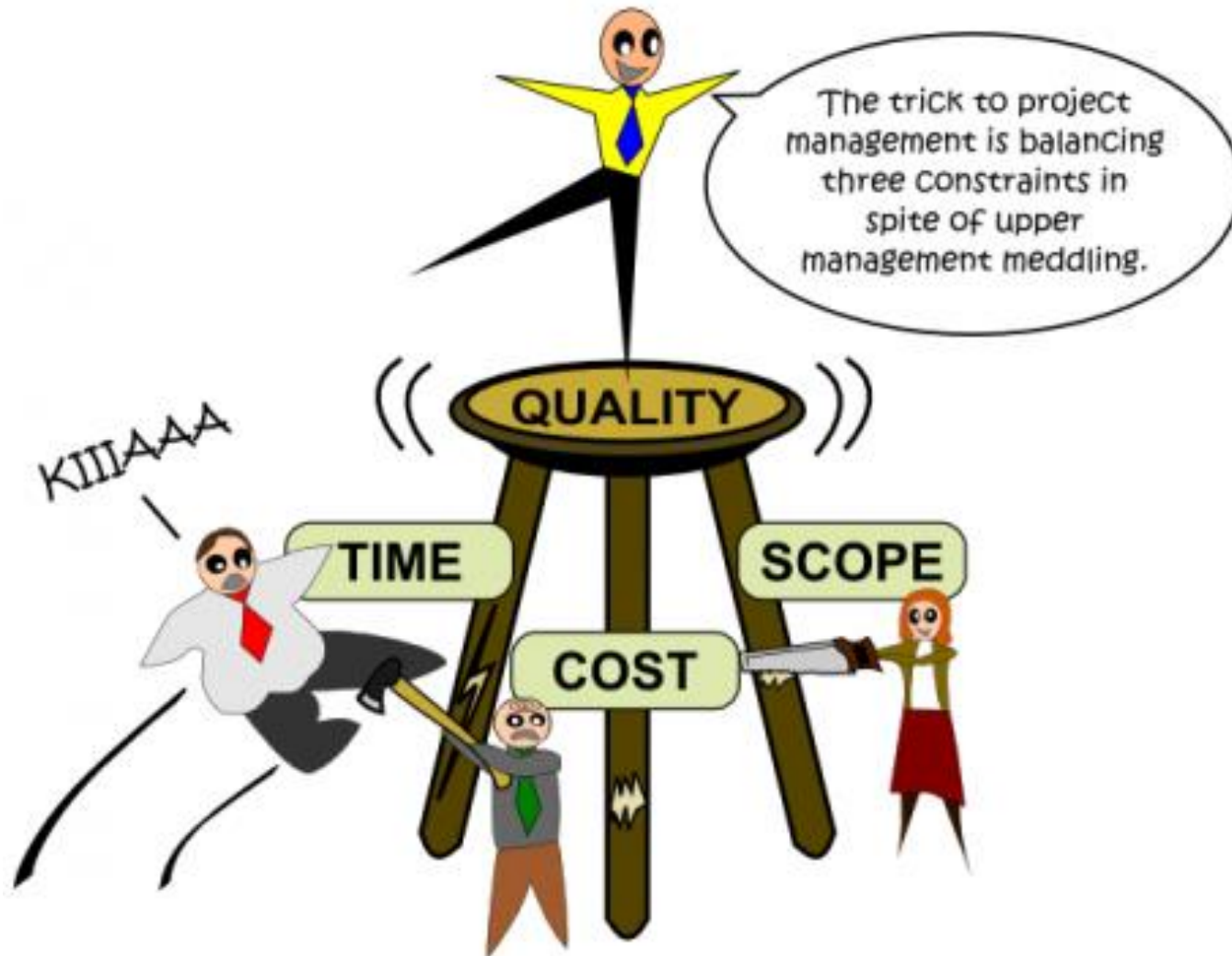




BIMU4098 – BİLİŞİM PROJE GELİŞTİRME

Dr. Öğr. Ü. Emel ARSLAN

Proje Yönetimi



Proje Türleri

Bireysel projeler:

- Rapor hazırlamak; düğün hazırlığı yapmak; yemek organize etmek;

Sanayi projeleri:

- Fabrika yapmak; sanayi bölgesi için gaz tedariki sağlamak; otomobil yolu inşa etmek; yeni araba tasarlamak;

Bilimsel-İş-Ticari projeler:

- Yeni bir yöntem geliştirmek; bilgisayar sistemi geliştirmek; yeni ürün geliştirmek; yıllık rapor hazırlamak; yeni bir ofis kurmak

Proje Türleri

- Her proje benzersizdir ve bağımsız olarak tanımlanmalıdır.
- Her projenin kendine has şekilde planlanması ve kontrol edilmesi gerekir.
- Farklı proje tiplerinin farklı aşamaları vardır.
- Büyük projeler, birden fazla aşamalara ayrıştırılabilir.





