



# BIMU4098 – BİLİŞİM PROJE GELİŞTİRME

Dr. Öğr. Ü. Emel ARSLAN

# Bu bölümde öğrenilecek kavramlar

- Yazılım ekip üyeleri
- Birlikte çalışma kaideleri
- Ekip psikolojisi
- Moral ve motivasyon
- Ekip ruhunu zedeleyen davranışlar
- İş ahlakı



# Proje Ekibi

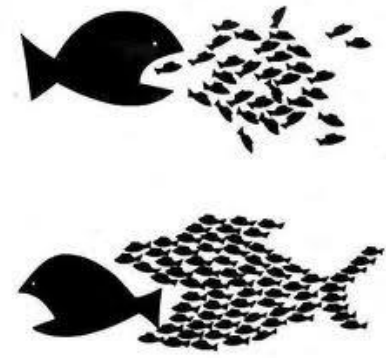
- Yazılım projesinin en önemli kaynağı **insandır** ve proje yönetimi büyük ölçüde **insan** yönetimidir.
- Yazılım geliştirme **kişilerin** fikir ve buluşlarına dayanır.
- Ekip, **insanlardan** oluşur.
- Önemli kararlar **insanlar** tarafından alınır; analiz ve tasarım gibi yazılım geliştirme aşamalarını **insanlar** yürütür.
- Yazılımın müşterisi yine **insandır**.



# Proje Ekibi

- Müşteri, bir yazılımcının 120 adam/ay emek gerektiren bir projeyi bitirmesi için on sene bekleyemez. Bu yüzden büyük yazılımlar, birçok farklı kişi ve görev tanımından oluşan ekipler tarafından gerçekleştirilir.
- Sadece aynı ortamda çalışmak ve aynı projede görev almak kişileri ekip yapmaz. **Ekip aynı hedefi paylaşan, birbirinden haberdar ve yardımlaşan kişilerin birlikte çalıştığı insan topluluğudur.** Ekip olmak ve ekip çalışmasıyla ilgili temel kaidelere uymak, proje gelistirmenin temel şartıdır.
- İnsanların kabiliyetleri süreç ve teknolojide karşılaşılan sorunların üstesinden gelmeyi sağlayabilir. Ancak çok iyi süreç ve teknolojiler, beceri ve motivasyonun eksikliğini gidermez [Fairiey-2009] .
- Bu yüzden proje ekibi **beceri, tecrübe, kabiliyet ve birlikte çalışmaya uyumu üst düzey** kişilerden oluşturulmalıdır. Ekibin devamlılığını sağlamak ve becerisini arttırmak için gerekli çalışmalar sürekli yapılmalıdır.

# Proje Ekibi



- Her yazılımcı ve her yazılım ekibi özellikleri itibarıyla başlı başına tektir. Bu yüzden her ekip özel yönetim gerektirir. *Yönetici* kendi çalışan ve ekibinin yapısını tanımalı, kuvvetli ve zayıf yönlerini çok iyi bilmelidir.
- Ekibin yapısını değiştirmek genellikle zordur. Bu yüzden plan hazırlanırken ve yazılım geliştirme yöntemi, kullanılacak yazılım geliştirme aracı ve diğer altyapılar seçilirken ekibin yapısı dikkate alınmalıdır.
- Üst yönetici, son kullanıcı ve müşteri de yazılım geliştirme ekibinin bir parçası olarak düşünülmelidir. Projenin başarı veya başarısızlığı, ilgili tüm kişilerin ortak bir ekip ruhuyla çalışmasına bağlıdır. **Ekip ruhuyla yapılan çalışmalar ayrı ayrı yapılan çalışmalardan her anlamda daha başarılı olur.**

# 1. Ekip Üyeleri

Yazılım projesi, birbiriyle bağlantılı birçok farklı görevin koordineli bir şekilde yapılmasıyla gerçekleştirilir. Bu görevler analiz, tasarım, kodlama ve test gibi alanlarda belli rollerle standartlaşmıştır. Farklı firmalarda rol tanımları değişebilir. Büyük projelerde her rol için ayrı bir ekip de oluşturulabilir.

Rol (uzmanlık) tanımı firmanın yapısı, projenin özellikleri ve yazılım geliştirmede kullanılan sürece göre netleşir. Ekibin her üyesi kurum, proje ve süreçteki yerini bilmelidir.

***Kişi, süreç içerisinde aynı anda birden fazla role sahip olabilir. Bu anlamda rol unvandan farklıdır[Kruchten-2003].***

Küçük bir projede proje yöneticisi, sistem mimarlığını da yürütebilir. Analiz aşamasında sistem analist olarak çalışan bir kişi, sonra test grubunda görev alabilir. Ancak her temel görevin doğrudan bir sorumlusu olmalıdır. Kişi aynı anda birden fazla göreve atanarak sürekli bölünmemelidir.

# 1. Ekip Üyeleri



## □ Proje Yöneticisi

*Proje yöneticisi,*

- projenin istenilen kalite ve sürede tamamlanması için gerekli planın hazırlanması,
- standartların belirlenmesi,
- teknik altyapı seçimi,
- plandaki faaliyetlerin koordinasyonu,
- görevlere uygun kişilerin atanması ve
- ekibinin plana uygun hareket etmesini

kontrol eden sorumludur. Proje yöneticisi temel kararları alır. Tüm işlere sürekli müdahale etmesi, ayrıntılarla uğraşması mümkün ve doğru değildir.

***Proje yöneticisi, başarı ve başarısızlık durumunda hesap verecek ilk kişidir.*** Üst yönetici ve müşterinin, yazılım ekibiyle birlikte yapacakları çalışmaları da proje yöneticisi koordine eder.

# 1. Ekip Üyeleri



## □ **Takım Lideri**

*Takım lideri, proje yöneticisine bağlı çalışan teknik yöneticidir.* Projenin teknik altyapısının isteklere ve hazırlanmış plana göre geliştirilmesinden sorumludur. Proje yöneticisiyle birlikte ihtiyaç ve teknik gereksinimler arasında bir köprü oluşturur. Proje ekibinin iç iletişimini ve yeni çalışanların intibakını sağlar.

Kaynaklarda genellikle proje yöneticisi, teknik yöneticilerin üzerinde bir koordinasyon makamı olarak konumlanır. Ancak küçük ve orta ölçekli projelerde proje yöneticisi ile takım lideri genellikle aynı kişidir. Proje büyüdükçe her modül için bir takım lideri tayin edilebilir. Proje yöneticisi birçok projeyle ilgilenebilirken takım lideri genellikle tek bir proje içerisinde çalışır.



A hand-drawn diagram illustrating the components of a database. At the top, a box labeled "DATABASE" is the central focus. Four arrows point upwards towards this box, each originating from a category of data. Below each category, a list of attributes is provided:

- Users:** password, username, email
- Groups:** group ID, name, role, status
- Entities:** entity ID, name, type, category ID
- Features:** feature ID, name, value, status

*Yazılım mimarı, kullanıcı ihtiyaçlarını karşılamak için nasıl bir yazılım mimarisi kullanılacağını belirler.* Yazılım mimarisi yazılımın üst düzey bir modelidir ve ayrıntılı tasarımın nasıl olacağı konusunda da yol gösterir. Analiz, tasarım ve gerçekleştirilme aşamalarının birbiriyle tutarlı olması için yazılımın mimari yapısı doğru kurulmalıdır.

Mimarinin belirlenmesi tecrübe gerektirir. Bu yüzden mimarinin tecrübeli kişilerden oluşan çekirdek bir ekip tarafından hazırlanması ve diğer yazılımcıların bunu kullanarak yazılımı gerçekleştirmesi önerilir.

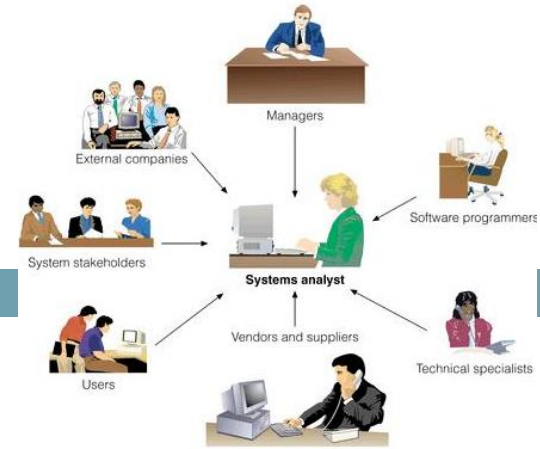
# 1. Ekip Üyeleri

## □ Sistem Analisti

*Sistem analisti*, kullanıcı ihtiyaçlarını tespit, analiz ve belgelemekle sorumludur. Gerçekleştirilen yazılım veya prototipi kullanıcıya sunarak geri beslemeleri yazılım ekibine aktarır.

*Sistem analisti*, yazılım ekibindeki en önemli rollerden birisidir. Analiz çalışması, yazılımın hizmet vereceği sektörde tecrübesi olan kişilerle birlikte yürütülmelidir.

Örneğin hukuk alanına hizmet edecek bir yazılım için bir avukatla çalışmak gibi...



# 1. Ekip Üyeleri

## □ Yazılım Geliştirici

*Yazılım Geliştirici*, yazılımın gerçekleştirilmesinden sorumludur. Genellikle programcı olarak anılır. Gerçekleştirme kodlama, ekran ve raporların yazılması gibi faaliyetleri kapsar. Gerçekleştirme yapılabilmesi için kod seviyesinde gerekli ayrıntılı tasarım ve modelin oluşturulması da programcı sorumluluğundadır.

Örneğin yapılacak bir sıralama işleminde hangi algoritmanın kullanılacağını programcı seçer.



# 1. Ekip Üyeleri

## □ Ara Yüz Geliştirici

*Ara Yüz Geliştirici*, kullanıcının yazılıma erişimini sağlayan arayüzlerin tasarımından sorumludur. Özellikle internet tabanlı yazılımlarda kod ve ara yüz geliştirme iki farklı rolle yürütülmektedir. Ara yüz geliştirici ekranın grafik tasarımı ve özellikle de görsel güzelliğinden sorumludur. Grafik tasarımcı ve kod geliştirici ortak standartlara dayalı çalışarak, kod ile ekrandaki görsel işlevlerin bütünleşik ve tutarlı olmasını sağlar.



# 1. Ekip Üyeleri

## □ Ürün Yöneticisi

*Ürün yöneticisi*, kullanıcı ihtiyaçlarından yola çıkarak yazılımın ticari bir ürün haline getirilmesi sürecini yönetir. Piyasadaki benzer ürünleri inceleyerek ürünün gelecekte alacağı şekli belirler. Ürünün sürüm stratejisini oluşturur. Bu işlemlerin yürütülebilmesi için gerekli belge ve bilgileri yazılım ekibine aktarır.



# 1. Ekip Üyeleri

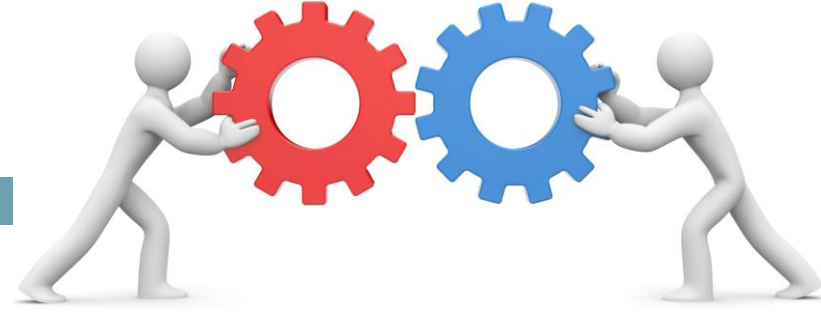


## □ Kalite/Test Mühendisi

*Kalite/Test Mühendisi*, projenin kullanıcı istek, teknik gereksinim ve standartları karşılayacak şekilde hatasız geliştirilmesi için gerekli kontrol ve testleri yapmaktan sorumludur.

Kalite ve test farklı hedeflere yöneliktir. Kalite istenilen alanda piyasadaki en iyi özelliklere sahip nitelikli bir ürün çıkarmayı, test hata bulmayı hedefler. Bu yüzden bazı şirketlerde kalite ve test farklı kişiler tarafından yerine getirilebilir.

# 1. Ekip Üyeleri



## □ Yardımcı Ürün Geliştirici

*Yardımcı Ürün Geliştirici*, projenin geliştirilmesi için kullanılan ürünlere yardımcı olacak, işleri hızlandıracak ve kolaylaştıracak yan ürünler teminini organize eden veya bizzat geliştiren kişidir. Yazılım ürün dosyalarının derleme ve dağıtımını kolaylaştıran sistemler, sürüm kontrol sistem ve proje yönetim sistemleri gibi ürünler bu kişiler tarafından araştırılır ve kuruma uyarlanır.

Yazılım geliştirmeye doğrudan görevli bir kişi, üzerinde çalıştığı işin daha kolay nasıl yapılabileceğini düşünemeyecek kadar yoğun olabilmektedir. Kendisine verilen görevi kendisine verilen araçlarla bitirmeye gayret eder.

Görevi yardımcı ürünlerle geliştirme olan bir uzman bulundurmak, işleri kolaylaştıracak ve otomatikleştirecek araçların tespitini sağlar. Böylece yazılımcılar rutin işlerden kurtulur ve proje için çok önemli zaman kazancı sağlanır.

