

Yazılım Geliştirme Süreçleri

Yazar: Gökmen Göksel, Semen Cirit, Renan Çakırerk, Beyza Ermiş
Sürüm: 0.1



Sürecin Amacı ve Politikası

Kullanıcının veya müşterinin istediği yazılım ürününü ortaya çıkarmak için, gereksinimleri analiz etmek, tasarımını yapmak, geliştirmek ve test etmektir.

Sürecin Kapsamı

Yazılım geliştirme aktivitelerini kapsar. Projenin kapatılması ile sona erer.

Sürecin Sahibi

Yazılımı geliştirme sorumluluğunu alan Proje Lideri ve birlikte çalıştığı ekip.

Süreç İlişkileri

Bu süreç, Dağıtım Koordinasyonu ile paralel bir şekilde gerçekleştirilir.

Süreç Girdileri/Çıktıları

Girdileri

- Proje Planı
- Müşteri veya kullanıcı tarafından teslim edilen ve projenin gerçekleştirilmesi için gerekli her türlü doküman
- Yazılım Test Planı

Çıktıları

- Sistem gereksinimleri Spesifikasyonu
- Yazılım Gereksinimleri Spesifikasyonu
- Yazılım Tasarımı Tanımlama Dokümanı
- Arayüz Tasarım Tanımlama Dokümanı
- Veritabanı Tasarım Tanımlama Dokümanı

Süreç Kaynakları

İnsan Kaynakları

- Teknoloji Sorumlusu ilgili proje için gerekli teknolojiye yön verme ve teknik alt yapı için son onayı vermeden sorumludur.
- Dağıtım Koordinatörü projenin yazılım süreçlerine uygunluğunu ve diğer projeler ve sürümler ile birlikte işin ilerleme ve zamanlamasını dengeleme ve kontrol etme işinden sorumludur.
- Proje Lideri, yapılacak olan işin içeriğine göre, Teknoloji Sorumlusu ve Dağıtım Koordinatörü tarafından seçilir.
- Projede yer alacak olan geliştiriciler, Proje Liderinin talebi üzerine Dağıtım Koordinatörü ile birlikte seçilir.
- Geliştiriciler ve Proje Lideri projenin yazılım gereksinimi ve rakip analizi, yazılım tasarımı, yazılım geliştirme, unit test, sistem entegrasyonu ve paketlenmesinden sorumludur.
- Proje Lideri ayrıca bu işlerin takipçisi ve bir üst birime (Dağıtım Koordinatörü ve Teknoloji Sorumlusu) raporlayıcısıdır.

Süreç Aktiviteleri

Proje Başlangıcı

Proje başlangıcı yazılımı talep eden kişi ya da kişilerin, yazılımın bir ya da iki cümle ile direkt olarak yapması gereken ana işlevi içeren gereksinim ile başlar. Bu cümle aşağı-

dakine benzer bir şekilde olabilir;

“Sistemde bulunan kullanıcıların yönetimi için bir grafik ara birim gerekmektedir.”

Yazılımı talep eden kişinin Dağıtım Koordinatörüne bu bilgiyi aktarması sonrasında, bu işle ilgili bir Proje Lideri belirlenir.

Proje Lideri belirlendikten sonra, proje için temsili bir kod isim belirlenir ve bu isim ile birlikte Pardus İş Takip sistemi (tracker.pardus.org.tr)nde, proje açılır. Proje gereksinimleri doğrultusunda başka bir projenin alt projesi olarak ifade edilebilir. Bu gibi durumlarda üst projenin sorumlusunun da söz konusu projeye -danışman olarak dahi- katılımı yerinde olacaktır. Ayrıca, projenin gizlilik durumuna bağlı olarak “internal” ya da “uludag” kod depolarından birinde konu ile ilgili alt dizinin altında bir çalışma dizini yaratılır. Yapılan tüm çalışmalar ile ilgili bilgi ve belgeler bu dizin altında toplanmalıdır. Tüm süreç, İş Takip sistemi üzerinden yürütülmelidir ve tüm bilgi ve belgeler proje çalışma dizininde tutulmalıdır. İhtiyaç olmadığı kesinleşse dahi, kayıt amaçlı olarak, dizine koyulan hiçbir belge silinmemeli, proje bitimine ya da kullanılabilir sürümüne geldiğinde “Proje Sonuç Raporu” dahilinde gerekli birimlere sunulmalıdır. Ayrıca projenin ihtiyacı olması durumunda, proje konularını tartışabilmek üzere bir elektronik posta listesi açılabilir.

