

# Gerçekleşmek

Başarılı bir web uygulaması oluşturmanın daha akıllı, daha hızlı ve daha kolay yolu

Basecamp tarafından

# İçindekiler

Bölüm 1: Gerçek Olan Ne? \_\_\_\_\_

Bölüm 2: Basecamp Hakkında \_\_\_\_\_

Bölüm 3: Uyarılar, sorumluluk reddi beyanları ve diğer önleyici uyarılar \_\_\_\_\_

4. Bölüm: Daha Az İnşa Edin \_\_\_\_\_

Bölüm 5: Sorununuz Ne? \_\_\_\_\_

Bölüm 6: Kendiniz Finanse Edin \_\_\_\_\_

Bölüm 7: Zamanı ve Bütçeyi Düzeltin, Esnek Kapsam \_\_\_\_\_

Bölüm 8: Bir Düşmanınız Olsun \_\_\_\_\_

9. Bölüm: Bir Angarya Olmamalı \_\_\_\_\_

Bölüm 10: Daha Az Kütle \_\_\_\_\_

Bölüm 11: Değişim Maliyetinizi Düşürün \_\_\_\_\_

Bölüm 12: Üç Silahşör \_\_\_\_\_

Bölüm 13: Kısıtlamaları Kucaklayın \_\_\_\_\_

Bölüm 14: Kendiniz Olun \_\_\_\_\_

---

## Öncelikler

Bölüm 15: Büyük Fikir Nedir? \_\_\_\_\_

Bölüm 16: Ayrıntıları Erkenden Yoksay \_\_\_\_\_

Bölüm 17: Sorun Olduğunda Sorun Olur \_\_\_\_\_

Bölüm 18: Doğru Müşterileri İşe Alın \_\_\_\_\_

Bölüm 19: Daha Sonra Ölçeklendir

Bölüm 20: Fikir Amaçlı Yazılım Yapın \_\_\_\_\_

---

## Öznitelik Seçimi

Bölüm 21: Yarım, Yarım Değil \_\_\_\_\_

Bölüm 22: Sadece Önemli Değil \_\_\_\_\_

Bölüm 23: Hayır ile Başlayın \_\_\_\_\_

Bölüm 24: Gizli Maliyetler \_\_\_\_\_

Bölüm 25: Başa Çıkabilir misin? \_\_\_\_\_

Bölüm 26: İnsan Çözümleri \_\_\_\_\_

Bölüm 27: Özellik İsteklerini Unutun \_\_\_\_\_

Bölüm 28: Mayo'yu Tutun \_\_\_\_\_

---

## İşlem

Bölüm 29: Yazılımı Çalıştırma Yarışı \_\_\_\_\_

Bölüm 30: Durulayın ve Tekrarlayın \_\_\_\_\_

Bölüm 31: Fikirden Uygulamaya \_\_\_\_\_

Bölüm 32: Tercihlerden Kaçının \_\_\_\_\_

Bölüm 33: "Bitti!"

Bölüm 34: Vahşi Doğada Test

Bölüm 35: Zamanını Küçült

---

## Organizasyon

Bölüm 36: Birlik

Bölüm 37: Yalnız Zaman

Bölüm 38: Toplantılar Zehirlidir

Bölüm 39: Küçük Zaferleri Arayın ve Kutlayın

---

## personel

Bölüm 40: Daha Az İşe Alın ve Daha Sonra İşe Alın

Bölüm 41: Lastikleri Tekmele

Bölüm 42: Sözler Değil Eylemler

Bölüm 43: Çok Yönlü Bireyler Olun

Bölüm 44: Sahte Coşku Yapamazsınız

Bölüm 45: Sözcüler

---

## Arayüz tasarımı

Bölüm 46: Önce Arayüz

Bölüm 47: Merkez Üs Tasarımı

Bölüm 48: Üç Devletli Çözüm

Bölüm 49: Boş Sayfa

Bölüm 50: Savunmaya Geçin

Bölüm 51: Tutarlılık Üzerine Bağlam

Bölüm 52: Metin Yazarlığı Arayüz Tasarımıdır

Bölüm 53: Bir Arayüz

---

## kod

Bölüm 54: Daha Az Yazılım

Bölüm 55: Mutluluk İçin Optimize Edin

Bölüm 56: Kod Konuşur

Bölüm 57: Borç Yönetimi

Bölüm 58: Açık Kapılar

---

## Kelimeler

Bölüm 59: İşlevsel Bir Özelliğin İşlevsel Hiçbir Şeyi Yok

Bölüm 60: Ölü Belgeler Yapma

Bölüm 61: Bana Kısa Bir Hikaye Anlatın

Bölüm 62: Gerçek Kelimeleri Kullan

Bölüm 63: Ürününüzü Kişiselleştirin

---

## Fiyatlandırma ve Kayıt

Bölüm 64: Ücretsiz Örnekler

Bölüm 65: Kolay Açma, Kolay Kapatma

Bölüm 66: Aptal Tavşan, Hileler Çocuklar İçindir

Bölüm 67: Daha Yumuşak Bir Kurşun

---

## Terfi

Bölüm 68: Hollywood Lansmanı

Bölüm 69: Güçlü Bir Promosyon Sitesi

Bölüm 70: Blog Dalgasına Binin

Bölüm 71: Erken Talep Etme

Bölüm 72: Eğitim Yoluyla Teşvik Edin

Bölüm 73: Öne Çıkan Yiyecekler

Bölüm 74: Günlüklerinizi Takip Edin

Bölüm 75: Sıralı Ek Satış

Bölüm 76: İsim Kancası

---

## Destek

Bölüm 77: Acıyı Hisset

Bölüm 78: Sıfır Eğitim

Bölüm 79: Hızlı Cevap

Bölüm 80: Zor Aşk

Bölüm 81: Güzel Forumda

Bölüm 82: Sıkıntılarınızı Yayınlayın

---

## Lansman Sonrası

Bölüm 83: Bir Aylık Ayarlama

Bölüm 84: Gönderilerin Gelmesine Devam Edin

Bölüm 85: Beta Değil, Daha İyi

Bölüm 86: Tüm Hatalar Eşit Yaratılmamıştır

Bölüm 87: Fırtınadan Kurtul

Bölüm 88: Jones'lara ayak uydurun

Bölüm 89: Şişkinlik Canavarı'na Dikkat Edin

Bölüm 90: Akışla Devam Edin

---

## Çözüm

Bölüm 91: Motorlarınızı Çalıştırın

# 1: Gerçek Olan Ne?

Başarılı bir web uygulaması oluşturmak ister misiniz? O zaman Gerçek Olma zamanı. Gerçek Olmak, yazılım oluşturmanın daha küçük, daha hızlı ve daha iyi bir yoludur.

