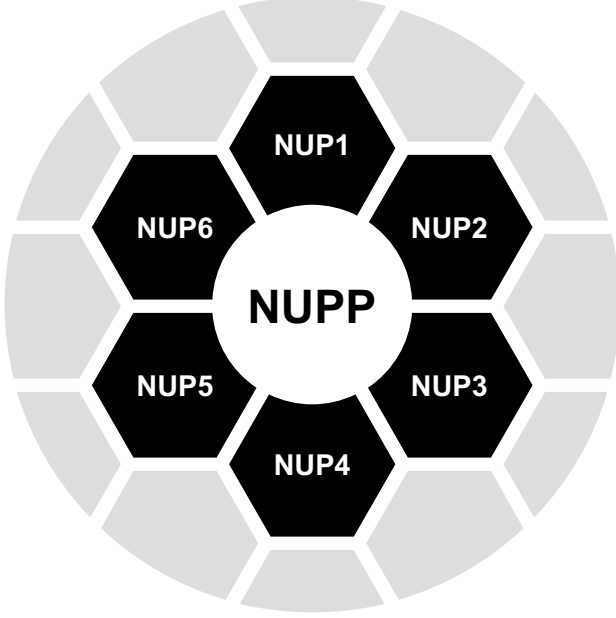


# NUPP

Projelerin Neredeyse Evrensel İlkeleri



Bu, 2026-07-02 tarihinde oluşturulmuş çevrimiçi kılavuzun (<https://omimo.org/tr/modules/nupp/>) indirilebilir sürümüdür. Daha yeni sürümler için web sitesine bakın.

NUPP, açık ve minimalist modüllerden oluşan bir aile olan OMIMO'dan (<https://omimo.org/tr/>) gelir.

Bu kılavuz, Creative Commons Attribution 4.0 International lisansı kapsamında serbestçe kullanılabilir ve dağıtılabilir.

Co-funded by the  
European Union



OMIMO, Avrupa Birliği tarafından eş finansmanla desteklenmektedir. Ancak burada ifade edilen görüş ve düşünceler yalnızca OMIMO'ya aittir ve Avrupa Birliği'nin veya EPOS VZW'nin görüşlerini ve düşüncelerini yansıtmak zorunda değildir. Bu nedenle ne Avrupa Birliği ne de hibeyi sağlayan kuruluş bu görüş ve düşüncelerden sorumlu tutulamaz.

Çeviren: [Ünsal Atasoy](#)

## giriş

---

NUPP, neredeyse evrensel proje prensiplerinden oluşan bir koleksiyondur: Başarımızı en üst düzeye çıkarmak için kullandığımız metodolojilere ve yaklaşımlara bakılmaksızın, tüm projelerde takip ettiklerimizdir.

Projeleri yürütmek için mevcut kaynakların ve yöntemlerin her biri, bu NUP'lardan bazılarına dayanır (neredeyse evrensel prensipler). Ancak, aşağıdaki noktaların akılda tutulması gerekir:

- Uygulayanların bir alt küme yerine tüm NUP'leri dikkate almaları yararlı olacaktır.
- Temel ilkeler genellikle kaynaklarda ve yöntemlerde yeterince açık hale getirilmez ve çoğu uygulayıcı pratik ayrıntılarla o kadar meşguldür ki, ilkeleri unutur ve onlarla uyumlu olmayan şeyler yapar.

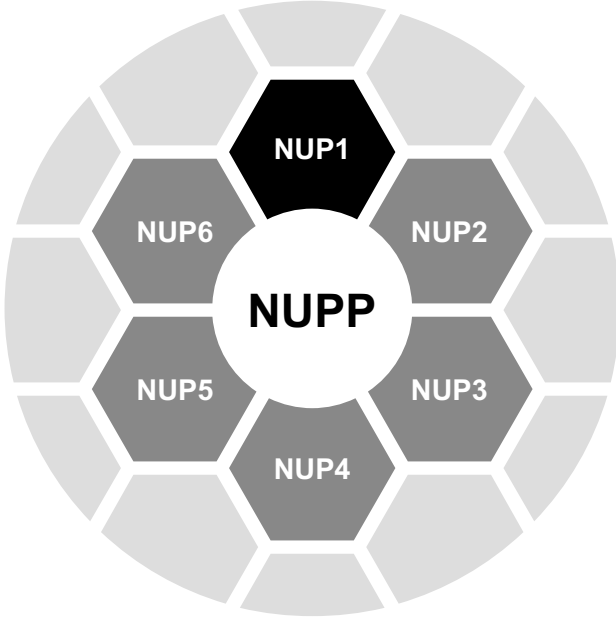
NUPP, PRINCE2®, PMBOK® Kılavuzu, P3.express, PM<sup>2</sup>, DSDM®, XP ve Scrum gibi tüm ana yöntemler, sistemler, kaynaklar ve çerçevelerle uyumludur. Yine de bu sistemlerin belirli yorumlarıyla/versiyonlarıyla uyumlu olmayabilir ve NUPP'in uygulayıcıları kendi yorumlarını yeniden gözden geçirmeye teşvik etmeye çalıştığı yer burasıdır.

NUPP, aşağıdaki NUP'lerin bir koleksiyonudur:

- [NUP1 — Sonuçları ve gerçekleri bağlılıklara tercih edin](#)
- [NUP2 — Enerji ve kaynakları koruyun ve optimize edin](#)
- [NUP3 — Her zaman proaktif olmak](#)
- [NUP4 — Bir zincirin ancak en zayıf halkası kadar güçlü olduğunu unutmayın](#)
- [NUP5 — Açık bir amaç olmadan hiçbir şey yapmayın](#)
- [NUP6 — Tekrarlanabilir öğeler kullanın](#)

## NUP1

### Sonuçları ve gerçekleri bağlılıklara tercih edin



Hepimizin gruplara ait olma konusunda doğal bir eğilimi vardır. Bu eğilim genellikle temel biçiminin ötesine geçer, güçlü bağlantılar yaratır ve sorunlara neden olur. İlişkilerden dolayı kazandığımızdan çok daha fazlasını kaybederiz. Kimliğimizi ve tercihlerimizi belirli gruplarla sınırlamazsak daha profesyonel ve etkili uzmanlar haline gelebiliriz.

#### Örnek: Çevik ve Şelale

Tahmine dayalı yaklaşımları kullanmanın norm olduğu bir zamanda, BT geliştirmede adaptif geliştirme yaklaşımlarını deneyecek kadar cesur olan, oldukça hevesli bir grup insan bir araya geldi ve yaklaşımlarını “Çevik” olarak adlandırdı. Seçenekleri, gerekli görünenlerle sınırlamamak için bu harika bir girişimdi. Çevik topluluğunda hala çok hevesli ve sonuç odaklı insan var, ancak maalesef bu toplulukta Çevik’i bir tarikata dönüştüren ve tüm dışarıdakileri düşman olarak gören bazı insanlar da var. Bu, aşağıdakiler dahil olmak üzere, birçok şekilde soruna neden olur:

- Kendi gruplarının dışından bir şeyler öğrenmelerine izin vermez
- Dışarıdakileri, kendilerinden öğrenmelerinden caydırır
- Gruba ait olmayı gerçek amaçtan daha önemli hale getirir ve bu da birçok üyesinin Çevikliğin gerçek anlamını öğrenmesini engeller.