Çalışma deposu, belirlenmiş kod adı ile birlikte projenin gizlilik durumuna bağlı olarak, “internal” ya da “projeler” depoları içerisinde açılmalıdır.

Gizli Projeler

Gizli Projelerin tümü ile ilgili tüm kod ve belgeler “internal” altındaki proje dizininde yer alır.

Açık Projeler

Projenin kodlanma sürecini takiben, proje için Pardus Git sunucusunda, proje kod adı ile bir Git deposu açılır. Bu aşamaya kadar hazırlanmış, geliştirilmiş tüm belgeler bu depoya taşınır. Bundan sonra geliştirilecek tüm kodlar ve belgeler bu depo içerisinde tutulur/güncellenir.

Projenin gizli bir proje olarak başlaması ardından açık bir proje haline getirilmesi kararı verildiği takdirde; bu noktaya kadar “internal” içerisinde tutulan belge ve bilgiler açık sunulan git deposuna taşınır.

Sistem Gereksinimlerinin Oluşturulması

1. İlk sistem gereksinimleri eğer yapılan proje dış projelere dahil ise Dış Projeler Koordinatörü tarafından, değil ise kullanıcılardan gelen yeni özellik istekleri doğrultusunda Dağıtım Koordinatörü tarafından iletilir.

2. Bu sistem gereksinim tablosu kullanıcı tarafında karşılanması hedeflenen öncelikli gereksinimlerin ayrıntılı bir şekilde ifade edilmesi ile oluşturulur (önceliği, hangi kullanıcı veya müşteri için istendiği vb.)
3. Proje Lideri, sistem mühendisi, dış Projeler Koordinatörü (dış proje olması durumunda), dağıtım koordinatörü tarafından sistem gerekleri gözden geçirilir ve ayrıntılandırılır.

Rakip Analizi

1. Proje Lideri, Teknoloji Sorumlusunun onayı ile analiz edilecek rakipleri belirler.
2. Rakip analizi için Proje Lideri, Dağıtım Koordinatörü onayı ile konu ile ilgili bir ekip belirler. (Bu ekip projeyi geliştirecek muhtemel ekiptir, daha sonra farklı eklemeler ve çıkarmalar olabilir.)
3. Rakip Analizi ile ilgili kesin sonuçlar alabilmek adına, konu ile ilgili kişi ya da kişiler tarafından ispat çalışmaları (proof of concept) gerçekleştirilebilir ve bu süreçlerin sonuçları yazılım gereksinimlerini büyük ölçüde şekillendirecektir.
4. Bu süreç içerisinde ayrıca teknik kısıtlar da projeyi gerçekleştirilecek ekip tarafından ifade edilerek, yazılım gereksinimleri üzerinde değişikliğe gidilebilir.

Yazılım Gereksinimlerinin Oluşturulması

1. Sistem gereksinimlerinden ve rakip analizinden yola çıkarak, konu ilgili deneyim sahibi geliştiricilerden oluşan bir grup ile bilgi paylaşımında bulunulur ve yazılım gereksinimleri oluşturulmaya başlanır.
2. Proje Lideri, Sistem Mühendisi ve Teknoloji Sorumlusu ile birlikte yazılım gereksinimlerinin ilk versiyonu gözden geçirilir ve ayrıntılandırılır.