- Gerçeğe Ulaşmak, gerçeği temsil eden her şeyi (tablolar, grafikler, kutular, oklar, şemalar, tel kafesler, vb.) atlamak ve gerçek olanı inşa etmekle ilgilidir.
- Gerçekleşmek daha azdır. Daha az kütle, daha az yazılım, daha az özellik, daha az evrak işi, gerekli olmayan her şeyden daha az (ve gerekli olduğunu düşündüğünüz şeylerin çoğu aslında değildir).
- Gerçekleşmek küçük kalmak ve çevik olmaktır.
- Gerçekleşmek, insanların kullanacağı gerçek ekranlar olan arayüzle başlar. Müşterinin gerçekte deneyimlediği şeyle başlar ve oradan geriye doğru inşa eder. Bu, yazılımı yanlış yapmadan hemen önce arayüzü almanızı sağlar.
- Gerçekleşmek, yinelemeler ve değişimin maliyetini düşürmekle ilgilidir. Gerçekleşmek tamamen başlatma, ince ayar ve sürekli iyileştirme ile ilgilidir, bu da onu web tabanlı yazılım için mükemmel bir yaklaşım haline getirir.
- Gerçekleşmek, müşterilerin ihtiyaç duyduğu her şeyi sağlar ve ihtiyaç duymadıkları her şeyi ortadan kaldırır.

## Gerçek Olmanın Faydaları

Gerçekleşmek daha iyi sonuçlar verir çünkü sizi bu problemler hakkındaki fikirleriniz yerine çözmeye çalıştığınız gerçek problemlerle uğraşmaya zorlar. Sizi gerçeklerle yüzleşmeye zorlar.

Gerçeği Almak, gerçek ekranlar oluşturmak için işlevsel özelliklerden ve diğer geçici belgelerden vazgeçer. Gerçek bir web sayfası gerçeklik iken, işlevsel bir özellik, bir inandırıcılık, bir anlaşma yanılsamasıdır. Müşterilerinizin göreceği ve kullanacağı şey budur. Önemli olan bu. Gerçek Olmak sizi oraya daha hızlı götürür. Ve bu, soyut kavramlar yerine gerçek şeylere dayalı yazılım kararları verdiğiniz anlamına gelir.

Son olarak, Gerçek Olmak, web tabanlı yazılıma ideal olarak uyan bir yaklaşımdır. Yazılımın bir kutu içinde gönderilmesinin ve ardından bir veya iki yıl sonra bir güncellemenin teslim edilmesini beklemenin eski okul modeli kayboluyor.

Yüklü yazılımların aksine, web uygulamaları günden güne sürekli olarak gelişebilir. Gerçek Olmak, bu avantajı tüm değeriyle kullanır.



## Güçlü Yazılım Nasıl Yazılır

Güçlü yazı özlüdür. Bir çizimde gereksiz satırlar ve makinede gereksiz parçalar olmaması gerektiği gibi, bir cümle de gereksiz kelimeler, paragraf gereksiz cümleler içermemelidir. Bu, yazarın tüm cümleleri kısaltmasını veya tüm ayrıntılardan kaçınmasını ve konuları yalnızca ana hatlarıyla ele almasını değil, her kelimenin anlattığını gerektirir.

—William Strunk Jr.'ın "The Elements of Style" kitabından.

## Artık şişkinlik yok

Eski yol: uzun, bürokratik, kıcımızı örtmek için bunu yapıyoruz süreci. Tipik sonuç: şişirilmiş, sıradanlık damlayan unutulabilir yazılım. Blech.

## Gerçeğe Başlarken kurtulmak...

- Aylar hatta yıllar süren zaman çizelgeleri
- Gökyüzünde pasta işlevsel özellikleri
- Ölçeklenebilirlik tartışmaları
- Bitmek bilmeyen personel toplantıları
- Düzinelerce çalışanı işe alma "ihtiyacı"
- Anlamsız sürüm numaraları
- Mükemmel geleceği öngören bozulmamış yol haritaları
- Sonsuz tercih seçenekleri
- Dış kaynaklı destek
- Gerçekçi olmayan kullanıcı testi
- gereksiz evrak
- Yukarıdan aşağıya hiyerarşi

Harika bir yazılım oluşturmak için tonlarca paraya, büyük bir ekibe veya uzun bir geliştirme döngüsüne ihtiyacınız yok. Bu şeyler yavaş, bulanık, değişmez uygulamaların bileşenleridir. Gerçekleşmek tam tersi bir yaklaşım gerektirir.

## Bu kitapta size göstereceğiz...

- Felsefeye sahip olmanın önemi
- Neden küçük kalmak iyi bir şeydir
- Nasıl daha az inşa edilir
- Fikirden gerçeğe hızlı bir şekilde nasıl gidilir

- Ekibinizi nasıl
- görevlendireceksiniz Neden içten dışa tasarım
- yapmalısınız Yazmak neden bu kadar önemlidir
- Rekabetinizi neden geride bırakmalısınız
- Uygulamanızı nasıl tanıtacaksınız ve başarılı
- desteğin sırları Lansmandan sonra ivmeyi devam
- ettirmeye ilişkin ipuçları ...ve çok daha fazlası
- 

Odak, büyük resim fikirleri üzerindedir. Sizi ayrıntılı kod parçacıkları veya css numaralarıyla boğmayacağız. Gerçekleşme sürecini yönlendiren ana fikirlere ve felsefelere bağlı kalacağız.