# 1. Ekip Üyeleri

## □ Risk Yöneticisi

*Risk Yöneticisi*, projede ortaya çıkabilecek olumsuz durumları tespit, takip ve düzeltici işlemleri belirlemekten sorumludur. Proje ekibinin hedefi projeyi biran önce bitirmek olduğundan, ekip içinde olumsuz durumlar göz ardı edilebilir. Bu yüzden risk yönetimi için harici bir ekip oluşturulmasında fayda vardır.





# 1. Ekip Üyeleri



## □ Son Kullanıcı Belge Geliştirici

*Son Kullanıcı Belge Geliştirici*, son kullanıcıların yazılım ürününü kullanabilmeleri için gerekli olan belgeleri geliştirme, güncelleme ve dağıtım konularında görev yapan kişidir.

Hazırlanan belgeler ilk anda kullanıcı eğitimlerinde kullanılabileceği gibi yazılımın çalışma sürecinde uzun vadeli yardım şeklinde de kullanılabilir. Belge içerikleri hedeflenen son kullanıcı yapısına göre belirlenir. Sistemi yöneten kullanıcı için teknik, son kullanıcı için işe yönelik belgeler hazırlanabilir.

Yazılım projesi, kendi kendilerini açıklayacak bir yapıda olmalıdır.

İlişkili fonksiyonlar birbirine yakın ve son kullanıcının anlayabileceği isimler taşınmalıdır. Menü, ekran sihirbazı ve kısa yol gibi fonksiyonlar kullanıcıya işi nasıl yapacağını göstererek yardımcı olmalıdır. Böylece son kullanıcı, programı fazla belge veya yardıma ihtiyaç duymadan kullanabilir ve belge geliştiricilerin işi kolaylaşır.

# 1. Ekip Üyeleri

## □ Destekleyici

*Destekçi veya sponsor*, projeye idari destek ve/veya mali kaynak sağlayan kişidir. Kurum üst-ara yöneticileri ve mali kaynak sağlayan üçüncü kurum yöneticileri bu kategoride değerlendirilebilir. Destekçi, genellikle ayrıntılarla değil projenin üst seviye vizyonu ve temel fonksiyonu ile ilgilenir.



# 1. Ekip Üyeleri



## □ Müşteri

*Müşteri*, yaptığı işlerde üründen fayda sağlayan kişi veya gruplardır. Müşteri daha çok ürünün sahip olması gereken fonksiyonların ilk tanımını yapan kişidir. Önce müşterilerle görüşülür ve genel kapsam tanımlanır. Bu tanım, son kullanıcılarla yapılacak ayrıntılı çalışmalarla son halini alır.



# 1. Ekip Üyeleri



## □ Son Kullanıcı

*Son kullanıcı*, proje sonunda ortaya çıkan ürünü doğrudan kullanacak kişidir. Müşteri ve son kullanıcının, beklentileri ve bu beklentilerin yazılım özellikleri üzerine yansımaları farklıdır.

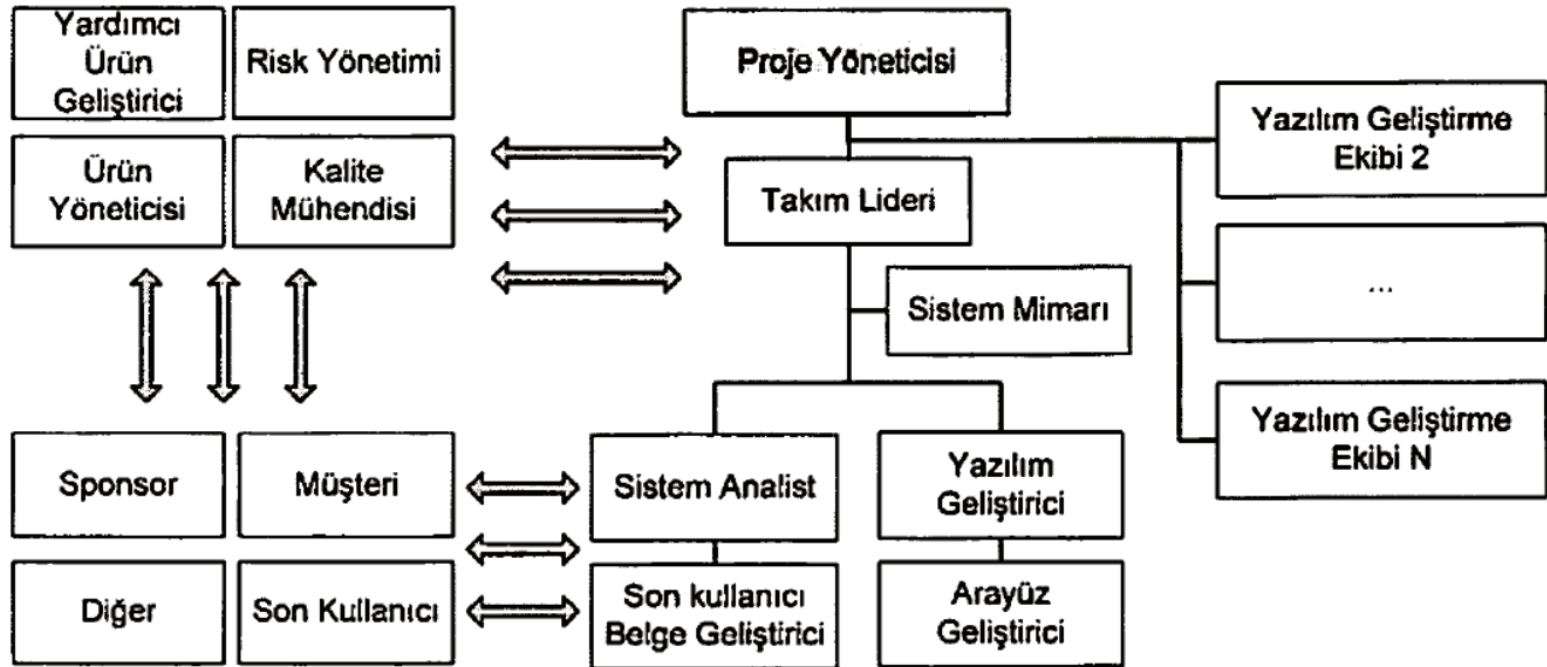
Örneğin bir firma, satışlarını internetten yapmasını sağlayacak bir yazılım teminine karar vermiş olsun. Bu durumda firma idari personeli *destekçi*, satış birimi yazılımın *müşterisi*, internet sitesi vasıtasıyla alışveriş yapan kişi ve programın firma içinde kullanan kişiler de *son kullanıcı* olacaktır.

**Genellikle müşteri ve destekçi *idealist*, son kullanıcı ise gerçekçidir** [McConnell-1997]. Müşteri ve destekçi yazılımla kuruma yeni bir düzen oturtmayı hedeflerken son kullanıcı mevcut duruma odaklanır. Yazılım ekibi tüm paydaşların memnuniyetini sağlayacak ortak noktalar bulmayı hedeflemelidir.

# 1. Ekip Üyeleri

## Yazılım Ekip Üyeleri Arasındaki İlişkiler

Yazılım geliştirme projelerinde takım lideri, sistem analisti, yazılım geliştirici, ara yüz geliştirici ve son kullanıcı belge geliştiriciler çekirdek ekibi oluşturur.



Yazılım ekibinde roller arası iletişim

# 1. Ekip Üyeleri

## Yazılım Ekip Üyeleri Arasındaki İlişkiler

Proje yöneticisi bağlantı noktasındadır. Yardımcı ürün geliştirici, risk yöneticisi, kalite mühendisi ve ürün yöneticisi geliştirilen ürünü destekleyen ve tamamlayan roller icra eder. Bu kişiler aynı anda birden fazla projeye hizmet verebilir.

Müşteri, son kullanıcı ve sponsorlar da geliştirme ekibinin (doğrudan olmasa da) bir parçası olarak düşünülmelidir. Özellikle son kullanıcıları temsilen ekip içerisinde birkaç kişi bulundurulması süreci çok hızlandıracaktır.



# 2.Birlikte Çalışma Kaideleri

İnsanların birlikte çalışması;

- ❑ ortak hedefe sahip olma,
- ❑ karşılıklı saygı, güven, dürüstlük,
- ❑ iletişim becerisi ve
- ❑ birlikte çalışmaya önem verme

gibi temel kaidelere bağlıdır.

Birlikte yapılan çalışmalarda eğer bu esaslara uyulursa, kişilerin ayrı ayrı yapacağı çalışmaların toplamından çok daha fazla iş üretilir.

*Birlikte çalışmak*, herkesin aynı şeyi düşünmesi değil, tartışmalara rağmen ortak bir noktada buluşabilmektir. Yöneticinin temel sorumluluklarından birisi birlikte çalışma kaidelerini belirlemek ve bunların uygulamasını sağlamaktır.

**Başarılı bir proje ancak birlikte çalışmayı bilen bir ekiple olasıdır!**





# 2.Birlikte Çalışma Kaideleri

## 1. Genel İnsani Kaideler

Karşılıklı saygı, güven, dürüstlük ve iletişim becerisi hangi iş yapılırsa yapılsın öncelikli ve önemli kaidelerdir. Bir kişi çok iyi kod da geliştirse, dürüst olmayan bir hareketi ekip ruhunu zedeler. Yazılımın karmaşık yapısı da dürüstlüğün önemini arttırır. Bir işin teknik olarak bitip bitmediğini anlayabilecek kişi sayısı sınırlıdır. Bu ortamda

- ❑ bitmeyen bir işlemi bitmiş olarak göstermek,
- ❑ kimsenin anlamadığı teknik gerekçeler ileri sürmek ve
- ❑ yapılan hataları gizlemek

gibi davranışlar kesinlikle doğru değildir.

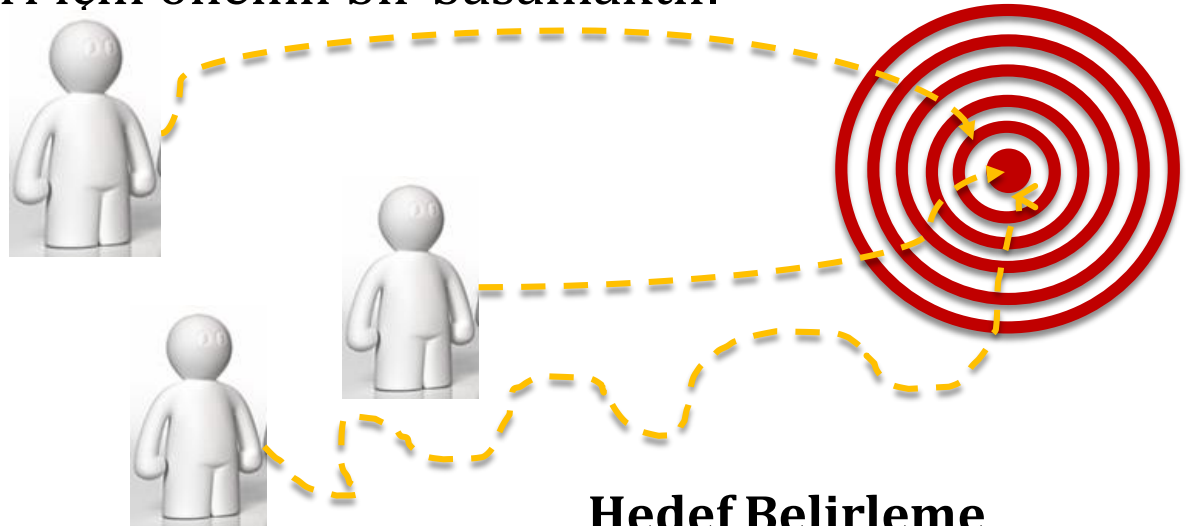




# 2.Birlikte Çalışma Kaideleri

## 2. Ortak Hedef

Kişiler birçok hedefe sahiptir ancak öncelikli hedefleri kendi hayat standartlarını idame ettirmek ve geliştirmektir. Geliştirilen yazılımla ilgili düşünceler farklı da olsa yazılım ekibi, **başarılı ve kaliteli bir yazılım ortaya çıkartma ortak hedefi etrafından birleşmelidir**; Başarılı bir proje, tüm ekip üyeleri için iyi bir referans ve kariyerleri için önemli bir basamaktır.



**Hedef Belirleme**

## 2. Birlikte Çalışma Kaideleri



### 3. Ekibin Proje Yöneticisini Benimsemesi

Her projede bir yönetici olmalı ve bu kişinin yetkileri kesin bir şekilde belirlenmelidir. Ekip proje yöneticisini kabul etmeli ve temel konularda onun tavsiyelerine uymalıdır. Yöneticisi belli olmayan bir projede, küçük bir fikir ayrılığı bile başarısızlığa zemin hazırlar. Bu durumda kişiler kendiliğinden oluşan sosyal gruplara göre çalışmaya başlar. Herkes kendi arkadaşının işine öncelik verir. Böylesi bir ortamda birlikte proje geliştirme ve ekip ruhundan söz edilemez.

Yetki ve sorumluluk arasındaki sınırlar net çizilmelidir. Aksi halde yönetim belirsizliği ortaya çıkar. Eğer ekipte proje yöneticisi ve takım lideri gibi birden fazla idari rol varsa, bir ekip üyesinin kime hangi konularda bilgi vereceği açıkça belirlenmelidir.