How the SAM expert explained it



How the project leader understood it



How the analyst designed it



How the processes were implemented



How the software publishers contributed



How the SAM tool provider described it



How the project was documented



What operations installed



How the customer was billed



How the project was communicated internally

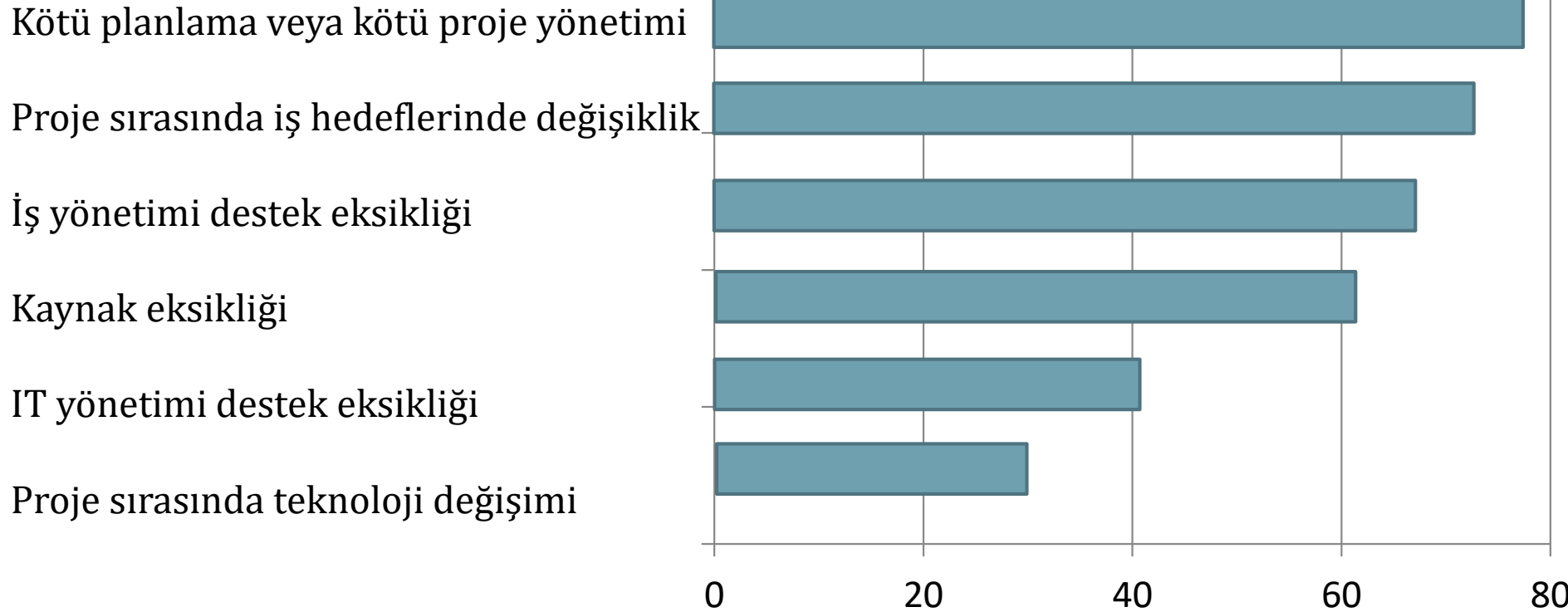


How the senior management team supported it



What the customer really needed

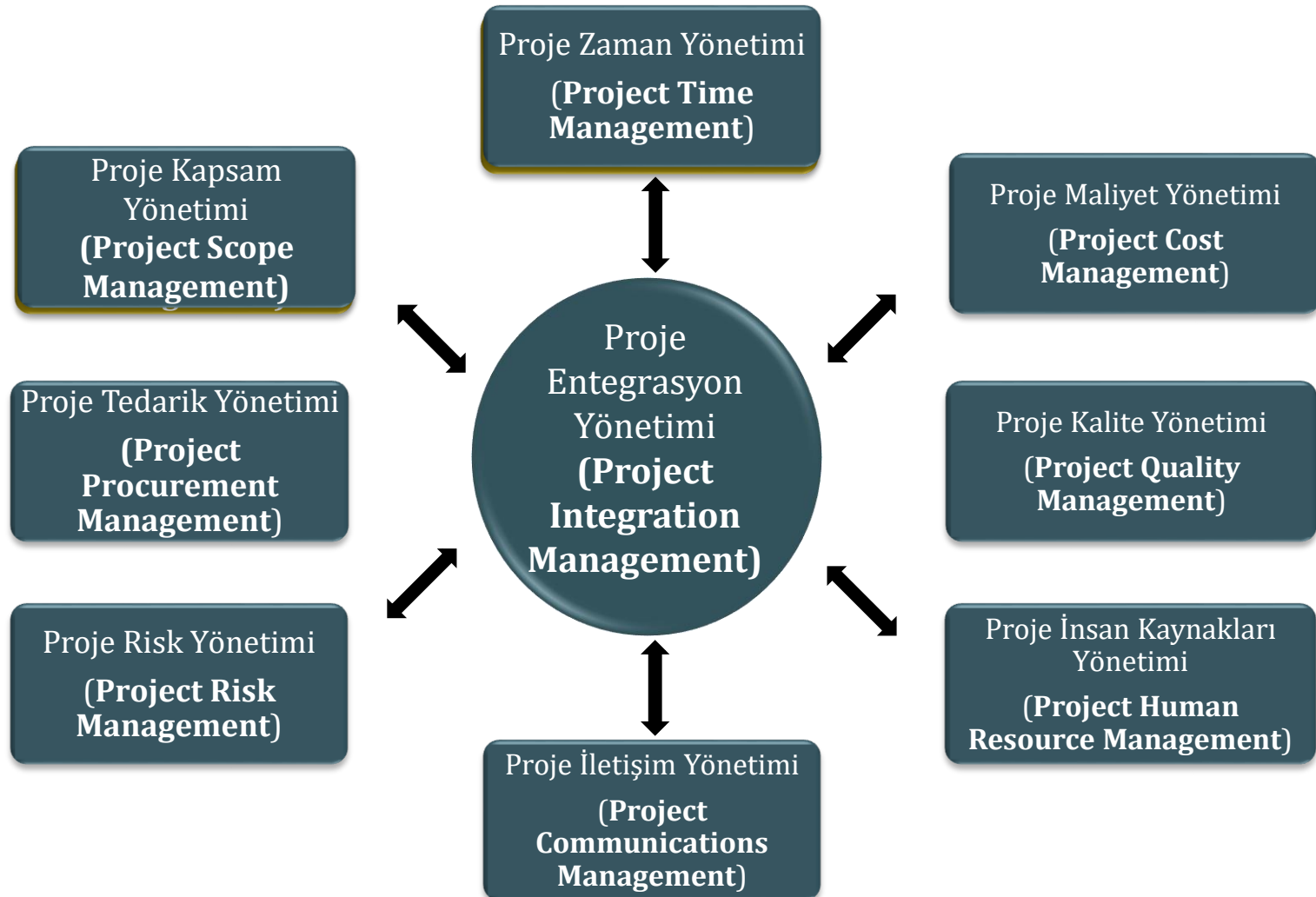
IT projelerinin programın gerisinde kalmasının veya başarısız olmasının en yaygın nedenleri



CLEARVISION



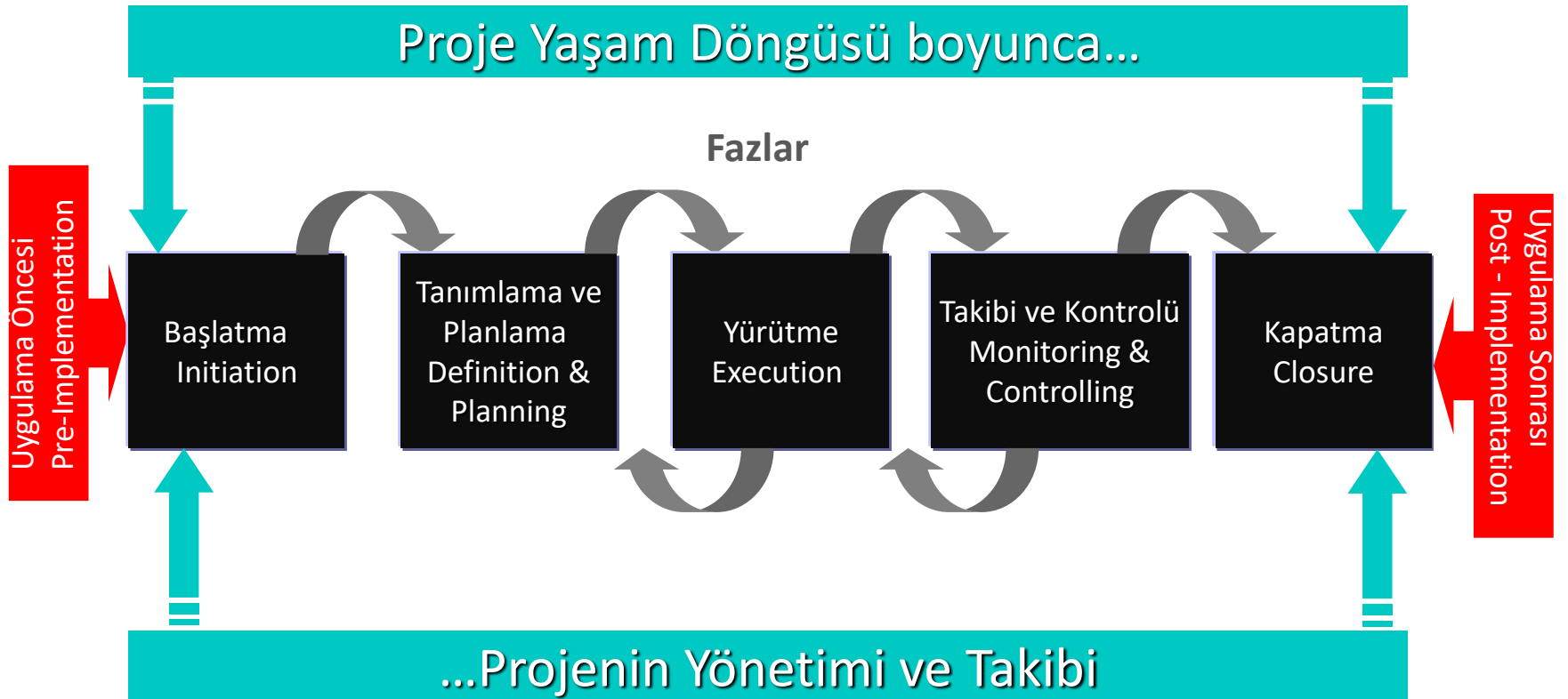
Proje Yönetimi Süreçleri



Proje Yaşam Döngüsü

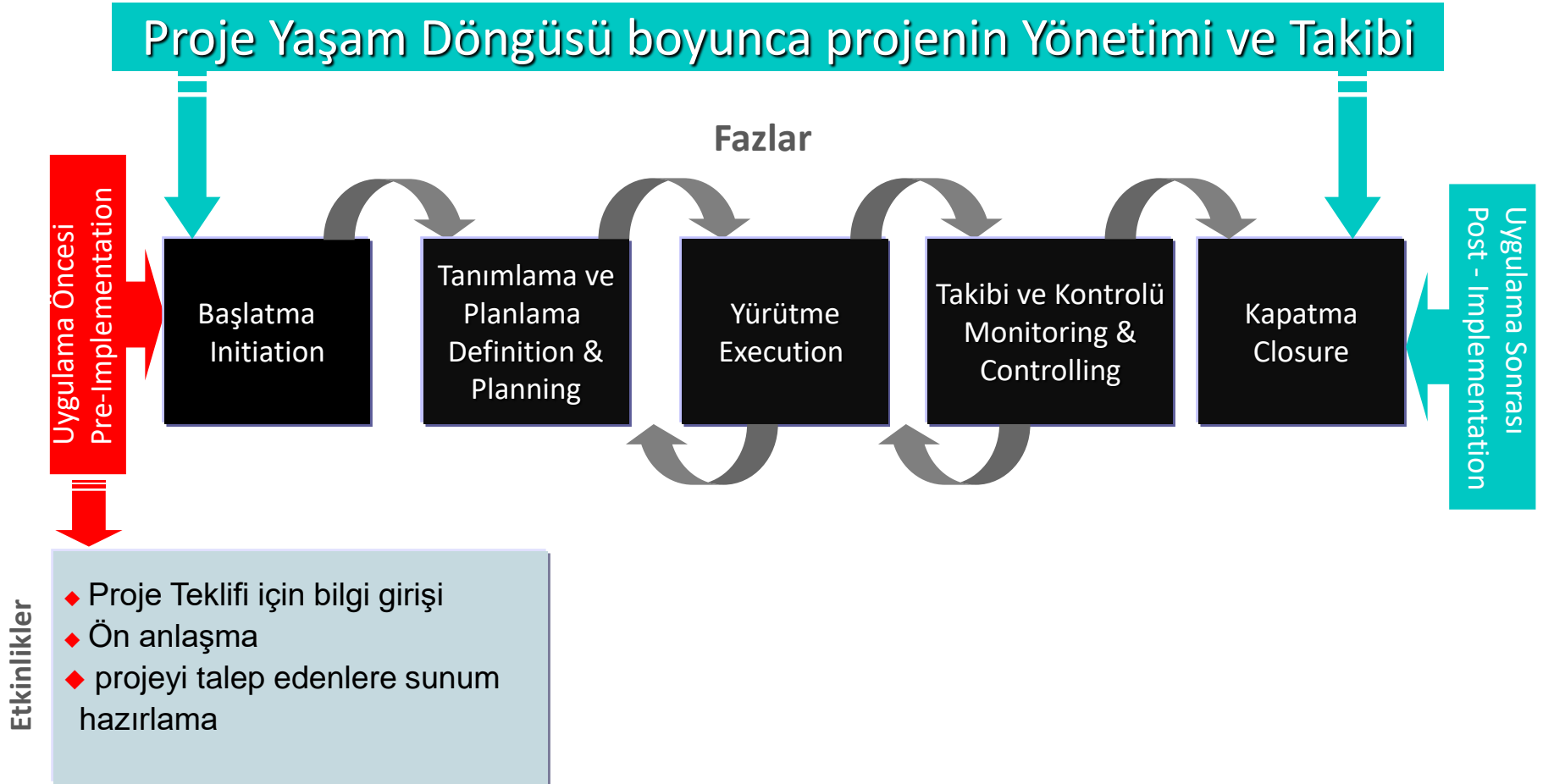
Project Life Cycle

Projenin başlatılması ve kapsam belirleme için tutarlı bir yaklaşım sergileyerek tanımlama, planlama, yürütme ve test aşamalarına geçilir. Uygulama süreci net hedefler ile uyumlu ve kontrollü bir ortamda yönetilir.