## Bu kitap senin için mi?

Büyük bir fikir üzerinde çalışan bir girişimci, tasarımcı, programcı veya pazarlamacısınız.

Eski kuralların artık geçerli olmadığını anlıyorsunuz. Yazılımınızı her yıl cd-rom'lara mı dağıtıyorsunuz?

Nasıl 2002. Sürüm numaraları? Pencereden dışarı. İnşa etmeniz, başlatmanız ve ince ayar yapmanız gerekiyor. Ardından durulayın ve tekrarlayın.

Ya da belki henüz çevik geliştirme ve iş yapıları konusunda fikir sahibi değilsiniz ama daha fazlasını öğrenmeye heveslisiniz.

Bu size benziyorsa, o zaman bu kitap tam size göre.

Not: Bu kitabın vurgusu bir web uygulaması oluşturmak olsa da, bu fikirlerin çoğu yazılım dışı etkinliklere de uygulanabilir. Küçük ekipler, hızlı prototip oluşturma, yinelemeler beklemek ve burada sunulan diğer pek çok öneri, ister bir iş kuruyor, ister bir kitap yazıyor, bir web sitesi tasarlıyor, bir albüm kaydediyor veya çeşitli başka işler yapıyor olun, size yol gösterici olabilir. . Hayatınızın bir alanında Gerçekleşmeye başladığınızda, bu kavramların çok çeşitli faaliyetlere nasıl uygulanabileceğini göreceksiniz.

## 2: Basecamp Hakkında

### ne yapıyoruz

Basecamp, basit, odaklanmış yazılımlar oluşturan küçük bir ekiptir. Ürünlerimiz işbirliği yapmanıza ve organize olmanıza yardımcı olur. Milyonlarca insan ve küçük işletme, işleri halletmek için web uygulamalarımızı kullanıyor. Wall Street Journal'dan Jeremy Wagstaff, "[Basecamp] ürünleri, Outlook ekranını bir işkence odasının yazılım eşdeğeri gibi görünmesini sağlayan güzel, basit, zarif ve sezgisel araçlardır" diye yazdı. Uygulamalarımız sizi asla rafa kaldırmaz.

### çalışma şeklimiz

Yazılımın çok karmaşık olduğuna inanıyoruz. Çok fazla özellik, çok fazla düğme, öğrenilecek çok şey var. Ürünlerimiz kasıtlı olarak rekabetten daha az şey yapıyor. Daha akıllı çalışan, daha iyi hissettiren, işleri istediğiniz gibi yapmanıza olanak tanıyan ve kullanımı daha kolay ürünler üretiyoruz.

### Bizim ürünlerimiz

Basecamp „proje yönetimini alt üst ediyor. Basecamp, Gantt çizelgeleri, süslü grafikler ve istatistik ağırlıklı elektronik tablolar yerine mesaj panoları, yapılacaklar listeleri, basit zamanlama, ortak çalışma ve dosya paylaşımı sunar. Şimdiye kadar yüz binlerce kişi bunun daha iyi bir yol olduğu konusunda hemfikir. Salon.com'dan Farhad Manjoo, “Basecamp, Web'deki yazılımın geleceğini temsil ediyor” dedi.

HEY, e-postanın temelden yeniden icadıdır. HEY'in yeni yaklaşımı, e-postayı uğraşmak zorunda kaldığınız bir şey yerine kullanmak istediğiniz bir şeye dönüştürür.

Geliştiriciler için Ruby on Rails, gerçek dünya uygulamalarını hızlı ve kolay bir şekilde yazmak için Ruby'de tam yığın, açık kaynaklı bir web çerçevesidir. Rails, yoğun işlerin üstesinden gelir, böylece siz fikrinize odaklanabilirsiniz. O'Reilly yayıncılık imparatorluğundan Nathan Torkington, “Ruby on Rails şaşırtıcı. Bunu kullanmak, bir düzine kötü kışkılcı çerçevenin küçük yeni gelenleri sadece çeşitli yaratıcı şekillerde ele almak için dövülmeye hazırlandığı bir kung-fu filmi izlemek gibidir. ” Bu alıntıyı sevmeliyim.

Ürünlerimiz ve firmamız hakkında daha fazla bilgiyi [basecamp.com](http://basecamp.com) adresindeki web sitemizde bulabilirsiniz.

### 3: Uyarılar, sorumluluk reddi beyanları ve diğer önleyici uyarılar

Sadece aradan çıkarmak için, ara sıra duyduğumuz bazı şikayetlere verdiğimiz yanıtlar şunlardır:

#### “Bu teknikler benim için çalışmayacak.”

Gerçekleşmek bizim için müthiş çalışan bir sistem. Bununla birlikte, bu kitaptaki fikirler güneşin altındaki her projeye uygulanmayacaktır. Bir silah sistemi, bir nükleer kontrol tesisi, milyonlarca müşteri için bir bankacılık sistemi veya başka bir yaşam/finansal-kritik sistem inşa ediyorsanız, laissez-faire tavrımızdan bazılarına karşı koyacaksınız. Devam edin ve ek önlemler alın.

Ve bu ya hep ya hiç önermesi olmak zorunda değildir. Gerçekleşmek'i tam olarak kucaklayamasanız bile, burada en az birkaç fikir olması gerekir ki, bu güçleri gizlice aşabilirsiniz.