Bu sorun, tamamen ortadan kaldırılmasa bile, üyeler içeren bir topluluktan ziyade geliştirme yaklaşımını kast eden bir etiket olarak “Çevik” kullanılarak; ve Çevik’i kimliklerinden ziyade basitçe etkinleştirici bir altın bilezik olarak gören, kendilerini yaratıcı, problem çözücü ve lider olarak gören insanlara sahip olarak önemli ölçüde azaltılabilir.

Gerçek profesyoneller için Çevik-Şelale savaşı yoktur.

## **Örnek: PRINCE2® ve PMBOK® Kılavuzu**

Proje yönetim topluluğunda, kendilerini PRINCE2® veya PMBOK® Guide (genellikle coğrafi konumları nedeniyle) ile ilişkilendiren ve diğerine aşına olmayan birçok profesyonel vardır. Hepimizin belirli kaynaklara yönelik tercihleri olabilir, fakat kimliğimiz olarak değil. Daha da önemlisi, bakış açımızı ve seçeneklerimizi genişletmek için hepsine alışmalıyız.

Gerçek profesyonel, tüm fikirlere açıktır, fikirleri arar, hakkında bilgi edinir ve herhangi bir bağlılık olmaksızın gerektiği gibi ve gerektiğinde kullanırlar.

## **Örnek: Sürekli öğrenme**

İlişkiler, oluşturdukları aidiyet duygusuyla kişiyi tatmin eder, ancak öğrenmeye zorlamaz ve hatta bazen kaybetme korkusuyla öğrenmekten vazgeçirir. Özgür bir kişi, bağları olmayan bir uzman, olduğunuzda boşluğu öğrenmeyle: yani sürekli öğrenmeyle doldurmanız gerekir.

Bugün inandığımız şey gerçek değil, sadece şu ana kadarki en iyi anlayışımız ve ilerledikçe iyileştirilmesi gerekiyor. Birinin fikirleri birkaç yıl öncekiyle tamamen aynıysa, yanlış bir şeyler vardır. Bu, NUPP için bile geçerlidir: Birkaç yıl sonra geri gelirsiniz ve aynı şeyi görürseniz, şüphelenmelisiniz.

## **Örnek: Açıklık**

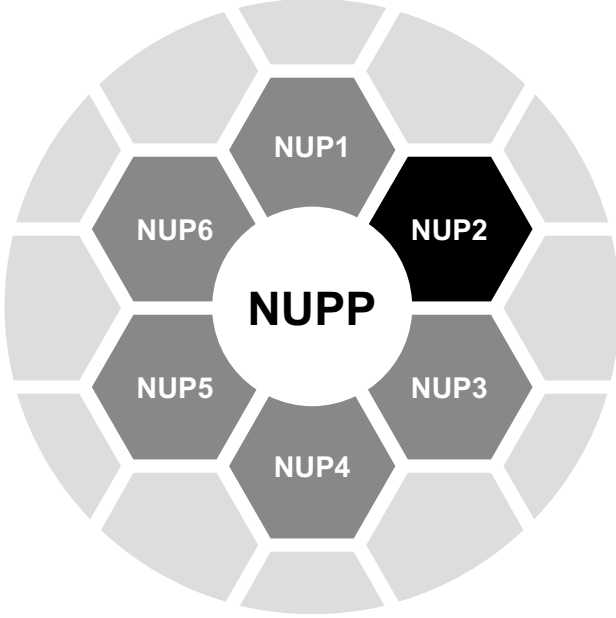
Birine itiraz ederken, itirazınızda kişiye değil fikre itirazı amaçladığınızdan emin olun. Bu, birçok gerilimi önlemeye yardımcı olur. Benzer şekilde, birisi size yada sizin hakkında itiraz ettiğinde, bunu size karşı bir savaş olarak yorumlamaya çalışmayın, daha çok fikirleriniz üzerine bir tartışma yapın ve ona açık olun. Cevap vermek için dinlemeyin, anlamak için dinleyin; ve fikri/konuyu geliştirmek için diğer kişiyle birlikte çalışın.

Bazı insanlar kasıtlı olarak fikir yerine sizi hedef alabilir, bu durumda, devam etmeden önce sizin yerinize fikre odaklanmalarına yardım etmeli ve konuşma boyunca bu şekilde tutmaya çalışmalısınız.

## NUP2

### Enerji ve kaynakları koruyun ve optimize edin

---



Kaynaklar sınırlıdır. İyi kararlar vermek zorunda olduğunuz zihinsel enerji gibi, proje için mevcut kaynaklar da sınırlıdır. Bu kaynağı kendiniz ve proje için korumalı ve optimize etmeli ve diğer ekip üyelerinin de aynısını yapmasına yardımcı olmalısınız.

#### Örnek: 80/20 kuralı

Proje yönetiminin olası faydalarının büyük bir kısmı, çabanın küçük bir kısmının harcanmasıyla elde edilebilir. Çoğu durumda, olası faydaların% 100'ünü hedeflemek çok pahalıdır ve gereksizdir. Yaptığınız her şeyde bu kuralı göz önünde bulundurmanız ve başkalarını da aynısını yapmaya teşvik etmeniz gerekir.

#### Örnek: Karar yorgunluğu

Her türlü kararı vermek ve iradeyi ifade etmek için tek bir zihinsel enerji kaynağı kullanıyoruz. Gereksiz veya önemsiz kararlar almak için bu kaynağın çoğunu kullanırsanız, önemli kararlar için daha az enerjiniz olur ve bu da kötü sonuçlara yol açabilir. Mikro yönetimden kaçınmanız gereken nedenlerden biri budur (PRINCE2®'nin "istisna ile yönetme" ilkesi).

Fikirlerle ilgili çatışmalar faydalı olabilir, ancak insanlarla ilgili çatışmalar genellikle projeye zarar verir ve sonuçlarından biri de ekip üyelerinin zihinsel enerjisini tüketmesidir. Böyle bir çatışma fark ederseniz, kök nedeni bulmak ve çözmek için elinizden gelenin en iyisini yapmalısınız.

#### Örnek: Kendinize iyi bakın!

Verdiğiniz kararlar ve ifade ettiğiniz irade, zihinsel enerji kaynağınızı tüketir. Öte yandan bu kaynak, düzenli uyuduğunuzda ve yemek yediğinizde enerji ile dolar. Bu nedenle, kendinize iyi bakmalısınız: yeterince uyuduğunuzdan, dinlendiğinizden ve iyi beslendiğinizden emin olun. Zararlı alışkanlıklarınız veya uyku ile ilgili sorunlarınız varsa, bununla tek başınıza uğraşmak zorunda değilsiniz; Bu tür

sorunları çözenize yardımcı olabilecek birçok uzman var. Bu konudaki çalışmalar henüz kesinleşmiş olmasa da, fiziksel aktivite de bu enerji kaynağına yardımcı olabilir.