Gereksinim Tablosu Sürümlendirme

Sistem ve yazılım gereksinimleri belirli bir olgunluğa ulaştığında (bu durumu projeyi gerçekleştirecek ve projeyi talep eden kişi ya da kişiler birlikte karar verecektir), Gereksinim Tablosu küçük düzeltmeler haricinde ve çok sıradışı bir durum olmadıkça değiştirilmez. Değiştirilmesi gerektiği durumlarda, her iki ekibin onayının alınması gerekir ve bu değişiklik gereksinim tablosunun yeni sürümü ile sunulabilir.

Gereksinimlerin Önceliklendirilmesi

Gereksinim tablosu dondurulduktan sonra, projeyi talep eden, Proje Lideri ve rakip analizini gerçekleştiren ekibin katılımı ile birlikte, gereksinimlerin önceliklendirilmesi ile ilgili bir süreç başlar. Bu süreç, projenin, talebi içeren ilk cümleyi en temel anlamda gerçekleştirdiği duruma gelene kadar yapılması gerekenlerin sıralamasını temel alır.

Proje Ekibinin Belirlenmesi

Yapılacak olan yazılım gereksinimlerinin büyüklüğüne ve aciliyetine göre Proje Lideri, Dağıtım Koordinatörü ile birlikte bir ekip oluşturur.

Teknik Analiz

Bu aşamadan sonraki adımlar, projeyi gerçekleştirecek ekibin sorumluluğundadır. Süreç teknik analiz ve uygulanabilecek olası yöntem/algoritma vs. nin proje ekibi tarafından belirlenmesi ile başlar. Bu süreç sonucunda kullanılmasında karara varılan yöntem, kütüphane ya da diğer uygulamaların, kullanıcı gereksinimlerinin tamamını karşıladığı kesin bir şekilde ifade edilmelidir. Bu işlem için gereksinim tablosu, onay listesi şeklinde kullanılabilir. Gerekli durumlarda yazılması/kullanıma uygun hale getirilmesi gereken modül ya da ek yazılımlar ile ilgili planlama ayrıca yapılmalıdır.

Görev Paylaşımı

Modüler yapıda geliştirilebilecek projeler için görev paylaşımı gerçekleştirilir. Bu paylaşım süreci, teknik ekipte bulunan kişi ya da kişilerin ve Proje Liderinin talepleri ile başlar. Modül sorumluları ve Görev Dağılımları kesin bir şekilde belirlendikten sonra Proje dokümanına ve iş takip sistemine kayıt edilir.

İş Takibi

Kodlamadan sorumlu kişi ya da kişilerin katılımı ile gerçekleştirilecek teknik toplantıların her birinde Proje Takvimi güncellenmeli, gerektiği durumlarda tarih kaydırma ya da erteleme durumları projeyi talep eden kişi ya da kişilerin onayı ile birlikte takvime işlenmelidir. Bu işlemler Proje Lideri tarafından gerçekleştirilir. Bu takvim Pardus İş Takip sistemi (tracker.pardus.org.tr) üzerinde işletilmelidir ve tüm geliştirme ekibinin bu takvim üzerinden iş paylaşımını sürdüreceği bilinmelidir. İş takip sisteminde belirlenmemiş durumların sorumluluğu, o kısım ile ilgili görevi almış olan kişiye aittir. Projeyi talep eden kişi ya da kişiler de proje durumu ile ilgili bilgiyi İş Takip sisteminden anlık olarak alabilir.

Pardus Yazılım Döngüsü ve İş Kırılımı

Kodlama Süreci

Yazılım Standart Dokümanlarının Hazırlanması

Standart yazılım geliştirme süreçlerini koruyabilmek ve kodlama sürecini bu standartların üzerinden gerçekleştirebilmek için aşağıda listelenmiş raporlar/dokümanlar, kodlama sürecinin başlangıcında hazırlanmalı ve proje dokümanlarına dahil edilmelidir. Tüm geliştirme sürecinin bu dokümanlar temel alınarak yapılacağı unutulmamalıdır. Bu dokümanlarda ihtiyaç dahilinde değişiklik yapılabilir ve kendi içerisinde bu dokümanların da sürüm numaraları olmalıdır.