#### "Bu fikri sen icat etmedin."

Bu teknikleri icat ettiğimizi iddia etmiyoruz. Bu kavramların çoğu, uzun süredir şu ya da bu biçimde var olmuştur. Tavsiyelerimizden bazılarını okursanız ve bu size filancanın web günlüğünde veya 20 yıl önce yayınlanmış bir kitapta okuduğunuz bir şeyi hatırlatırsa, sinirlenmeyin. Kesinlikle mümkün. Bu teknikler hiç de Basecamp'a özel değildir. Size sadece nasıl çalıştığımızı ve bizim için neyin başarılı olduğunu söylüyoruz.

#### "Siyah beyazı çok fazla değerlendiriyorsun."

Tonumuz çok bilmiş görünüyor, bize katlanın. Fikirleri bu konuda karamsar olmaktansa cesur vuruşlarla sunmanın daha iyi olduğunu düşünüyoruz. Bu kendini beğenmiş ya da kibirli olarak geliyorsa, öyle olsun. Her şeyi “duruma göre...” ile sulamaktansa kışkırtıcı olmayı tercih ederiz. Elbette bu kuralların esnetilmesi veya çiğnenmesi gereken zamanlar olacaktır. Ve bu taktiklerden bazıları sizin durumunuz için geçerli olmayabilir. Kararınızı ve hayal gücünüzü kullanın.

#### "Bu benim şirketimde işe yaramaz."

Gerçek Olmak için çok büyük olduğunu mu düşünüyorsun? Microsoft bile Gerçekleşiyor (ve onlardan daha büyük olduğunuzdan şüpheliyiz).

Şirketiniz genellikle büyük ekiplerle uzun vadeli planlar üzerinde çalışsa bile, yine de gerçekçi olmanın yolları vardır. İlk adım, daha küçük birimlere bölünmektir. İşin içinde çok fazla insan olduğunda hiçbir şey yapılmaz. Ne kadar yalın olursanız, işler o kadar hızlı ve daha iyi yapılır.

Verilmiş, biraz satıcılık gerektirebilir. Şirketinizi Gerçekleşme sürecine sokun. Onlara bu kitabı göster. Onlara daha kısa sürede ve daha küçük bir ekiple elde edebileceğiniz gerçek sonuçları gösterin.

Gerçekleşmenin yeni kavramları test etmek için düşük riskli, düşük yatırımlı bir yol olduğunu açıklayın. Kavram kanıtı olarak daha küçük bir projede ana gemiden ayrılıp ayrılamayacağınıza bakın. Sonuçları gösterin.

Ya da gerçekten cesur olmak istiyorsanız, gizlice gidin. Radarın altında uçun ve gerçek sonuçları gösterin.

Microsoft'ta Gerçekleşirken Start.com ekibinin kullandığı yaklaşım budur. "Start.com ekibinin çalışmasını izledim. İzin istemiyorlar," diyor Microsoft'ta Teknik Evangelist Robert Scoble. "Hava koruması sağlayan bir patronları var. Ve her seferinde biraz ısırlar ve bunu yaparlar ve geri bildirim yanıt verirler."

## Microsoft'un Start.com'unu Gönderme

Büyük şirketlerde süreçler ve toplantılar normdur. Müşteri için neyin "doğru" olduğu konusunda herkesin bir anlaşmaya varması amacıyla, özellikleri planlamak ve ayrıntıları tartışmak için aylar harcanır.

Bu, shrink ambalajlı yazılımlar için doğru yaklaşım olabilir, ancak web ile inanılmaz bir avantajımız var. Sadece gönder! Kullanıcının size doğru olup olmadığını söylemesine izin verin ve değilse, hey, isterseniz düzeltebilir ve aynı gün web'e gönderebilirsiniz! Müşterinininkinden daha güçlü bir kelime yoktur - uzun soluklu toplantılara ve tartışmalara katılma dürtüsüne direnin. Sadece gönder ve bir noktayı kanıtla.

Söylemesi yapmaktan çok daha kolay - bu şu anlama gelir:

Aylarca planlama gerekli değildir.

Spesifikasyonların aylarca yazılması gerekli değildir - spesifikasyonların temelleri sağlamlaştırılmalı ve geliştirme aşamasında ayrıntılar belirlenmeli ve rafine edilmelidir. Geliştirme başlamadan önce tüm açık sorunları kapatmaya ve her ayrıntıyı tespit etmeye çalışmayın.

Daha az özellik, ancak kaliteli özellikler gönderin.

Yepyeni bir sürüm ve birçok özellik ile büyük patlama yaklaşımına ihtiyacınız yok. Kullanıcılara sindirebilecekleri bayt boyutunda parçalar verin.

Küçük hatalar varsa, temel senaryoları çiviledikten hemen sonra gönderin ve hata düzeltmelerini bundan sonra yavaş yavaş web'e gönderin. Kullanıcı geri bildirimini ne kadar hızlı alırsanız o kadar iyi. Fikirler kağıt üzerinde kulağa harika gelebilir, ancak pratikte yetersiz olduğu ortaya çıkıyor. Bir fikirde yanlış olan temel sorunları ne kadar erken öğrenirseniz o kadar iyi.

Hızlı bir şekilde yinelediğinizde ve müşteri geri bildirimlerine tepki verdiğinizde, bir müşteri bağlantısı kuracaksınız. Hedefin müşteriye istediklerini inşa ederek kazanmak olduğunu unutmayın.

—Sanaz Ahari, Start.com Program Yöneticisi, Microsoft

## 4: Daha Az İnşa Edin

# rekabeti alt üst et

Geleneksel bilgelik, rakiplerinizi yenmek için onları bire bir geçmeniz gerektiğini söyler. Dört özelliği varsa, beş (veya 15 veya 25) gerekir. X harcıyorlarsa, xx harcamanız gerekir. 20'leri varsa, 30'a ihtiyacınız var.

Bu tür tek yönlü Soğuk Savaş zihniyeti bir çıkmaz sokaktır. Ürün üretmenin pahalı, savunmacı ve paranoyak bir yolu. Savunmacı, paranoyak şirketler ileriye düşünemezler, sadece gerisini düşünebilirler. Önderlik etmezler, takip ederler.

