Yazılım yaşam döngü modellerini birçok başlıkta inceleyebiliriz ama ben ana başlıkta incelemek istiyorum. İlk olarak yazılım yaşam döngülerinin ne olduğuyla başlayalım. Günümüzde birçok yazılım yaşam döngüsü vardır neredeyse hepsinin benzerlikleri vardır bunun sebebi ise hepsinin amacının aynı olmasıdır. Yazılımı daha kolay, sistematik, verimli ve hızlı geliştirebilmektir. Bazı yazılım yaşam döngülerinin açıklaması şu şekildedir. Gelişigüzel yazılım modeli : Herhangi bir süreç veya model takip edilmez yazılımı geliştiren kişinin inisiyatifine göre ilerlenir bundan dolayı genelde tekil geliştirmelerde kullanılır.1960’lı yıllarda kullanılmıştır. Yazılım geliştirmede gelişigüzel yaşam döngüsü kullanıldığında kişinin inisiyatifine göre ilerlendiği için başkasının sürece dahil olması veya devam etmesi oldukça zordur. Barok yaşam döngüsü : Bu model 1970’lerde kullanılmıştır. Model herhangi bir döngüye sahip değildir daha çok doğrusal bir geliştirme yapılır. Model geliştirme aşamasına daha çok önem verdiği için belgeleme aşaması ancak test süreci bittikten sonra yapılır. Waterfall yaşam döngüsü ; Türkçesi şelale yaşam döngüsü olarak bilinen bu döngü günümüzde kullanılan yaşam döngülerinin atasıdır. Altı aşamalıdır gereksinim aşaması problemin ne olduğunu ve kullanıcının isteklerini ve bu istekleri karşılayabilmek için gereksinimleri belirlediğimiz aşamadır. Analiz aşaması toplanan gereksinimlerin ve geliştirilecek yazılımın ne yapacağını analiz ettiğimiz aşama. Tasarım aşaması geliştirilen yazılım için tasarım yapılır. Gerçekleştirme aşaması tasarım aşamasında belirlenen bölümler kodlanır ve kodlanan her bölüm için ayrı bir test yapılır en son yazılımın bütünü test edilir ve kullanıcının test etmesi için teslim edilir. Teslim sonrası aşaması teslim sonrasında yazılımın bakımı ve geliştirilmesi için yapılacak işlemlerdir. ve bir aşama tamamlanmadan diğer aşamalara geçilmez ve her aşama en az bir kere tekrar edilir bu da bir takım sorunlara yol açmaktadır. Bunlar ise ; yaşam döngüsünde kodlama sadece bir kısımda kullanıldığı için ekip sıkılabilir. Farklı gruplar farklı basamakları yapması durumda ekipten birinin dahil olmadığı bir aşamada yapılan hata çok sonralarda tespit edilebilir. Basit ve kısa süreli bir yaşam döngüsü olduğu için uzun sürecek veya yüksek maliyetli projelerde genelde kullanılmaz V süreç modeli temel olarak 3 modelden oluşmuştur. Bunlar kullanıcı modeli mimari model ve gerçekleştirim modelidir. Kullanıcı modelinde gereksinimler bulunur ve maliyet hesabı yapılıp teslim edilir. Mimari modelde ise projenin tasarımı ve deneme işlemleri yapılır. Gerçekleştirim modelinde gerekli kodlama işlemleri ve denemeleri yapılır. Helezonik model spiral model olarak da bilinir. Temelde 4 aşamadan oluşur ve her aşamada risk analizi yapılır bu yüzden hatalar erken fark edilir ve çözümlenmesi kolay olur. Planlama aşaması her aşama planlanır ve aşama sonlarında oluşacak ürünler planlanır. Risk analizi aşaması geliştirme süresince alınacak risklerin araştırılması ve çözümlenmesi bu aşamada yapılır. Üretim aşaması projenin kodlanması tasarlanması vb. işlemler bu aşamada yapılır ve sonucunda bir ürün veya ara ürünü elde edilir. Kullanıcı değerlendirme aşaması oluşturulan ürünün kullanıcı tarafından test edilmesi ve alınan geri dönütlerin değerlendirilmesi. Bu model her aşamada risk analiz yapıldığı için küçük projeler için pahalı ve uzun süreli bir süreç gerektirir bu yüzden tercih edilmez. Arıtımsal geliştirme süreç modeli bu modelde