Proje Yaşam Döngüsü

Project Life Cycle

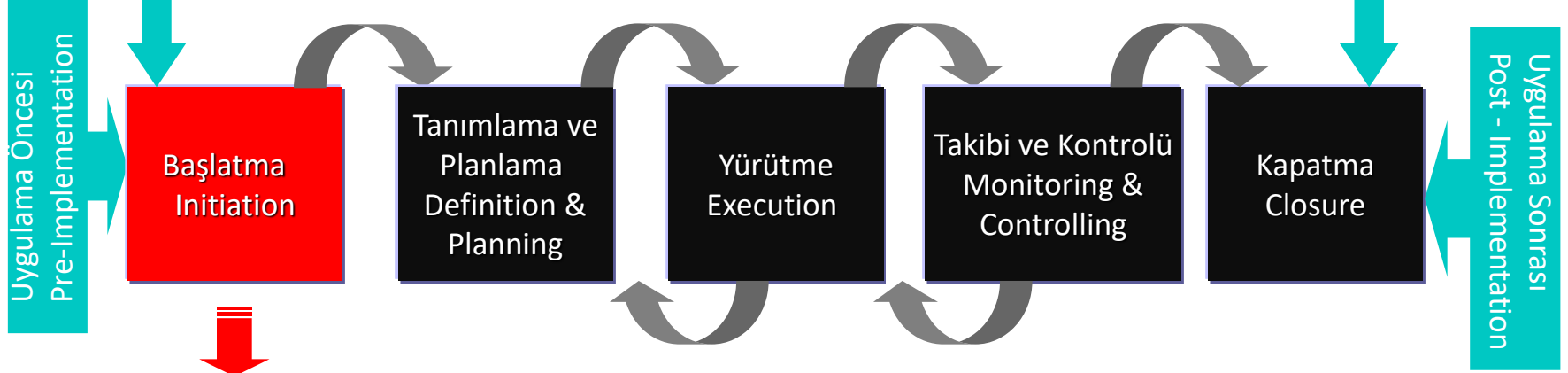


Proje Yaşam Döngüsü

Project Life Cycle

Proje Yaşam Döngüsü boyunca projenin Yönetimi ve Takibi

Fazlar



➤ Proje Entegrasyon Yönetimi
(Project Integration Management)

➤ Proje İletişim Yönetimi
(Project Communications Management)