Takip eden bir şirket kurmak istiyorsanız, bu kitabı hemen bırakabilirsiniz.

Öyleyse ne yapmalı? Cevap daha az. Onları yenmek için rakiplerinizden daha azını yapın. Basit sorunları çözün ve kılı, zor, nahoş sorunları herkese bırakın. Bire bir yükselmek yerine, bir kere indirmeyi deneyin. Aşırıya kaçmak yerine, alt etmeyi deneyin.

Bu kitap boyunca daha az kavramını ele alacağız, ancak yeni başlayanlar için daha azı şu anlama gelir:

- Daha az özellik
- Daha az seçenek/tercih
- Daha az insan ve kurumsal yapı
- Daha az toplantı ve soyutlama
- Daha az vaat

# 5: Sorununuz Ne?

## Kendiniz için yazılım oluşturun

Yazılım geliřtirmenin harika bir yolu, kendi problemlerinizi çözerek başlamakdır. Hedef kitle olacaksınız ve neyin önemli neyin önemsiz olduđunu bileceksiniz. Bu size, çıđır açan bir ürün sunma konusunda harika bir başlangıç sağlar.

Buradaki anahtar, yalnız olmadıđınızı anlamaktır. Bu sorunu yaşıyorsanız, muhtemelen yüz binlercesi aynı gemidedir. İşte pazarınız. Bu kolay deđil miydi?

Basecamp bir sorundan kaynaklandı: Bir tasarım firması olarak müşterilerimizle projeler hakkında iletişim kurmak için basit bir yola ihtiyacımız vardı. Bunu, manuel olarak güncelleyeceđimiz müşteri ekstranetleri aracılıđıyla yapmaya başladık. Ancak bir projenin güncellenmesi gerektiđinde html'yi elle deđiřtirmek işe yaramıyordu. Bu proje alanları her zaman bayatlamış gibi görünüyordu ve sonunda terk edildi. Sinir bozucuydu çünkü bizi dađınık bıraktı ve müşterileri karanlıkta bıraktı.

Bu yüzden diđer seçeneklere bakmaya başladık. Yine de bulduđumuz her araç 1) ihtiyacımız olanı yapmadı ya da 2) faturalandırma, katı erişim kontrolleri, çizelgeler, grafikler vb. gibi ihtiyacımız olmayan özelliklerle dolup taşıdı. Daha iyi bir yol olması gerektiđini biliyorduk. bu yüzden kendimizinkini kurmaya karar verdik.

Kendi probleminizi çözdüđünüzde tutkulu olduđunuz bir araç yaratırsınız. Ve tutku anahtardır. Tutku, onu gerçekten kullanacađınız ve önemseyeceđiniz anlamına gelir. Ve bu, başkalarının da bu konuda tutkulu hissetmesini sağlamanın en iyi yolu.

## Kendi kařıntını kařmak

Açık Kaynak dünyası bu mantrayı uzun zaman önce benimsedi - buna "kendi kařıntını kařmak" diyorlar. Açık kaynak geliřtiriciler için bu, istedikleri araçları istedikleri şekilde teslim ettikleri anlamına gelir. Ancak fayda çok daha derine iner.

Yeni bir uygulamanın tasarımcısı veya geliřtiricisi olarak her gün yüzlerce mikro kararla karşı karşıya kalıyorsunuz: mavi mi yeřil mi? Bir masa mı iki mi? Statik mi, dinamik mi? İptal mi yoksa kurtarma mı? Bu kararları nasıl alıyoruz? Önemli olduđunu bildiđimiz bir şeyse, sorabiliriz. Gerisi, tahmin ediyoruz. Ve tüm bu tahminler, uygulamalarımızda bir tür borç oluřturuyor - birbirine bađlı bir varsayımlar ađı.

Bir geliřtirici olarak bundan nefret ediyorum. Yazdıđım uygulamalardaki tüm bu küçük ölçekli saatli bombaların bilgisi stresimi arttırıyor. Açık Kaynak geliřtiricileri, kendi kařıntılarını kařıyarak bundan etkilenmezler. Kendi kullanıcıları oldukları için soruların %90'ına dođru cevapları biliyorlar.



vermek zorunda oldukları kararlar. İnsanların zor bir kodlama gününden sonra eve gelip ardından açık kaynak üzerinde çalışmasının nedenlerinden birinin bu olduğunu düşünüyorum: Rahatlatıcı.

—Dave Thomas, Pragmatik Programcılar

## İhtiyaçtan doğmuş

Campaign Monitor gerçekten ihtiyaçtan doğdu. Yıllarca, oradaki e-posta pazarlama seçeneklerinin kalitesi bizi hayal kırıklığına uğrattı. Bir araç x ve y yapar ama asla z yapmaz, bir sonraki y ve z çivilenmiş ancak x'i doğru yapamamış. Kazanamadık.

Programımızı temizlemeye ve hayallerimizdeki e-posta pazarlama aracımızı oluşturmaya karar verdik. Bilinçli olarak başkalarının ne yaptığına bakmamaya karar verdik ve bunun yerine bizim ve müşterilerimizin hayatlarını biraz daha kolaylaştıracak bir şey inşa ettik.

Görünüşe göre, oradaki seçeneklerden mutsuz olan sadece biz değildik. Herhangi bir tasarım firmasının kullanabilmesi için yazılımda birkaç değişiklik yaptık ve bunu yaymaya başladık. Altı aydan kısa bir süre içinde binlerce tasarımcı, kendileri ve müşterileri için e-posta bültenleri göndermek için Campaign Monitor'ü kullanmaya başladı.

—David Greiner, kurucu, Campaign Monitor

## bununla ilgilenmen gerek

Bir kitap yazarken, ilginç bir hikayeden daha fazlasına sahip olmanız gerekir. Hikayeyi anlatmak için bir arzunuz olmalı. Bir şekilde kişisel olarak yatırım yapmanız gerekiyor. İki yıl, üç yıl, hayatının geri kalanı boyunca bir şeyle yaşayacaksan, onunla ilgilenmelisin.