## 2.Birlikte Çalışma Kaideleri



### 4. Birlikte Yapılan İşe Öncelik Verilmesi

Yazılım ekibindeki kişiler, her ne kadar istenmese de aynı anda birden fazla işle uğraşmak zorunda kalmaktadır.

Örneğin bir yazılım geliştirici tam yeni bir kod üzerinde çalışırken, önceden teste gönderdiği işlerden geri dönüşler olur. Ayrıca bu kişi yine önceki projelerden gelen destek işleriyle de uğraşmak zorunda kalmaktadır.

Bu karmaşa içinde proje yöneticisi tüm işlere vakıf olamaz. Bu yüzden kişiler, önceliği diğer kişilerle birlikte çalıştıkları işlere vermeyi yöntem olarak benimsemelidir.

# 2.Birlikte Çalışma Kaideleri



## 4. Birlikte Yapılan İşe Öncelik Verilmesi

Proje yöneticisinin önceliği, diğer kişilerle birlikte yürüttüğü faal proje olmalıdır. Tek başına yürüttüğü yeni proje hazırlıkları öncelik sırasında daha sonra gelir.

Yazılımcılar için de benzer durum geçerlidir. Örneğin bir yazılım geliştirici çalışmasını bitirsin ve işini test eden personele göndersin. Test edilecek fonksiyon sistem entegrasyonu için önemli bir göreve sahip olsun. Test sonucunda bir hata olduğunda yazılımcı elindeki yeni geliştirme işini bırakıp, hatayı düzeltmek için harekete geçmelidir. Aksi halde bu durum entegrasyonun bitmesini bekleyen ve yapılmamasından doğan eksiklikten etkilenecek birçok kişinin işi de engelleyecektir.

Birlikte yapılan işlere öncelik vermekten kişinin sürekli işini bölmesi anlaşılmamalıdır. Ancak kişinin tek başına yaptığı işlere makul sürelerde ara verip ortak yapılan işi öncelikle tamamlaması, ekip olarak ilerlemeyi kolaylaştırır. Aksi halde herkesin birbirinden iş beklediği bir kısır döngü oluşması kaçınılmazdır.

# 2.Birlikte Çalışma Kaideleri

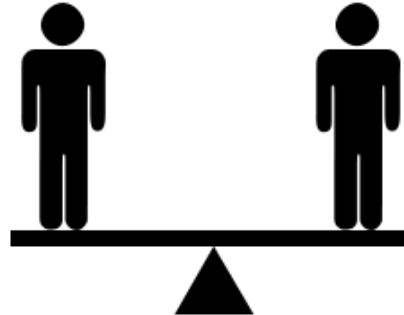


## 5. Denge

Çalışanların sosyal ve teknik becerileri arasında denge sağlanmış bir ekip yapısına ulaşmak hedeflenmelidir. Bu öncelikle karakter ve kişiliğinde bu dengeyi sağlamış çalışanları ekibe kazandırmakla mümkündür. Maalesef bazı kişilerin teknik becerileri arttıkça sosyal ve iletişim becerileri azalmaktadır. Bu durumda proje ekibi seçilirken yakın çalışacak kişiler birbirini dengeleyecek şekilde belirlenmelidir.

Örneğin **meseleleri hızlı çözmeye odaklı bir kişi ile ayrıntılara önem veren bir kişinin birlikte çalışması sağlanabilir**[Turner-2006].

Böylece iş hızlı çözülürken, ayrıntıların atlanmasının önüne geçilmiş olur.



## 2.Birlikte Çalışma Kaideleri

### 6. Birlikte Başarı, Birlikte Başarısızlık İnancı

Başarısızlıktan da başarıdan da tüm ekip pay sahibidir. Projenin başarılı olması kişilerin tek tek başarılı olması manasına da gelir. Bunun tersi de geçerlidir. Başarısız bir projede, bir personel teknik olarak ne kadar başarılı çalışmalar yaparsa yapsın başarısızlıktan pay alır. Tüm ekip bunun bilincinde olarak problemleri açıkça ortaya koymalı ve ortak çalışmalara destek vermelidir.



## 2.Birlikte Çalışma Kaideleri

### 7. Herkesin En İyi Bildiği İşi Yapması

**Kişi, en uzman olduğu konuda görevlendirilmelidir.** Yeni bir iş yapma heyecanını hissettiren tasarım çalışmalarının talibi özellikle çok olur. Ancak bu konudaki görevin tecrübe ve becerisi en yüksek yazılımcılara verilmesi riski azaltır.

Yazılım geliştirme sürecindeki görevler birçok noktada birbiriyle kesişmektedir. Yapılan ihtiyaç analizinin çıktısı yazılım tasarımına dönüşür. Bu noktada analizde bir hata bulan tasarım ekibi analizi kendisi yapmaya kalkmadan, hatayı analiz ekibine bildirmelidir. Tasarım ekibi en iyi bildiği işi bırakıp analist rolüne talip olmamalıdır.



# 2.Birlikte Çalışma Kaideleri

## 8. Yapılanlardan ve Sorunlardan Tüm Ekibin Haberi Olması

Proje geliştirilirken ortaya birçok sorun çıkar. Sorunların ekip üyelerinin hepsinin görebileceği bir ortamda paylaşılması faydalıdır. Örneğin bir testin veya gözden geçirmenin sonucu, sadece ilgili programcının değil tüm ekibin görebileceği bir ortamda paylaşılabilir. Bu durum izlenebilirliği ve ekip içi yardımlaşmayı arttıracaktır.

Bilginin duyurulmasında ise bir öncelik sırası belirlenmelidir. Bilgi öncelikle doğrudan ilgili kişilere aktarılır. Bunun için doğrudan görüşme veya elektronik bir postayla uyarı gönderilebilir. Doğrudan sorunla ilgili olmayan kişilerin bölünmemesi için, onlara uyarı göndermek yerine, bilgiler ortak bir internet uygulamasında gözlem ekranları veya raporlar vasıtasıyla gösterilebilir.





## 2.Birlikte Çalışma Kaideleri

### 9. Eleştirilerin Çözümü de İçermesi

Ekip içerisinde (kendi işini sağlıklı yapan) her kişinin, diğer çalışmalarını eleştirme hakkı vardır. Proje yöneticisi olumlu eleştirileri teşvik etmelidir. Eleştiride kritik nokta eleştirirken çözümü de ortaya koymaktır. Tartışmalarda kişisel çatışmalardan mümkün olduğunca kaçınılıp, önerilen tekniğin fayda maliyet analizi üzerinde durulmalıdır. Bu yıkıcı eleştirilerin de önünü kesecektir.



# 2.Birlikte Çalışma Kaideleri

## 10. Ortak Çalışma Mekanı

En etkili iletişim yüz yüze iletişimdir. Bu yüzden ekip üyelerinin aynı veya yakın fiziksel mekanda çalışması ekip içi paylaşım ve dolayısıyla birlikte çalışma imkanlarını artıracaktır. Bu sayede kişilerin el ve yüz mimikleri dahi iletişimde kullanılabilecek mesajlara dönüşür. İnternet ve video konferans gibi araçlar da iletişim açısından kullanılabilecek faydalı teknolojilerdir. Ancak günümüzde hiçbir teknoloji, ekibin mekan yakınlığıyla aynı yararı sağlamamaktadır!



# 2.Birlikte Çalışma Kaideleri

## 10. Ortak Çalışma Mekanı

**Aynı proje için dağınık çalışma ortamlarında çalışmak, ortak mekanda yapılan geliştirmelere göre daha fazla çaba gerektirir.** Yapılan araştırmalar, aynı iş için gösterilmesi gereken çaba ve çalışma mekanı ilişkisini ortaya koymuştur [Boehm-2000].

Tek bir şirkette ve ortak mekanda yapılan bir proje;

- Eğer birden fazla şirkete ve mekana dağılırsa %25 emek artışı gerekir.
- Eğer milletlerarası bir ekiple yapılırsa %40 emek artışı gerekir.

Yazılım geliştirme yüksek konsantrasyon gerektiren bir çabadır. Ortak mekanda çalışırken, kişilerin kendilerine ait bir çalışma ortamı olması faydalı olacaktır. Böylece sürekli bölünerek işlerin yarım kalması tehlikesi ortadan kalkar.



# 2.Birlikte Çalışma Kaideleri

## **11. Dağınık Çalışma Mekanı- Modül Tabanlı Mekan Ayrımı**

Yazılım ekibinin farklı mekanlarda çalışması iletişim, yönetim ve koordinasyon sorunlarına yol açar. Bu sorunlara çözüm için mekan ayrımının, yazılım süreç aşama veya ürün-modül yapısını yansıtması tercih edilmektedir.

Süreç tabanındaki ayrımında, yazılım sürecindeki aşamalara göre mekan tahsisi yapılır. Analist ekibi bir fiziksel mekana toplanmıştır. Kodlama uzak ve ayrı bir mekanda yapılır. Bu çalışma şekli sürekli irtibat halinde olması gereken analist, tasarımcı ve geliştirici arasındaki iletişimin kopmasına sebep olur.

# 2.Birlikte Çalışma Kaideleri

## 11. Dağınık Çalışma Mekanı- Modül Tabanlı Mekan Ayrımı

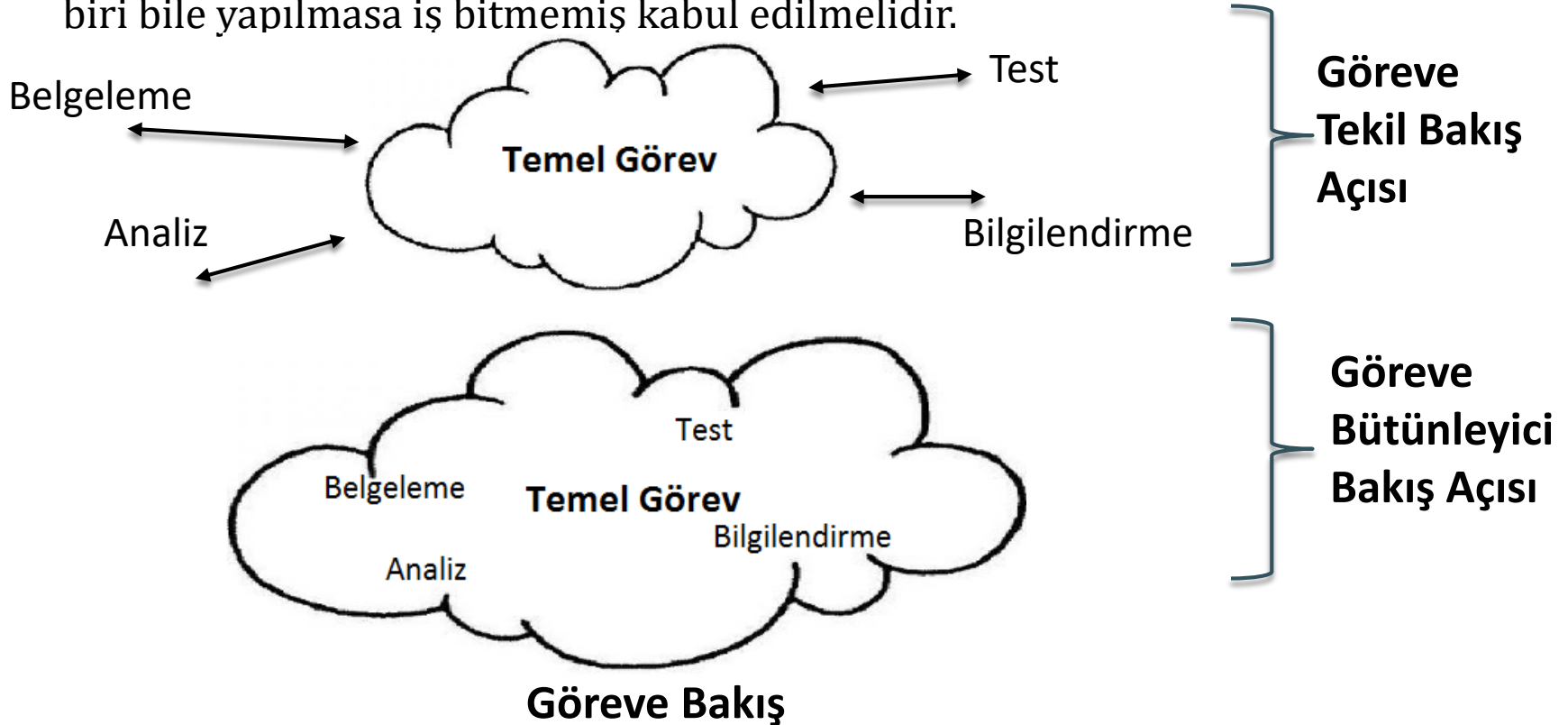
Yazılımın modüllere ayrılması ve her modül için bir fiziksel mekan tahsisi süreç tabanlı ayrıma göre daha faydalı olacaktır. Örneğin kurumsal kaynak planlama sisteminde insan kaynakları modülünü geliştiren ekip bir mekanda, mali işler modülü geliştiren ekip bir mekanda toplanabilir. Böylece ekip yapısı, modül içi iletişimi artırmak ve modüller arası iletişimi azaltmak kuralına uygun kurulmuş olacaktır.

Proje bakım ve destek çalışmalarını yürüten ekiplerin merkezileştirilmesi de çok tercih edilen diğer bir yöntemdir. Hatalar ortak bir merkezden alınıp gözetilmeye çalışılır. Bu merkezde giderilemeyen hatalar ürün sorumlularına iletilir.

