



SDLC

SOFTWARE DEVELOPMENT LIFE CYCLE

(Yazılım Geliştirme Yaşam Döngüsü)

4. Ders
08.10.2022



Bu derste neler öğreneceğiz?

- 1- Kanban
- 2- Epic / Feature / User Story
- 3- Kahoot



KANBAN

Kanban uzakdoğu kökenli bir kelimedir. Japonca'da “sinyal tahtası”, Çince de ise “büyük görsel tahta” anlamına gelmektedir.

Agile yaklaşımı içinde de en çok tercih edilen metotlardan birisidir.

Temelini de 1950li yıllarda kullanılmaya başlanan Toyota Üretim Sistemine dayanmaktadır.

2000li yılların başında ise David Anderson tarafından da bir değişim yönetim metodu olarak geliştirilmiştir.

Kanban metodu evrimsel bir metot olduğundan mevcuttaki çalışma metodunuza uyarlanabilmektedir. Waterfall sürecinde de kanban prensipleri uygulayabildiğiniz gibi Scrum çerçevesinde de uygulayabilirsiniz.

Hatta son zamanlarda Scrum ve Kanban pratiklerinin beraber kullanıldığı “Scrumban” metotuda sıkça kullanılmaktadır.





KANBAN

Kanban, ekip yapınızda herhangi bir deęişiklik yapmadan süreçleri ve iş akışı verimliliğini iyileştirmek için kullanılabilir. **Kanban Metodu**'nu uygulamadan önce, temel ilkelerini anlamak ve benimsemek önemlidir:

- Ne yapıyorsanız oradan başlayın** - Kanban, belirli bir kurulum gerektirmez ve doğrudan mevcut iş akışınıza uygulanabilir. Bu, mevcut süreçlerinizi deęiştirmenize gerek olmadığı için uygulamayı kolaylaştırır.
- Deęişimi kabul etmelisiniz** - Kapsamlı deęişiklikler ekipleri rahatsız edebilir, akışı bozabilir ve performansa zarar verebilir. Kanban, sürekli, artan ve evrimsel deęişiklikleri teşvik ederek minimum dirençle karşılaşacak şekilde tasarlanmıştır.
- Mevcut sürece, rollere ve sorumluluklara saygı gösterin** - Başlangıçta herhangi bir organizasyon deęişikliği olmamalıdır. Kanban, mevcut süreçlerin, rollerin ve sorumlulukların deęeri olabileceğini ve korunmaya deęer olduğunu kabul etmelisiniz.
- Her düzeyde liderliği destekleyin** - Kanban, tüm üyeler arasında liderliği ve karar almayı destekler. En alt sıradaki ekip üyesinin parlak bir fikri varsa, bu kabul edilmeli ve benimsenmelidir.



KANBAN PRATİKLERİ NELERDİR?

1

GÖRSELLEŞTİR



2

DEVAM EDEN İŞLERİ
LİMİTLE



3

AKIŞI YÖNET



4

POLİTİKALARINI
AÇIK HALE GETİR



5

GERİ BİLDİRİM
DÖNGÜLERİ KUR



6

İŞBİRLİĞİ İLE İYİLEŞ,
DENEYSELLİKLE EVRİMLEŞ
(BİLİMSEL YÖNTEM VE MODELLERİ KULLAN)

