YAZILIM YAŞAM DÖNGÜ MODELLERİ

Yazılım yaşam döngüsü, bir yazılım projesinin hayatının farklı aşamalarını belirleyen ve yöneten bir süreçtir. Bu aşamalar, yazılımın tasarımından, geliştirilmesine, test edilmesine, dağıtılmasına ve bakımına kadar uzanır. Birçok farklı yazılım yaşam döngü modeli vardır ve bunlar genellikle proje ihtiyaçlarına ve geliştirme ekibinin tercihlerine göre seçilir.

Yazılım yaşam döngü modelleri, yazılım geliştirme sürecindeki her bir aşamada alınması gereken adımları ve bu adımların sıralamasını belirler. Bir yazılım yaşam döngü modeli, geliştirme ekibi tarafından uygulanarak yazılımın doğru bir şekilde tasarlanmasını, geliştirilmesini ve test edilmesini sağlar.

İşte bazı yazılım yaşam döngü modellerinden bazıları:

1. **V Süreç Modeli:** V-Model sistem veya yazılım geliştirme sürecinin farklı aşamalarını tanımlayan bir modeldir. V-Model, adını geliştirme sürecinin V şeklinde bir şekille gösterilmesinden alır. Bu modelde, her bir geliştirme aşamasının bir karşılığı vardır ve bu aşamaların tamamlanması sırasıyla yapılır.

V-Modelin Avantajları:

- Doğruluk: V-Model, her bir aşamayı tamamlamadan önce doğrulama ve onaylama sürecini içerir, bu nedenle hataları erken tespit edebilir ve düzeltmelerin maliyetini azaltabilir.
- Şeffaflık: V-Model, projenin her aşamasındaki işlemleri ve çıktıları açıkça tanımlar, bu nedenle proje ilerlemesi kolayca izlenebilir.
- Müşteri Odaklılık: V-Model, kullanıcının gereksinimlerini anlamak için gereken analiz ve test süreçlerini içerir, bu nedenle müşteri odaklı bir yaklaşım sunar.
- Tekrar Kullanılabilirlik: V-Model, bir sonraki projede kullanılabilecek test senaryoları, tasarım şablonları ve belgeler gibi tekrar kullanılabilir varlıkların oluşturulmasını sağlar.

V-Modelin Dezavantajları:

- Esneklik eksikliği: V-Model, bir sonraki adımın yalnızca bir önceki adım tamamlandıktan sonra gerçekleştirilmesini gerektirir. Bu, süreçte esnekliğin olmamasına neden olur. Bu nedenle, değişen ihtiyaçlara yanıt vermek için ekstra çaba harcanması gerekebilir.
- Yüksek maliyet: V-Model, test aşamalarını yoğunlaştırdığından, test maliyetleri yüksek olabilir. Ayrıca, her test aşaması için ayrı bir test ekibi ve test araçları gerektirir.
- Kullanılabilirlik ve kullanıcı deneyimi: V-Model, kullanılabilirlik ve kullanıcı deneyimi konularını atlar veya azaltır. Bu, son kullanıcı ihtiyaçlarını karşılamayan bir ürüne yol açabilir.
- Dokümantasyon yükü: V-Model, belgelendirme ve dokümantasyona ağırlık verir. Bu, geliştirme sürecinin daha uzun ve karmaşık hale gelmesine neden olabilir.

2. **Spiral (Helezonik) Model:** Spiral model, yazılım geliştirme sürecinin planlamasında kullanılan bir çerçevedir. Bu model, bir döngüsel işlem olarak tasarlanmıştır ve her döngüde bir önceki döngüdeki bilgilerin geri bildirimi ile bir sonraki döngüye geçilir. Bu model, risk yönetimi ve güvenlik odaklıdır. Spiral model, aşağıdaki dört temel öğeyi içerir:

    - Planlama: Bu aşamada, projenin amaçları ve hedefleri belirlenir ve geliştirme süreci için bir plan oluşturulur. Riskler ve kısıtlamalar değerlendirilir ve projenin takvimi belirlenir.
    - Risk Analizi: Bu aşamada, projenin riskleri belirlenir ve analiz edilir. Bu riskler, projenin tamamlanmasını engelleyebilecek faktörlerdir. Riskler değerlendirilir ve önceliklendirilir ve risklerle ilgili önlemler alınır.
    - Mühendislik: Bu aşamada, yazılımın tasarımı, geliştirilmesi, test edilmesi ve entegrasyonu gerçekleştirilir. Bu aşamada, yazılımın gereksinimleri karşıladığından emin olmak için düzenli testler yapılır.
    - Değerlendirme: Bu aşamada, projenin başarısı değerlendirilir. Bu aşamada, müşteri geri bildirimleri toplanır ve projenin riskleri ve başarısı değerlendirilir. Bu değerlendirmeler, gelecekteki projelerin iyileştirilmesine yardımcı olmak için kullanılabilir.

    Spiral (Helezonik) Model Avantajları:

    - Esneklik: Spiral model, esnek bir modeldir. Bu model, yazılım geliştirme sürecinde yapılan değişikliklere kolayca uyum sağlar.
    - Risk Yönetimi: Spiral model, risklerin belirlenmesi ve yönetilmesi için etkili bir modeldir. Her spiralin bir risk analizi bölümü vardır. Bu sayede, risklerin önceden belirlenmesi ve önlenmesi için gerekli önlemler alınabilir.
    - Prototip Geliştirme: Spiral model, prototip geliştirmek için de kullanılabilir. Prototip geliştirme, yazılım geliştirme sürecinin daha iyi anlaşılmasını sağlar ve müşteri geri bildirimlerinin hızlı bir şekilde alınmasını sağlar.
    - Müşteri Katılımı: Spiral model, müşteri katılımını sağlamak için uygun bir modeldir. Müşteri, yazılım geliştirme sürecinin her aşamasında yer alabilir ve geri bildirimlerini doğrudan verebilir.

    Spiral (Helezonik) Model Dezavantajları:

    - Karmaşıklık: Spiral model, diğer modellere göre daha karmaşıktır. Bu nedenle, bir proje yöneticisi ve takımının spiral modelini anlamaları ve uygulamaları zaman alabilir.
    - Maliyet: Spiral model, projenin riskleriyle ilgili olarak her döngüde risk analizi ve prototip geliştirme gibi ek aktiviteler gerektirdiği için daha fazla maliyete neden olabilir.
    - Kaynak Yoğunluğu: Spiral model, birçok farklı aktivite ve döngü içerdiği için kaynak yoğunluğuna neden olabilir. Bu nedenle, projenin sınırlı kaynakları varsa, spiral modelinin kullanılması zor olabilir.

3. **Çevik Model:** Çevik Model, yazılım geliştirme sürecinde bir iterasyonel ve inkrementel yaklaşıma dayalı bir yazılım yaşam döngüsü modelidir. Bu modelde,

yazılımın kullanıcı ihtiyaçlarına ve değişen gereksinimlere uygun olarak hızlı ve esnek bir şekilde geliştirilmesi hedeflenir.
Çevik Modelde, yazılım geliştirme süreci, kısa süreli "sprint" adı verilen zaman dilimlerine bölünür. Her sprint, belirli bir hedef veya işlevsellik eklenmesiyle sonuçlanır. Sprintler, ekip üyelerinin düzenli toplantılar düzenleyerek süreci değerlendirmesi ve gerektiğinde planları ayarlamasıyla yönetilir.
Bu modelde, müşteri veya kullanıcılar, yazılımın geliştirilmesi sürecine dahil edilir ve yazılımın ihtiyaç duyduğu değişiklikleri belirler. Bu şekilde, yazılımın tam olarak müşteri ihtiyaçlarına uygun olarak geliştirilmesi hedeflenir.