# 2.Birlikte Çalışma Kaideleri

## 12. İş Bir Bütün Olarak Kabul Etmek

Ekip ortamında yapılan bir iş, kişilerin yalnız başına yaptıkları kod geliştirme gibi teknik çalışmalardan ibaret değildir. Bilgilendirme, belgeleme, test gibi destekleyici işlemler için doğrudan bir parçası kabul edilmelidir. Bu işlemlerden biri bile yapılmasa iş bitmemiş kabul edilmelidir.



# 2.Birlikte Çalışma Kaideleri

## 12. İşi Bir Bütün Olarak Kabul Etmek

Proje yöneticisi, kişilerin belgeleme ve bilgi verme gibi ana görevle bütünleşik faaliyetlerini takip etmelidir. Bunların projenin sağlıklı yürümesine sağladığı yararlar ekip elemanlarına sürekli izah etmelidir. Örneğin bir programcının yazacağı küçük bir raporla test ekibine zamanında bilgi vermeyi ihmal etmesi, test ekibinin konuyu tekrar tekrar sormasına yol açar. Raporu yazmamak ilk anda belki birkaç dakika kazandırır fakat sonradan çok daha büyük zaman kaybına yol açar. Elbette gerçekten yararsız bürokratik işler varsa bunlar kaldırılmalı veya yararlı şekle dönüştürülmelidir.

Ekte her iş ilgili kişilerin anlayacağı ve kabul edeceği mahiyette olmalıdır. Örneğin kalitesiz kod, sadece teknik olarak hata çıkaran kod demek değildir. Standartlara uymadığı için diğer ekip üyelerince anlaşılamayan kod da kalitesiz kabul edilmelidir.



# 2.Birlikte Çalışma Kaideleri

## 13. Ortak Çalışma Altyapıları Seçimi

Ortak çalışmaları sağlıklı yürütebilmek için bunu destekleyen altyapılar kullanılmalıdır. Birlikte çalışmaya katkı sağlayan altyapılar olarak;

- ▣ uygulama yaşam döngüsü yönetim,
- ▣ proje yönetim,
- ▣ görev takip,
- ▣ e-posta ve
- ▣ belge yönetim sistemleri

sayılabilir. Büyük projelerde bu tür altyapılar kullanılması kaçınılmaz bir mecburiyettir.

Ortak çalışma aracı kullanmak, en küçük iş parçasından başlayarak standartlaşmalı ve teşvik edilmelidir. Yapısal programlamada fonksiyon ve paket; nesneye yönelik tasarımda nesne ve bileşen kütüphanesi; veri tabanı paketi bu tür altyapılardır.



# 2.Birlikte Çalışma Kaideleri

## 13. Ortak Çalışma Altyapıları Seçimi

Bilgiler sürüm kontrolü de sağlayan merkezi belge yönetim ve paylaşım sistemlerinde saklanmalıdır. Böyle bir sistem yoksa en azından dosya sunucusu kullanılabilir. Ancak hiçbir durumda bilgi sadece kişisel bilgisayarlara hapsedilmemelidir.

e-posta araçları kolayca bilgi gönderme ve toplantı düzenleme gibi fonksiyonlarla birlikte çalışmaya katkı sağlar. e-posta bilgileri belgeleme amacıyla da kullanılabilir.

Proje yöneticisi ve yardımcı ürün geliştirici, sürekli ekibin çalışma altyapısını ortak çalışma araçlarıyla zenginleştirecek çalışmalar yapmalıdır. Altyapının tespitinden sonra ekibin çalışma şeklini açıklayıcı standartlar oluşturulur. Gerekli eğitim verildikten sonra yazılım ekibinden bu altyapıları kullanması beklenir.

## 2.Birlikte Çalışma Kaideleri

### 14. Kişi Sayısının Düşük Tutulması

**Ekip içi iletişimi kolaylaştırmak için ekip sayısı 6 en fazla 10 kişiyi geçmemelidir.** Büyük projelerde yüzlerce kişi çalışabilir. Bunların tek bir ekip olarak çalışması halinde, iletişim ihtiyacı artar ve bu ihtiyaç projeyi yönetilmesi zor bir noktaya sürükler. Bu durumda ekip, küçük alt ekiplere bölünmelidir. Ekibi daha küçük ekiplere bölmek iletişim ihtiyacını azaltmaz. Sadece iletişimi daha sistemli ve kolay yapmayı sağlar. Proje yöneticisinin ekipler arası koordinasyon vazifesi her zaman önemlidir.

## 2.Birlikte Çalışma Kaideleri

### 15. Projeye Uygun Kişilerin İstihdamı

**Ekte yeterli sayıda ve yapılacak işe uygun tecrübede çalışan istihdam edilmelidir.** Projenin gerektirdiği adam gün sayısı ayrıntılı şekilde belirlendikten sonra istenen sürede tamamlanması için gerekli tecrübe ve sayıda personel ekibe dahil edilmelidir. Yeterli kaynak sağlanmayan projelerin başarısız olacağı açıktır. Bu tür projelerde birlikte çalışma imkanları da azalır. Çünkü ekip üzerindeki baskı artar ve kişiler işi bir an önce bitirme kaygısıyla yalnız çalışmayı tercih eder.

Ekteki üye ve unvanlar gerçekleştirilecek proje büyüklüğüne ve yazılım firmasının imkanlarına göre değişkenlik gösterebilir. Küçük projelerde bir kişi birden fazla rol üstlenebilirken büyük projelerde daha özel roller bulunur. Küçük çaplı bir projede programcı analist fonksiyonlarını da üstlenebilirken geniş kapsamlı bir projede performans yöneticisi ve performans analisti gibi farklı roller vardır.

## 2.Birlikte Çalışma Kaideleri

### 15. Projeye Uygun Kişilerin İstihdamı

Ancak **bir yazılım projesinde en az bir proje yöneticisi, yazılım analisti-kalite sorumlusu ve yazılımcı bulunmalıdır. Yazılımcı ve yazılımı test eden kişi farklı olmalıdır.** Çünkü yazılımcı projeyi çalışır hale getirmeye, test yapan kişi ise programı kırmaya ve hata bulmaya çalışır. Bu kişinin kendi projesini test etmesinin önünde psikolojik bir engeldir.

Yazılım ekibinde önemli bazı pozisyonlar muhakkak doldurulmalıdır. Diğer meslek dallarında bu konunun anlaşılması yazılım geliştirmeye göre kolaydır. Örneğin futbol takımı kaleci olmadan nasıl kurulamazsa yazılım ekibinde de test sorumlusu olmaması düşünülemez. Yazılım alanında tecrübe sahibi olmayan idari yöneticilerin, proje yönetimini doğrudan yürütmek istemeleri de benzer bir hatadır. Mali ve idari bazı mazeretlerle önemli mevkilerin boş bırakılması projeyi başarısızlığa sürükler.

## 2.Birlikte Çalışma Kaideleri

### 16. Ekibin Uzun Süre Birlikte Çalışması

Bilişim ve yazılım sürekli gelişen, yeni fırsatlar ortaya çıkan bunun sonucunda da kişilerin sıklıkla iş değiştirdikleri alanlardır. **Birlikte çalışma ortamı, teorik bilgilerden ziyade uzun süreli beraber çalışma sayesinde kişilerin birbirini tanımasıyla oluşur.** Yazılım ekibindeki tecrübeli ve kritik rollerdeki kişilerin uzun süreli istihdam edilmesi için gerekli tedbirlerin alınmalıdır. Bu tedbirlerden bazıları kişiye özel kariyer planlama, uygun ücret politikası ve eğitim imkanları şeklinde sıralanabilir.

## 2.Birlikte Çalışma Kaideleri

### **17. Birlikte İş Harici Faaliyetler**

Birlikte çalışma sadece resmi görüşmelerle sağlanamaz. Kurumsal yemek, sportif faaliyet ve geziler ekibin birbirini tanımasını sağlayarak aralarındaki iletişimi artırır. Kişiler arası resmi olmayan iletişim, resmi iletişimi de kolaylaştırır.

Yazılım alanında resmi iletişim kadar gayri resmi ve serbest iletişim de önemlidir. Çünkü kararların önemli bir kısmı kişisel yargılara dayanır.

# 3.Ekip ruhunu zedeleyen davranışlar

Ekip ruhunu kuvvetlendiren davranışlar yanında bunu zedeleyen davranışlar da mevcuttur. Proje yönetici bu tür durumlara müsaade etmemeli, oluşmuşsa sebeplerini araştırarak tekrarının önüne geçmelidir.

# 3.Ekip ruhunu zedeleyen davranışlar

## 1. Plansızlık

Plansızlık veya var olan planın izlenmemesi ekibin ne iş yapacağını, hedefin ne olduğunu bilememesine ve dolayısıyla kişilerin kendi başına çalışmasına yol açar. Kimsenin diğer kişilerin ne yaptığını bilmediği bir ortamda birlikte çalışmaktan bahsedilemez. Belirsizlik bir sis gibi proje ekibinin üzerine çöker kimse önünü göremez. Ekip olarak hareket edebilmek için herkes, çalışmasını genel plana bağlı olarak yapmalıdır. Yine herkesin, genel plana bağlı özel bir planı olması da faydalıdır.

Kişilerin planı benimsememeleri de önemli bir problemdir. Plandaki işin zamanında bitmemesinden rahatsızlık duymayan yazılımcılar mevcuttur. Bu kişiler uygun dille ikna edilerek kazanılmaya çalışılmalıdır. Bu imkan yoksa proje ve hatta işten uzaklaştırmaya kadar varan kararlar alınması zorunlu olabilir.



# 3.Ekip ruhunu zedeleyen davranışlar

## 2. Bitince Gösterelim Değil Birlikte Geliştirelim

"*Bitince gösterelim*" cümlesi genellikle yapılan bir çalışmayı görmek isteyen proje yöneticisine yazılımcı tarafından veya bir müşteriye yazılım geliştirme firması tarafından söylenen tehlikeli bir ifadedir. Proje yöneticisi bu tür sözler karşısında dikkatli olmalıdır. Elbette bitmemiş bir yazılımın gösterilmesi risklidir. Ancak nasıl insan adım adım yetişkin oluyorsa, her proje de aşama aşama büyür. Projenin her aşamasında proje yönetici veya müşterinin göreceği (görmesi gerekli) şeyler vardır.

Proje yöneticisinin görevi sadece sonucu görmek değildir. Sonuca giden yolda doğru yöntemin izlenmesini sağlamak da bu makamın görevidir. Müşterinin de birçok ara aşamada yapılanları görmesi gerekir. Müşteri, üst yönetim ve yazılımcılar ortak bir ekip olmalı ve birlikte hareket etmelidir.

# 3.Ekip ruhunu zedeleyen davranışlar

## 2. Bitince Gösterelim Değil Birlikte Geliştirelim

Projenin sürekli çalışır bir şekilde geliştirilmesi yukarıda birçok problemi engeller. *Microsoft®* firmasının başlangıçtan itibaren her akşam çalışan bir sürüm oluşturmaya dayanan stratejisi buna bir örnektir [McConneii-1993]. **Bu sayede yazılım ekibi gelişmeleri yakından izleyebilir ve birçok fonksiyon erken test edilebilir.**

Adım adım geliştirmek sadece bütün proje için değil projenin müstakil adımları için de geçerlidir. Örneğin analiz belgesi hazırlama için bir ay ayrılmış olsun. Bu belgeyi müşteriye 29. gün sunmak doğru değildir. ilk haftanın sonunda gösterilecek ve tartışılacak birçok şey olacaktır. "*Bitirince gösterelim*" değil "*Her aşamada gösterip istişare edelim ve birlikte bitirelim.*" anlayışı tercih edilmelidir.

# 3.Ekip ruhunu zedeleyen davranışlar

## 3. Hatayı Kendi Dışında Görmek

Her yazılımcı yaptığı işleri sahiplenmeli ve hata varsa çözmeye çalışmalıdır. Bazı durumlarda altyapı yazılımındaki hatadan (*bug*) kaynaklanan sorunlara dahi alternatif çözümler bulmak gerekir. "*Bu işletim sisteminin hatası benim kodum çalışıyor*" denemez. Kod, işletim sistemi hatasına takılmayacak şekilde, tekrar düzenlenmelidir.

"*Kullanıcı hatası*" neredeyse her yazılımcının dilindeki bir cümledir. Bu cümle en tehlikeli yaklaşımlardan birisidir çünkü var olan hatayı sadece çözmeyi değil, görmeyi dahi engeller. Elbette kullanıcı hata yapabilir. Ancak bu hatalardan birçoğu yazılımların buna izin vermesinden kaynaklanır. Örneğin kullanıcı ancak yazılımda bir sorun varsa bilgi tutarlılığını zedeleyen hatalı kayıt girebilir.

# 3.Ekip ruhunu zedeleyen davranışlar

## 4. Hatayı Kişilerde Görmek

Yazılım, bir ekip ürünüdür. Süreç içerisinde olgunlaşır. Kodlamada yapılan bir hata ancak testte fark edilmezse devreye alınan ürüne taşınır. Çıkan ürünlerdeki hata, tüm sürecin iyi işlememesi ve sürecin doğru yönetilememesinden kaynaklanır. Hatalı üründen tüm ekip sorumludur. Kişisel suçlamalar, kişiyi yaptığı hatayı gizlemeye iter. Mümkün olduğunca kişi değil, ürün ve süreç değerlendirilmelidir.