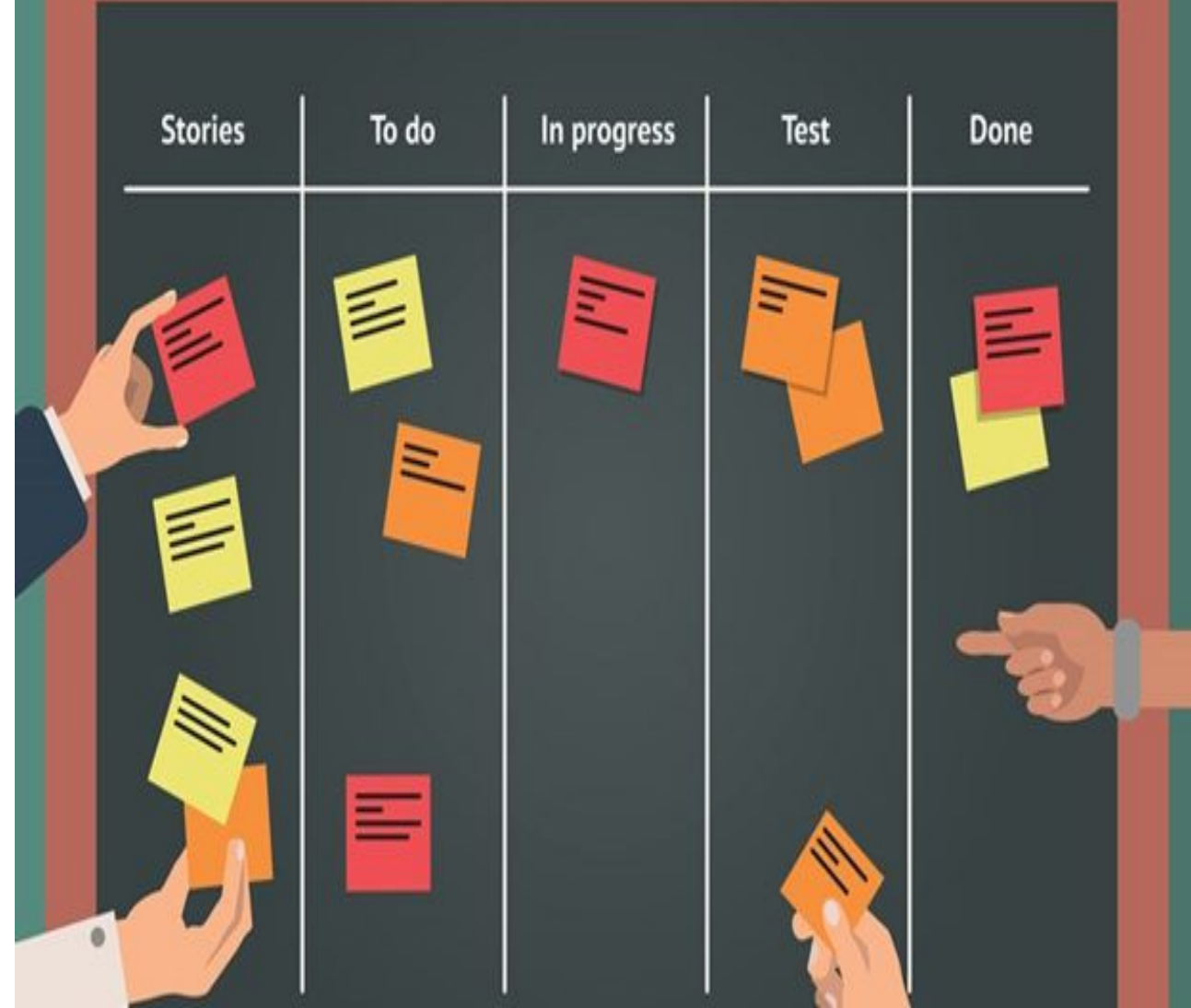




KANBAN

Başarılı Kanban uygulaması için gözlemlemeniz gereken altı temel uygulama vardır.

İş akışını görselleştirin - İlk ve en önemli görev, mevcut iş akışını anlamaktır - bir öğeyi talepten teslim edilebilir bir ürüne taşımak için uygulanacak adımların sırası nedir? Bu, kartları ve sütunları olan bir Kanban panosu kullanılarak yapılır: her sütun iş akışınızdaki bir adımı temsil eder ve her kart bir iş öğesini temsil eder.