proje parçalara bölünür ve geliştiricinin inisiyatifine göre öncelik sırasına göre sıralanır ardından her parça çözülür ve bir ürün elde edilir. Elde edilen ürün sonraki parçaya eklenerek tekrar çözülür. Evrimsel geliştirme modeli ilk tam ölçekli modeldir çoklu kullanıma uygundur. Yazılımın başarısı ilk evirilen modele bağlıdır. Geliştirmeden sonra değişiklik yapmak veya geliştirmek oldukça zordur.Agile geliştirme modeli Türkçe çevirisi çevik geliştirme modelidir. Şelale modelinin eksiklerini kapatmak amacıyla geliştirilmiştir. Modelin temel mantığı kısa geliştirmeler yaparak ürün elde edilmesi ve kullanıcıdan alınan geri dönüşler vasıtasıyla ürünlerin geliştirilmesine dayanmaktadır. Kısa ve etkili geliştirmeler yapıldığı için ekip içi iletişimin ve moral durumunu yüksek tutmaktadır. SCRUM aslen Agile geliştirme modelinin bir alt basamağı olarak kabul edilen SCRUM günümüzde en çok tercih edilen geliştirme modelidir. Proje parçalara bölünür ve her parçaya sprint adı verilir. Agile geliştirme modelinde olduğu gibi sürekli geri dönüşlere verilen yanıtlar ile geliştirme yapılır. Ekip içi iletişim çok önemli olduğundan neredeyse her gün SCRUM meetings denilen toplantılar yapılır. SCRUM da 3 prensip üzerinde durulur şeffaflık yani projenin ilerleyişi net görülebilir olmalıdır denetleme yani projenin ilerleyişi devamlı kontrol edilir uyarlama alınan geri dönüşler vasıtasıyla yapılacak değişikliklere uyum sağlayabilmelidir ayrıca SCRUM da 3 temel aşama bulunur bunlar roller toplantılar ve bileşenlerdir. Roller product owner yani ürün sahibi SCRUM master yani scrum yöneticisi ve development team yani geliştirme takımından oluşur product owner müşteri ve takım arasındaki diyaloglardan sorumludur proje için müşterinin isteklerine göre projenin özlelliklerini tanımlar projenin önceliklerine göre product backlogu yani proje için gereksinim listesini oluşturur ve projenin ilerleyişine göre sprintlerin seçimi ve iptali yetkisine sahiptir.SCRUM master ise scrumun kurallarını ve pratiklerini iyi bilir takımın buna uygun hareket etmesinden sorumludur. Development team ise bir sprinte alınan ve yüklenen bütün görevlerden sorumludurlar kendi kendilerini yönetirler.Toplantılar ise sprint planning yani sprint planlaması daily SCRUM yani günlük SCRUM toplantıları sprint review yani sprint gözlemi ve sprint retrospective yani sprint geri dönütü olarak adlandırılır.sprint planning toplantılarında projenin product backlogunda belirlenen gereksinimleri karşılamak için oluşturalacak modüller küçük görevlere ayrılır ve takımdaki herkes kendi becerisine göre bu görevleri alır.Daily SCRUM toplantılarında her gün geliştirme takımındaki herkes kendi görevinde ve geliştirilen sprintlerde nereye kadar geldiğinin yarın nereye kadar gelineceğinin veya geliştirmelerini engelleyen sorunların olup olmadığını diğer geliştirme grup üyeleriyle paylaşır.Sprint review her sprint sonunda yapılır ve geliştirilen sprint gözden geçirilir.Sprint retrospective sprint boyunca yapılan işlerin kalitesinin gözlendiği toplantılardır sprint review toplantılarına göre daha kapsamlı ve daha detaycı toplantılardır.Bileşenler aşamasında ise product backlog içerisinde bulunan gereksinimler incelenir ve sprint sonunda sprintlerin uygunluğu kontrol edilir.SCRUMU diğer modellerle karşılaştıracak olursak SCRUM ve gelişigüzel yazılım modelini karşılaştıracak olursak SCRUM bir ekip çalışmasını gerektirdiği için maliyeti daha fazla olacaktır ama gelişigüzel yazılım modelinde çalışabilecek