—Malcolm Gladwell, yazar ( A Few Thin Slices of Malcolm Gladwell'den )

## 6: Kendinize Fon Sağlayın

### Dışarıdaki para B planıdır

Birçok girişimin ilk önceliği yatırımcılardan fon elde etmektir. Ancak unutmayın, finansman için yabancılara yönelirseniz, onlara da cevap vermeniz gerekir. Beklentiler yükseltilir. Yatırımcılar paralarını hızlı bir şekilde geri istiyorlar. Üzücü gerçek şu ki, para kazanmak çoğu zaman kaliteli bir ürün üretmeye başlar.

Bu günlerde yuvarlanmak çok fazla sürmüyor. Donanım ucuzdur ve birçok harika altyapı yazılımı açık kaynak kodlu ve ücretsizdir. Ve tutku bir fiyat etiketi ile gelmez.

Bu yüzden elinizdeki nakitle yapabileceğinizi yapın. İyi düşünün ve neyin gerçekten gerekli olduğunu ve onsuz neler yapabileceğinizi belirleyin. On yerine üç kişiyle ne yapabilirsiniz? 100 bin dolar yerine 20 bin dolar ile ne yapabilirsiniz? Altı yerine üç ayda ne yapabilirsiniz? Günlük işinizi sürdürürseniz ve uygulamanızı yan tarafta oluştursanız ne yapabilirsiniz?

### Kısıtlamalar yaratıcılığı zorlar

Sınırlı kaynaklarla çalıştırın ve kısıtlamaları daha erken ve daha yoğun bir şekilde hesaba katmak zorunda kalacaksınız. Ve bu iyi bir şey. Kısıtlamalar inovasyonu tetikler.

Kısıtlamalar ayrıca fikrinizi bir an önce doğaya salmaya zorlar - başka bir iyi şey. Kapıdan bir veya iki ay sonra, bir şeyin üzerinde olup olmadığına dair oldukça iyi bir fikre sahip olmalısın. Eğer öyleyseniz, kısa süre içinde kendi kendini idame ettirebilecek duruma geleceksiniz ve dışarıdan nakit paraya ihtiyacınız olmayacak. Fikriniz bir limonsa, çizim tahtasına geri dönme zamanı. En azından, yolun aşağısındaki aylar (veya yıllar) yerine şimdi biliyorsunuz. Ve en azından kolayca geri dönebilirsin. Yatırımcılar işin içine girince çıkış planları daha da zorlaşıyor.

Sadece hızlı para kazanmak için yazılım oluşturuyorsanız, bunu gösterecektir. Gerçek şu ki, hızlı bir ödeme pek olası değildir. Bu nedenle, sizin ve müşterilerinizin uzun süre birlikte yaşayabileceği kaliteli bir araç oluşturmaya odaklanın.

### iki yol

[Jake Walker yatırımcı parası olan (Disclive) ve parasız (The Show) bir şirket kurdu.

Burada iki yol arasındaki farkları tartışıyor.]

Tüm sorunların kökü para toplamak değil, onunla birlikte gelen her şeydi.

Beklentiler sadece daha yüksek. İnsanlar maaş almaya başlar ve motivasyon onu inşa etmektir.

ya da ilk yatırımcıların paralarını geri kazanmaları için başka bir yol bul. İlk şirket söz konusu olduğunda, olduğumuzdan çok daha büyük davranmaya başladık - zorunluluktan...

[The Show ile] daha az maliyetle çok daha iyi bir ürünü ancak daha fazla zamanla teslim edebileceğimizi fark ettik. Ve insanların hız yerine kaliteyi beklemeye istekli olacağı konusunda kendi paramızın bir kısmıyla kumar oynadık. Ancak şirket küçük bir operasyon olarak kaldı (ve muhtemelen olmaya devam edecek). Ve o ilk projeden beri, tamamen kendi kendimizi finanse ettik. Satıcılarımızdan aldığımız birazcık yaratıcı terimlerle, operasyona kendi paramızın çoğunu koymamız gerekmiyor. Ve beklenti büyüyüp satmak değil, büyüme adına büyümek ve bundan finansal olarak yararlanmaya devam etmektir.

—Sinyale Karşı Gürültü'den bir yorum

## 7: Zamanı ve Bütçeyi Düzeltin, Esnek Kapsam

### Zamanında ve bütçede lansman yapın

İşte zamanında ve bütçe dahilinde başlatmanın kolay bir yolu: sabit tutun. Bir soruna asla daha fazla zaman veya para harcamayın, sadece kapsamı küçültün.

Şuna benzer bir efsane var: Zamanında, bütçede ve kapsamda lansman yapabiliriz. Neredeyse hiç olmaz ve yaptığında kalite genellikle zarar görür.

Her şeyi ayrılan süre ve bütçeye sığdıramıyorsanız, zamanı ve bütçeyi genişletmeyin. Bunun yerine, kapsamı geri çekin. Daha sonra bir şeyler eklemek için her zaman zaman vardır - sonra sonsuzdur, şimdi kısaktır.

Kapsamı planlanandan biraz daha küçük olan harika bir şey başlatmak, vasat ve boşluklarla dolu bir şey başlatmaktan daha iyidir çünkü sihirli bir zaman, bütçe ve kapsam penceresine ulaşmanız gerekiyordu. Sihri Houdini'ye bırakın. Gerçek bir işletmeniz ve gerçek bir ürününüz var. teslim etmek.

Zamanı ve bütçeyi sabitlemenin ve kapsamı esnek tutmanın faydaları şunlardır:

#### önceliklendirme

Gerçekten neyin önemli olduğunu bulmalısın. Bu ilk sürüme ne katacak?

Bu, sizi kıvırmak ve havlamak yerine zor kararlar almaya itecek bir kısıtlamaya neden olur.