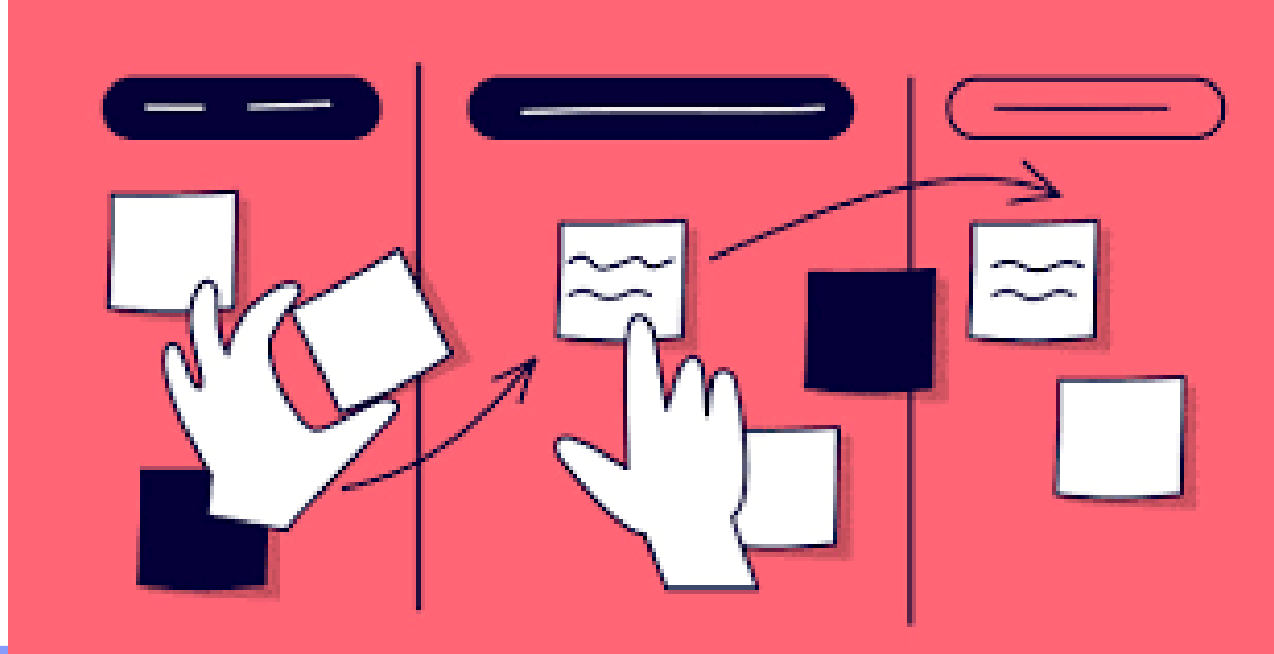




KANBAN

Devam Eden Çalışmayı Sınırla

Odaklanma kaybı, ekibinizin performansına ciddi şekilde zarar verebilir, bu nedenle bu uygulama, devam eden işe sınırlar koyarak kesintileri ortadan kaldırmaya odaklanır. Devam Eden Çalışma için sınırlar uygulayan ekipler, yeni işe başlamadan önce olağanüstü işleri bitirmeye odaklanır. Devam Eden Çalışma'nın sınırlandırılması, Kanban'ın başarılı bir şekilde uygulanması için kritiktir.





KANBAN

Akışı yönetin - Akış verimliliğini gözlemleyerek ve analiz ederek, sorunlu alanları tanımlayabilirsiniz. Kanban'ı uygulamanın temel amacı, teslim sürelerini iyileştirerek ve gecikmeleri önleyerek sorunsuz bir iş akışı oluşturmaktır.

Süreç politikalarını açık hale getirin - Süreç, ekipteki herkes için açıkça tanımlanmalı, yayınlanmalı ve onaylanmalıdır: İnsanlar, yararlı olacağını düşünmedikçe bir şeyin parçası olmak için motive olmayacaklar. Herkes açık politikalardan haberdar olduğunda, her kişi performansınızı artıracak iyileştirmeler önerebilir.

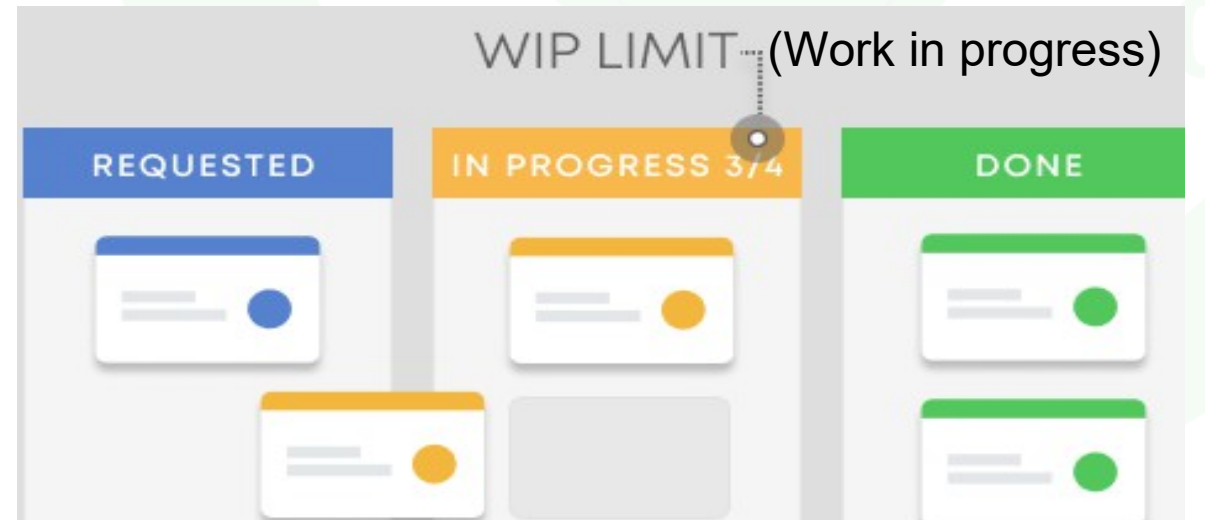
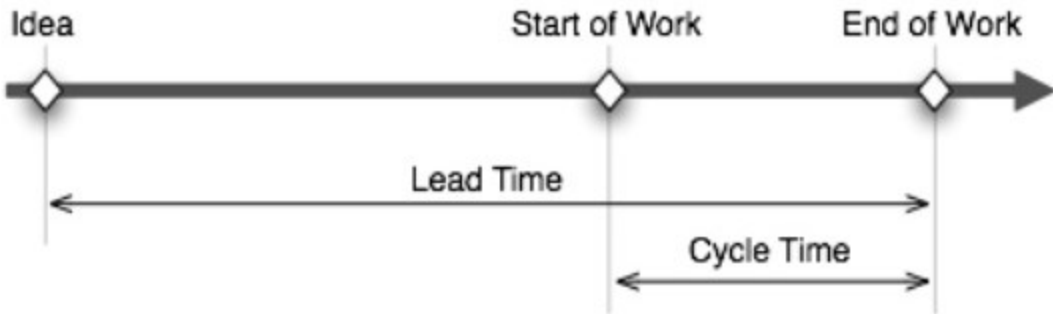




KANBAN

Geri bildirim döngülerini kullanın - Olumlu değişimin gerçekleşmesi için, tüm ekibe önemli geri bildirimler sağlamak için düzenli toplantılar gereklidir. Bu toplantıların sıklığı değişir, ancak fikir, düzenli olarak, belirli bir zamanda olmalarıdır.