Bu dokümanları hazırlama görevi, proje ekibi içerisinde belirlenen kişi ya da kişiler ile birlikte, gerekli görüldüğü takdirde projeyi talep eden kişi ya da kişilerin katılımı ile de gerçekleştirilebilir. Diğer tüm işlerde olduğu gibi bu dokümanların hazırlanması ve geliştirilmesi süreci ile ilgili İş Takip sistemi kullanılmalıdır. Modül geliştiricileri arasında bu dokümanlar paylaşılabılır.

- *UML Diyagramı*
- *Akış Diyagramı*
- *Use-Case Diyagram*
- *Nesne Diyagramı Geliştirici ekip özelinde gerçekleştirilir*
- *Senaryolar Sınıflar ve Sınıflar arasındaki ilişkilerin belirlenmesi*
- *Metotlar, değişkenler, parametreler Bu tip tanımlamalarda iskelet kod ve bu kodun üzerinden erişilebilecek belgelendirme yöntemi izlenebilir.*
- *Veri alanlarının belirlenmesi (Veritabanı Seçimi, Tablolar, Indexler ve Tablo ilişkileri) Proje gereksinimlerinde belirtilen Veritabanı gereksinimleri temel alınmalıdır.*

Sınıfların Tanımlanması

Kodlama sürecinin başlangıcında kullanılması planlanan tüm işlevler için detaylı bir sınıf tablosu hazırlanır, bu tablo üzerinden, mümkün olduğunca esnek bir şekilde kullanılacak şekilde ve projenin gereksinimleri doğrultusunda API/kitaplık tasarımı belirlenir. API/kitaplık tasarımı ayrıca belgelendirilmeli ve projenin teknik anlamda en kapsamlı belgesi olacağı unutulmamalıdır.

Ara birimlerin Tanımlanması

Kullanılacak yöntem, özellik vb. için tanımlanmış olan sınıf tablosunun bir benzeri de, gerektiği takdirde arayüzler için ayrıca yapılır. Bu tablo, kullanıcı gereksinimlerinde belirtilen gereksinimleri karşılamak üzere oluşturulması gereken ara birimleri, bu ara birimlerin ana görevlerini ve ek olarak ara birim için hazırlanacak bir çizimi (mockup) içerebilir. Ayrıca çoklu ara birim gereksinimine ihtiyaç duyan projelerde ara birim geçişleri de bir akış diyagramı üzerinde belirtilmelidir.

Arayüz tasarlanırken belirlenen özellikler, kullanıcı gereksinimlerini karşılamak üzere hazırlanmalıdır. Olası farklılıklar da Kullanıcı Gereksinimleri tablosuna geri dönüş yapılabilir fakat bu tercih edilen bir süreç değildir. Arayüzlerin temel hedefinin öncelikle gereksinimleri karşılaması gerekliliği unutulmamalıdır.

Dış Bağımlılıkların Belirlenmesi

Dışarıdan kullanılması gereken tüm kitaplıklar ve bunların bağımlılıkları ile ilgili bir rapor hazırlanmalı ve bu rapor her yapılan değişiklik için güncellenmelidir. Bu raporun içeriği, Gereksinim tablosunda belirtilmiş olan Kısıtlar ile çakışmamalıdır. Örneğin;

“Ürün Linux Sistemlerde çalışabilmelidir.”

cümlesindeki gibi bir kısıta sahip olan bir projenin dış bağımlılıklarının da Linux sistemler ile uyumlu olması gereklidir.

Test Süreçleri

Kodlama süresince, gerçekleştirilen her modül için; öncelikle modülün geliştiricisi tarafından kod üzerinde test yapılır. Bu testler “Yazılım Testleri” dokümanında belirtildiği şekilde gerçekleştirilmeli ve her modül sürümünden önce gerçekleştirildiği konusunda, geliştiricisinin onayı alınmalıdır.