Ekip üyelerini sizin yaptığınız gibi yapmaya teşvik etmeye çalışın. Öncelikle, sürdürülebilir bir hızda ve fazla mesai yapmadan çalıştıklarından emin olun. Ardından, eğer imkanınız varsa, çalışma saatlerinde enerjik, sağlıklı yiyecekler, atıştırmalıklar ve içecekler sunmaya çalışın.

### **Örnek: İş-yaşam dengesi**

Çoğumuz yaptığımız işi seviyoruz ama yine de hayatta sahip olmamız gereken her şey bu değil. İşinizi optimize ettiğiniz gibi, kişisel yaşamınızı neşeli ve mutlu kılacak şekilde fikirleri planlamalı ve uygulamalısınız. Daha mutlu olduğunuzda, işte de daha başarılı olabilirsiniz. Mümkünse, ekip üyelerinizi de aynısını yapmaya teşvik etmeye çalışın.

### **Örnek: Motivasyon**

Motivasyon karmaşık bir kavramdır. Konuyla ilgili bazı ilginç ve faydalı kaynakların yanı sıra, birçok bilim dışı kaynak da var. Yinede, ekibin zihinsel enerjisini ve proje için daha iyi sonuçlar elde etme olasılığını artırdığı için, bunları öğrenmeli ve sürekli kullanmalısınız.

Motivasyon, insanlara iyi çalışmalarını farkettiğinizi nazik bir gülümsemeye veya basit bir “teşekkür ederim” ile bildirmek kadar basit olabilir. Ancak, küçük parasal ödüller gibi pek çok yaygın motivasyon biçiminin olumsuz etkiye sahip olduğu konusunda dikkatli olmalısınız.

### **Örnek: İşbirliği ve takım çalışması**

İşbirliği yapan insanlar bazen daha iyi sonuçlar yaratma gücüne sahip olabilir, fakat daha da önemlisi, insanlar sosyaldır ve bir grubun parçası olmaktan zevk alır. Ekip çalışmasının olumsuz yönlerini ortadan kaldırabilir ve hoş bir ortam yaratabilirseniz, projede daha mutlu ekip üyeleri olacaktır.

Yine de dikkatli olmalısınız, çünkü insanlar farklıdır ve bazılarının diğerlerinden daha rahat, odaklanmış ve yalnız kalmaya ihtiyacı vardır; genellikle dengelenmesi gerekecektir.

### **Örnek: Tek takım kültürü**

Farklı paydaşların alt gruplar oluşturma veya bunları dikkate alma ve diğer gruplarla gerginliğe neden olma eğilimi vardır; örnek olarak, projenin iş yönlerine odaklanan kişilerle ürünü oluşturan kişiler verilebilir. Bu gerilim her iki taraftan da çok fazla enerji tüketir, bu herkesin aynı hedef için birlikte çalıştığı tek takım kültürü oluşturmaya çalışmamızın nedenlerinden biridir.

### **Örnek: Kalabalıkların bilgeliği**

Çeşitliliğe sahip çok sayıda insan bir araya geldiğinde ve kolaylaştırılmış bir ortamda çalıştığında, çok iyi sonuçlar, fikirler ve çözümler elde etme potansiyeli vardır, bu tek bir uzmandan gelenlerden daha iyi olabilir.

Fırsatınız varsa, ekip üyelerinden projedeki zor sorunları çözenize yardımcı olmalarını istemeyi düzenli olarak kullanabilirsiniz. İyi sonuçlar alma olasılığının yanı sıra, ekip üyelerinin fikirlerinin takdir edildiğini ve projede önemli bir rol oynadıklarını bilmelerini sağlar ve bu da enerji seviyelerini artırır. P3.express'in E02 numaralı etkinlik, projelerde kalabalıkların bilgeliğini kullanmanın bir örneğidir.

### **Örnek: Baş proje kolaylaştırıcısı**

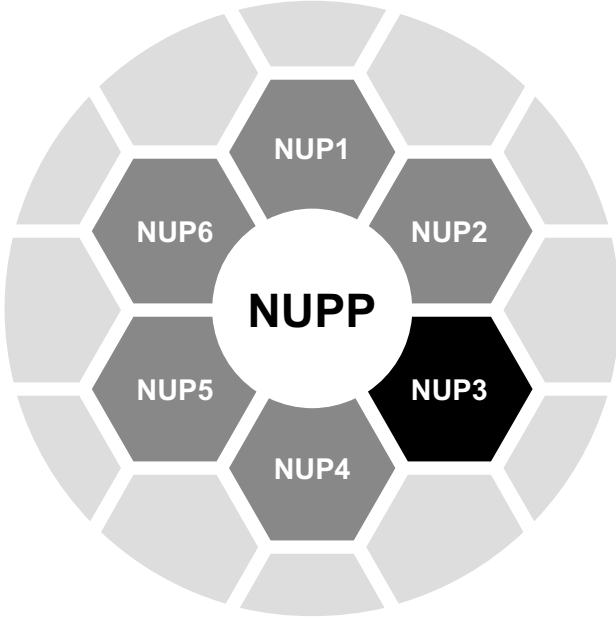
Eğer bir proje yöneticisi iseniz, yaptığınız şeylerin çoğu kolaylaştırıcı niteliktedir (veya en azından olmalıdır). Öte yandan, ekip üyelerinin geçmişte proje yöneticileriyle kötü deneyimleri olduğunu ve bu deneyimlerin sizinle ilişkilerini etkilediğini görebilirsiniz: enerjilerinin bir kısmını size güvenmek yerine

potansiyel tehditler için davranışınızı analiz etmeye harcarlar. Bu durumda, ünvanınızı proje yöneticisinden Baş Proje Kolaylaştırıcısı olarak değiştirebilirsiniz. Sonuçta, projede gerçekten yaptığınız şey budur.

## NUP3

### Her zaman proaktif olmak

---



İçimizde reaktif olma yönünde doğal bir eğilim vardır. Bu eğilim, önemsiz konularla uğraşırken enerjimizi korumamıza yardımcı olabilir veya tamamen yetersiz olduğumuz bir şeyle uğraşırken bize daha iyi sonuçlar verebilir. Bu durumlar projelerimizden farklıdır ve burada proaktif olarak daha iyi sonuçlar alabiliriz.

#### Örnek: Planlama

Yeni bir yere gitmek istiyorsanız ve geç kaldıysanız, “zamandan tasarruf etmek” için hemen araba kullanmaya başlayabilir ve ortaya çıkan olası sorunlarla başa çıkabilirsiniz. Proaktif yaklaşım, başlangıçta biraz zaman ayırarak navigasyon sisteminizi, trafiğe, olası kazalara ve tıkanmalara göre size en hızlı rotayı verecek şekilde ayarlamak ve ardından sürmektir; Bu, daha sonraki sorunlardan kaçınmak ve böylece sonunda zamandan tasarruf etmek için direkt uygulamadan önce zaman harcamaktır.

Bazı insanların çevik projeler hakkında düşündüklerinin aksine, planlama her zaman gereklidir. Yalnızca planlardaki ayrıntıların türü ve düzeyi farklıdır. Uygulamadan önce planlama, proaktif bir yaklaşımdır.