İşbirliği yaparak geliştirin - Kanban, sürekli değerlendirme, analiz ve iyileştirme gerektirir. Takımlar süreç hakkında ortak bir anlayışa sahip olduklarında, herhangi bir sorun çıkması durumunda bir fikir birliğine varma olasılıkları daha yüksektir.





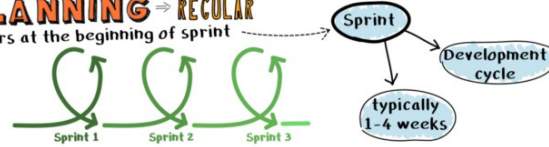
SCRUM

VS

KANBAN

PLANNING → REGULAR

occurs at the beginning of sprint



ESTIMATIONS of TIME



BEFORE start of sprint

items should be small to finish within sprint
if not, split them to smaller pieces



CHANGES TO WORK SCOPE



should wait for next sprint

ROLES



MEETINGS

SPRINT PLANNING
1-4 hour collaborative session

SPRINT REVIEW
0.5-2 hours review the results

DAILY SCRUM
10-15 min everyday
everybody talks about achievements/issues

RETROSPECTIVE
0.5-2 hours what went well and what did not

OWNERSHIP



WHEN TO USE

- small items - small value
- adding increments possible
- requirements in a good shape
- Roadmap is clear
- more cross-dependent teams



BOULDER

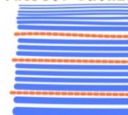


ROCK

PEBBLE

BOARDS / ARTIFACTS

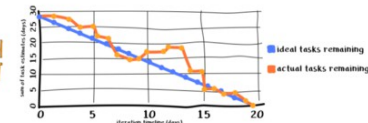
PRODUCT BACKLOG



SPRINT BACKLOG



BURNDOWN CHART



PLANNING

NOT PRECISE planning routine

PLAN WHEN they FINISH items

demand planning



CONTINUOUS FLOW



ESTIMATIONS of TIME

optional
when items are completed



teams simply **PULL** next item from backlog and implement it

LIMIT

how many items can be in working columns at the same time

CHANGES TO WORK SCOPE

added **AS NEEDED**



ROLES

AS NEEDED

MEETINGS

NONE REQUIRED



OWNERSHIP

DEPENDS on defined roles and necessities



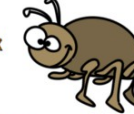
WHEN TO USE

Changes are **too fast**

support/maintenance
work (operational level)



bugfix



BOARDS / ARTIFACTS

KANBAN BOARD

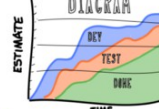


LEAD and CYCLE TIME DIAGRAM

presents the average amount of time it takes for a task to be processed from the specific start to the finish point