Çevik Model Avantajları ve Dezavantajları

- Çevik Modelin avantajları arasında, müşteri memnuniyetinin artması, hızlı ve etkili bir şekilde yanıt verilebilmesi, sürekli gelişim sağlanabilmesi, ekip üyeleri arasındaki iletişim ve iş birliğinin artması gibi faktörler yer alır. Ancak, bu modelin dezavantajları arasında, projenin bütçe ve zaman kısıtlamalarına uygun şekilde yönetilememesi, sürekli değişen gereksinimlerin yönetim zorluğu, belirsizliklerin oluşması gibi durumlar yer alabilir.

4. **Şelale Modeli (Waterfall Model):** Şelale Modeli (Waterfall Model), yazılım geliştirme sürecinde kullanılan en eski ve en yaygın modeldir. Bu model, yazılım geliştirme sürecini adım adım ve sırayla ilerleyen bir dizi aşamaya ayırır.
Şelale Modelinde, yazılım geliştirme süreci aşağıdaki gibi sıralı aşamalardan oluşur:

   - Gereksinim Analizi: Müşteri gereksinimlerinin toplanması, analizi ve belgelendirilmesi.
   - Tasarım: Gereksinimlere dayalı olarak sistemin mimarisi, alt sistemleri ve modülleri tasarlanır.
   - Geliştirme: Kodlama, birim testleri, modül testleri, entegrasyon testleri ve sistem testleri gibi aktivitelerle yazılımın geliştirilmesi.
   - Test: Yazılımın işlevselliği, performansı ve hataları gibi konuların test edilmesi.
   - Uygulama: Yazılımın canlı sistemlere uygulanması, kullanıcılara sunulması ve bakımının yapılması.

   Her aşama tamamlanmadan bir sonraki aşamaya geçilmez. Ayrıca, bu modelde herhangi bir aşamaya geri dönüş yapılamaz ve her aşama tamamlandığında bir sonraki aşamaya geçilir.
   Şelale Modeli Avantajları:
   - Kolay anlaşılır: Şelale Modeli, yazılım geliştirme sürecinin sıralı bir şekilde ilerlediği basit bir yaklaşımı benimsediği için kolayca anlaşılabilir.
   - Öngörülebilirlik: Bu model, her aşamanın sonunda belirli bir çıktı verdiği için, sürecin öngörülebilirliği artar.

- Dokümantasyon: Her aşama tamamlandıktan sonra, dokümantasyon yapılarak projenin kontrol edilmesi ve takip edilmesi kolaylaştırılır.
- Ölçülebilirlik: Her aşama tamamlandığında, işin yapılıp yapılmadığı kolayca ölçülebilir.

Şelale Modeli Dezavantajları:

- Esneklik: Şelale Modeli, değişen gereksinimleri karşılamakta esnek değildir. Bir aşama tamamlandıktan sonra, geri dönüp değişiklik yapmak zordur.
- Geri Bildirim: Bu modelde, müşteri geri bildirimleri işin tamamlandığı son aşamada alınır. Bu da müşteri geri bildirimlerinin daha geç alınmasına neden olabilir.
- Kaynak: Bu modelde, her aşama tamamlanmadan diğerine geçilemez. Bu da gereksiz kaynak kullanımına ve proje maliyetlerinin artmasına neden olabilir.
- Uyumlu Değil: Bu model, modern yazılım geliştirme süreçlerine uygun değildir. Çevik veya esnek geliştirme modelleri, şelale modeline göre daha tercih edilir hale gelmiştir.

