirmenize yardımcı olur. Akışı yönetip geliştirdikçe, ekibinizin hızı daha tahmin edilebilir hale gelir. Bu, müşterilerinize taahhüt vermenizi kolaylaştırır. Ve kanban felsefesinde güvenilir taahhütlerde bulunmak çok önemlidir.

\*\* Lead time is the period between creating a task in your workflow and its final departure from the kanban board. Therefore, kanban helps you monitor workflow, identify blockages, and make adjustments to improve the flow for reducing the lead time.

\*\* Teslim süresi, iş akışınızda bir görev oluşturma ile kanban panosundan son ayrılması arasındaki süredir. Bu nedenle, kanban, iş akışını izlemenize, tıkanmaları tanımlamanıza ve sağlama süresini azaltmak için akışı iyileştirmek için ayarlamalar yapmanıza yardımcı olur.

4-Making Process Policies Explicit

Making process policies, basic principles and working methods explicit is the fourth practice of kanban. Being explicit and clear on these issues allows the team to follow the process easily and make proposals for the necessary improvements. In other words, the team members who are not clear about the existing process can not improve the system.

4-Süreç Politikalarını Açık Hale Getirme

Süreç politikalarını, temel ilkeleri ve çalışma yöntemlerini açıklığa kavuşturmak, kanban'ın dördüncü uygulamasıdır. Bu konularda açık ve net olmak, ekibin süreci kolayca takip etmesini ve gerekli iyileştirmeler için önerilerde bulunmasını sağlar. Başka bir deyişle, mevcut süreç hakkında net olmayan ekip üyeleri sistemi geliştiremezler.

5-Implementing Feedback Loops

During the third practice, we discussed reducing the lead time by managing the flow. However, rapid delivery can not solve all the problems. You should also know what your customers and the end-users think, and how much the product contributes to the value of your company.

This is where fifth practice comes into the picture, you need to get feedback from people not involved in your system. Feedback loops are also required within a system to ensure that the desired functionality is provided with the high quality. That's where different kinds of tests come out. Automated continuously running tests are preferred as they shorten feedback loops.

5-Geri Bildirim Döngülerini Uygulama

Üçüncü uygulama sırasında akışı yöneterek teslim süresini azaltmayı tartıştık. Ancak, hızlı teslimat tüm sorunları çözemez. Ayrıca müşterilerinizin ve son kullanıcıların ne düşündüğünü ve ürünün şirketinizin değerine ne kadar katkıda bulunduğunu bilmeniz gerekir.

Burada beşinci uygulama devreye giriyor, sisteminize dahil olmayan insanlardan geri bildirim almanız gerekiyor. Ayrıca, istenen işlevselliğin yüksek kalitede sunulmasını sağlamak için bir sistem içinde geri besleme döngüleri gereklidir. Farklı tür testler burada ortaya çıkar. Geri besleme döngülerini kısalttıkları için sürekli çalışan otomatik testler tercih edilir.

6-Improving Collaboratively, Evolving Experimentally (Using the Scientific Methods)

Because the kanban is an evolutionary improvement process you should evaluate your system continuously and improve it constantly. That allows you to adopt minor changes in the process and improve the workflow at a constant pace. While improving your workflow, kanban encourages you to use scientific methods like you form a hypothesis and test it.

Then you make changes depending on the results of the test. If you are implementing agile principles, your main task is to continuously evaluate your process and improve it when needed. You can observe and measure the impact of the changes that you make by scientific methods. By using these methods, you can assess whether a change helps you improve or not. Then you make a decision about whether to try some other solutions or to keep the current status.

6-Ortak Çalışmayı Geliştirme, Deneysel Olarak Gelişme (Bilimsel Yöntemleri Kullanarak)

Kanban evrimsel bir iyileştirme süreci olduğundan, sisteminizi sürekli değerlendirmeli ve sürekli iyileştirmelisiniz. Bu, süreçte küçük değişiklikler yapmanıza ve iş akışını sabit bir hızda geliştirmenize olanak tanır. Kanban, iş akışınızı geliştirirken, sizin gibi bir hipotez oluşturup test ettiğiniz gibi bilimsel yöntemleri kullanmanızı önerir.

Ardından, testin sonuçlarına bağlı olarak değişiklikler yaparsınız. Agile ilkeler uyguluyorsanız, asıl göreviniz sürecinizi sürekli olarak değerlendirmek ve gerektiğinde iyileştirmektir. Yaptığınız değişikliklerin bilimsel yöntemlerle etkisini gözlemleyebilir ve ölçebilirsiniz. Bu yöntemleri kullanarak, bir değişikliğin gelişmenize yardımcı olup olmadığını değerlendirebilirsiniz. Ardından başka çözümler denemek veya mevcut durumu korumak konusunda karar verirsiniz.

\*\* **Q**: Are there any drawbacks of the Agile model? If yes, please explain.  
 **A**: Yes, there are some drawbacks of the Agile method, some of them are as follows:  
1- It is not easy to make an estimation of the effort required to complete a task. It becomes more complex in the case of large projects as it becomes difficult to make a prediction about the total effort required.  
2- In case the desired requirements of the client are not understood properly, the final project will not meet the customer requirements. Thus, this will lead to customer dissatisfaction.  
3- Only the leader who has considerable experience in the Agile model is capable to take important decisions. The team members with less or no experience are not involved in the decision-making process, thus they don’t have a chance to advance their knowledge.

\*\* S: Agile modelinin dezavantajları var mı? Evet ise, lütfen açıklayınız.

C: Evet, Agile yönteminin bazı dezavantajları vardır, bazıları aşağıdaki gibidir:

1- Bir görevi tamamlamak için gereken çabayı tahmin etmek kolay değildir. Büyük projeler söz konusu olduğunda, gerekli toplam çaba hakkında bir tahmin yapmak zorlaştıkça daha karmaşık hale gelir.

2- Müşterinin istenen şartlarının tam olarak anlaşılmaması durumunda, nihai proje müşteri şartlarını karşılamayacaktır. Böylece, müşteri memnuniyetsizliğine yol açacaktır.

3- Sadece Agile modelinde önemli deneyime sahip lider önemli kararlar alabilir. Deneyimi az olan veya hiç olmayan ekip üyeleri karar verme sürecine dahil değildir, bu nedenle bilgilerini ilerletme şansları yoktur.

Formun Üstü

Formun Altı