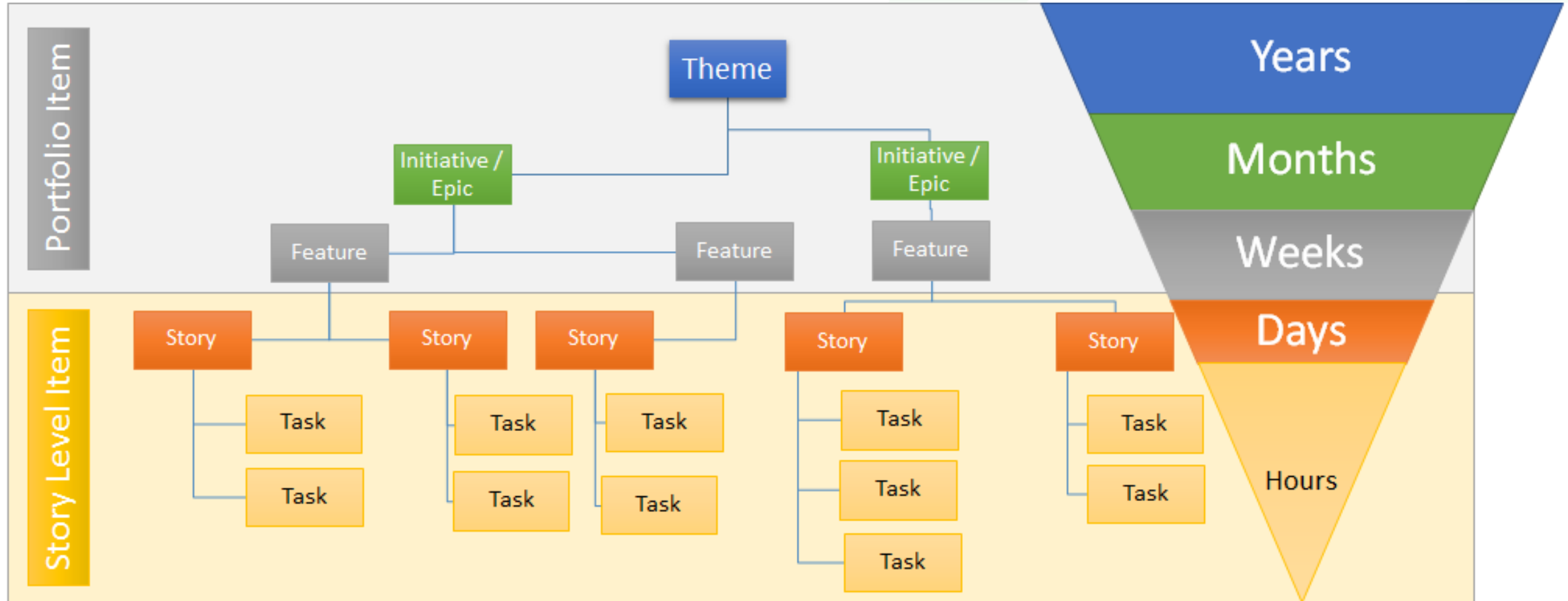


CUMULATIVE FLOW DIAGRAM



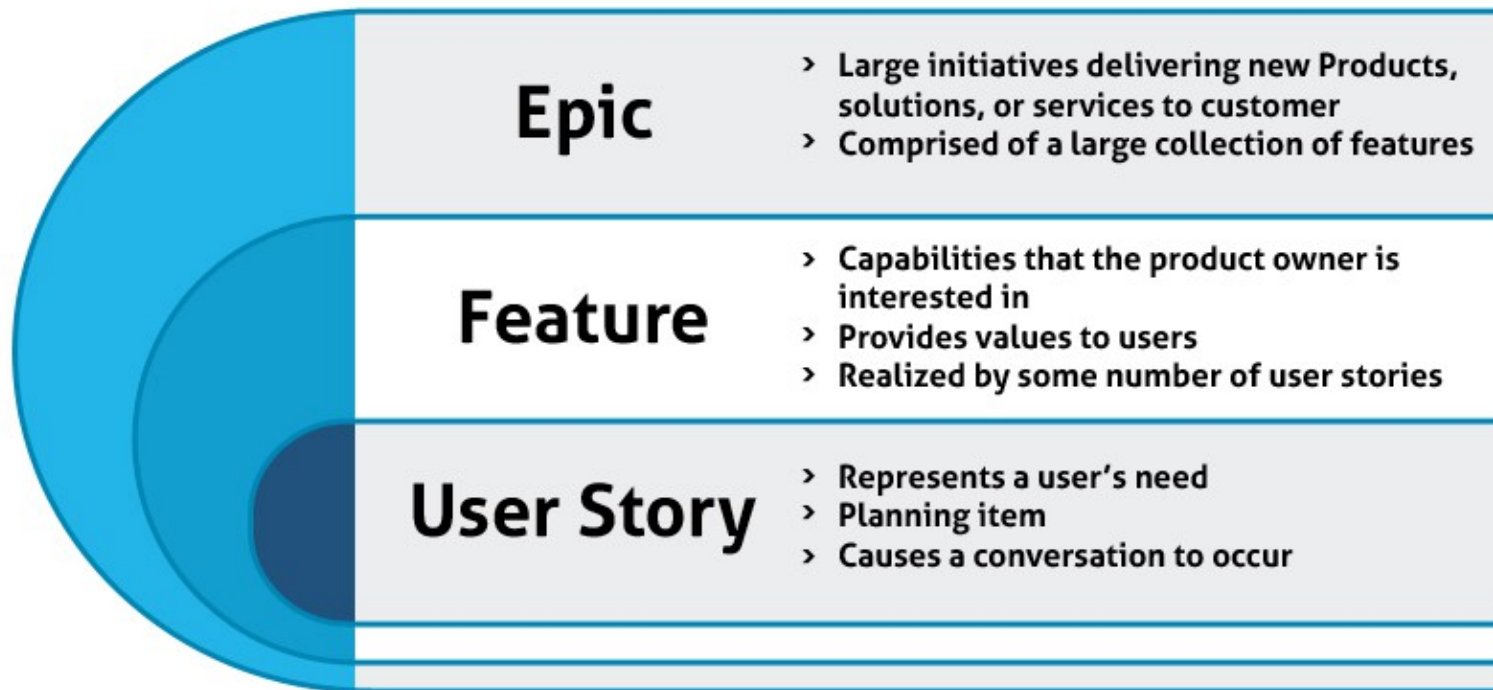


EPICS , FEATURE , USER STORY , TASK

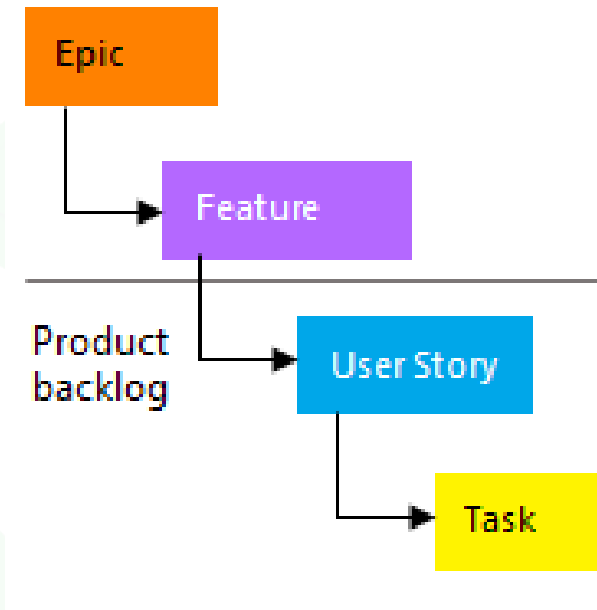




EPICS , FEATURE , USER STORY , TASK



Portfolio backlog





EPIC , FEATURE NEDİR?

EPIC : Belirli bir vizyona ulaşabilmek için şirketlerin metrikler ile tanımlı, ulaşması gereken hedefleri vardır. Ekipler ise bu hedeflere ulaşabilmek için yeni girişimler/özellikler ile gelir ve sahip oldukları üründe/ürünlerde yeni güncellemeler yayınlarlar. Epics, yapılması planlanan bir veya birden fazla yeni fonksiyonların yahut özelliklerin gruplanmasıdır.

Product ekibinde yapılan her şey son kullanıcıya sunulmaz veya yeni bir özellik olarak sonuçlanmaz. Bu yüzden özellikler(features) olarak değil epics olarak adlandırılırlar. Ayriyeten bir sprint'ten daha uzun süren çalışmalar da “*epics*” olarak adlandırılmaktadır.

FEATURE : Epicler içerisinden PO tarafından seçilerek Product Backlog'a eklenen, birden fazla user story içeren, birden fazla haftada tamamlanacak modüllerdir. Son kullanıcıya tamamlanmış ürün olarak ulaşır.



USER STORY NEDİR?

User storyler 3 bölümden oluşur. Bunlar;

- Kullanıcının hikaye tanımı (Value statement)
- Kabul kriterleri (Acceptance criterias)
- Bitti tanımı (Definition of done)

User Stories, Agile metodolojisinin bir parçasıdır. Son kullanıcıya sunulacak herhangi bir özelliği veya ürünü açıklama yollarından biridir.

user story template

| | |
|---|---------------------------------|
| WHO are we building it for? Who is the user? | As a <type of user> |