Lincon'dan şu alıntıyı hatırlayın: “bir ağacı kesmem için bana altı saat verin, ilk dördünü baltayı keskinleştirmek için harcarım.”

Tahmine dayalı bir proje ise, 4 saatinizi baltanızı keskinleştirmek için harcayabilirsiniz, çünkü bir ağacı keseceğinizden eminsiniz. Çevik bir projede, bir ağacı kesip kesmeyeceğinizden, kırık dalları toplayacağınızdan, çim biçeceğinizden, kömür çıkaracağınızdan veya başka bir şeyden emin değilsiniz. Yine de tüm bunlar için genel bir hazırlığa (en yakın nalburun nerede olduğunu bilmeye) ve belirli bir çözüme odaklanacağınız zaman özel bir hazırlığa (keskinleştirmeye) ihtiyacınız var; bu planlamadır.

## Örnek: Planlamanın planlanması

Projeyi yürütme şeklimizi planlamak proaktif bir yaklaşımdır. Bu proaktivite, yürütmeyi nasıl planlayacağımızı planlayarak bile genişletilebilir; bu, PMBOK® Kılavuzunun yönetim planı kavramı, PRINCE2®'nin yönetim stratejileri ve DSDM®'deki yaklaşımlardır.

## Örnek: Sürekli planlama

Gerçek, planladığımızla nadiren örtüşür ve bu doğaldır, ancak gerçekçi ve pratik kalmalarını sağlamak için planlarımızı sürekli olarak adapte etmeliyiz. Bunu, bir sorunla karşılaştığımızda değil, adaptasyona ihtiyaç duyulduğu anda yapmalıyız. Bu proaktif bir yaklaşımdır.

## Örnek: Risk yönetimi

Tüm risk yönetimi kavramı proaktiviteye dayanır: Belirsiz olaylarla karşılaştığımızda, ne olacağını görmek için beklemek ve sonra onlara tepki vermek yerine, olasılıklar ve etkiler hakkında düşünür, yanıtları düşünür ve muhtemelen gerçekleşmeden önce bununla ilgili bir şeyler yaparız.

Projelerde yaptığımızın ciddi olduğunu unutmayın; bu bazen insanların hayatları ile ilgilidir.

## Örnek: Roller ve sorumlulukları tanımlayın

Proje ekibi üyelerini net roller ve sorumluluklar olmadan çalışmaya bırakabilirsiniz. Er ya da geç bir tür rol ve sorumluluk ortaya çıkacaktır, ancak bu yöntem çok maliyetlidir ve sonuçta pek işe yaramayabilir. Proaktif yaklaşım, rol ve sorumlulukları erken tanımlamak ve gerektiğinde uyarlamaktır. Böyle yapmak, herkes için çalışmayı kolaylaştırır ve kimin ne yapacağına karar vermek yerine üretmeye odaklanabilir.

Rollerin sayısı ve çeşitliliği projenin türüne ve büyüklüğüne bağlıdır; Scrum'daki gibi basit bir tanım, P3.express'deki gibi makul bir tanım veya DSDM® ve PRINCE2®'deki gibi kapsamlı bir tanım olabilir. Ancak, bu yöntemlerdeki rol tanımlarının yalnızca yönetim faaliyetleriyle ilgili olduğunu ve her zaman teknik yönler için de rol açıklamaları eklemeniz gerektiğini unutmayın.

## Örnek: Mevcut seçenekler

Projeyi erken mi kapatmalısınız yoksa devam mı etmelisiniz?

Soru bunu ima etse bile, nadiren sadece iki seçenek vardır. Karar vermeden önce proaktif bir yaklaşıma sahip olmanız ve tüm seçimlerinizi gözden geçirmeniz gerekir. Belki projeyi yeniden gözden geçirebilirsiniz; belki başka bir şey netleşene kadar duraklatabilirsiniz; veya belki proje yaklaşımını değiştirebilirsiniz (örneğin, dış kaynak kullanımı), vb.

## Örnek: Eleştirel düşünme

Hepimizin, bir yandan hayatta kalmamıza yardımcı olan, diğer yandan da kötü kararlar vermemizi sağlayan, birçok önyargısı var. Proje hakkında önemli kararlar vermek söz konusu olduğunda, bir süre ara vermek ve kararımızı etkileyebilecek tüm önyargıları sorunlara neden olmadan önce değerlendirmek en iyisidir.

Referans olarak [Wikipedia](#)'da verilen bilişsel önyargıların listesini kullanabilirsiniz.

Daha iyi kararlar almak için kullanabileceğiniz karar verme çerçeveleri bile var. İlk başta, onları kullanmak dikkat dağıtıcı ve hatta can sıkıcı olabilir, ancak kısa sürede bunlara alışır ve fazla bilinçli bir çaba harcamadan onlardan avantaj elde edersiniz.

## Örnek: Şeffaflık

Projeye geç kalmayı veya başka bir problem yaşamayı sevmeyiz, ancak bu onu saklamamız gerektiği

anlamına gelmez. Şeffaf olmalısınız ve paydaşların bilmesini sağlamalısınız, çünkü onlardan bazıları size yardımcı olabilir. Ve dahası, sorunları ve sonuçlarını er ya da geç öğrenecekler. Bunlardan bazıları kendi taraflarından erken eylemler gerektirebilir (örn: olumsuz sonucu kabul etmek).

### **Örnek: Etkili iletişim kurun**

Paydaşlara rapor gönderdiğiniz ve size herhangi bir geri bildirim verilmeyen birçok durum olabilir. Öyle olmadığı halde, olumsuz geri bildirim olmadığı için her şeyin yolunda olduğuna inanabilirsiniz. Proaktif olmanız ve raporu gerçekten kullanıp kullanmadıklarını ve amaca hizmet edip etmediğini kontrol etmeniz ve bu girdiyi iletişim yönteminizi ayarlamak için kullanmanız gerekir; aksi takdirde, bu gizli sorun daha sonra düzeltilmesi çok zor olduğunda ciddi sorunlara neden olabilir.

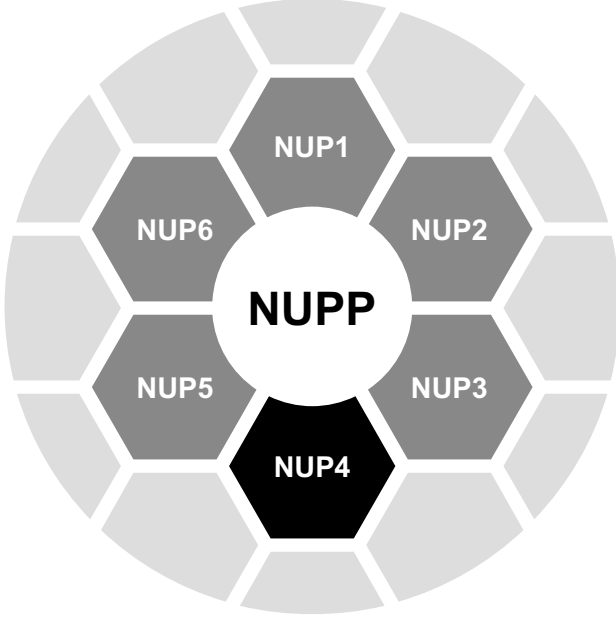
### **Örnek: Sorumluluk alın**

Kötü sonuçlar için başkalarını suçlamak kolaydır. Örneğin, kuruluşunuzun size projedeki her şeyi değiştirip mükemmel şekilde yapmanız için tam yetki vermesini isteyebilirsiniz, ancak vermezler ve sonuç olarak proje başarısız olur. Bu proaktif bir yaklaşım değildir.