# 3.Ekip ruhunu zedeleyen davranışlar

## 5. İşi Aşırı Sahiplenme

Yazılımcılar, yapmış oldukları projeleri sahiplenirler. Bu temelde çok gerekli ve güzeldir. Bazı yazılımcılar yazılımı çocukları gibi görür. Ancak sahiplenmede aşırıya gidilmesi doğru değildir. İşin planlanan sürede bitmemesi ve hataların zamanında çözülmemesi gibi durumlar, görevin başka bir yazılımcıya aktarılmasını gerekli kılabilir. Böylesi durumlarda bile genellikle yazılımcılar, ellerindeki görevi başkasına vermek istemez ve bunu önemli bir başarısızlık olarak görürler.

Görev bitiyor mu sorusuna karşı, "*Ben bunu yapamıyorum.*" cevabıyla neredeyse hiç karşılaşılmaz. Genellikle bitmek fiilinin şimdiki zaman "*Bitmek üzere*", "*bitiyor*", veya gelecek zaman çekimleri "*yarın bitecek*", "*bir iki gün içinde biter*" cevabı alınır. Proje yöneticisi bu cevapları kabul etmemelidir!

# 3.Ekip ruhunu zedeleyen davranışlar

## 5. İşi Aşırı Sahiplenme

Kişilerin o ana kadar yaptıkları sonraki yapacakları işin bir göstergesidir. Uzun süre bir yazılımı bitirememiş bir kişi, bir anda her şeyi çözemez. Tüm ekibin planlarını bitti/bitmedi şeklinde yapmaya alışması çok önemli bir kazançtır.

Zamanında bitmeyen işler için çözüm yolları:

- **Yardımcı olmak:** İşin nasıl yapılabileceği konusunda proje yöneticisinin kendisi yardımcı olabilir, görevi tekrar açıklayabilir veya farklı uzmanların görüşlerinden faydalanılabilir.
- **Süre vermek:** Yeni bir tarih vermek (Haftaya Salıya kadar bitirilmeli gibi).
- **Görev değişikliği:** Eğer sürekli sorun yaşıyorsa görevin kişiden alınması gerekecektir. Kişiye farklı görev verme, başka projede görevlendirme, farklı birimde görevlendirme ve nihayet işten çıkarma ihtimal dahilindedir.

# 3.Ekip ruhunu zedeleyen davranışlar

## 6. İşi Bütünüyle Sahiplenme

Yazılım geliştirme süreci, yeni teknolojilerin araştırılması gibi kariyer açısından faydalı işler olduğu gibi veri aktarmak gibi rutin, zaman alan ve yeni bir şey öğrenme imkanı kısıtlı işlere de sahiptir.

Her işin hamaliyesi vardır. Bir kilo süttten sadece birkaç gram kaymak çıkması gibi yazılım geliştirmede de kişiyi geliştirici işler sınırlıdır. Ancak sıkıntı verici rutin işleri yapmadan bunlara sıra gelmeyecektir. Ayrıca her yapılan işin içinde gelişime açık çeşitli yönler bulmak da önemli bir yetenektir.

Yazılımcı yaptığı işle ilgili her ayrıntıyı dikkate almalıdır. Proje yöneticisi tarafından ayrıntıların tek tek hatırlatılması gerekmemelidir. Ekibe işi verip sonradan gönül rahatlığıyla neticesini bekleyebilmek, bir proje yöneticisi için en büyük bahtiyarlıklardan birisidir. Her yapılan işin başında, ortasında ve sonunda kişileri ikaz etmek zorunda olmak, hem yönetim hem de işi yapan için ciddi bir sıkıntıdır.

# 3.Ekip ruhunu zedeleyen davranışlar

## 7. Değişimleri Esneklikle Karşılama

Yazılım ekipleri çok farklı müşteri ve kullanıcı istekleriyle yüz yüze gelebilmektedir. Bazen istekler arasında tutarsızlık da olabilir. Bu durum yazılım ekibini müşterinin kendi iç çatışmalarının içerisine sürüklememelidir. Dikkatli incelenirse isteklerdeki tutarsızlığın çoğunun esnek bir tasarımla aşılabileceği görülecektir. Örneğin avans işlemi için mali işler ve personel birimi farklı ekranlar istesin. Bu istek yazılımın genel standartlarına uymayabilir. Ancak günlerce bunu tartışmak yerine aynı altyapı ile farklı ekran ve raporlar hazırlanarak sorun çözülebilir.

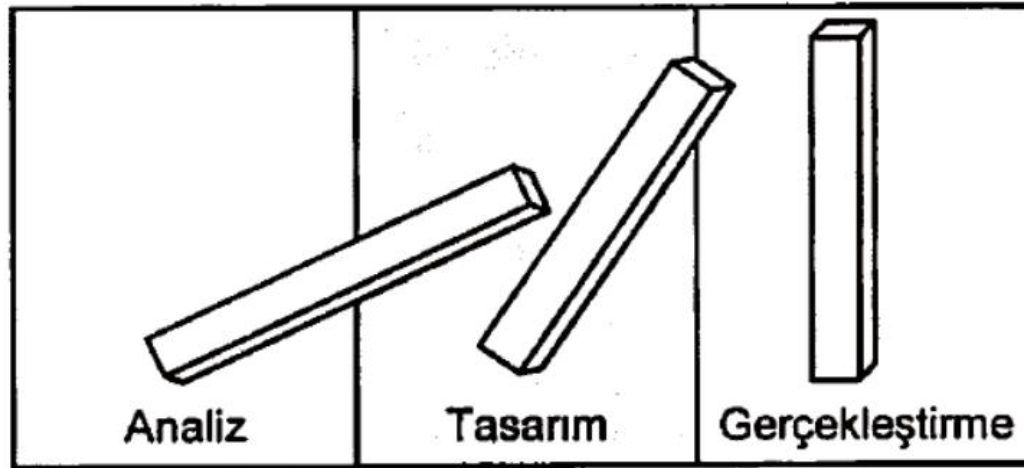
Proje yöneticisi ekibini müşterilerin kendi iç çatışmalarından mümkün olduğunca uzak tutmalıdır. Müşterinin de yazılım ekibinin bir parçası olduğu unutulmadan, esnek ve uzlaşmaya açık bir yaklaşım izlenmelidir. Tartışma, ancak yazılımdaki esnekliğin çözemeyeceği durumlar olduğunda, üst yönetime aktarılmalıdır.



# 3.Ekip ruhunu zedeleyen davranışlar

## 8. Görev Kayması

Yazılım geliştirme sürecinin birbirine bağlı adımlardan oluşması, yazılım ekibindeki kişilerin çalışmalarını da birbiriyle bağımlı hale getirir. Ekipteki bir kişinin sorumlu olduğu aşamada oluşan sıkıntı, diğer kişilere ve aşamalara da doğrudan yansır.



**Görev Kayması**

# 3.Ekip ruhunu zedeleyen davranışlar

## 8. Görev Kayması

**Belli kişilerin görevlerini yapmadığı veya görev paylaşımının sağlıklı yapılmadığı ortamlarda bir görev kayması yaşanır.** Bu genellikle kişinin diğer çalışanların işlerini üstlenmesine, bazen de kendi düzeyinin altındaki işlerde çalışmasına sebep olur. Proje yöneticisi, yazılım mimarı veya programcı gibi çalışmaya başlar. Programcı ise sistem analistinin işlerini üstlenir. Kullanıcıyla görüşür hataları alır ve çözmeye çalışır. Sistem analisti ise kullanıcı gibi çalışmaya hatta sisteme kayıt girmeye başlar.

Bilgi işlem yöneticisi, proje yönetimine müdahale ederse proje yöneticiliğine doğru kayar. Proje yöneticisi ekipte yazılım mimarı veya danışman pozisyonuna geçer. Görev kayması özellikle belgeleme ve test gibi destekleyici işlemleri aksatır.

# 3.Ekip ruhunu zedeleyen davranışlar

## 8. Görev Kayması

Görevi tam yapmama sorunu, hangi seviyede olursa olsun, derhal düzeltme yoluna gidilmezse neticesinde görev kayması olur. Proje yöneticisi görev kaymasına müsaade etmemeli ve sorunlu işlerin bir parçasını kendi üzerine almasının görev kaymasına yol açmamasına dikkat etmelidir.

Proje yöneticisi çeşitli zamanlarda yeni teknoloji araştırmalarında görev alabilir. Ancak bu işler proje yöneticisinin bıraktığı noktada kalmamalı ve ekip tarafından sahiplenilmelidir.

# 3.Ekip ruhunu zedeleyen davranışlar

## 9. Meselelerin Kişiselleştirilmesi

**Yazılım geliştirme sürecindeki tartışmalar, teknik olarak en doğru çözüme ulaşma hedefiyle yapılmalıdır.** Ancak yazılım teknik ve insani meselelerin iç içe geçtiği bir alandır. Maalesef birçok durumda mesele teknik olarak kabul edilmek yerine, kişisel fikir ayrılığı ve çatışmaya dönmektedir.

Örneğin bir kod standartlara uymamasına rağmen çalışıyorsa, geliştiren kişiye bir sorun olduğunu anlatmak zordur. Konunun şahsi olmadığını anlatmak, standartları net ortaya koymak ve ekipteki herkesten standartlara uygun çalışma beklemek gerekir.

Proje ekibi birbirini teknik doğrular ekseninde samimi bir şekilde ikaz etmelidir. İkaz edilen kişi, bunu kişiliğine saldırı olarak değil projenin iyiliği için gerekli bir eleştiri kabul eder ve hata varsa düzeltirse, ekip çalışması çok daha üst düzeye çıkar.

# 4.Kiři ve Ekip Psikolojisi

***Kiřiler düşünerek, sorun çözerek ve araştırarak yazılım geliştirirler.***  
Bu kişilere, uygun ve rahat bir çalışma ortamı sağlanması gerekir. Bu ortam kişilerin işlerinin sürekli bölünmemesi, rahat ofis ortamlarında çalışmaları, moral ve motivasyonlarının yüksek olmasıyla oluşturulur.

# 4.Kişi ve Ekip Psikolojisi

## 1. Ekibi Tanımak

**Proje yöneticisi kendi ekibini tanımak ve proje planını buna göre belirlemek zorundadır.** Ekibin birbirini tanıması kişilerin nasıl davranış göstereceğinin tahmin edilebilmesini sağlar.

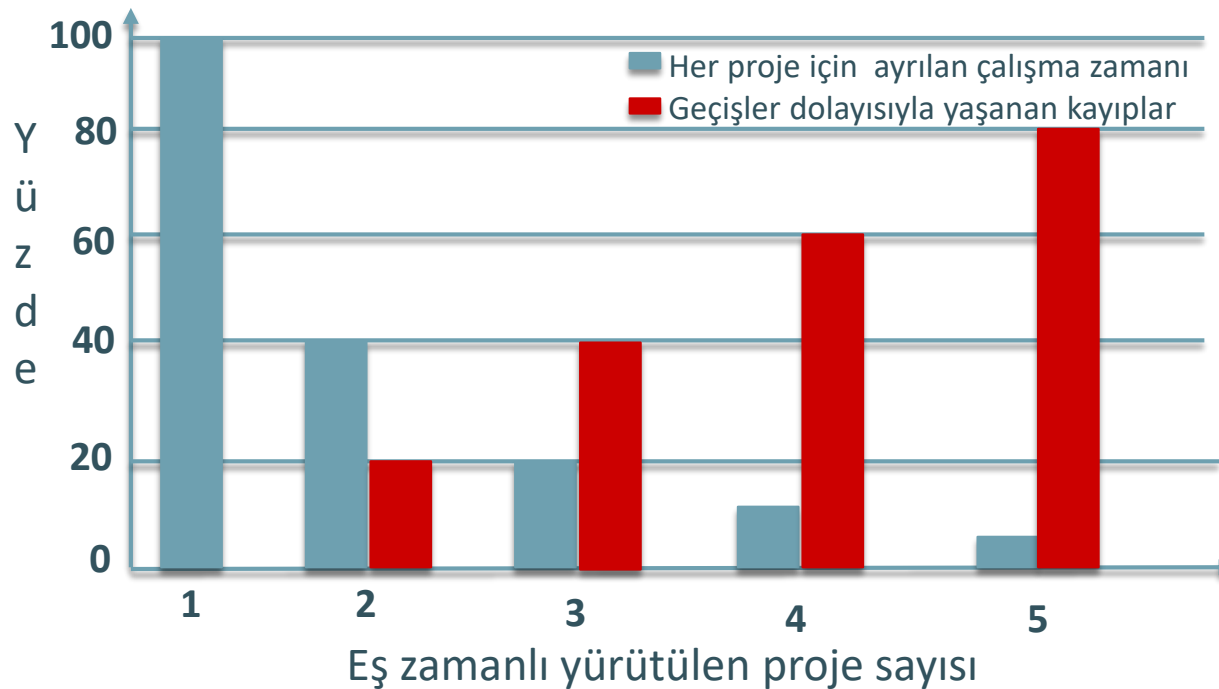
Kişi ve ekipler, görevleri ancak kendi tecrübe ve kabiliyetleri ölçüsünde yapabilir. Bu yüzden ekip yapısına uymayan bir planın başarı şansı azalır. Örneğin mevcut ekip *Microsoft® Visual Studio.NET* teknolojisiyle uygulama geliştirsin ve yeni bir ürün için Java platformunda çalışmalara başlanmış olsun. Proje planına ekibin yeni platforma uyum süresi de eklenmelidir.