| WHAT are we building? What is the intention? | I want <some goal or objective> |
| WHY are we building it? What is the value for the customer? | So that <benefit/value> |



USER STORY #1

Hikâye Tanımı (Value Statement)

Ürün ekibi yetkilisi olarak, paket yenilemelerini arttırabilmek için; üyelerimize paketlerinin bitmesine 1 ay kala “yenileme yapmak ister misiniz” mail gönderimi yapılmasını, mail şablonu içerisinde X, Y, Z paket detayları ve ilgili müşteri temsilcilerinin ad, soyad, iletişim bilgilerinin yer almasını istiyorum.

Kabul kriterleri (Acceptance criterias)

- Paket bitimine 1 ay kalan kullanıcılara mail gönderimlerinin başarıyla yapılması
- Üye – müşteri temsilcisi eşleşmesinin doğru yapılmış olması
- Gönderimlerin sadece mail gönderimine izin verilen kullanıcılara yapılması
- Mail opt-out kurgularının sağlıklı bir şekilde çalışması

Bitti Tanımı (Definition of Done)

- Kod kontrol edildi (Code review)
- Birim testleri yapıldı (Unit tests)
- Kabul kriterleri karşılandı (Acceptance criterias)
- Product Owner(PO) story kabulü (PO accepts user story)



| User Story ID | Description | Acceptance Criteria |
|---------------|---|---|
| US 0001 | ATM kullanıcısı olarak, nakit çekebilmem için PIN kodumu girmek istiyorum | PIN dört basamaklı olmalıdır |
| | | PIN özel karakterlere (sembollere) izin vermemelidir |
| | | PIN'in 30 saniye içinde girilmesi gerekir, aksi takdirde işlem iptal edilir |
| US 0002 | Facebook login (Giris sayfası) geçersiz kimlik bilgileriyle erişilmemelidir | Gecersiz kullanıcı adı ile erişim sağlanamaz |
| | | Geçersiz şifre ile erişim sağlanamaz |
| | | Geçersiz kullanıcı adı ve şifre ile erişim sağlanamaz |