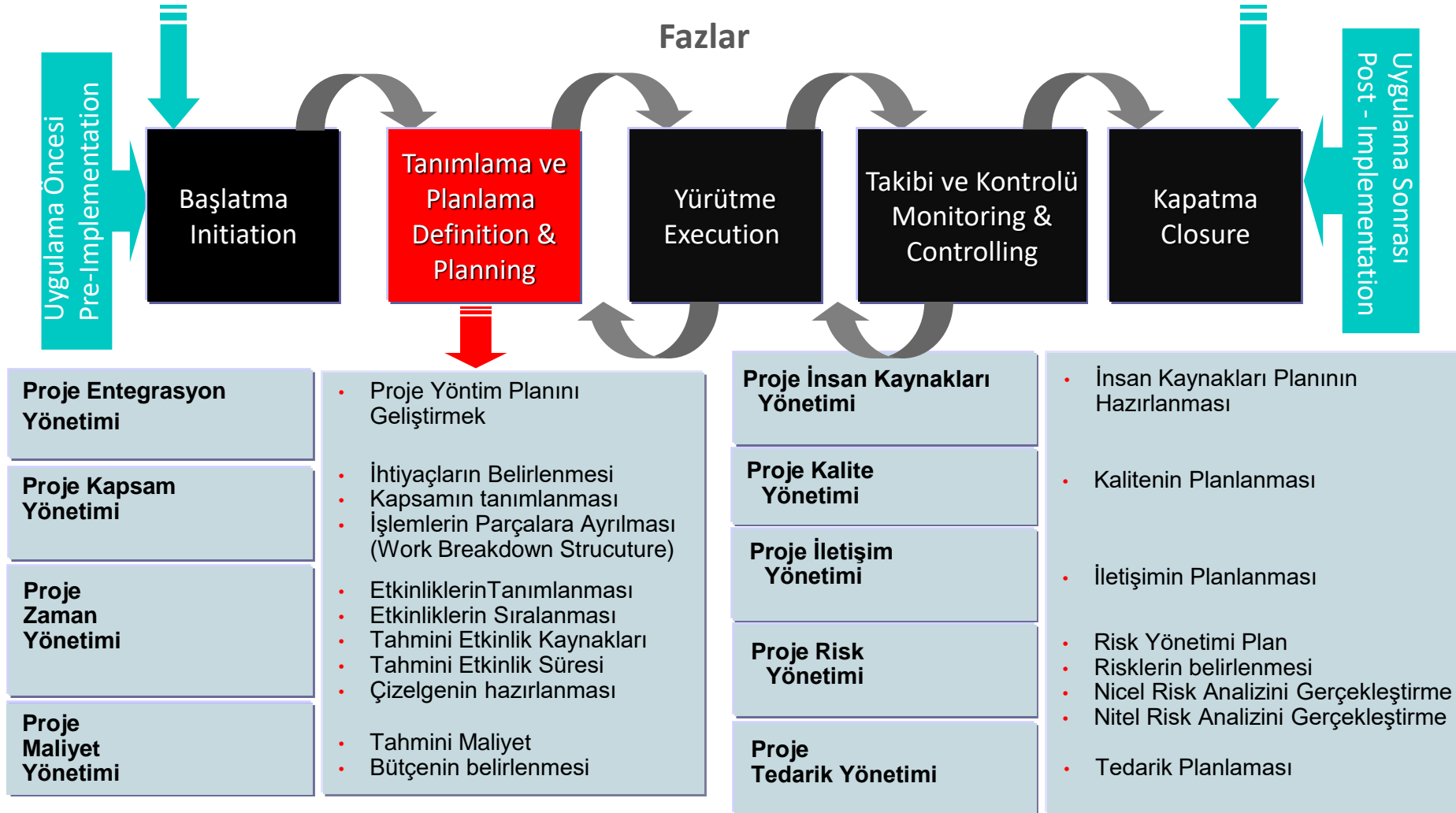
Proje Sözleşmesini geliştirmek

Paydaşların tespiti

Proje Yaşam Döngüsü

Project Life Cycle

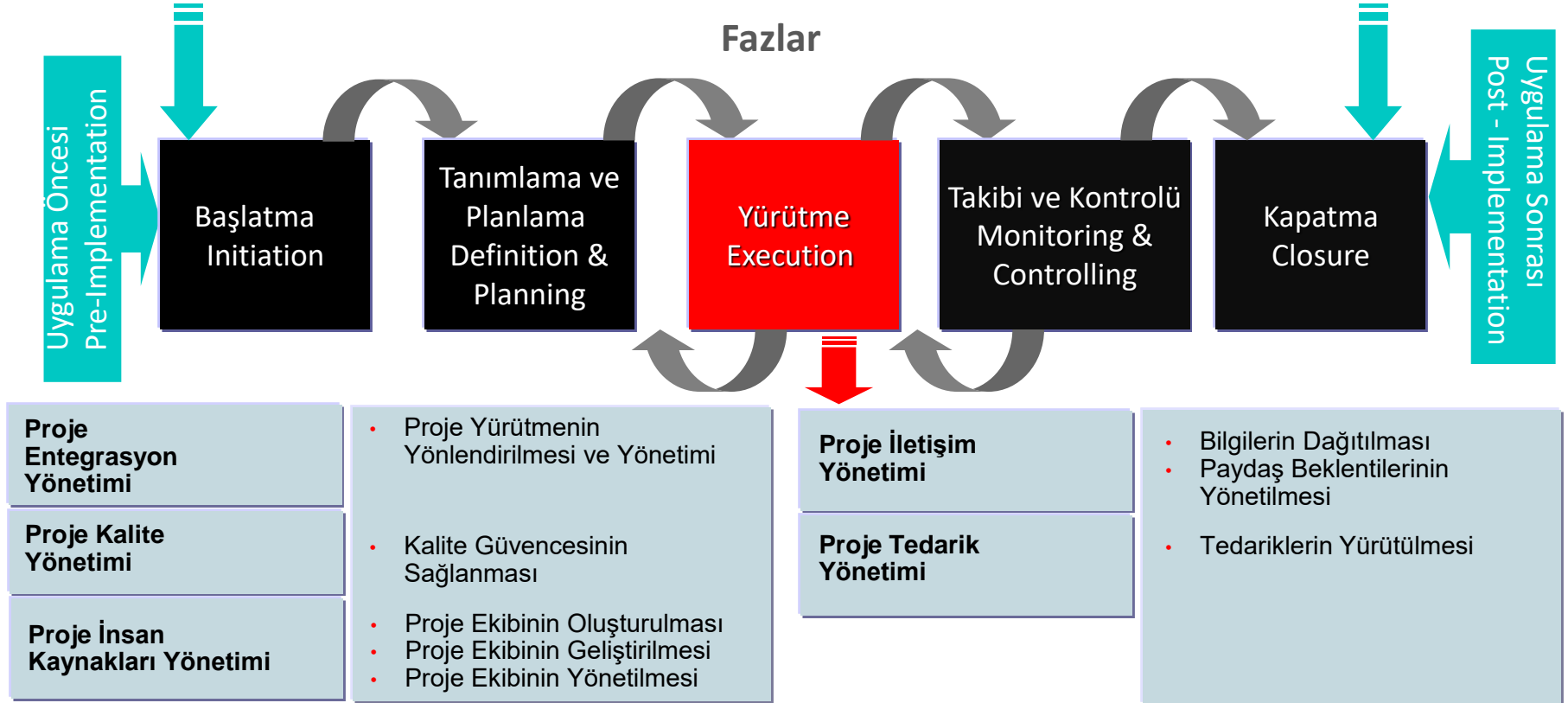
Proje Yaşam Döngüsü boyunca projenin Yönetimi ve Takibi



Proje Yaşam Döngüsü

Project Life Cycle

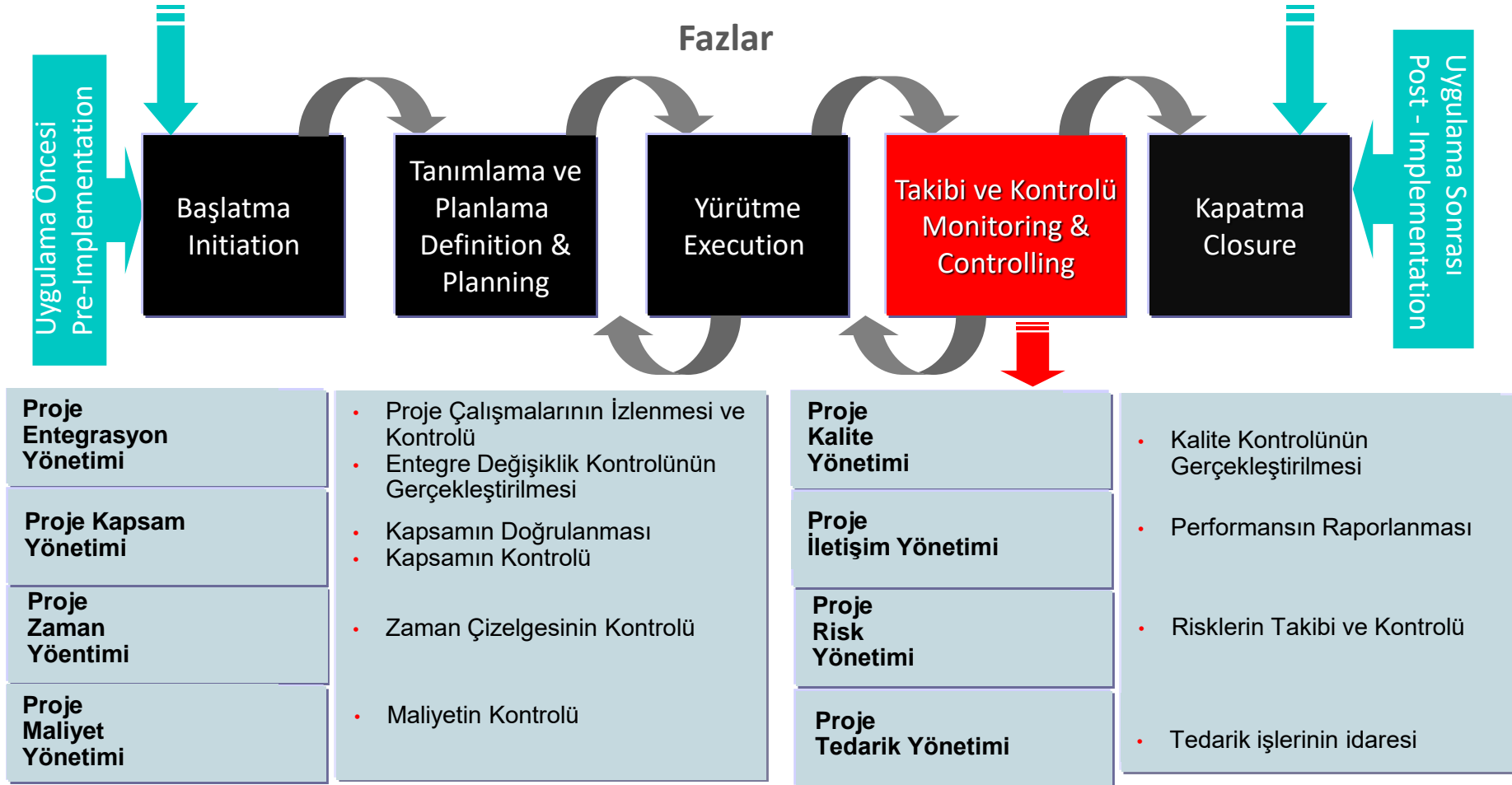
Proje Yaşam Döngüsü boyunca projenin Yönetimi ve Takibi



Proje Yaşam Döngüsü

Project Life Cycle

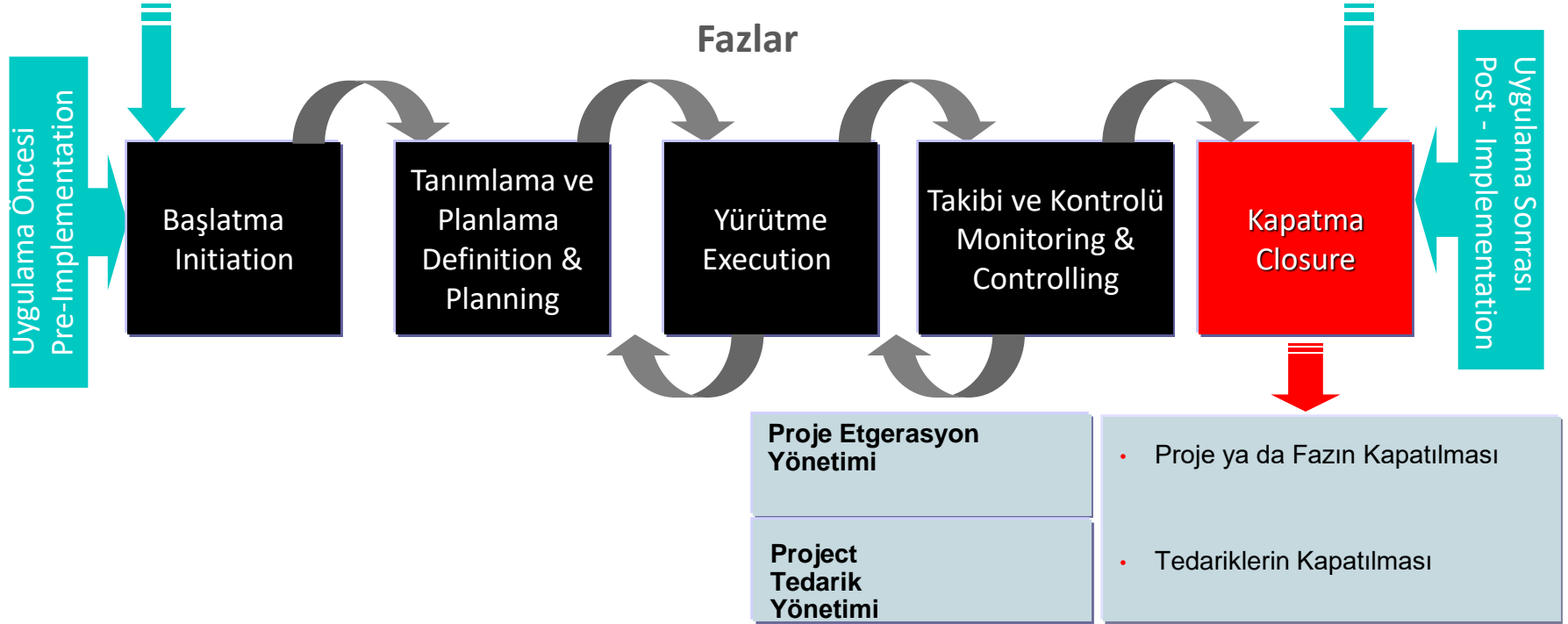
Proje Yaşam Döngüsü boyunca projenin Yönetimi ve Takibi