Sonradan ekibe katılan proje yöneticisinin ilk yapması gereken iş, ekibi tanımaya çalışmaktır. Bunun için proje ekibiyle toplantılar yapmak, ekibin özgeçmişlerini ve projede yapılmış işleri incelemek gerekir.

# 4. Kiři ve Ekip Psikolojisi

## 2. Srekli Blnme

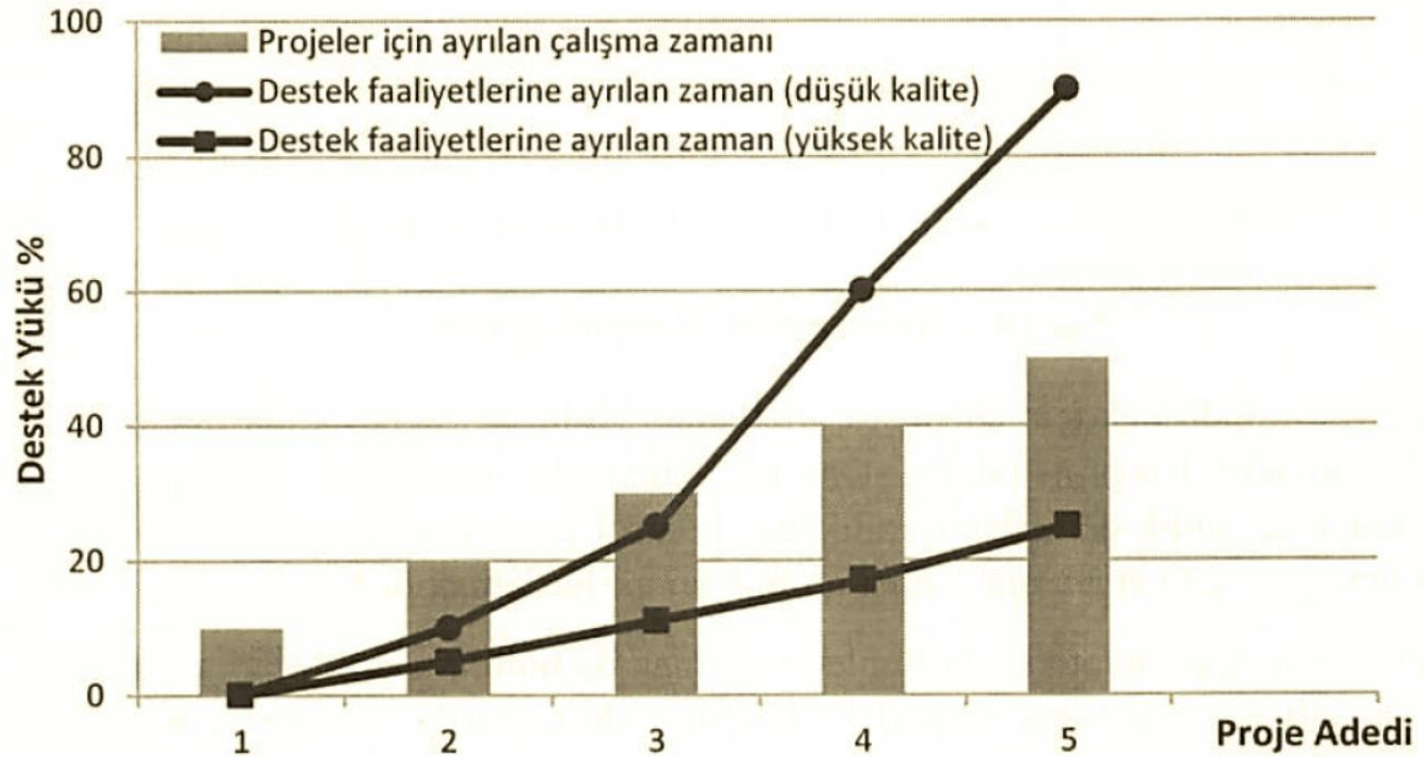
Yazılım geliřtirme, yksek yoęunlařma gerektiren bir abadır. Kiřinin 15-20 dakikalık kesintisiz bir alıřmadan sonra tam olarak yoęunlařtıęı ve karmařık problemleri zebilecek zihni seviyeye eriřtięi eřitli arařtırmalarla gsterilmiřtir [Demarco-1999].



Aynı anda  
yapılan  
projenin  
zaman  
kullanımına  
etkisi  
[Weinberg-  
1991]

# 4.Kiři ve Ekip Psikolojisi

## 2. Sürekli Bölünme

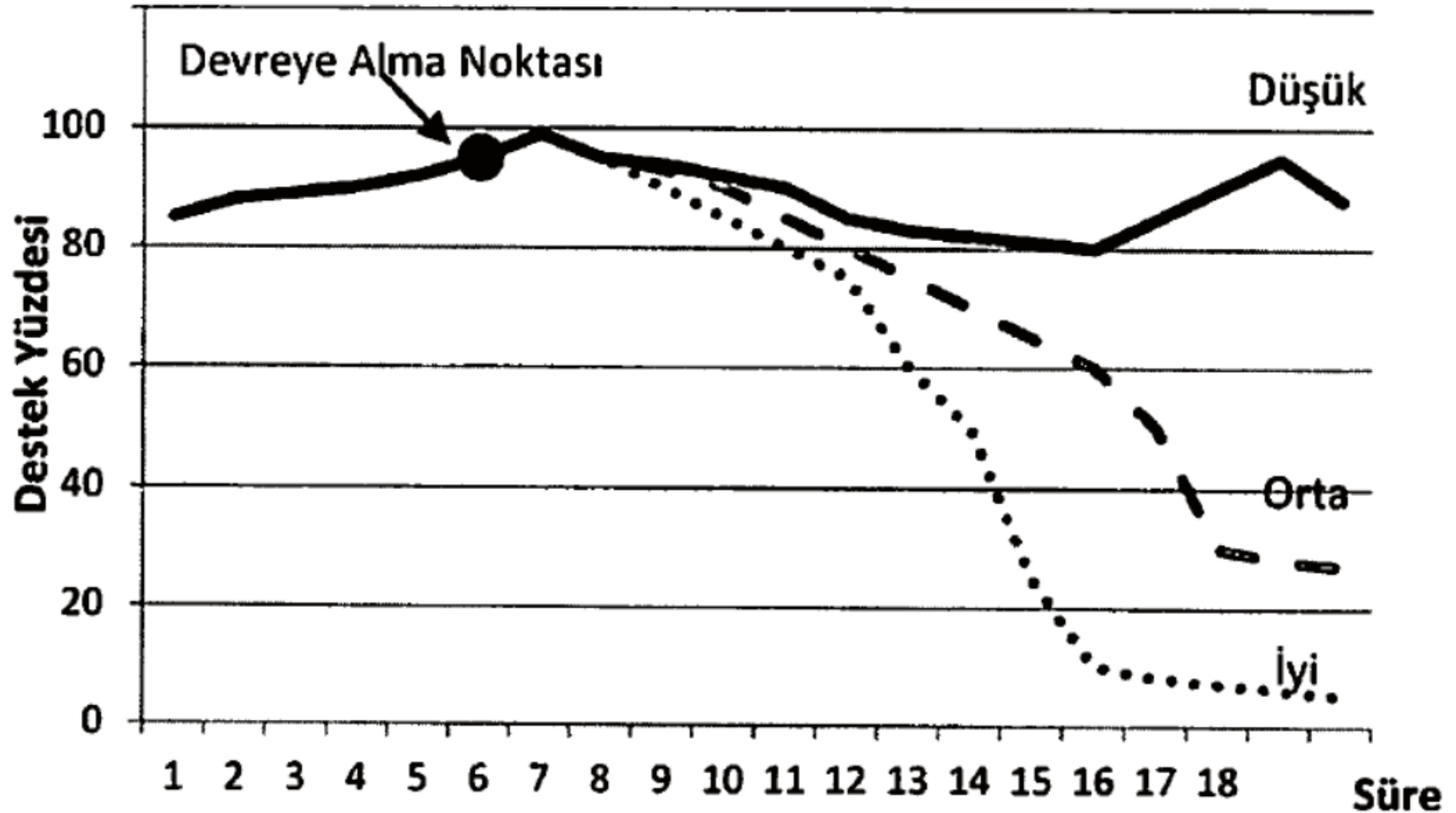


**Yazılımcıların yaptıkları projeler ve toplam destek yükleri**



# 4. Kiři ve Ekip Psikolojisi

## 2. Srekli Blnme



# 4.Kiři ve Ekip Psikolojisi

## 2. Sürekli Bölünme

Bölünmeyi aza indirmek için izlenecek yöntemlerden birisi de, bir işi bitirmeden diğeri işe başlamamaktır. Bir proje takip sistemi üzerinden işler atanıyorsa, kişinin kendisine gelen yeni işleri anlık değil belli aralıklarla kontrol etmesi sürekli bölünmeyi önler. Yeni iş gelir gelmez hemen mevcut işi bırakıp bu işe başlamak doğru değildir.

Yapılan işleri küçük mantıksal bölümlere ayırmak da bölünme etkilerini azaltacaktır. Nesne tabanlı programlama tekniğı kullanıldığı durumlarda, geliřtirmelerin metod temelli yapılması buna bir örnektir. Böylesi bir hareket, yapılan birim işlerin sürelerini azaltacağından konsantrasyon kaybının etkisini azaltır. Bir metodun gerçekleştirilmesi bitinceye kadar diğeri işlerle ilgilenmemek uygun olacaktır.

Kişilerin konsantrasyon kaybına yol açan diğeri etkenler ortak çalışma mekanı, sürekli çalan telefon ve hatanın hemen çözülmesini bekleyen müşteri olarak sıralanabilir. Kurum içi internet kullanımı, sosyal medya ve haber siteleri de önemli kesinti sebepleri olmaya başlamıştır. Bu konuda birçok kurum çeşitli seviyede kısıtlayıcı önlemler almaktadır.

# 4.Kişi ve Ekip Psikolojisi

## 3. Moral ve Motivasyon

**Moral kişilerin çalışmaktan duydukları neşe, heyecan ve enerji duygusudur.**

**Motivasyon ise kişileri işi yapmaya teşvik eden dış etkenlerdir.**

Öncelikle yüksek morale sahip bir ortam ve sonrasında iş motivasyonu hedeflenmelidir[McConneii-2004.2].

# 4.Kiři ve Ekip Psikolojisi

## 4. Ekip Yapısını Benimseme

Ekipteki alıřanların, ekibin ierisinde kendi grevini benimsemiř olması ve diğerk alıřanların unvanlarını kabul etmeleri gerekir. zellikle proje yneticisi ve diğerk yneticiler ekip tarafından kabul edilmiř olmalıdır. Zaman zaman ekipteki bazı kiřilerin proje yneticilerine tavır aldıkları grlebilmektedir. Yazılım gibi iř kalitesi ve kiřisel performansı lmenin zor olduėu bir alanda kontroln kaybına sebep olan bu durum, kabul edilemez. Bu davranıřı gsteren kiřiler iin ekip deėiřikliėi ve birim deėiřikliėi gibi tedbirler gerekebilir.

# 4.Kiři ve Ekip Psikolojisi

## 5. Ekibe Aidiyet Hissi ve İletişime Önem Verme

Yazılımcının asosyal ve içine kapanık bir kiři olarak tasvir edilmesi özellikle ölkemizde yaygın ama yanlış bir anlayıştır. Yazılımcı ekip çalışmasına yatkın olmalıdır. Bu birlikte çalışmayı sevmek, iletişime önem vermek, başarı ve başarısızlığı paylaşmaya dayanır. **İletişim bilgiye ulaşma, üretme ve ekip çalışmasının olmazsa olmaz şartıdır.** Yazılım geliştirme süreci birbirine bilgi aktaran birçok aşamadan oluşur ve ancak iletişimi sağlam bir ekiple yürütülebilir. Planlama ve planın yürütölmesi de büyük ölçüde iletişime bağılıdır. İletişim sorunu yaşayan yazılımcının kariyer gelişimi de yavaşlar. Yapacakları kişisel olarak yapabildikleri ile sınırlı kalır.

# 4.Kiři ve Ekip Psikolojisi

## 5. Ekibe Aidiyet Hissi ve İletişime Önem Verme

Bir personelin proje yöneticisine ve ekip arkadaşlarına bilgi vermemesi, işi zamanında bitirmemesinden daha büyük ve önemli bir problemdir!

Sözlü iletişim becerisi önemli olduğu kadar yazılı iletişim becerisi de önemlidir. Düşüncelerini yazıyla ifade etme becerisi, şirket üst yönetimine rapor sunmak ve müşterilere yapılacak bir teklifin hazırlanması gibi işler için en lüzumlu araçtır. Yazılım şirketi ve bilgi işlem birimi kendisini diğer insan, birim ve şirketlere yazılı veya sözlü anlatabilmelidir.