Proaktif yaklaşım, sorumluluk almak ve kısıtlamalar dahilinde yapabileceğiniz her şeyi yapmaktır. Kuruluşun size tamamen güvenmesini ve iyi sonuçlar elde etme umuduyla her şeyi vermesini bekleyemezsiniz, özellikle de çok sayıda başarısız proje gördüklerinde. Yapmanız gereken şey, belirlenen kısıtlamalar dahilinde küçük bir iyileştirme yapmak, bunu biraz güven kazanmak, birkaç kaynak ve kısıtlamalarda biraz daha fazla esneklik sağlamak için kullanmak ve sonra bunu biraz daha büyük bir iyileştirme için kullanmak şeklinde optimum hedefe ulaşana kadar devam etmektir.

## NUP4

Bir zincirin ancak en zayıf halkası kadar güçlü olduğunu unutmayın



Projelerde çeşitli alanlar vardır ve hepsinin dikkate alınması gerekir; projenin bütüncül bir perspektifine sahip olmalıyız. Görünüşte önemli bir alana (ör. Zaman) dikkat etmek yeterli değildir, çünkü tüm alanlar etkileşimde bulunur ve hepsi yeterli ilgi görmedikçe düzgün çalışmazlar.

### Örnek: Her şey terminle ilgili!

Olimpiyat Oyunları için bir şeyler inşa ettiğinizi varsayalım. Zaman yönetimini çok önemli kılan, çok ciddi bir teslim tarihine sahiptir. Peki bu doğru mu? Ya kalite tekrar çalışmayı gerektirecek kadar düşükse. Bu, zamanı etkileyecektir, bu da zamanı ve kalite anlamına gelir. Projenin orjinal tanımında listelenmiş süslü bir bahçeniz olabilir ama bilirsiniz ki, eğer yeterli zaman yoksa, -bu olasılığı zamanında düşündüğünüz ve hazırladığınız sürece-, onu atlayıp sadece çim ile örtebilirsiniz. Bu nedenle kapsam yönetimi de önemlidir. Bu durumda, dikkatimizin merkezinde kapsam, zaman ve kalite alanlarına sahip oluruz.

Başkan Kennedy'nin NASA'da bir kapıcıyla tanıştığı ve ona ne yaptığını sorduğu ve kapıcının ise ona "Bir adamı aya götürmesine yardım ediyorum" diye yanıt verdiği o ünlü örneği duydunuz mu? Projede bu tür insanların olması son teslim tarihini tutturmanıza yardımcı olmaz mı?

Proje devam ederken, projedeki her bir alanın zaman yönetimine katkıda bulunduğunu ve tüm alanları dikkate almadığınız sürece, kabul edilebilir bir kesinlik düzeyiyle son teslim tarihini karşılayamayacağınızı fark edersiniz.

### Örnek: Cımbızlayarak toplama

İnsanlar çeşitli yöntemlerle karşı karşıya kaldıklarında, bazen cımbızlayarak toplamaya başlarlar ve farklı sistemlerden ilginç görünen her şeyin bir karışımını oluştururlar. Bu genellikle işe yaramaz çünkü öğeler tek başlarına çalışmazlar ve birbirleriyle uyumlu olmaları gerekir. Bir sisteme yapılacak herhangi bir ekleme veya değişiklik bütüncül bir bakış açısıyla yapılmalıdır.

Bu nedenle bazen farklı yöntemlerde çelişkili unsurlar görüyoruz; bazı şeyler bir sistemde iyi çalışır ve tersi de başka bir sistemde iyi çalışır. Bir unsur tek başına doğru ya da yanlış değildir.

En güvenli yaklaşım, proje için bir metodoloji seçmek, onu uyarlamak ve ardından tüm sistemin tutarlılığını göz önünde bulundurarak dikkatlice yeni unsurları eklemektir.

### **Örnek: Süreç karşıtı yaklaşım**

Çevik yöntemlerin yaptığı en iyi şeylerden biri dikkatleri insana çekmektir. Çevik Manifesto, süreçlere ve araçlara kıyasla bireylere ve etkileşimlere daha fazla değer verir, ancak bu adil bir karşılaştırma olmayabilir. Hemen hemen tüm yöntemler, insanın önemli olduğunu söyler. Çevik yöntemlerle gerçek fark, bunun basit bir öneriden ziyade süreçlerinin yerleşik bir parçası olmasıdır. Yani, insani yön ve süreçler arasındaki bir rekabetle değil, daha çok insani yönlerinin sistemde görülme şekli ile ilgilidir.

Hiç şüphe yok ki, bazı insanlar daha sofistike süreçlerle insani yönleri değiştirmeye çalışıyor, ama bu sadece yanlış bir kullanım. Bunun tersi bile var: Süreçleri insan yönleriyle değiştirmeye çalışan insanlar, ki bu da pek işe yaramıyor.

### **Örnek: Bunlar ihtiyacınız olan tüm alanlardır.**

Alanları düşünürken hiçbirini kaçırmamaya dikkat etmelisiniz. Peki, nedir bunlar? Temel kaynakları kontrol ederseniz, farklı cevaplar alırsınız; ve yine de hiçbirisi tek başına doğru değildir.

PRINCE2® temaları alanlardır, ancak bunlar yalnızca metodolojide anahtar rol oynayan alanlardır. Diğer alanlar yalnızca ima edilmektedir.

PMBOK® Kılavuzu bir metodoloji değildir ve alanları on bilgi alanıyla çok daha iyi formüle edebilir. Ancak bunlar, tarafsız bir perspektif yerine PMBOK® Kılavuzunun proje perspektifine dayalı olarak tüm alanların yorumlarıdır (mutlaka tarafsız bir perspektif olduğu anlamına gelmez). Örneğin, insani yönler PMBOK® Kılavuzunda çok fazla ilgi görmez.

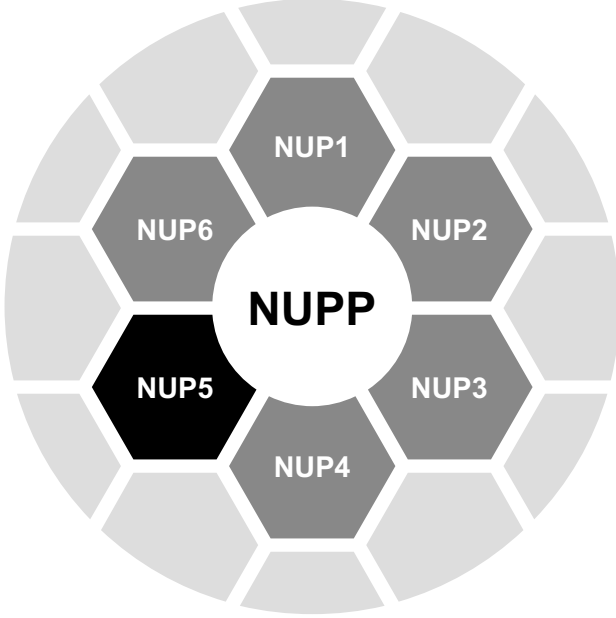
Alanlar hakkında iyi bir bilgi kaynağı ICB'dir. Ancak alanlarla ilgili değil, projede gerekli olan yetkinliklerle ilgilidir. Alanlarla bire bir ilişkisi yoktur, ancak onları tanımlamada çok yardımcı olur.