Modül testleri aşağıdaki test süreçlerini içerir. Bu test süreçlerini gerçekleştirecek kişi ya da kişiler proje dokümanında öndecen belirtilmelidir.

- *Birim testleri*
- *Gereksinim testleri*
- *Kurulum testleri*
- *Test ekibinin testleri*
- *Kullanıcı grubu testleri*

Yukarıda tanımlanmış test süreçleri ayrıca modüllerin birleştirildiği tüm sistem için, belirli sürümler/hedefler öncesinde Proje Lideri kontrolünde gerçekleştirilir.

Sürümlerin Belirlenmesi

Kabul Edilen Sürümler

Kodlama süreci içerisinde sürekli güncellenebilecek ve yenilenebilecek bir şekilde, aşağıdaki sürüm durumları için belirli hedef tarihleri belirlenir.

Prototip Sürüm

- *Bu sürüm kod deposunda esnek bir şekilde tutulur, bu sürümün geliştirilmesi sırasında bir kod dizin yapısı kullanılır, bu yapı ile ilgili ayrıntılı bilgi Pardus Kodlama Standartları dokümanında belirtilmiştir.*
- *Bu sürüm için her geliştirici kendi çalışma dizini (branch) yaratabilir, bu süreç projenin gereksinimleri doğrultusunda geliştirici başına ya da gerçekleştirilmesi planlanan işlevler başına ayrıca tanımlanabilir.*
- *Bu sürüm kendi içerisinde bir sürüm numarasına sahip olabilir. (1.0, 2.0 gibi..) Bu sürüm numaraları nihai sürümün numaralarından bağımsız bir şekilde ele alınır.*
- *Proje için kullanılacak depoda eğer destekleniyorsa (Git desteklemektedir) etkin bir şekilde etiketleme (tag) mekanizması kullanılır. Bu etiketleme her prototip sürümü için gerçekleştirilebilir. Bu sürümlendirmenin standartları da Sürüm Numaraları başlığı altında bulunur..*

- Bu sürümde projeden beklenen, öncelikle proje başlangıç cümlesini yerine getirebilmesidir. Bu cümleden yola çıkarak ortaya konulmuş detaylı gereksinimlerin bazıları için de Prototip Sürüm hedef olarak belirlenebilir.
- Prototip Sürüm, projenin gidişatı konusunda net veriler ortaya koyabilir ve bazı durumlarda projenin gereksinimler sürecine geri dönmesine dahi sebep olabilir. Bu gibi durumların belirlenebilmesi için, Prototip Sürüm'ün olası testlerine, projeyi talep eden kişi ya da kişiler de dahil edilmelidir.
- Prototip Sürüm'de geliştirilmiş API/Kitaplık ya da arabirimler, nihai sürüm için temel teşkil ediyor olsa da, Teknoloji Sorumlusu, Test Ekibi ve projeyi talep eden kişi ya da kişilerin katılımı ile gerçekleştirilecek "Prototip Sunumu" toplantıları ardından tamamen değiştirilebilir. Bu gibi durumların proje takvimini olumsuz yönde etkileyeceği unutulmamalıdır.
- Prototip Sürüm son kullanıcı için hazır olmayabilir, arabirim standartlarına ya da kodlama standartlarına uymayan kodlar/arabirimler içerebilir. Çıktıların en temel seviyede olması yeterlidir. Kod, kullanıcı dokümanları eksik olabilir.
- Yazılım Tasarım Belgesi prototip sürüm içerisinde hazırlanır ve developer.pardus.org.tr'de yayınlanır.