#### gerçeklik

Beklentileri belirlemek anahtardır. Zamanı, bütçeyi ve kapsamı düzeltmeye çalışırsanız, yüksek kalitede teslimat yapamazsınız. Elbette, muhtemelen bir şeyler sunabilirsiniz, ancak gerçekten iletmek istediğiniz “bir şey” mi?

#### Esneklik

Değiştirme yeteneği anahtardır. Her şeyin sabit olması, değişmesini zorlaştırır. Kapsam esnekliği enjekte etmek, ürünü oluştururken gerçek deneyiminize dayalı seçenekler sunacaktır. Esneklik senin arkadaşın.

Önerimiz: Kapsamı azaltın. Yarım ürün yapmaktansa yarım ürün yapmak daha iyidir (bundan daha sonra bahsedeceğiz).

Bir, iki, üç...

| Bir proje nasıl programın bir yıl gerisinde kalır? Günün birinde.

—Fred Brooks, yazılım mühendisi ve bilgisayar bilimcisi

## 8: Bir Düşmanınız Olsun

### Bir dövüş seç

Bazen uygulamanızın ne olması gerektiğini bilmenin en iyi yolu, ne olmaması gerektiğini bilmektir. Uygulamanızın düşmanını anlayın ve gitmeniz gereken yere ışık tutacaksınız.

Proje yönetimi yazılımı oluşturmaya karar verdiğimizde, odadaki gorilin Microsoft Project olduğunu biliyorduk. Gorilden korkmak yerine onu motive edici olarak kullandık. Basecamp'ın tamamen farklı bir şey olacağına karar verdik, anti-Project.

Proje yönetiminin çizelgeler, grafikler, raporlar ve istatistiklerle ilgili olmadığını, bunun iletişimle ilgili olduğunu fark ettik. Aynı zamanda bir proje yöneticisinin yüksekte oturup bir proje planı yayınlamasıyla da ilgili değil. Projeyi yürütmek için herkesin birlikte sorumluluk almasıyla ilgili.

Düşmanımız, Proje Yönetimi Diktatörleri ve onların kırbaç kırmak için kullandıkları araçlardı. Proje yönetimini demokratikleştirmek istedik - bunu herkesin (müşteri dahil) parçası olduğu bir şey haline getirmek. Herkes ortak mülkiyeti aldığında projeler daha iyi sonuç verir. işlem.

İş Writeboard'a geldiğinde, çok sayıda vızıltı özelliğine sahip rakipler olduğunu biliyorduk. Bu yüzden bunun yerine "yaygın olmayan" bir açığı vurgulamaya karar verdik. İnsanların temel olmayan özelliklerle boğmadan fikirlerini paylaşmasına ve üzerinde ortak çalışmasına izin veren bir uygulama oluşturduk. Eğer gerekli değilse, onu dışarıda bıraktık. Lansmandan sadece üç ay sonra 100.000'den fazla Yazı Tahtası oluşturuldu.

Sırt Çantası'na başladığımızda düşmanımız yapı ve katı kurallardı. İnsanlar, önceden biçimlendirilmiş bir dizi ekrana veya çok sayıda gerekli form alanına dayalı olarak değil, bilgilerini kendi yöntemleriyle düzenleyebilmelidir.

Bir düşmana sahip olmanın bir bonusu, çok açık bir pazarlama mesajıdır. İnsanlar çatışmalardan ürüyor. Ayrıca bir ürünü başkalarıyla karşılaştırarak anlarlar. Seçilmiş bir düşmanla insanlara duymak istedikleri bir hikayeyi besliyorsunuz. Sadece ürününüzü daha iyi ve daha hızlı anlamakla kalmayacak, aynı zamanda taraf tutacaklar. Ve bu, dikkat çekmenin ve tutkuyu ateşlemenin kesin yolu.

Tüm bunlarla birlikte, rekabete fazla kafayı takmamak da önemlidir.

Diğer ürünleri aşırı analiz edin ve düşünme şeklinizi sınırlamaya başlayacaksınız. Bir göz atın ve ardından kendi vizyonunuza ve kendi fikirlerinize geçin.

lideri takip etmeyin

Pazarlamacılar (ve tüm insanlar) lideri takip etmek için iyi eğitilmiştir. Doğal içgüdü, rekabet için neyin işe yaradığını bulmak ve sonra onu geçmeye çalışmaktır - fiyatta rekabet eden rakibinizden daha ucuz veya hız konusunda rekabet eden rakipten daha hızlı olmak. Sorun şu ki, bir tüketici bir kez başka birinin hikayesini satın alıp bu yalana inandığında, tüketiciyi değiştirmeye ikna etmek, onu yanıltıldığını kabul etmeye ikna etmekle aynı şeydir. Ve insanlar yanıltıklarını kabul etmekten nefret ederler.

Bunun yerine, farklı bir hikaye anlatmalı ve dinleyicileri, hikayenizin şu anda inandıkları hikayeden daha önemli olduğuna ikna etmelisiniz. Rekabetiniz daha hızlıysa, daha ucuz olmalısınız. Sağlık hikayesini satıyorlarsa, kolaylık hikayesini satmalısınız. Konumlandırma x/y eksenini türünden “Biz daha ucuzuz” iddiası değil, halihazırda anlatılan hikayeden tamamen farklı gerçek bir hikaye.

—Seth Godin, yazar/girişimci ( Be a Better Liar'dan)

## Anahtar sorun nedir?

Başınızı belaya sokmanın en hızlı yollarından biri, rakiplerinizin ne yaptığına bakmaktır. Bu özellikle BlinkList'te bizim için geçerliydi. Başlattığımızdan bu yana, başlatılan yaklaşık 10 başka sosyal imleme hizmeti daha var. Bazı insanlar, özellik karşılaştırması yoluyla ayrıntılı bir özelliklerle çevrimiçi elektronik tablolar oluşturmaya bile başladılar.

Ancak bu, kişiyi kısa sürede yoldan çıkarabilir. Bunun yerine, büyük resme odaklanın ve kendimize, çözmeye çalıştığımız temel sorunun ne olduğunu ve onu nasıl çözebileceğimizi sormaya devam ederiz.