# 4.Kiři ve Ekip Psikolojisi

## 6. Tecrübe

Kiři, yazılım geliştirme tecrübesinin büyük kısmını mesleęe başladıktan sonra geçen ilk birkaç yıl içerisinde kazanır. Kazanılan alışkanlıklar genellikle bir ömür boyu sürer. Öncelikle doğru alışkanlıklar kazanmak gerekir. Yanlış alışkanlıkların sonradan düzelmesi oldukça zordur. Günümüzde bu konuda yardımcı olabilecek analiz, kod, tasarım kalitesiyle ilgili birçok standart ve kaynak eser mevcuttur.

# 4.Kiři ve Ekip Psikolojisi

## 6. Tecrübe

Alışkanlık ve tecrübe esnekliđi azaltabilir. Yazılım teknolojilerindeki sürekli gelişim esnek olmayan bir tecrübeyi dezavantaja dönüştürecektir. Komut satırlarından görsel programlamaya, yapısal programlamadan nesne ve servis tabanlı mimariye, istemci sunucu mimariden internet ve mobil teknolojilerine doğru ilerleyen gelişim çizgisinde, birçok kod satırı veya geliştirme ortamının bilinmesi gerekir. Tecrübeyi merkeze alan ancak genişlemeye müsait bir dairede hareket edilmelidir.

**Esneklik tecrübesi, kazanılması en değerli tecrübelerden birisidir.**

Proje ekibi tecrübeli ve yeni programcıların dengeli birleşiminden oluşmalıdır. Tüm ekibin tecrübeli kişilerden oluşması esnekliđi ve yeni yöntemlere uyumu azaltabilir. Ekibin yeni yazılımcılardan oluşması ve sürekli yeni şeyler denemek de çalışan bir yazılım inşasını zorlaştıracaktır.



# 4.Kişi ve Ekip Psikolojisi

## 7. Yetenek

Yazılım kişilere; dolayısıyla kişisel yetenek, beceri ve kabiliyetlere bağlı bir alandır. Planlama, karmaşık sorunları çözümü ve sistem mimarisinin oluşturulması ancak konuya hakim, yetenekli ve becerikli kişiler tarafından yapılabilecek işlerdir. Ancak maalesef bazen iyi personeli yönetmek daha zor olabilmektedir. Çünkü bağımsız karar almak, kişiyi tamamen bağımsız hareket etmeye sürükleyebilmektedir.

**Zekice (ve karmaşık) tasarımlar yapmayın. Genellikle bu tür tasarımları anlaşılması ve desteği zordur. Basit ve kolay anlaşılan tasarımları tercih edin [McConnell-2004.1 ].**

Sistemi herkesin anlayabileceği standartlar üzerine bina etmek en uygundur. Bu yenilik aramamak değildir. Elbette yeni çözüm yöntemlerini kişilerin anlaması zaman alır.

# 4.Kiři ve Ekip Psikolojisi

## 7. Yetenek

Belki řöyle de denebilir.

Standartlarda herkesin anlayabileceęi zekice geliřtirmeler yapın veya zekanızı herkesin anlayabileceęi yeni standartlar geliřtirmek için kullanın  
[Nizam-2011]

Proje yöneticisi kişisel keřiflerin önünü açarken, proje bütünlüğünün bozulmasına izin vermemelidir. Farklı çözüm önerileri, yapılacak fayda maliyet analizinden sonra planlanan süreyi aşmamak ve ekibin dięer kısmının da anlamasını sağlamak kaydıyla dikkate alınmalıdır.

# 4.Kiři ve Ekip Psikolojisi

## 8. Azim

Sürekli yeni keşiflere ve sorun çözümüne dayalı projeler geliřtirmek birçok insan için zor olabilir. Yazılımcı karşılařtığı sorunu çözene kadar arařtırmalarını azimle ve kararlılıkla sürdürmelidir. Çözemediğı bir sorun için farklı yollar denemeli, internette arařtırmalı, ekiple tartıřmalı, daha tecrübeli yazılımcılara sormalı kısaca çözüme ulaşana kadar arařtırmasına devam etmelidir. Azim teşvik edilmelidir.

İři kendi yapma isteğinin de fazlası zararlıdır. Birçok yazılımcı bir sorunla karşılařtığında, bunu kendi çözebileceğini ispatlamak ister. Kendi içine kapanıp, küçük farklarla aynı şeyi tekrar tekrar deneyerek zaman kaybeder. Ancak biliřim "Saman altından su yürütülebilecek" bir alan değildir. Çoğı sorun iletişim kurmadan çözülemez. Aslında her sorun yeni şeyler öğrenmek, yeni keşifler ve birlikte çalışmak için birer fırsattır.

# 4.Kiři ve Ekip Psikolojisi

## 9. Özel Bir İş Yapma Duygusu

Her yazılım projesi özeldir. Yazılım projesi yapan her kişinin işi de özeldir ve bu duyguyu hissetmelidir. Analiz aşaması özeldir; çünkü bir yazılım analisti kullanıcının yıllarca süren çalışmalarla kazandığı kurumsal tecrübeyi, birkaç saatlik toplantılarla anlamaya çalışmaktadır. Yazılımı tasarlayan kişi, kullanıcının kendisinin dahi tam olarak tanımlayamadığı istekleri onun memnuniyetini sağlayacak bir mimaride resmeder. Kod geliştiricisi birçok farklı bileşeni kullanarak, kendisine verilen tasarımı hatasız ve hızlı çalışacak projelere dönüştürmeye çalışır.

Geliştirilen ürünün etkinliği de yazılım geliştirmeyi özel bir alan yapar. Yazılımcıların diğer mühendislerle ortaklaşa ortaya çıkardıkları internet gibi altyapılar, tüm dünyadaki bilgi erişim şekli ve iş yapma yöntemlerini değiştirecek derecede önem kazanmıştır. Makine ve inşaat gibi mühendisliklerin yıllar önce oynadığı dönüştürücü rolünü şu anda bilgi işlem sektörü üstlenmiştir.

# 4.Kiři ve Ekip Psikolojisi

## 10. Kariyer Planlama ve Önünü Görmek

Günümüz bilgi çalışanı, kendi kariyer planını kendisi yapmak istemektedir. Bir yazılım geliştiricinin uzun süre bir şirkette devam etmesinin, yüksek moral ve motivasyona sahip olmasının birinci şartı, şirketin kendisine mesleki ilerleme ve kariyer imkanı sunmasıdır. Kariyer basamakları adım adım çıkılır.

Kiři iş hayatına genellikle kendisine verilen görevleri yaparak başlar. ilk aşamada görevi yapmak için diğer yazılımcıların yardımına ihtiyaç duyulur. Kiři daha sonra yaptığı göreve benzeyen diğer işleri yardımsız yapabilecek seviyeye gelir.

# 4.Kişi ve Ekip Psikolojisi

## 10. Kariyer Planlama ve Önünü Görmek

Kişinin görev seviyesindeki tecrübesi arttıkça, projenin temel tasarımıyla ilgili bilgi sahibi olmaya başlar. Kendisine verilen benzer bir yazılım projesinde, uzmanlığıyla ilgili karar verebilecek ve çeşitli görevler yapabilecek seviyeye gelir. Ancak bu kişi hala yeni özellikler içeren bir projeye uyum sağlamakta güçlük çekebilir[McConneii-2004.2]



**Kişisel tecrübenin gelişimi**

# 4.Kiři ve Ekip Psikolojisi

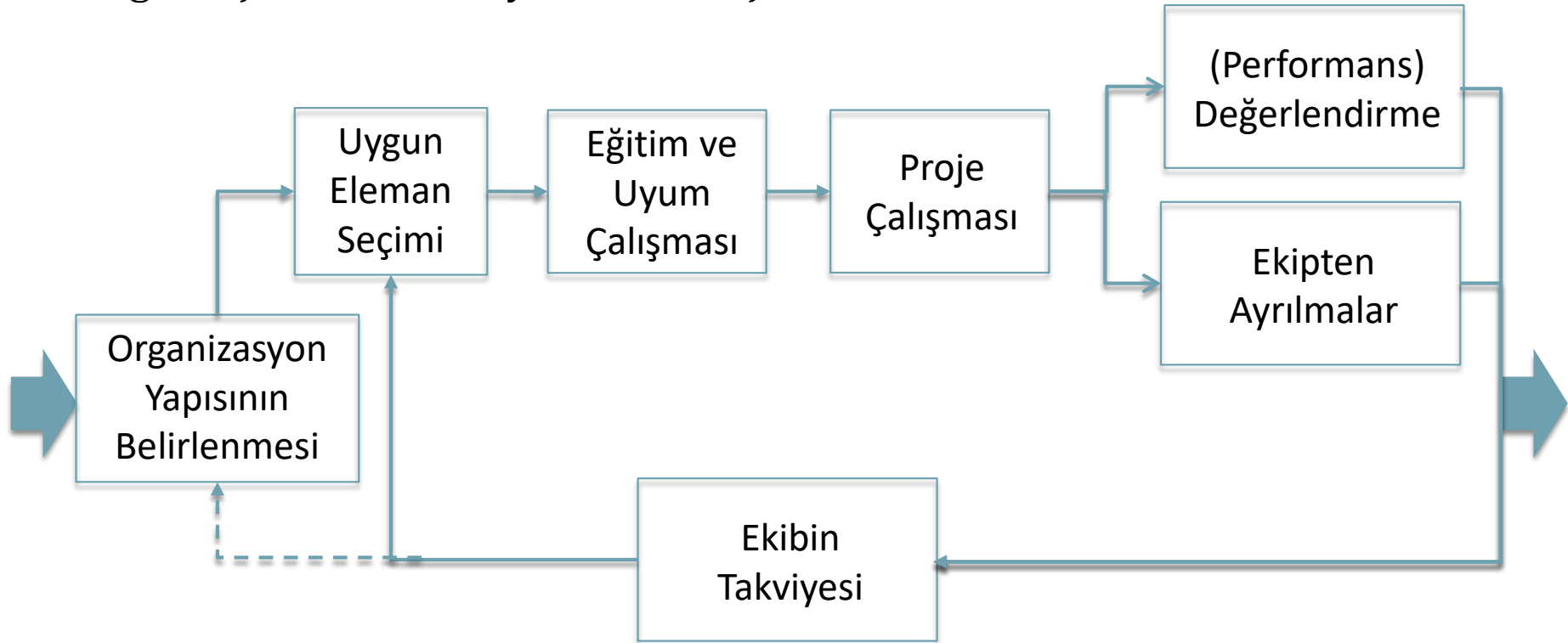
## 11. Kiřisel Sorunlar

Kiřisel sorunlar teknik gerekçelere dayanmayan fikir ayrılıkları, iře yansıyan ailevi ve insani sorunlardır. Proje yöneticisi zamanının büyük bir kısmını bu tür konular için harcamak zorunda kalmaktadır. Eğer çözüm bulunmazsa bu sorunlar önemli etkinlik ve verim kaybına yol açabilir.

Öncelikle sorunun tam olarak kiřisel olduğunun tespit edilmesi gerekir. Çünkü bu durumda izlenecek yöntem, teknik sorunlar için izlenecek yöntemden farklıdır. Kiřisel sorunların en önemli belirtisi, kiřilerin hemen hemen aynı fikri savunmalarına rağmen anlaşılamamasıdır. Teknik tartışmalar yerine kiřisel suçlamalar yapılmaya başlanır.

# 5.Ekibin Oluřturulması

Kaliteli yazılım, kaliteli ve birbirine uyumlu çalışan yazılımcılardan teşekkül eden bir ekiple gerçekleştirilir. Ekip oluřturma, ekibin olgunlařması ve takviyesi bir süreçtir.



**Yazılım ekiplerinin istihdamı**



# 5.Ekibin Oluřturulması

Proje bařlangıcında ekipteki tm kiřilerin rol, grev, sorumluluk ve iletiřim yntemleri planlanmalıdır. nce ekipte ihtiya duyulan roller ve roller arası iliřkiler belirlenir. Sonrasında bu rollere uygun kiřilerin kurum dıřından istihdamı, kurum ierisinden grevlendirilmesi veya danıřmanlıkla temini yoluna gidilir.

Personel temininden sonra proje iin gerekli eēitim ve uyum alıřmaları yrtlmelidir. Bu ařamadan sonra kiřilere projede grev verilir. Performans deēerlendirmeleri ve ekipten ayrımlar neticesinde ekibe takviye gerekebilir.

# 5.Ekibin Oluşturulması

Ekibin bileşiminde tecrübe ve yeni personel dengesi olmalıdır. Kişilerin tecrübelerine göre projelerde görevlendirme amacı değişebilir:

- **Doğrudan sorumlu:** Projeyle ilgili kendilerine atanan görevleri yapan kişilerdir. Analiz, kodlama, belgeleme gibi temel işleri yerine getirirler.
- **Yardımcı ve danışman pozisyonu:** İşin nasıl yapılacağıyla ilgili yardım alınan kişidir. Çözümü göstermek için mimari tasarımı yapar veya temel bazı kodları yazar. Ancak temel işin devamındaki rutin işleri yapmaz.
- **Eğitim:** Yeni personelin ekibe ve projeye intibak için görevlendirilmesidir. Bunlar tecrübeli ekip üyeleri nezaretinde görev yapar.

# 5.Ekibin Oluřturulması

Her yazılım projesi kendine özgü karakteristikler taşır. Bu yüzden her proje için ekibin yapısı tekrar gözden geçirilmelidir. Özellikle yeni alanlara yatırım yaparken bu çok önemlidir. Ekip proje başlangıcında yeterli sayıda ve uygun şekilde oluşturulmalıdır. Ekibe sonradan katılan yeni bir elemanın, karmaşık bir projeyi öğrenmesi ve diğer arkadaşlarını tanıması zaman alır. Bu kişi proje alanında tecrübe sahibi de olsa kurumu, ekibin diğer üyelerini tanıması ve geliştirilen projeyi öğrenmesi birkaç ay alabilir.