User Stories üç ortak özelliği barındırır:

- **Kısa ve açıklayıcı olma:** Ürün ya da özelliğin kullanarak ne yapılmak istendiğini kısaca anlatabilmenizi sağlar.
- **Kullanıcı tipini belirleme:** Kullanıcı hakkında detaylı bilgi verilmese de, kullanıcının karakteristik özelliklerinin ürün ya da yazılımla ilişkisi vurgulanır.
- **Problemi tanımlama:** Kullanıcı hikayeleri, ürünün ya da yazılımın çözmesi gereken problemleri ortaya çıkarır.



WHO WHAT WHY
As a **role**, I want **action** (so that **benefit**)

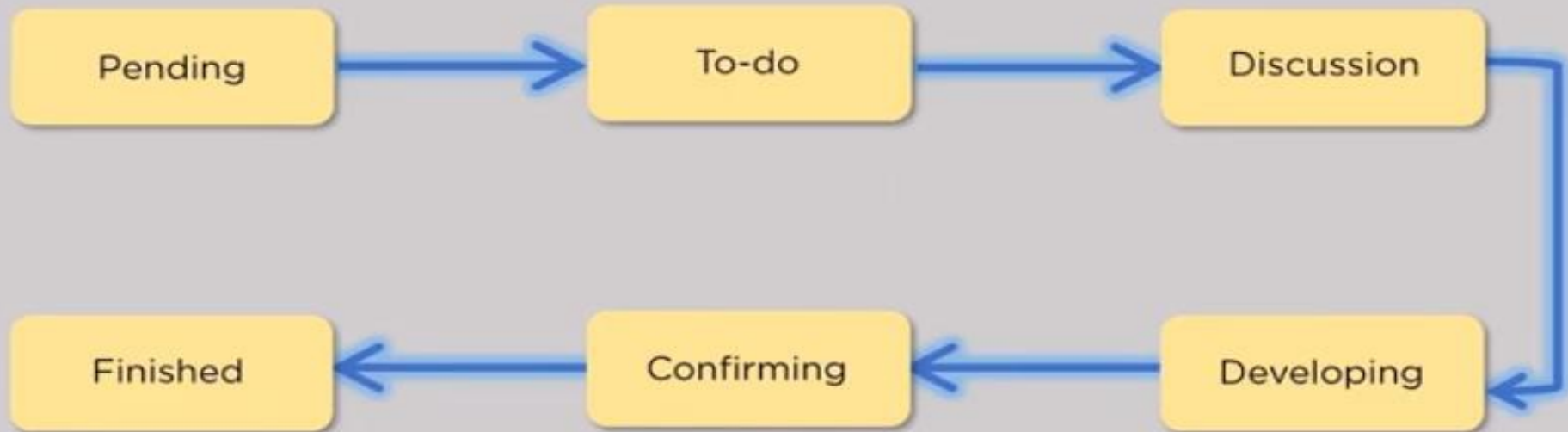
KİM? NE? NEDEN?
Rol olarak, **aksiyon** yapmak istiyorum, çünkü **yarar**.

- Bir kullanıcı olarak şifre oluştururken, güvenlik amacıyla, güçlü bir şifre oluşturmak istiyorum.
- Bir yönetici olarak, çalışanlarımın bilgilerini, filtreleyerek daha çabuk ulaşmak istiyorum.
- Bir kullanıcı olarak birden fazla kişiye, daha hızlı olacağı için aynı anda mesaj göndermek istiyorum
- “Sıkça sinemaya giden bir izleyici olarak, yakınımıdaki sinema salonlarında vizyona yeni girecek filmler hakkında bütün bilgilere telefonumdan ulaşabilmek, böylece her filmin web sitesini ayrı ayrı kontrol etmeye ihtiyaç duymamak istiyorum.”