Alfa Sürüm

- Bu sürüm Prototip Sürüm'ün kabul edilmesi ardından çıkartılacak ilk sürümdür ve bu sürümün çıkarılmasının ardından projede geri dönüş yapılamaz.
- Bu sürüm kod deposunda "Pardus Kodlama Standartları" dokümanına göre hazırlanmış bir şekilde tutulur. Prototip sürümden devralınan kod ağacı, gerektiği takdirde söz konusu dokümanda belirtildiği şekilde yeniden tasarlanabilir. Alfa sürüm tamamlandıktan sonra kod dizin ağacı yapısında bir değişiklik yapılamaz.
- Bu sürüm için her geliştirici kendi çalışma dizini (branch) yaratabilir, bu süreç projenin gereksinimleri doğrultusunda geliştirici başına ya da gerçekleştirilmesi planlanan işlevler başına ayrıca tanımlanabilir.
- Bu sürüm, ana sürüm ile birlikte sürüm numarası almalıdır. "0.1" gibi bir sürüm numarasının yanına "a" ibaresi eklenebilir; "0.1a" gibi. Bu sürüm numaraları nihai sürümün numaralarından bağımsız bir şekilde ele alınamaz.
- Bu sürüm sırasında Final Sürüm'de sunulması planlanan tüm modüllerin belirlenmiş, gereksinimleri çıkarılmış ve prototip çalışmalarının tamamlanmış olması gerekir. Alfa sürümü sonunda, yeni bir özellik istenmesi durumunda, Final sürümün çıkışından sonra, yeni bir sürüm

hedef alınır.

- Alfa sürümü gereksinimlerse bulunan özelliklerin tamamlanması ile son bulmaktadır.
- Bu sürüm, geri bildirim almak üzere belirli bir kullanıcı kitlesine, alacakları riskleri içeren bir sözleşmeyi onaylamaları karşılığında sunulabilir.

Beta Sürüm

- Beta Sürüm, projenin gelmiş olduğu son noktayı gösterir ve Final Sürüm'e giden yolda özellik kümesi değiştirilemez. Beta Sürüm'den sonra Gereksinimler seviyesinde bir noktaya geri dönüş mümkün değildir. Bu gibi durumlarda, Final Sürüm'ün ortaya konması ardından, yeni bir sürüm (versiyon numarası) hedef alınır.
- Beta Sürüm'de geliştirilmiş API/Kitaplık ya da arabirimler, Final Sürüm'de farklı olamazlar. Final Sürüm'e giden süreçte, Beta Sürüm'de sadece hata düzeltilmesi, çeviri güncellemesi ya da grafik güncellemesi yapılabilir. Bu sebeple, Beta Sürüm proje için büyük önem taşır.
- Bu sürüm sırasında Alfa Sürüm'de geliştirilmiş tüm prototiplerin ürün haline getirilmiş olması beklenir.
- Bu sürüm, ana sürüm ile birlikte sürüm numarası almalıdır. "0.1" gibi bir sürüm numarasının yanına "b" ibaresi eklenebilir; "0.1b" gibi. Bu sürüm numaraları nihai sürümün numaralarından bağımsız bir şekilde ele alınamaz.
- Bu sürüm, geri bildirim almak üzere belirli bir kullanıcı kitlesine, alacakları riskleri içeren bir sözleşmeyi onaylamaları karşılığında sunulabilir.
- Beta Sürüm son kullanıcı için hazır olmalıdır, arabirim standartlarına ya da kodlama standartlarına uymayan kodlar/arabirimler içeremez. Kod, kullanıcı dokümanları tamamlanmış olmalıdır. Bu gibi durumlar sürüm için engelleyici durumlardır.
- Prototip Sürüm içerisinde hazırlanan "Yazılım Tasarım Belgesi", Beta Sürüm'den sonra güncellenemez. Güncellenmesini gerektiren durumlarda Final Sürüm'ün ardından geliştirilecek yeni sürüm (versiyon) beklenmelidir.

Final Sürüm

- Bu sürümde, Beta Sürüm sırasında tespit edilmiş olan ölümcül hatalar ve çeviri ve görsel eksiklikler gibi iyileştirme hataları giderilmiş olmalıdır. Alt yapıyı tamamı ile değiştirecek ve yeni özellik eklemeye kadar gidebilecek olan hatalar için, Final sürümün çıkışından sonra, yeni bir sürüm hedef alınır.