Proje Yaşam Döngüsü

Project Life Cycle

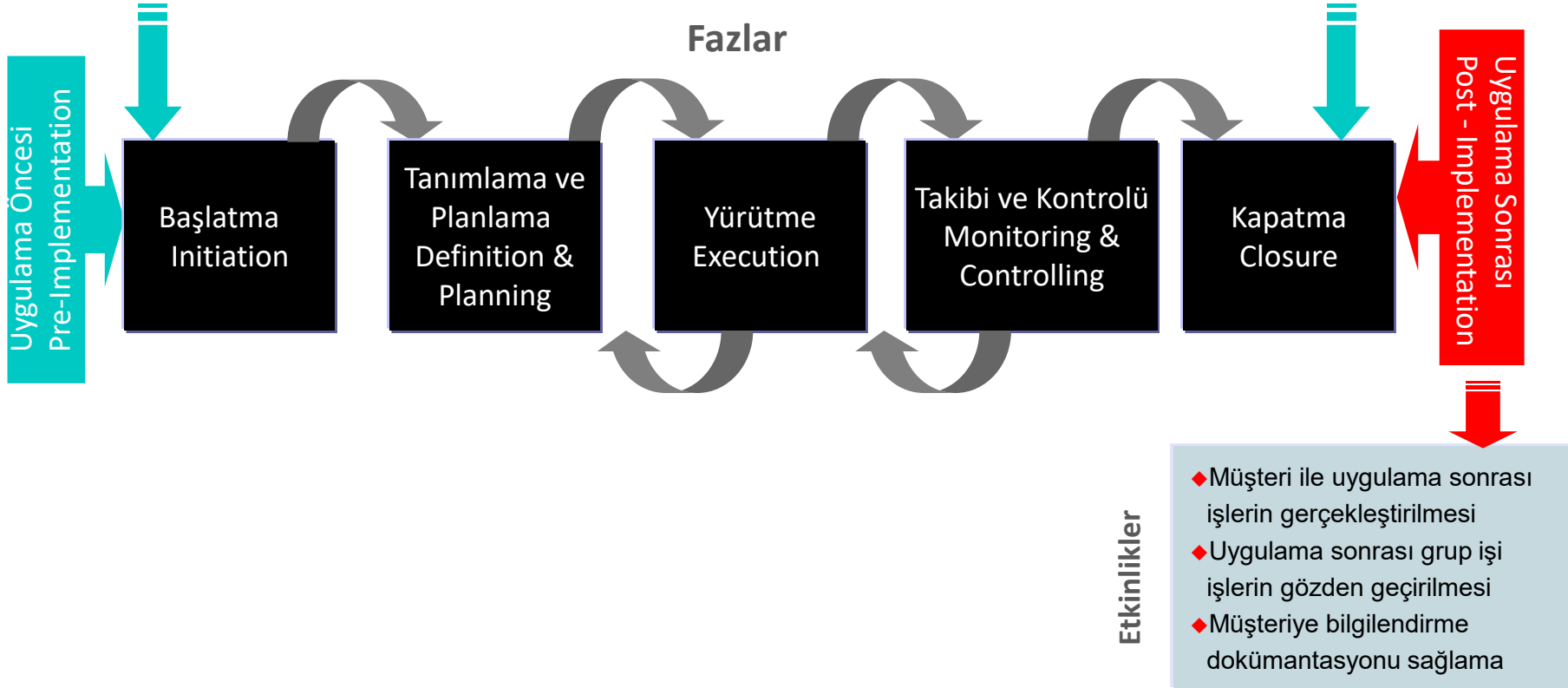
Proje Yaşam Döngüsü boyunca projenin Yönetimi ve Takibi