| | A | B | C | D | E |
|----|---------------|---|--|----------|------------------------|
| 1 | User_Story_ID | Description | Acceptance Criteria | Priority | Validation |
| 2 | US_001 | System should allow any user to register with valid credentials validating the success message | There should be a valid SSN respecting the "-" where necessary, it should be 9 digits long | High | UI and Backend Testing |
| 3 | | | There should be a valid name that contains chars and cannot be blank | | |
| 4 | | | There should be a valid lastname that contains chars and it is a required field | | |
| 5 | | | You can provide chars and digits to describe a valid address along with zip code | | |
| 6 | | | User should provide 10 digit-long mobilephone number as a required field respecting the "-" | | |
| 7 | | | User cannot use just digits, but can use any chars and digits along with them and of any length | | |
| 8 | | | You should provide a valid email format that contains "@" sign and "." extensions | | |
| 9 | | | | | |
| 10 | US_002 | System should not allow anyone to register with invalid credentials seeing the message "If you want to sign in, you can try the default accounts:- Administrator (login="admin" and password="admin") - User (login="user" and password="user"). | Any field on the registration page should not be left blank | High | UI and Backend Testing |
| 11 | | | SSN number cannot be of any chars nor spec chars except "-" | | |
| 12 | | | Mobilephone number cannot be of any chars nor spec chars except "-" | | |
| 13 | | | email id cannot be created without "@" sign and "." extensions | | |
| 14 | | | | | |
| 15 | US_003 | Registration page should restrict password usage to a secure and high level passcode | There should be at least 1 lowercase char for stronger password and see the level chart change accordingly | High | UI and Backend Testing |
| 16 | | | There should be at least 1 uppercase char and see the level chart change accordingly | | |
| 17 | | | There should be at least 1 digit and see the level chart change accordingly | | |
| 18 | | | There should be at least 1 special char and see the level bar change accordingly | | |
| 19 | | | There should be at least 7 chars for a stronger password | | |
| 20 | | | | | |
| 21 | US_004 | Login page should accessible with valid credentials | There should be a valid username and password validating the success message to login | High | UI and Backend Testing |
| 22 | | | There should be an option to cancel login | | |
| 23 | | | | | |
| 24 | US_005 | Login page should not be accessible with invalid credentials | User cannot login with invalid username validating the error message | High | UI and Backend Testing |
| 25 | | | User cannot login with invalid password validating the error message | | |
| 26 | | | User cannot login with invalid username and password validating the error message | | |
| 27 | | | User with invalid credentials should be given an option to reset their password | | |
| 28 | | | User should be given the option to navigate to registration page if they did not register yet | | |
| 29 | | | | | |



LIFECYCLE OF A USER STORY





DEFINATION OF DONE (DoD) TANIMI

Development Team, kendilerinden tamamlanması beklenen işleri Product Owner'a sunmadan önce bir kalite-kontrol süreci gibi kontrol listesi hazırlar. Bu liste tüm User Story'ler için ortaktır ve biten tüm işler bu prosedürden geçmesi gerekir. Buna kısaca "Done" veya "DoD — Definition of Done (Bitti Tanımı)" denir. Bu liste genellikle şu kontrolleri içerir:

- Unit testleri
- QA testleri
- User Story'deki tüm dökümanların güncellenmesi
- Kodun review edilmesi
- User Story kabul kriterlerinin karşılanması
- Product Owner onayı

DoD checklisti, Sprint veya Release sonunda belirli tamamlama kriterlerinin karşılanıp karşılanmadığını kontrol etmek için de kullanılmak üzere hazırlanabilir.



DEFINATION OF DONE (DoD) TANIMI

Definition of Done

2 / 5

Add a new item

- ☒ ~~Functionality described in the story is implemented and tested by the QA. Technical tasks are accepted by the Component lead.~~
- ☐ ~~Change has been deployed to the test server~~
- ☒ ~~Unit tests are written for any new code~~
- ☐ ~~The documentat~~
- ☐ ~~CHANGLOG entry is added for all services impacted during work on ticket.~~

Checked by Claribel Spring on Mon Mar
25 2019 14:40:03 GMT+0100



TAKIM KAPASİTESİ VE PUANLAMA

User storyler sprint backlog'a alınırken takım tarafından puanlanır (**POINT**)

- Sprint başlangıç toplantısı veya refinement toplantısında User Story okunur ve development team'in tüm üyeleri user story'e point verir

- Verilen pointler kartlara yazılarak gösterilir

- Pointler Fibonacci Serisindeki sayılara göre verilir.

1 1 2 3 5 8 13 21 34

- Her üye niçin bu point'i verdiğini açıklar ve team'in ortak kararıyla user story'nin point'i belirlenir

- Team'deki developer ve QA sayısına göre takım (team) kapasitesi (capacity) belirlenir.

1 point = 1 günlük çalışma yani 8 saat





TAKIM KAPASİTESİ VE PUANLAMA

Örnek: 5 developer'in takım kapasitesi nedir?

$5 * 1 = 5$ (5 developer'in günlük kapasitesi 5 point)

1 haftalık kapasitesi $\Rightarrow 5 * 5 = 25$ point

2 haftalık kapasitesi = 50 point

Örnek: 2 QA'in takım kapasitesi nedir?

$2 * 1 = 2$ (2 developer'in günlük kapasitesi 2 point)

1 haftalık kapasitesi $\Rightarrow 5 * 2 = 10$ point

2 haftalık kapasitesi = 20 point

Development team'de 1 kişinin 2 haftalık sprintti yapacağı iş miktarı $5 + 5 = 10$ point'tir.