*Ekip oluşturmak*, genellikle proje yöneticisinin yetkisini aşar. Ancak proje yöneticisi yapacağı önerilerle uygun bir ekibin oluşturulmasında katkı sağlamalıdır. Proje yöneticisi ilk anda kendisine verilen ekiple projeyi bitirmeyi taahhüt etmişse, sonradan yeni eleman talebi üst yönetimde hoşnutsuzluğa yol açar. Bu talepler, üst yönetim tarafından yanlış giden işlere mazeret arayışı gibi anlaşılabilir. Bu yüzden elemanla ilgili talepler ilk aşamalarda yoğunlaşmalıdır.

# 5.Ekibin Oluřturulması

## 1. Organizasyon Yapısının Belirlenmesi

Bir yazılım ekibi, geliştirilecek proje yapısı ve hedeflerine göre organize edilmelidir. Ekip organizasyonu görev tanımları, görevler arası ilişkiler, dięer ekip ve birimlerle birlikte çalışma esaslarından oluşur. Önce görev tanımları ve organizasyon netleşir sonra uygun kişiler istihdam edilir. İşlemleri yapacak kişilerin ilk aşamada rol temelli belirlenmesi, sonraki aşamalarda projenin personel deęişiminden daha az etkilemesini sağlar.

**Genellikle rutin işlerin devamı için fonksiyonel, büyük çaplı projeler için matris, tamamen yeni veya farklı mekanlarda yürütülecek projeler için proje temelli örgütlenme tercih edilmektedir.** Proje organizasyonu belgelemek, organizasyon şeması oluşturmak ve yayınlamak birçok iletişim sorununu giderecektir.

# 5.Ekibin Oluřturulması

## 2. Alınabilecek En İyi Personelin İstihdamı

Personel seçimi en zor kararlardan birisidir. **Kiřiden beklenen en düşük yetenek ve bilgiyi açıklayan bir iş tanımı oluşturulmalıdır** [Whitaker-2009]. Projeye alınacak personelin teknik yeterlilikleri, problem çözme yaklaşımı, iletişim ve ekiple çalışma becerisi değerlendirilmelidir. Eğer mümkünse bir deneme süresi de öngörülebilir.

Ekibe, teknik ve sosyal olarak en fazla fayda sağlayacak kişilerin istihdam edilmesi tercih edilmelidir. Yazılımı sadece teknik bir iş olarak görmek ve eleman takviye ederken sadece bunu göz önüne almak, ekip çalışmasına uygun olmayan kişilerin görevlendirilmesine sebep olan önemli bir problemdir. Yazılım ancak birlikte çalışmayı önemseyen kişilerce geliştirilebilir.

Birçok araştırma tecrübeli ve becerikli yazılım geliştiricilerin üretkenliğinin, diğerlerine oranla 10 katı ve hatta daha fazla olduğunu göstermektedir. Elbette kalitenin bir maliyeti vardır. 10 kat üretkenliğin yanında tecrübeli yazılımcılara %50 daha fazla ücret verilmesi normal karşılanmalıdır.

# 5.Ekibin Oluřturulması

## 2. Alınabilecek En İyi Personelin İstihdamı

Tecrübesiz kişiler elinde aylarca süren bazı işlerin, tecrübeli programcılar tarafından birkaç günde tamamlandığı karşılaşılan bir durumdur. Bundan daha büyük bir tehlike de tecrübesiz kişiler elindeki işin hiç bitmeme ihtimalidir.

Kişilerin gelişen teknolojiler karşısında tutumları farklı olabilmektedir. Bazı kişiler yeni yazılımlara kolayca uyum gösterebilirken diğerleri bu konuda zorlanmaktadır. Kurumun yeni bir alana veya teknolojiye yatırım yaparak farklı proje geliştirilmesi, ancak uyum kabiliyeti yüksek personel istihdamıyla mümkün olur. Destek işlerinde ise rutin işlerde çalışmayı seven kişiler görevlendirilebilir.

# 5.Ekibin Oluřturulması

## 3. Ekipteki Kiři Sayısının Belirlenmesi – Karekk Kuralı

**Yazılım projesini istenen zamanda bitirmek iin en uygun kiři sayısını belirlemek ok nemlidir.** 120 adam/aylık bir projede 10 kiřiyle 1 senede veya 30 kiřiyle 6 ayda tamamlamak gibi farklı alternatifler olabilir. Her projede sırayla yapılması gereken işler vardır ve 120 kiři görevlendirerek 1 ayda bu projeyi tamamlamak mümkün değildir.

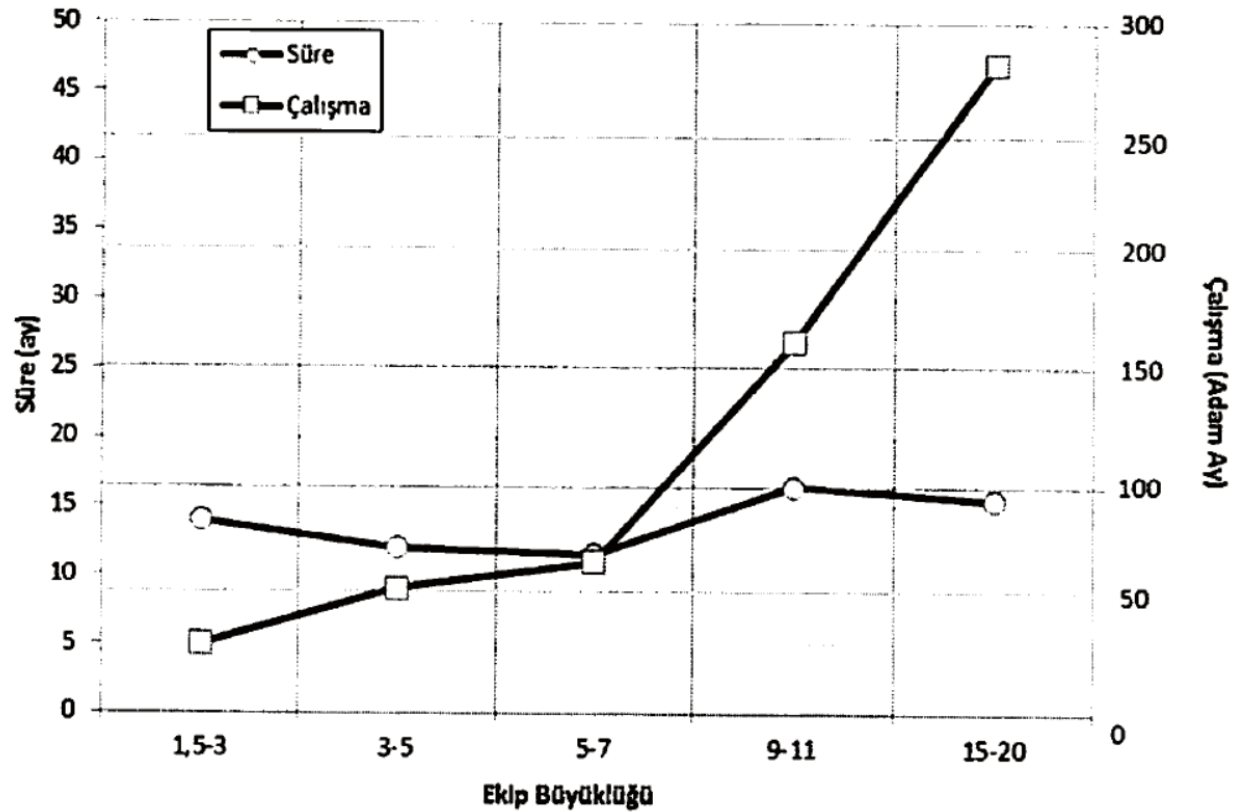
Kiři sayısının proje büyüklüğüne baėlı belirlenmesi iin karekk kuralı, toplam emeėin (adam/ay türünden) karekkünün alınması önerilmektedir. Projelerde nce proje büyüklüğü, sonra sırasıyla harcanacak toplam emek ve zaman belirlenir. rneėin toplam emek 120 adam/ay olarak belirlenmiř bir proje ( $\sqrt{120}$ ) yaklařık 11 kiři gerektirir.

Bir yazılım projesinde alıřacak kiři sayısı 6-7 kiřiyi ařmamalıdır. 6-7 kiřilik ekiplerde ekip lideri planın izleme ve sürecin doėrulama işlemlerini birebir yapabilir. Bu sayı ařıldığında ekip lideri iin koordinasyon, iletişim ve planlama güçleşir. Ekip büyürse, ekip lideri üyelere daha fazla sorumluluk vererek ara yönetim kademeleri oluřturmalı ve proje yöneticisi teknik işlerden yönetim aėırlıklı işlere doėru kaymalıdır.

# 5.Ekibin Oluřturulması

## 4. Ekip Sayısı ile Proje Süresi İliřkisi

Proje süresini kısaltmak için ekip sayısını arttırmak klasik bir çözümdür. Ancak yapılan arařtırmalar proje süresinin belli bir seviyeden fazla kısaltılmasının mümkün olmadığını göstermiştir. 35000 satır ve 95000 satır arasında 500 projede yapılan arařtırma sonuçları bu durumu göstermektedir.



**Ekip büyüklüğünün süre ve çalışmaya-emeğe etkisi**



# 5.Ekibin Oluřturulması

## 4. Ekip Sayısı ile Proje Süresi İliřkisi

Ekip sayısı ortalaması 1,5-3 kiřiden 5-7 kiřiye kadar artarken projeyi bitirmek için gerekli süre olumlu yönde azalır. Ancak ilginç řekilde ekip sayısı daha da artarsa proje süresi de ciddi oranda artacaktır. Ekip sayısıyla paralel artan iletiřim ihtiyaçı bunun en önemli sebebidir.

Süreyi azaltmak fazladan çaba gerektirir. Çünkü iřleri paralel olarak yapabilmek için aynı anda daha fazla personel çalışmalıdır. Sürenin uzaması da harcanan çabayı arttıracaktır. Çünkü kiřiler projeye daha uzun süre çalışmak zorunda kalır. Proje yöneticisi en uygun süre-kiři sayısı dengesine ulaşmayı hedeflemelidir.

# 5.Ekibin Oluřturulması

## 5. Ekibin Sürekli Takviyesi

Ekibin ilk başta doğru kurulması yeterli değildir. Deęişen şartlar, performans değerlendirmeleri ve ekipten ayrılmalar karşısında ekibe gerekli takviyeler yapılmalıdır. Yazılım karmaşık ve birçok parçadan oluşan bir üründür. Daha önce benzer projelerde bulunmuş tecrübeli kişilerin dahi yeni bir projeye uyum sağlaması zaman alır. Bu yüzden ekibe yeni katılan bir kişinin projede alacağı ilk görevlerde eğitim ve intibak süresi düşünölmelidir.

Büyük çapta kurumsal projelerin bir kuruma uyarlaması, en başta geniş çaplı bir ekip tarafından yapılmakta sonrasında bu sayı azalmaktadır. Ancak sayı azaltılırken çok dikkatli davranılmalı ve özellikle proje süreklilięi için önemli tecrübeli personelin istihdamına devam edilmelidir.

# ÖZET

Ekip olarak çalışabilmek büyük yazılım projelerinin vazgeçilmez şartıdır. İyi bir ekip, ortak bir hedef etrafında kenetlenmiş kişilerden müteşekkildir. Başarılı bir proje için mümkün olan en tecrübeli kişiler istihdam edilerek uyumlu bir ekip oluşturulmalıdır.

Yazılım projeleri, birbiriyle ilişkili birçok farklı görevin koordineli bir şekilde yapılmasıyla gerçekleştirilir. Bu görevler belli unvanlarla standartlaşmıştır. Kodlama, analist, tasarım, test gibi alanlarda uzmanlıklar gelişmiştir. Bu uzmanlıklar yazılım geliştirme süreci içinde anlam kazanır. Ekibin her elemanı süreçteki yerini bilmelidir.

Proje yöneticisi ekibin birlikte çalışması, iletişim, koordinasyon, moral ve motivasyonunun artmasında aktif rol oynamalıdır. Ekibi tanımalı, çalışanların sürekli bölünmesini mümkün olduğunca engellemeli ve çalışanlara özel bir iş yapma duygusunu hissettirmelidir. Ekip hedefleri ve kişisel kariyer hedefleri arasında bir denge kurulmalıdır.

# ÖZET

Kaliteli yazılım, kaliteli ve birbirine uyumlu çalışan yazılımcılarla gerçekleştirilir. Proje yöneticisi, bu konuda yapacağı önerilerle uygun bir ekibin oluşturulmasında öncülük eder. Proje için yeterli sayı ve tecrübede eleman ayrılmalıdır. Ekip sayısı ve tecrübesi, proje süresini olumlu ya da olumsuz etkiler. Sayı artışı olumludur ancak belli bir noktadan sonra iletişim ihtiyacını çok fazla arttırarak proje süresinin uzamasına sebep olabilir. Ekip oluşturduktan sonra çalışmalar sürekli değerlendirilmeli ve gerekirse ekip takviye edilmelidir.