NUPP'de bir alan listesi yoktur, çünkü öncelikle bir sistemden ziyade bir meta-sistemdir ve ayrıca alanların sınıflandırılması projenin türüne ve ortamına bağlıdır; Örneğin, rutin bir inşaat projesi, yaratıcı bir araştırma projesinden farklı bir bakış açısına ihtiyaç duyabilir.

## NUP5

### Açık bir amaç olmadan hiçbir şey yapmayın

---



Açık bir amacı olmadıkça hiçbir şey yapmamalısınız. Yapmayı düşündüğünüz şey dışında her şeyin aynı olduğu iki paralel dünya hayal edin: Bu dünyalar ne kadar farklı olurdu? Fark, o şeyi yapmak için harcanan çabaya değer mi?

Aklınızda net bir amacınız yoksa ve sadece herkes yaptığı için yapıyorsanız veya herkes bunu yapmanın önemli olduğunu söylediği için yapıyorsanız, o zaman

- sizin durumunuzda gerçek bir yararı olmayabilir ve bu nedenle bundan hiçbir şey elde edemeyebilirsiniz; veya
- sizin durumunuzda yine de potansiyel faydaları olabilir, ancak aklınızda bir amaç olmadığı için, bunu yapma şekliniz faydaların farkına varmanıza yardımcı olmayabilir.

#### Örnek: Portföyler ve programlar

Projelerin seçilmesi ve başlatılmasıyla ilgileniyorsanız, üründen çok, faydalara ve çözülmesi gereken sorunlara odaklandığınızdan emin olmalısınız. Klasik örnek, asansörlerinin hızıyla ilgili şikayetler alan ve bu nedenle tam bir başarı olmadan hızı artırmak için birden fazla proje yürüten asansör üreticisidir. Bu sorun, “doğal” çözüm (daha hızlı asansörler) yerine soruna (insanların can sıkıntısı veya rahatsızlığı) odaklanmaya karar verene kadar devam etti. Sonuç olarak asansörlere aynalar eklediler ve problemi basit ve ucuzca çözdüler.

Proje yönetiminin temelde işleri doğru yapmakla ilgili olduğunu, portföy yönetiminin ise doğru şeyleri yapmak olduğunu unutmayın. Yanlış projeleri yapıyorsanız, projelerinizi ne kadar iyi yürüttüğünüz önemli değildir. Her şey bir amaca sahip olmakla ilgilidir.

#### Örnek: Bir bütün olarak proje

Ürünün esnekliği projeden projeye değişiklik göstermektedir. Bazı BT geliştirme projelerinde ürün

tamamen esnektir ve ürünün nihai şekli, uyarlanabilir (Çevik) bir yaklaşım gerektiren, proje sırasındaki artışların oluşturacağı geri bildirimlere bağlıdır. Bu, pratik anlamda portföy, program ve proje katmanlarının bir birleşimidir ve nihai hedefe azami dikkat gösterilmesini gerektirir. Amacı belgelemek ve erişilebilir kılmak iyi bir fikirdir; bazı Scrum projelerinde kullanıldığı şekliyle “ürün vizyonu”nun amaçlarından biridir. Çevik projelerde iş değerine odaklanmak, amaçla hizalanmayı garantilemenin yoludur.

Ürünün nispeten sabit olduğu ve belirlenen ürünün amacı karşılamasını sağlayacak başka mekanizmaların bulunduğu diğer proje türlerinde, proje ekibi üyelerinin odaklarının büyük bir bölümünü amaçtan ürüne kaydırmaları mümkündür ( dolayısıyla PRINCE2®’nin “ürünlere odaklanma” ilkesi). Ancak inşa edilenin amacı karşıladığından emin olmak için amaca en azından asgari düzeyde dikkat edilmesi gerekir, bu da öngörülen faydalar ile beklenen faydalar karşılaştırılarak yapılabilir ( dolayısıyla PRINCE2®’deki “sürekli iş gerekçesi” ilkesi ve “iş durumu” teması).

Harici müşteri için bir proje yürütüldüğünde, müşterinin proje için kendi amacı ve şirketinizin farklı bir amacı olur. Gerçekten başarılı olmak için her ikisini de anlamalı ve kullanmalısınız.

## Örnek: Proje izleme

Nihai amaca odaklanmak gereklidir, ancak günlük hayatta çoğu için çok soyut olabilir. Bu nedenle, daha pratik hale getirmek için bir amaç/motivasyon hiyerarşisine ihtiyaç vardır. İlk olarak, projenin gerekçesi ve faydaları nihai amaca dayalı olarak tanımlanır ve ardından gerekçeyi karşılamak için proje değişkenleri (örn., zaman, maliyet ve kalite) için açık ve örtülü hedefleriniz olacaktır ve sonuç olarak nihai amaç sağlanacaktır. Bunlar, günlük görevlerimizin çoğu için yararlı olan alt düzey amaçlardır.

İzleme söz konusu olduğunda, nihai amacı izlemek mümkün olmayabileceğinden, projenin düşük seviyeli izlemesi en düşük seviyedeki değişkenler kullanılarak yapılacaktır. Bu durumda, yine de amaçları aklınızda tutmalısınız: İzlemenin amacı nedir?

Normal ve kabul edilebilir bir cevap, doğru yolda olup olmadığımızı görmek ve değilse, bizi tekrar yola getirecek veya hedefleri ayarlayacak belirli bir tepkiyi tetiklemek ve nihai amacı hala karşılayıp karşılayamayacağını görmektir. Bu nedenle ölçümlerimiz bu düşük seviyeli amaca yardımcı olmalıdır ve uygun ölçümler, nihai değişkenler için tahminlerdir; örneğin, projeyi ne zaman bitirebiliriz? Projeyi bitirmek için ne kadar paraya ihtiyacımız olacak?

Bugüne kadar planlanan ve gerçekleşen değerler gibi diğer tüm ölçümler, yönetim amacıyla kullandığınız nihai değerler değil, tahminleri hesaplamak için ihtiyaç duyduğunuz ara değerlerdir.