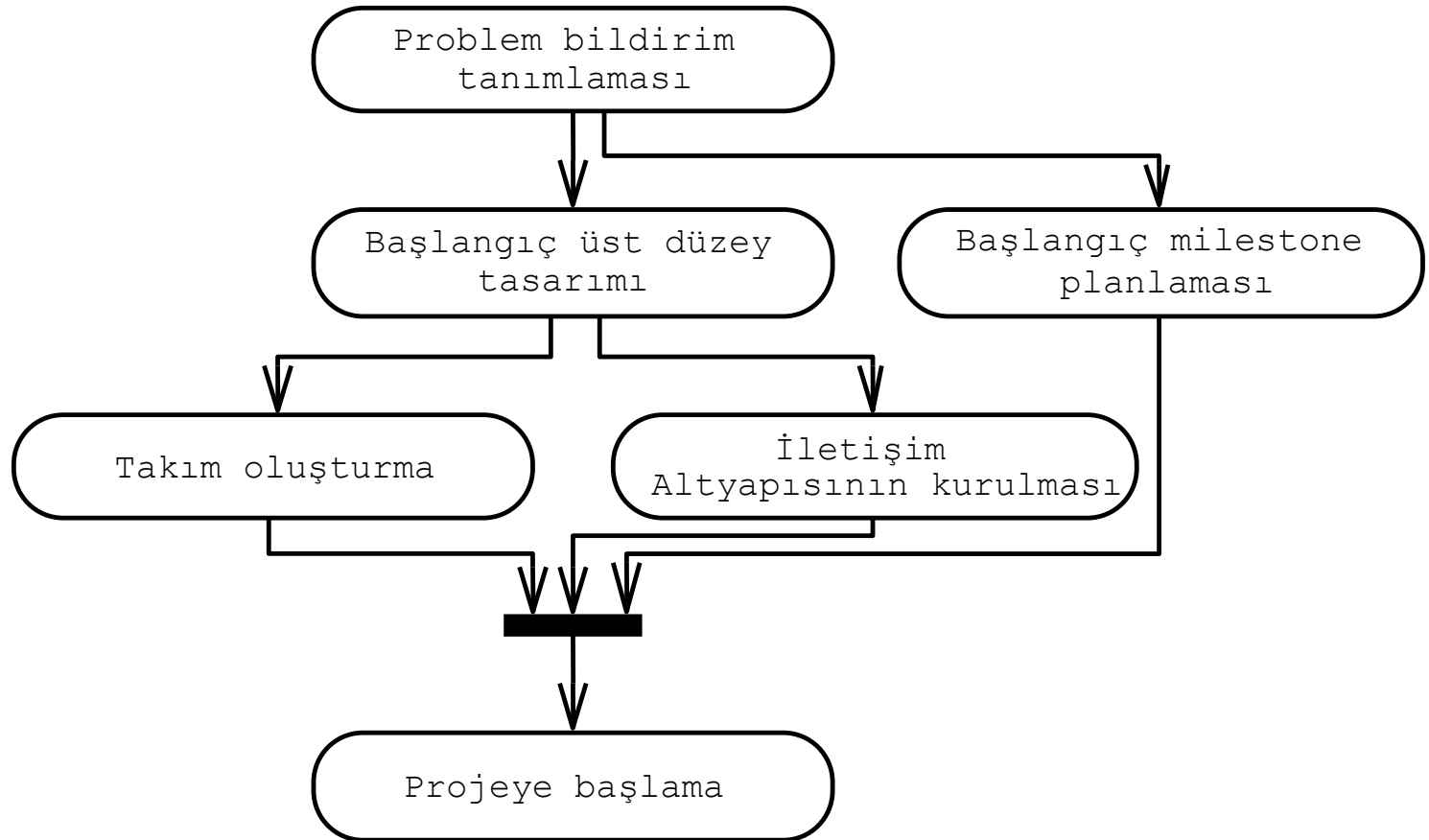
Proje Yaşam Döngüsü

Project Life Cycle

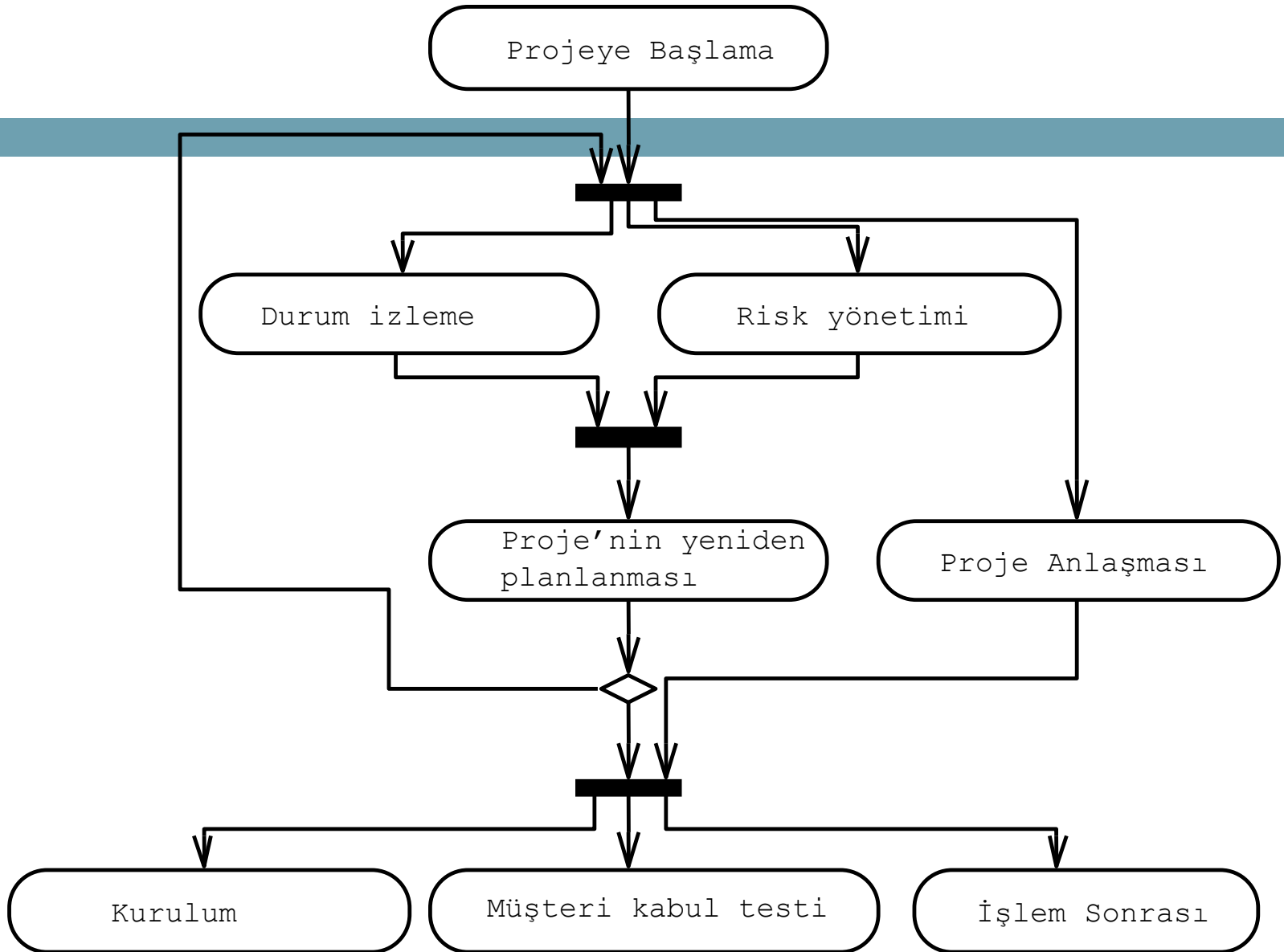
Proje Yaşam Döngüsü boyunca projenin Yönetimi ve Takibi



Proje Yönetimi Faaliyetleri



Devamı sonraki slaytta →



Projede görev alanlar

- Planlamacı
- Analist
- Tasarımcı
- Programcı
- Tester
- Sürdürme
- Eğitimci
- Belge Editörü
- Ağ Yöneticisi
- Yapılandırma Yöneticisi
- Group lideri
- İrtibat
- Minute Taker
- Proje Yöneticisi



Proje yöneticisinin görevleri - 1

- Kapsam tanımlama ve planlama
- Etkinliklerin planlaması ve sıralaması
- Kaynakların Planlaması
- Zaman Çizelgesi Hazırlama
- Süre Tahmini
- Maliyet Tahmini
- Bütçenin hazırlanması
- Dokümantasyon
- Grafikler ve Zaman Çizelgesinin Hazırlanması
- Risk Analizi



Proje yöneticisinin görevleri - 2

- Risk ve sorunları yönetme
- İşleyişi izleme ve raporlama
- Takım Liderliği
- Stratejik etki
- İş Ortaklığı
- Bayiler ile çalışma
- Ölçeklenebilirlik , birlikte çalışabilirlik ve Taşınabilirlik Analizi
- Kalite Kontrolü
- Gerçekleşme Faydaları (Benefits Realisation)

Bir Proje Yöneticisinin sahip olması gereken en önemli özellikler;

- Müzakere Yeteneği
- Etkileme Yeteneği
- İletişim Yeteneği
- Problem çözme Yeteneği



Yazılım Proje Yönetimi

Yazılım proje yönetimi diğer proje yönetimlerinden şu sebeplerle ayrılır:

- Ürünün fiziksel varlığı yoktur
 - ▣ Proje yöneticisi ilerlemeyi görebilmek için diğerlerinin ürettiği belgelendirmeye ihtiyaç duyar
- Yazılım geliştirme süreci standart değildir.
 - ▣ Yazılım süreçleri kurumdan kurumdan farklılık gösterir
- Çoğu yazılım projesi bir defalık projedir
 - ▣ Projelerin kendilerine has özellikleri vardır
 - ▣ Hızlı teknolojik değişiklikler proje yöneticisinin deneyimini geçersiz kılar

Yazılım Geliştirme Süreci

Yazılım geliştirme temel hedef ve kapsamın tespitinden başlayarak, planlama, analiz, mimari tasarım, gerçekleştirme, test, kurulum ve bakım aşamalarından oluşan bir süreçtir.

- Projenin genel hedef ve süresi özet bir plan çerçevesinde belirlenir.
- Kullanıcı ihtiyaçları analiz edilir.
- Analiz edilen ihtiyaçların yazılımla nasıl karşılanacağı modellenir.
- Bu model prototip olarak son kullanıcıya gösterilip, fikri alınarak olgunlaştırılabilir.
- Gerçekleştirme aşamasında nesne, ekran ve rapor gibi yapıların kodları geliştirilir.
- Yapılanlar test edilir, hatalar düzeltilir ve test sonucu başarılı olan proje devreye alınır.
- Devreye alındıktan sonra gelen istekler doğrultusunda proje bakım çalışmaları devam eder.

Yazılım Geliştirme Süreci

Proje Yöneticisi, yazılım ekibinin de katkılarıyla yazılım geliştirme sürecini planlar, izler ve yönetir. Bunun için öncelikle yazılım geliştirme sürecine hakim olmak gerekir. Sürecin her aşamasında yapılan çalışmaları ve aşamalar arasındaki geçiş yöntemlerini bilmek, projeyi sağlıklı planlama ve yürütmenin ön şartıdır.

Yazılım Geliştirme Süreci

Yazılım geliştirme süreci proje, ekip ve kurum yapısından kaynaklanan bir çok farklı ihtiyaca cevap vermelidir. Bu durum doğrusal, yinelemeli ve çevik geliştirme gibi farklı yazılım süreç yönetim modelleri ortaya çıkmasına neden olmuştur.

Tercih edilen model, proje yönetiminin nasıl yapılacağını da büyük ölçüde etkiler. Proje yöneticisi geliştirecek yazılıma, ekibinin tecrübesine, kuruma ve müşteri beklentilerine en uygun yöntemi seçmelidir.

Yazılım Projesi

Proje yöneticisi yazılım, yazılım geliştirme süreci, proje kavramlarını ve bu kavramların birbirleriyle ilişkilerini bilmelidir.

Yazılım geliştirme : Kişi ve kurum ihtiyaçlarının bilgisayar programı haline getirilmesi sürecidir. Yazılım projesi bu dönüşümü gerçekleştirmek için yapılması gerekenleri tanımlar.

Proje : Yeni bir ürün, hizmet veya sonuç elde etme amacına yönelik belirli sürede gerçekleştirilen benzersiz çabadır.

Yazılım Projesi : Kişi veya kurumun belirlenmiş bir ihtiyacını analiz ederek elektronik ortamda karşılamak için geliştirilen, belli bir gelişme süresi olan ve daha önce yapılmamış bir yazılım geliştirme çabasıdır.

Yazılım Projesi

İhtiyaç tespitinden devreye almaya uzanan tüm süreç bütünlüklük yönetilmelidir.

Yazılım Proje Yönetimi : Müşteri isteklerinin verilen kaynaklarla, belirlenen süre ve maliyet kısıtları içerisinde tamamlanması için gerekli planlama, yönetim, koordinasyon, bilgilendirme, gözden geçirme ve çıkan sorunları çözmek için gerekli düzeltmelerle yapma faaliyetlerini kapsar.

Tüm projelerin genel hedefi : Belirtilen kapsamı, sunulan kaynaklarla, istenen sürede ve kalitede tamamlamaktır.

Yazılım Projesi

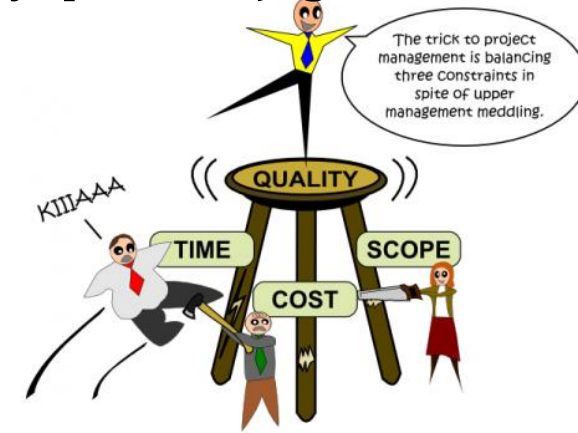
Proje genellikle **zaman**, **maliyet** ve **kapsam** değişkenleriyle temsil edilir.

Maliyet: Projenin gerçekleştirilmesi için gerekli tüm kaynakların maliyetidir.