5. **SCRUM:** SCRUM Modeli, yazılım geliştirme sürecindeki çevik yaklaşımın en popüler uygulamalarından biridir. Bu model, yazılım geliştirme sürecini takım tabanlı bir yaklaşımla yöneterek, projenin müşteri ihtiyaçlarına ve değişen gereksinimlere uygun şekilde hızlı ve esnek bir şekilde geliştirilmesini hedefler. SCRUM Modelinde, yazılım geliştirme süreci, "sprint" adı verilen kısa zaman dilimlerine bölünür. Sprint, genellikle 2-4 hafta sürer ve belirli bir hedef veya işlevsellik eklenmesiyle sonuçlanır. Her sprint, bir takım üyesi tarafından yönetilir ve takım üyeleri düzenli olarak toplantılar düzenleyerek sprintlerin ilerlemesini değerlendirir ve gerektiğinde planları ayarlar. SCRUM Modelinde, takımın görevleri ve sorumlulukları açık bir şekilde tanımlanır. Takım üyeleri, düzenli aralıklarla birbirlerine geri bildirim verir ve gelişimlerini takip eder. Bu modelde, müşteri veya kullanıcılar, yazılımın geliştirilmesi sürecine dahil edilir ve yazılımın ihtiyaç duyduğu değişiklikleri belirler.

SCRUM Avantajları ve Dezavantajları

- SCRUM Modelinin avantajları arasında, müşteri memnuniyetinin artması, sürekli gelişim sağlanabilmesi, ekip üyeleri arasındaki iletişim ve iş birliğinin artması, takım motivasyonunun yüksek olması gibi faktörler yer alır. Ancak, bu modelin dezavantajları arasında, belirli bir disiplin ve öz disiplin gerektirmesi, projenin bütçe ve zaman kısıtlamalarına uygun şekilde yönetilememesi, belirsizliklerin oluşması gibi durumlar yer alabilir.

SCRUM Günümüzde Neden Popüler?

SCRUM, günümüzde yazılım geliştirme alanında en popüler proje yönetim metodolojilerinden biridir. SCRUM'ın popülerliği, birçok faktörden kaynaklanmaktadır.

Birincisi, SCRUM, çevik bir yönetim metodolojisi olarak tanımlanır. Bu, iş gereksinimlerinin ve müşteri ihtiyaçlarının sürekli olarak değiştiği bir ortamda kullanılabilir olmasını sağlar. SCRUM, proje yönetiminde sıkça karşılaşılan bir sorun olan, değişen gereksinimlerin yönetimini kolaylaştırır.

İkinci olarak, SCRUM, projelerin hızlı bir şekilde teslim edilmesine olanak tanır. SCRUM'ın odaklandığı kısa döngüler, yani sprintler, projenin daha hızlı bir şekilde ilerlemesini sağlar. Bu, müşterilerin, projenin ilerlemesini takip edebilmelerini ve hızlı geri bildirimde bulunabilmelerini sağlar.

Üçüncü olarak, SCRUM, proje ekibinin etkili bir şekilde çalışmasını sağlar. SCRUM'ın düzenli toplantıları, proje ekibinin birbirleriyle iletişim kurmalarını ve birbirlerinden geri bildirim almalarını sağlar. Bu, projenin başarı şansını artırır ve ekibin birlikte çalışma becerilerini geliştirir.

Dördüncü olarak, SCRUM, proje risklerinin azaltılmasına yardımcı olur. SCRUM'ın sprintleri, projenin belirli bir hedefe odaklanmasını sağlar. Bu, proje risklerinin belirlenmesini ve bu risklerin ele alınmasını kolaylaştırır. SCRUM'ın düzenli toplantıları ve geri bildirimleri, projedeki risklerin daha erken tespit edilmesine ve ele alınmasına yardımcı olur.

Sonuç olarak, SCRUM, çevik bir proje yönetim metodolojisi olarak günümüzde popülerdir. SCRUM, değişen iş gereksinimlerine ve müşteri ihtiyaçlarına uyum sağlayabilir, projelerin hızlı bir şekilde teslim edilmesine olanak tanır, ekibin etkili bir şekilde çalışmasını sağlar ve proje risklerinin azaltılmasına yardımcı olur. Bu nedenlerden dolayı, SCRUM, günümüzde popüler bir proje yönetim metodolojisi olarak kabul edilir.

Yusuf Batmaz

220601024

Github hesap adresim: https://github.com/YusufBatmaz

Medium hesap adresim: https://medium.com/@YusufBatmaz

LinkedIn hesap adresim: https://www.linkedin.com/in/yusuf-batmaz-180610251/