## Örnek: Belgeler

Projede hangi geliştirme yaklaşımını kullanırsanız kullanın, planlama her zaman gereklidir. Asıl soru, her bir plan türündeki ayrıntı düzeyidir. Planda yeterli ayrıntı yoksa, plan yeterince katkıda bulunamayacak ve projenin yürütülmesi entegre, bütünsel kararlardan ziyade geçici kararlara dayanacaktır. Öte yandan, birçok insan dikkatli olmaya ve planlarına çok fazla ayrıntı eklemeye çalışır, bu da hazırlamayı ve bakımı çok zorlaştırır ve projenin belirsizlikleri için çok katıdır. Sonuç olarak, plan gerçek dışı ve işe yaramaz hale gelir.

Ayrıntı düzeyine karar vermenin en iyi yolu, plandaki her öge için bir amacı göz önünde bulundurmadır. Örneğin, plana kaynak eklemeyi düşünüyorsanız, bunun için bir amacınız olmalı: Nasıl kullanacaksınız? Projeye nasıl yardımcı olacak? Ne kadar çaba gerektirecek? ve buna değer mi?

Bu, bazen planlarınızda belirli bir ögeye sahip olmak isteyip istemediğinize, bazen de bir şeyi nasıl planlamak veya hazırlamak istediğinize karar vermekle ilgilidir. İster bir iş vakası, ister bir proje başlatma belgesi veya bir rapor olsun: yine de kendinize bu belgeyi neden hazırladığınızı ve bunun

projeye nasıl yardımcı olabileceğini sormalısınız.

Bir “şablon” aramak, bir amaca dayalı bir şey yapmanın tam tersidir.

### **Örnek: Durum raporlama**

Birçok projede gerçekten uzun durum raporlarına sahip olmak yaygındır. Bu NUP’a dayanarak, kendimize raporun amacının ne olduğunu ve diğer insanların çoğu ne yapıyor olursa olsun bu amaca nasıl ulaşabileceğimizi sormalıyız.

Bu, çoğu durumda, uzun raporlar yerine, paydaşların gerçekten okuyup anladığı çok basit tek sayfalık raporlar hazırlamamıza neden olabilir. Bu, amaca dikkat etmektir.

Ancak tek sayfalık raporlar hazırlarsanız, bazı kişiler size itiraz edebilir ve “uygun” bir izleme sisteminizin olmadığını düşünebilir. Pratikte bu sizin için ikinci bir amaç yaratır (paydaşların projenin durumunu anlamalarına yardımcı olan birinci amacın yanında) ve bunu tatmin etmek için çok uzun ikinci bir rapor türü oluşturabilirsiniz. Ancak, bunu ilk raporla karıştırmamalısınız, çünkü bu iki amaç aynı değildir.

### **Örnek: İş gerekçesi ve proje başlatma belgesi**

İş gerekçesi ve proje başlatma belgesi gibi belgeler hazırlamak çoğu insan için genellikle sıkıcı, sonuçsuz, bürokratik bir gereklilik iken, bu belgelerin hepsinin çoğu proje için geçerli olan amaçları vardır. Bir “şablon” bulmaya çalışırsanız ve onu doldurursanız, iş boşa gider; oysa ki bunun yerine bu belgelerin gerçek amacını kontrol edebilir, projenize nasıl yardımcı olabileceklerini görebilir ve ardından belgeyi istediğiniz şekilde hazırlayabilirsiniz, sadece bu amaçları yerine getirmek için: bu doğru belge olacaktır.

Belgeleri amaçlarına uygun şekilde nasıl hazırlayabileceğinizi düşünürken, her senaryoyu düşünmeyebilir ve bir şeyleri kaçırabilirsiniz. Bunu önlemek için PRINCE2®, PMBOK® Guide, P3.express ve DSDM® gibi kaynakların ne önerdiğini kontrol edebilir ve ardından bu önerileri amaçlara göre değerlendirebilirsiniz.

### **Örnek: Proje sonrası**

Projenin belirli bir amacı olsa da, bu amacı proje boyunca göz önünde bulundurur ve amacın gerçekleşmesi esas olarak proje sırasında yapılan tahminlere göre değerlendiririz. Ancak proje bittiğinde bu amacı unutmamalıyız. Proje bittikten sonra hedeflerin gerçekleşip gerçekleşmediğini kontrol etmek önemlidir, çünkü

- nihai hedeflere nasıl ulaşıldığını görebilir ve bundan gelecekteki projeler için öğrenebiliriz ve
- bazen hedeflere ancak projenin tamamlanmasından sonra bazı ek görevler gerçekleştirdiğimizde ulaşılır ve bu ekstra görevlerin gerekliliğini anlamak, izlemeyi gerektirir.

Proje sonrası izleme, amaca yönelik olmak için gerekli bir adımdır. Bu, esas olarak program ve portföy yönetim sistemlerinin sorumluluğundadır ve çoğu şirketin bu tür yönetim katmanları olmadığı için bu adım genellikle ihmal edilir. Bu nedenle P3.express ve DSDM® gibi yöntemler proje yönetimi metodolojilerine bu adımı ekler.

### **Örnek: Sadelik**

Dünya karmaşık ve kaotiktir, ancak modellerimiz dünyanın parçalarını yansıtan soyut yaklaşımlardır ve bu nedenle basit olabilirler. Öte yandan, basit sistemler genellikle daha iyi çalışır çünkü ana fikre odaklanabiliriz.

Çoğumuz, pratikte çalışmayı zorlaştırırsa ve genellikle amacı yerine getirmemizi engellese de, dünyanın

karmaşıklıđıyla eŖleŖeceđini umarak sistemlerimize karmaşıklık ekleyerek daha iyi sonular elde etmeye alıŖırız.

### **Örnek: Uyarlama**

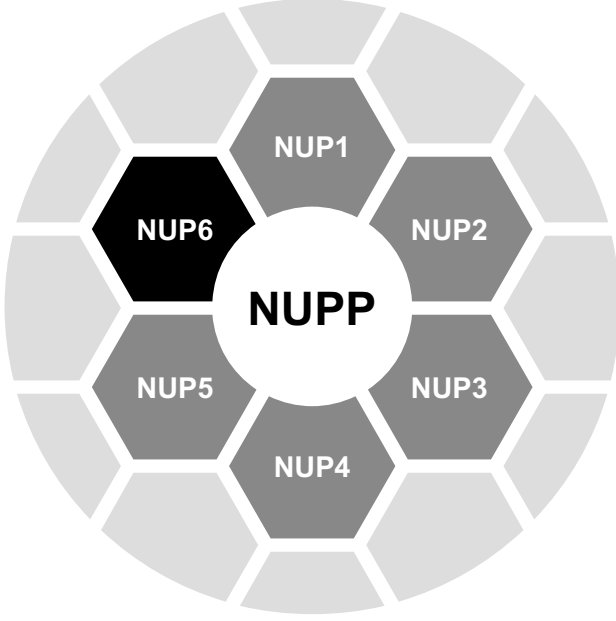
Müzik yayın akıŖı için tasarlanmış bir yazılım, insanların hayatlarının buna bađlı olabileceđi bir hastanede veya uakta ekipman için kullanılacak yazılımdan veya yıllarca ökmeden alıŖması gereken bir uyduda kullanılacak yazılımdan ok farklı bir koŖula sahiptir. Ve bunların hepsi bir yazlık villa, bir itfaiye istasyonu ve bir iŖleme tesisi inŖa etmekten farklıdır.