Kapsam:

Ürün Kapsamı: İstenen özellik, fonksiyon ve kalite şartlarından oluşur.

Proje Kapsamı: İstenen ürünü teslim etmek için yapılacak iş, görev ve aşamalar şeklinde tanımlanır.



Yazılım Projesi

Proje Geliřtirmede denge kavramı

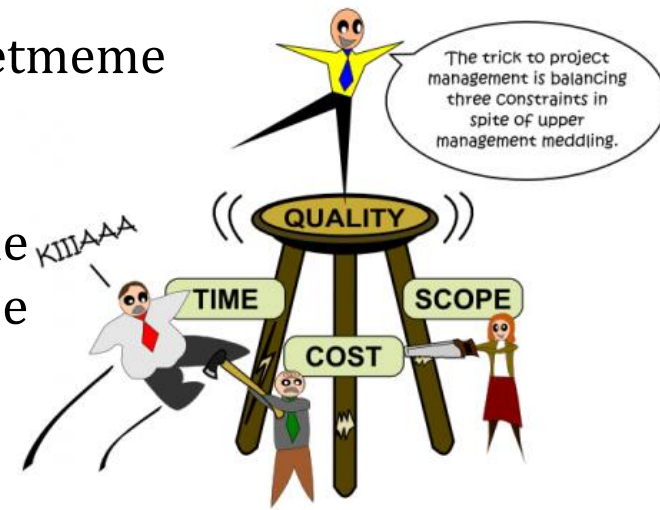
Saęlıklı bir proje geliřtirilebilmesi için hedef, kaynak ve kısıtlar arasında tam bir denge olmalıdır.

Denge karřılıklı ödünlerle deęil mümkün olan en üst seviyede saęlanmalıdır. Örneęin performanstan ödün vermeyen esnek bir tasarım tercih edilmelidir.

Saęlıklı kurulamayan denge projede başarısızlıęa neden olur.

- Süreyi aşırı kısa tutmak,
- Maliyeti düşürmek için yeterli ekip istihdam etmeme
- Aşırı beklentiler

Bu sebeplerin birden fazlasının aynı projede bulunması halinde başarısızlık neredeyse kaçınılmazdır.



Yazılım Geliştirme Temel İlkeleri

Yazılımın kalitesi belli temel ilkelere dayanır. Geliştirme için hangi yöntem kullanılırsa kullanılsın bu ilkelere dikkat edilmelidir. Bu ilkeler hem ortaya çıkacak ürünle hem de ürünün geliştirildiği süreçle ilgilidir.

Proje yöneticileri geliştirilen projelerde bu hususlara dikkat edilmesini sağlamakla vazifelidir.

Yazılım Geliştirme Temel İlkeleri

- **Basitlik** yazılımda gereksiz karmaşıklıktan kaçınmak olarak tanımlanabilir. Yazılımın basit ve kolay anlaşılır olması planlama açısından da önemlidir.
 - ▣ Gereksiz zorluklar için para ve zaman gibi kaynaklar israf edilmez
 - ▣ Planlama da kolaylaşır çünkü plan yapılacakların bir yansımasıdır.
 - ▣ Kişiler tasarımı daha kolay anlayabilir
 - ▣ Kişilere atanan görevler basitleşir ve daha kolay gerçekleştirilebilir.

Yazılım Geliştirme Temel İlkeleri

- **Tekrar Kullanılabilirlik:** Yazılımı oluşturan kod, nesne ve bileşenlerinin tekrar kullanılabilir şekilde tasarlanması, orta-uzun vadede zaman ve maliyet kazancı sağlar.

Nesneye yönelik tasarım, web servis, servis ve bileşen tabanlı mimari gibi mimariler tekrar kullanılabilirliği artıracak şekilde gelişmektedir. Denge ihtiyacı tekrar kullanım için de söz konusudur. Aşırıya kaçmamak gerekir.

Sadece kod ve nesneler değil, yöntem ve planlar da tekrar kullanılabilir. Ancak her yazılım projesinin kendine özgü şartları vardır. Dolayısıyla bir plan hiç değiştirilmeden başka bir projede tekrar kullanılamaz.

Yazılım Geliştirme Temel İlkeleri

- **Süreklilik:** Yazılım ürünü sistemin çökmesine yol açabilecek hatalardan arındırılmış olmalıdır.

Ayrıca artan kullanım ve kullanıcı sayısı karşısında performanslı çalışabilmelidir.

Sürekliliğinde problem olan bir yazılım, yazılım ekibi için önemli bir bakım ve destek yüküdür.

Yazılım Geliştirme Temel İlkeleri

- **İzlenebilme:** Geliştirme süreci ve bu süreçte olgunlaşan yazılım teknik ekip, yönetici ve müşteri tarafından performans ölçümleriyle izlenebilecek parçalara bölünmelidir.

Görev tanımı basit ve anlaşılır, görev çıktı ve bitiş kriterleri net olmalıdır.

Müşterinin süreci izleyebilmesi çok önemlidir. Çünkü yazılım geliştirme sadece bilim adamlarının anlayabileceği özel bir alan değil müşteriye hitap eden bir mühendisliktir.

Yazılım ürününü izlenebilecek bir yapıda tasarlamak, proje planında kabul kriter ve kontrol noktaları oluşturulmasına katkı sağlar. Geline nokta ürünün ne kadarının geliştirildiğini bilmek, geleceğe yönelik süre ve maliyet tahminini de kolaylaştırır.

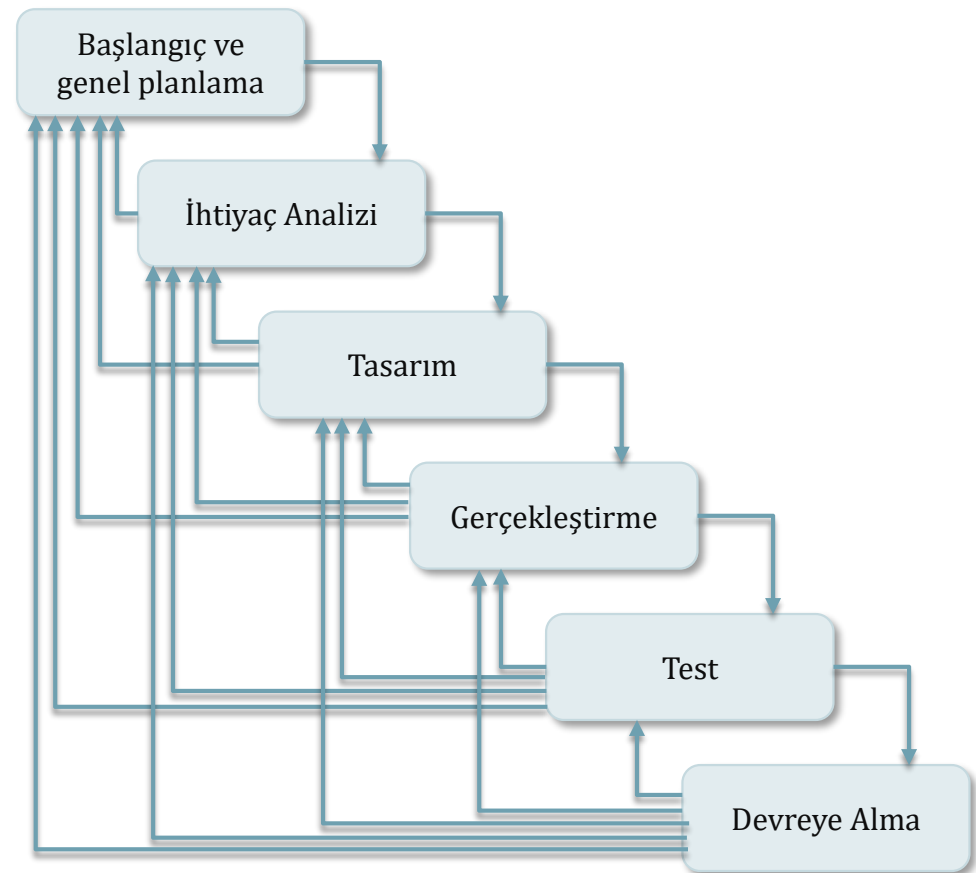
Yazılım Geliştirme Temel İlkeleri

- **Güvenlik:** Yazılımın önemi ve yazılıma emanet edilen bilgi miktarındaki artış güvenliğin önemini de arttırmaktadır. Güvenlik iki aşamalı ele alınabilir:
 - ▣ **Geliştirilen yazılımın çalışma anında güvenli olması:** Yazılım, müşteri verilerine yetkisiz erişimlere izin vermeyecek ve özellikle internet üzerinden gelecek saldırılarla çökmeyecek şekilde modellenmelidir.
 - ▣ **Geliştirme sürecinin güvenliği:** Kod ve üretilen diğer belgeler, sadece yetkili kişilerin erişebileceği güvenli bir ortamda saklanmalıdır.

Yazılım Geliştirme Süreci

□ Geliştirme aşamaları yazılım mühendisliğinde kısaca:

- Planlama
- İhtiyaç Analizi
- Tasarım
- Gerçekleştirme
- Test
- Devreye alma
- Bakım



Yazılım Geliştirme Döngüsü

Yazılım Geliştirme Süreci

Belgeleme: Yazılım geliştirmenin her aşamasında bir çok bilgi ortaya çıka ve çeşitli kararlar verilir. Belgeleme soyut tartışmaları somut kararlar haline dönüştürür.