- *Final Sürüm'den sonra Gereksinimler seviyesinde bir noktaya geri dönüş mümkün değildir. Bu gibi durumlarda, Final Sürüm'ün ortaya konması ardından, yeni bir sürüm (versiyon numarası) hedef alınır.*
- *Bu sürüm ile birlikte, projenin dağıtım süreçlerinin de (paketleme, depo gereksinimleri, bağımlılıklar) tamamlanmış olmalıdır.*
- *Bu sürüm, ana sürüm numarası alır. Herhangi bir son ek almaz.*
- *Kod, kullanıcı dokümanları, kurulum dokümanları, teknik destek dokümanları, proje web sayfası tamamlanmış olmalıdır. Bu gibi durumlar sürüm için engelleyici durumlardır.*

Sürüm Numaralandırma

Alfa Sürüm ve sonrasında uygulanacak sürüm numaralandırma standardı aşağıdaki gibidir;

(Ana Sürüm Numarası).(Alt Sürüm Numarası).(Revize Numarası).(Derleme Numarası)

Ana Sürüm Numarasının değişmesi gereği, proje ekibi tarafından bir sürüm öncesine göre yapılmış değişikliklerin içeriğine göre belirlenir. Bir önceki sürümde mevcut olmayan yeni bir modülün sisteme eklenmiş olması, Alt Sürüm Numarasını arttırmak için yeterli bir sebeptir. Revize Numarası, uygulamaya dahil edilmesi gereken her değişikliğin ardından arttırılabilir, bu sayı için depodaki değişiklik numarası temel alınabilir. Derleme Numarası ise geliştiricilerden ziyade kullanıcılar ve test ekibi için önem taşıyan bir numaradır, bu numara ürünün paketlenmesi, kullanıcıya/testçiye yeniden gönderildiği her durumda arttırılır. Ve sürümlendirmeden ziyade, paketlemenin yapıldığı paket yönetim sistemine ve bulunduğu depo şartlarına bağlı olarak projeden bağımsız bir şekilde güncellenir.

Başarı Ölçütleri

1. Yazılım Gereksinimleri dondurulduktan sonra gereksinim değişim oranı (%)
2. Proje İlerleme Raporlarında yer alan yazılım ile ilgili problem/hata sayısı
3. Geliştirici testlerinde bulunan hata sayısı
4. Müşteri kabul testlerinde bulunan (yazılım) hata sayısı
5. Proje boyunca yazılım sürecinden sapma sayısı (yerine getirilmeyen pratikler anlamında)
6. Tekrar üretilen yazılım iş/ürün sayısı

Belgelendirme

Kodlama süresince gerçekleştirilecek kod belgelendirmesi, modülün geliştirilmesi sorumluluğunu alan geliştiriciye aittir. Bu belgelendirme "Pardus Kodlama Standartları" dokümanında belirtildiği şekilde yapılmalıdır.

Modüllerin kullanıcıya sunulması için hazırlanması gereken Yardım Belgeleri de yine modül sorumlusunun iş planı içerisinde yer almaktadır. Tüm sistem ile ilgili Yardım dokümanları için tüm ekibin katılımı ile gerçekleştirilecek toplantılar ya da ortak çalışmalar gerçekleştirilebilir.

Kurulum ve sistemin hazır hale getirilmesi için gereken dokümanların hazırlanması sorumluluğu geliştirme ekibine ait olmak ile birlikte, projeyi talep eden kişi ya da kişilerin önerilerine de başvurulabilir. Bu belgelerin sonucunda ortaya çıkacak kurulum gereksinimleri, Proje Gereksinimlerinde tanımlanan kısıtlara uygun şekilde ifade edilmelidir.

Bakım için gereken tüm adımlar ve teknik gereklilikler, Bakım Dokümanında belirtilmeli ve bu dokümanın hedef kitlesinin Teknik Personel olduğu unutulmamalıdır.