Amaları göz önünde bulundurduğunuzda, sistemleri ve yapıları farklı projeler için nasıl uyarlayacağınızı daha iyi anlarsınız.

## NUP6

### Tekrarlanabilir öğeler kullanın

---



Projeye özel bir yaklaşım çok fazla enerji ve kaynak gerektirir ve her zaman gerekli unsurların bazılarını kaçırma riskini taşır. Yapılması gerekeni basitleştirmenin en iyi yolu, tekrarlanabilir öğeler kullanmak ve tercihen bunları tekrarlanabilir döngülerde ele almaktır.

#### Örnek: Kalite kontrol listeleri

Bir kontrol listesi, birçok insanın kişisel ve profesyonel yaşamlarında kullandığı potansiyel olarak tekrarlanabilir bir öğenin basit bir örneğidir. Örneğin: Bir teslimatın kalite kriterlerini ele alın:

- İlk olarak, bir planlama şekli olan tüm kriterlerin bir kontrol listesini oluşturabilirsiniz.
- NUP6'nın önerdiği şey, bunu genelleştirmeye çalışmaktır: projede benzer başka çıktılar var mı? Bu durumda, o çıktı kategorisi için genel bir kalite kontrol listesi hazırlayın ve bunu hepsi için kullanın. Bazı varyasyonlar varsa, genel listeyi saklayın ve bireysel teslimatlar için birkaç ekstra öğe ekleyin. Artık tekrarlanabilir kontrol listelerine sahipsiniz.
- Çeşitli teslimat türleri için genel kontrol listeleri hazırladıktan sonra, bunlar arasında tekrar eden ve onlar için sanal bir ana kategori öneren öğeler bulabilirsiniz. Bu durumda, tüm bu genel kontrol listeleri için maddeleri tekrarlamak yerine, onları çıkarabilir ve bir ana kontrol listesine koyabilirsiniz. Sonunda, muhtemelen tüm proje için tek bir genel kontrol listeniz olacak. Scrum'ın "bitti tanımı", kalite için proje düzeyinde kontrol listelerinin kullanılmasına bir örnektir (diğer şeylerin yanı sıra mümkündür). Bunu yaptığınızda, her teslimat bir kategori hiyerarşisine ait olacaktır ve zincirlerindeki tüm kategorilerin kontrol listelerinde görünen öğeleri karşılamalıdır.

Bunu yaparak, ana kontrol listesindeki bir öğe, altındaki tüm çıktılar için tekrarlanabilir hale gelecek ve bu da planlama ve uygulamada zaman ve enerji tasarrufu sağlayacaktır.

Daha da önemlisi, bunu bir kez bir proje için yaptığınızda, bunu gelecekteki tüm benzer projeler için

uyarlayabilir ve kullanabilirsiniz; bu, birden fazla proje için tekrarlanabilir bir planlama şeklidir.

### **Örnek: Süreçler ve iş akışları**

Bazı çıktılar veya bunlarla bağlantılı hedefler, standart hale getirilebilecek ve tekrarlanabilir hale gelebilecek belirli adımlara ihtiyaç duyar. Örneğin, çıktıların bireysel olarak tasarlanması ve onaylanması gerekiyorsa, tüm adımları, ilgili kişileri ve yaklaşık süreleri netleştiren basit bir iş akışı hazırlayabilir, böylece birçok zorluktan kaçınabilirsiniz. Bununla birlikte, olumsuz bir sonucu olacağından, iş akışlarını ve süreçleri çok karmaşık veya dokümantasyonda çok yoğun hale getirmemeye dikkat etmelisiniz. Projeye dahil olan tüm kişiler, iş akışlarını ve süreçleri, gerçek işlerini engelleyen bürokratik belgeler olarak değil, işlerini destekleyen ve onlar için her şeyi kolaylaştıran bir şey olarak görmelidir.

Çevik projeler, her özellik için belirli geliştirme faaliyetlerinin tekrarlandığı yinelemeli geliştirme yaklaşımlarında tekrarlanabilir unsurlara sahiptir; Örneğin. XP'deki genel günlük rutin (eXtreme Programlama): eşleştirin, bir öge seçin, beyaz tahtada tasarlayın, test komut dosyalarını ve kodu oluşturun, kodu entegre edin, vb.

Teknik faaliyetler için kullanılacak tekrarlanabilir iş akışlarının yanı sıra proje yönetimi faaliyetleri için de tekrarlanabilir unsurlara sahip olabilirsiniz. PMBOK® Guide, PRINCE2® ve DSDM®'deki süreçler, P3.express'teki aktiviteler ve Scrum'daki olaylar bu kavramın örnekleridir.

### **Örnek: Döngüler**

Projeyi yönetmek için tekrarlanabilir ögelere sahip olmak faydalıdır. Bu, onları tekrarlanabilir döngülere koyarak daha da kolay hale getirilebilir. Bu döngüler, projenin yönetimine ve liderliğine dahil olan kişilerin günlük faaliyetlerini önemli ölçüde basitleştirir. Birden fazla aşamaya sahip bir projede kullanıldığında PMBOK® Kılavuzundaki süreç gruplarının döngüleri, PRINCE2®'de aşamalar, P3.express'te günlük, haftalık ve aylık döngüler, DSDM®'de yinelemeler ve zaman kutuları ve Scrum'da Sprint'lerin tümü bu kavrama örnektir.

Daha kısa döngüleri anlamak ve kullanmak, uzun döngülere göre daha kolaydır; örneğin, PMBOK® Kılavuzuna göre aşamaların aksine Scrum'da Sprintler. Ancak, çok kısa döngüler belirli proje türlerini desteklemek için yeterli olmayabilir ve çözüm; DSDM®'nin daha uzun yineleme döngüleriyle birlikte kısa zaman kutusu döngülerini kullanması veya P3.express günlük, haftalık ve aylık döngüler gibi birden çok döngünün kullanılması olabilir.

### **Örnek: Yöntemler**

Bir projeyi yürütmek için bir metodoloji veya çerçeve kullanmak, tekrarlanabilir öğelerin başka bir kullanımınıdır. Bu, PRINCE2®, P3.express, DSDM® veya Scrum gibi mevcut bir sistem veya sizin özelleştirdiğiniz veya kendiniz kurduğunuz bir sistem olabilir. Ancak, sıfırdan inşa etmektense, normalde mevcut yöntemlerden biriyle başlamak ve onu ihtiyaçlarınıza uyarlamak daha iyi bir fikirdir.

Herhangi bir tekrarlanabilir öge soyuttur ve onu gerçek dünyaya uyarlamak için özelleştirmeye ihtiyaç duyar. Yine de bir soyutlama yelpazesine ve özelleştirmeye ihtiyaç var: Spektrumun bir ucunda en az soyutlama ve uyarlama ihtiyacı olan küçük, nispeten somut kalite kontrol listeleri bulunurken, metodolojiler diğer uçta en yüksek uyarlama ihtiyacına sahiptir. Her zaman uyarlama ihtiyacına dikkat etmelisiniz, aksi takdirde, tekrarlanabilir öge ihtiyaçlarınızı tam olarak karşılamayacaktır