olan bir kişinin yapacağı işi daha kısa sürede ve daha kaliteli yapabileceği için maliyet bakımından daha tercih edilebilir olacaktır.Küçük çaplı ve az gereksinim istenen işlerde SCRUM için bir ekip toplamak yerine gelişigüzel yazılım modeli kullanılması daha uygundur. SCRUM ve barok yaşam modelini karşılaştıracak olursak yine büyük çaplı işlerde gerek doğrusallığı gerek belgeleme yetersizliği ve yapılan hataların daha geç farkedilebilir olmasından kaynaklı barok yaşam modeli daha az tercih edilebilirdir.SCRUM ve waterfall yaşam modelini kıyaslayacak olursak yine büyük çaplı işlerde kullanılması tercih edilmeyen ama kullanılabilen waterfall yaşam modeline göre daha sistematiktir ve yapılan hataların kontrolü çözümü ve farkedilmesi daha kolay olacaktır.SCRUM ve v süreç modelini kıyaslayacak olursak SCRUM’un anlık olarak maliyetinin daha fazla olduğu düşünülebilir fakat yapılan proje 0 hatasız olamayacağı için uzun vadede SCRUM daha az maliyetli daha dinamik ve daha iyi bir geliştirme sergileyecektir.SCRUM ve helezonik yaşam modeli kıyaslayacak olursak helezonik yaşam modeli hata ayıklama ve hataların çözümlenmesinde SCRUMdan geri kalmayacaktır ama küçük çaplı işlerde helezonik yaşam modeli hata bulunduktan sonra neredeyse en başından başlanması gerektiği için SCRUMa göre daha maliyetli olacaktır.SCRUM ve agile yaşam modelini kıyaslayacak olursak SCRUM agile yaşam modelinin alt basamağı olarak kabul edilebileceği için böyle bir kıyaslama yapmak çok doğru değildir ama SCRUM agile yaşam modeline göre çok daha sistematik olduğu için SCRUM agile yaşam modeline net bir üstünlük sağlayacaktır.SCRUM’un günümüzdeki popüleritesini kazanmasındaki en büyük etken günümüzde yazılımın bir grup çalışması olduğunun idrak edilmesidir.SCRUM yazılım alanı haricinde farklı alanlarda büyük çaplı projlerin yönetiminde ve gerçekleştirilmesinde kullanılır bunun nedenleri ise SCRUM kullanılırken zamandan ve paradan büyük oranda tasarruf edilmesidir.Görece esnek bir model olduğu için gelecekte kullanılabilecek teknolojilere kolaylıkla uyum sağlayabilir. Çok büyük çaplı projelerde gereksinimler net olarak belirlenmemişse bile uygulanabilir olması. Ekip içi moral ve motivasyon yüksek tutulduğu için projede az hata çıkması ve hataların erken farkedilmesi.Müşteri ve kullanıcıdan sürekli geri bildirim alındığı için proje teslim edilirken müşteri memnuniyetinin fazla olması.Diğer bazı yazılım modelleri gibi doğrusal olmadığından daha esnek ve rahat geliştirme olanağı sağlaması ve diğer bazı yazılım modelleri gibi yinelemeli olması.Projenin küçük parçalara bölünmesi ve parçaların gruptan kendini yeterli gören grup üyeleri tarafından alınması.Projede müşterinin gereksinimleri değiştirmesi durumunda proje parçalara bölündüğü içi projeye baştan başlamak yerine gerekli parçaların değiştirilmesi ile sorun çözülebilmesi.Hangi projede hangi model kullanılması gerektiği düşünüldüğünde çok çok düşük maliyetli ve küçük çaplı bir iş olmadığı sürece SCRUM her projede kullanılabilir ama genel olarak düşük maliyetli ve küçük çaplı işlerde waterfall yaşam modeli veya v süreç modeli kullanılmalıdır. Büyük çaplı ve yüksek maliyetli işlerde en fazla verim SCRUM kullanılarak alınacaktır ama agile geliştirme modelide kullanılabilir.

Ad-Soyad:Burak Ünsal

Öğrenci no:210601038

Linkedin hesap adresi: <https://www.linkedin.com/in/burak-%C3%BCnsal-754a47222/>

Githup hesap adresi: <https://github.com/unsalburak>

Medium hesap adresi: <https://medium.com/@burak_unsalce>