- İşin bir parçasıdır ve ayrı bir süreç adımıyla ziyade sürecin her aşamasıyla bütünleşik yapılması gereken bir işlemdir.
- Bir plan dahilinde yapılmalıdır. Bu planda: belge geliştirmekten sorumlu kişiler, belgelerin hedeflediği izleyici grubu, standart şablonlar, belgelerin güncellenme sıklığı ve sürüm kontrol yöntemi gibi bilgiler olmalıdır.
- Belgeleme bütün önemine rağmen sonuçları doğrudan görülmediğinden; zaman alıcı, külfetli ve sıkıcı bir iş olarak görülebilmektedir. Belgelemenin sağlıklı yürütülebilmesi büyük ölçüde proje yöneticisi ve diğer üst düzey yöneticilerin destek ve sorumluluğundadır.

1. Yazılım Projesi Başlangıç Noktası

Yazılım projesi yazılımla giderilecek bir ihtiyacın ortaya çıkmasıyla başlar. Ardından kapsam belirlenir ve planlama yapılır. Yeni bir projenin başladığını fark etmek, gerekli kaynakları ayırmak açısından çok önemlidir.

Planlı İstek

Kurum üst yönetimi ihtiyacın farkındadır ve bunun giderilmesi için harekete geçer. Projenin hedefleri ilk anda belirlidir, ismi konulmuş ve kaynaklar ayrılmıştır. Bu tür projelerin planlı bir şekilde başlama ihtimali daha fazladır.

Belirsiz başlangıçlar

İlk anda basit ve kolay görülen bir isteğin yeni bir yazılım projesi başlatmasıdır. Müşterinin basit görünen bir isteği bir çok talep eklenerek büyür ve bir projeye dönüşür. Bu şekilde başlayan projelerin bir çoğunda ciddi planlama eksikliği ve hatta tamamıyla plansızlık görülür.

- Mevcut Projede yapılan bir istek
- Proje kapsamının değişmesi

2. Planlama

Genel olarak proje hedeflerine ulaşmak için, belirlenmiş süre ve mali kısıtlarını dikkate alarak, izlenecek yöntem, yapılacaklar, kullanılacak kaynaklar ve süre takvimlerinin belirlenmesidir. Yazılım, temel bir ihtiyaç etrafında şekillenir. Temel ihtiyacın ortaya çıkmasını takiben kapsam belirleme ve planlama süreci başlar.

İstekler ve ürün özellikleri ayrıntılara bölünerek yapılacak görevler ortaya çıkar. İhtiyaç duyulan kaynakları belirlemek, tedarik etmek ve görevlendirmek bunu takip eder.

Ayrıca projenin hedef kitlesi olan kullanıcılar da belirlenmelidir. Müşteri veya üst yönetimin istediği süre içinde projeyi tamamlamak esastır.

3. İhtiyaç Analizi

Paydaş ihtiyaç ve beklentilerinin ayrıntılı şekilde tanımlanmasıdır. İhtiyaç veya beklenti; müşteri tarafından sistemin gerçekleştirmesi istenen temel görev veya fonksiyonlardır. Bu ihtiyaç aşağıdaki özellikleri taşımalıdır:

- **Gerçekleştirebilme:** Mevcut insan ve mali vb. kaynaklarla yapılabilirliği
- **Doğrulanabilme:** Tarafsız ve ölçülebilir kriterlerde yapıldığını anlayabilme
- **Mananın tam anlaşılması:** Belirsiz ve birden fazla manaya gelebilecek ifadelerden kaçınılması
- **Çözümleri değil istekleri ifade etme:** Bu özellik ihtiyaç analizinin hedefini de açıkladığı için çok önemlidir.
- **Tutarlılık:** Diğer ihtiyaçlarla tezat teşkil etmemelidir.
- **Seviyesi veya yeri doğru olmalı:** Bir ihtiyaç sistem hiyerarşisinde doğru yerde tanımlanmalıdır. Basit bir ihtiyacı çok öne çıkartmak veya karmaşık bir ihtiyacın en alt seviyede tanımlanması ürün yapısına uygun değildir.

4. Prototip Geliştirme

Prototip veya sunum sürümü temel fonksiyon ve ara yüzlerin, sistemi açıklama amacıyla kısmen geliştirilmesi ve proje paydaşlarına gösterilmedir.

Prototip için geliştirilen kodun ana sistemin geliştirilmesinde kullanılmaması önerilir.

Prototip geliştirirken, görsel sunuma yönelik çizim ve metin yazılımı kullanmak yerine hızlı uygulama geliştirme aracı da kullanılabilir.

Kullanıcı ve müşteri açısından yararları: Birçok kullanıcı, istemiş olduğu sistemin bilgisayarda nasıl geliştirileceğini ancak görerek anlayabilir ve isteğinin karşılandığına prototip sayesinde karar verir.

Programlama Ekibi Açısından Yararları: Kullanıcı ihtiyacının anlaşılması, iletişim ve ortak mutabakat sağlama görünen ara yüzler sayesinde kolaylaşır ve hızlanır. Projenin tüm kodlaması tamamlanmadığından değişiklik ve geri dönüşler bitmiş halden çok daha kolay olacaktır.

Projenin amacı doğru anlatılmalıdır.

5. Tasarım

Tasarım, yapılan analizin yazılım geliştirme araçlarının imkan, kural ve sınırları dahilinde geliştirilecek yazılımın üst seviye modeline dönüştürülmesi işlemidir. Tasarım için ilişkisel veri modeli, nesne modeli, UML şeması, iş akış şeması gibi birçok farklı yöntem kullanılır.

İhtiyaç analizinde yön ihtiyacı anlamaya, tasarımda ise çözüme dönüktür. Analiz esnasında yazılım geliştirme araçlarının kapasite, kural ve bir anlamda sınırları düşünülmemelidir.

Proje yöneticisi, analiz bitmeden tasarıma geçilmesini veya iki aşamanın kontrolsüzce birleştirilmesini önlemelidir.

Analiz ve tasarım için ayrı kişiler görevlendirmek, muhtemel sorunları oluşmadan engeller.

6. Gerçekleştirme

Gerçekleştirme aşamasında, müşteri talepleri için tasarlanan model, yazılım geliştirme araçları vasıtasıyla yazılım ürününe dönüştürülür. Bu aşamada;

- Fiziksel veri yapılarının veri tabanında oluşturulması
- Kod
- Bileşen
- Kod kütüphanesi
- Kullanıcı ara yüzü
- Raporların yazımı gibi işlemler yapılır.

Gerçekleştirme aşamasında da belli ölçüde tasarım yapılır. Bu yüzden IEEE/EIA 10207 gibi bazı yaklaşımlarda tasarım özet ve ayrıntılı tasarım şekline iki aşamaya bölünmüş olarak gösterilmektedir. Programcının kendi geliştirdiği ürünle ilgili birim test yapması da gerçekleştirme aşaması içinde görülebilir.

7. Test

Yapılanların kontrol edilmesidir. Kontrol iki aşamalıdır. Hedef ve yapılanlar test edilmelidir.

İstenilenler doğru anlaşılmış mı?

Yapılanlar doğru çalışıyor mu?

Yapılanın isteği karşılayıp karşılamadığının **onaylanması (validation)** ve düşünülenin yapılıp yapılmadığının **doğrulanması (verification)** gerekir.

Doğrulama, yazılım ekibinin kendi içinde yaptığı testlerle gerçekleştirilir. Bunlar arasında metot veya nesne seviyesinde yapılan birim testi, bileşenler arası entegrasyon testi, bütünleşik sistem testi, performans ve yük testi yapılabilir. Test neticesine göre yazılımın tasarım ve kodunda değişiklikler yapılabilir.

Onaylama, kullanıcı ile müşterinin dahil olduğu sunum ve testlerle belirlenir. Bu aşamada kişisel bakış açıları geçerlidir. Kişisel memnuniyeti sağlamak gerekir ki bu oldukça zordur. Bu test neticesinde ihtiyaç analizi değişebilir.

8. Gözden Geçirme

Yapılan çalışmanın, yapan kişi dışında ekibin diğer üyeleri veya ekip harici kişiler tarafından kontrolüne **gözden geçirme** denir. Testten farklı olarak yapılanın çalışmasından ziyade kalitesine odaklanılır. Gözden geçirme;

- Kod teftişi
- Genel gözden geçirme
- Müşteri veya üst yönetim gözen geçirmesi
- Denetim

gibi yöntemlerle yapılır. Gözden geçirme hatayı çözmekten çok oluşmasını önlemeye odaklanır.

9. Devreye alma

Test aşaması tamamlandıktan sonra kullanıcı eğitimi, pilot kurulum ve nihai proje kurulumu yapılarak proje devreye alınır.

Devreye aldıktan sonra proje destek çalışmaları başlar.

Artan kullanıcı ve veri sayısı performans sorunu oluşturabilir.

Devreye alım anında, tüm proje ekibi sistemi dikkatle izlemelidir.

İlerleyen zamanlarda da sistem sürekli izlenir ve destek anlaşmasına göre varsa yeni istekler gerçekleştirilir.