—Michael Reining, kurucu ortak, MindValley & Blinklist

## 9: Bir Angarya Olmamalı

### Tutkunuz - ya da eksikliğiniz - parlayacak

Uygulamanız ne kadar az inşa edilecek bir angaryaysa, o kadar iyi olur. Sürecin tadını çıkarabilmeniz için küçük ve yönetilebilir tutun.

Uygulamanız sizi heyecanlandırmıyorsa bir sorun var demektir. Sadece para çekmek için üzerinde çalışıyorsanız, görünecektir. Aynı şekilde, uygulamanız hakkında tutkulu hissediyorsanız, nihai üründe ortaya çıkacaktır. İnsanlar satır aralarını okuyabilir.

#### tutkunun varlığı

Anlamanın genellikle tartışmalı bir şekilde öznel veya acı verici bir şekilde anlaşılabilir olduğu tasarımda, tutkunun varlığından daha açık ve net çok az şey vardır. Bu, bir ürünün tasarımı sizi memnun etse de sizi üşütse de doğrudur; her iki durumda da onu inşa eden ellerin duygusal yatırımını tespit etmemek zor.

Coşku elbette kendini hemen gösterir, ancak kayıtsızlık da aynı şekilde silinmezdir. Bağlılığınız elinizdeki işe karşı gerçek bir tutku içermiyorsa, ne kadar özenle veya çekici tasarlanmış olursa olsun, gizlemesi neredeyse imkansız bir boşluk haline gelir.

—Khoi Vinh, [Subtraksiyon.com](http://Subtraksiyon.com)

## Fırın

Bu noktada Amerikan işi gerçekten bir fikir geliştirmek, onu kârlı kılmak, kârlıken satmak ve sonra dışarı çıkmak veya çeşitlendirmekle ilgilidir. Bu sadece her şeyi emmekle ilgili.

Benim fikrim şuydu: Pişirmenin tadını çıkar, ekmeğini sat, insanlar beğensin, daha çok sat. Fırını devam ettirin çünkü iyi yemek yapıyorsunuz ve insanlar mutlu.

—Ian MacKaye, Fugazi üyesi ve Dischord Records'un ortak sahibi  
([Salon.com](http://Salon.com) People | [Ian MacKaye](http://IanMacKaye.com)'den)



## 10: Daha Az Kütle

### Ne kadar yalın olursan, değiştirmek o kadar kolay olur

Bir nesne ne kadar büyükse, yönünü değiştirmek için o kadar fazla enerji gerekir. Bu, fiziksel dünyada olduğu kadar iş dünyasında da geçerlidir.

Web teknolojisi söz konusu olduğunda, değişim kolay ve ucuz olmalıdır. Anında değişemezseniz, değiştirebilecek birine zemin kaybedersiniz. Bu yüzden daha az kütle için ateş etmeniz gerekiyor.

### Kütle artar...

- Uzun vadeli sözleşmeler
- Fazla personel
- Kalıcı kararlar
- Diğer toplantılarla ilgili toplantılar
- kalın süreç
- Envanter (fiziksel veya zihinsel)
- Donanım, yazılım, teknoloji kilitlenmeleri
- Özel veri formatları
- Geçmiş geleceği yönetiyor
- Uzun vadeli yol haritaları
- Ofis kuralları

### Kütle azalır...

- Anında düşünme
- Çok görevli ekip üyeleri
- Kısıtlamaları kucaklamak, onları kaldırmaya çalışmak değil
- Daha az yazılım, daha az kod
- Daha az özellik
- Küçük takım boyutu
- Basitlik
- Kısıtlanmış arayüzler
- Açık kaynaklı ürünler
- Açık veri formatları
- Hataları kabul etmeyi kolaylaştıran açık bir kültür

Daha az k t le, y n  hızlı bir Őekilde deęiŐtirmenizi saęlar. Tepki verebilir ve geliŐebilirsiniz. İyi fikirlere odaklanabilir ve k t leri bırakabilirsiniz. M Őterilerinizi dinleyebilir ve yanıtlayabilirsiniz. Yeni teknolojileri daha sonra deęil, Őimdi entegre edebilirsiniz. Bir u ak gemisi yerine bir sigara teknesini y netiyorsunuz. Bu ger eęin tadını  ıkarın.

 rneęin, daha az yazılım ve daha az  zellik i eren bir  r n geliŐtiren yalın, daha az kitlesel bir Őirket d Ő nelim. Dięer tarafta,  nemli  l  de daha fazla yazılım ve daha fazla  zellięe sahip bir  r ne sahip olan daha kitlesel bir Őirket var. O zaman diyelim ki Ajax gibi yeni bir teknoloji ya da etiketleme gibi yeni bir konsept ortaya  ıktı. Kim  r nlerini daha hızlı adapte edebilecek?

Daha fazla yazılım ve daha fazla  zellik ve 12 aylık bir yol haritası olan ekip mi yoksa daha az yazılım ve daha az  zellik ve daha organik bir “Őu anda neye odaklanmamız gerektięine odaklanalım” s reci olan ekip mi?

A ık ası, daha az k t leye sahip Őirket, pazarın ger ek taleplerine uyum saęlamak i in daha iyi bir konumdadır. Daha fazla k t leye sahip Őirket, daha az k t leye sahip Őirket ge iŐ yaptıktan  ok sonra bile deęiŐiklikleri tartıŐıyor veya b rokratik s recinden ge irmeye devam edecek. Daha az k t leli Őirket iki adım  nde olacak, daha  ok k t leli Őirket hala nasıl y r neceęini bulmaya  alıŐıyor.

 evik,  evik, daha az k t leli iŐletmeler, t m iŐ modellerini,  r nlerini,  zellik setlerini ve pazarlama mesajlarını hızla deęiŐtirebilir. Hatalar yapabilir ve bunları hızlı bir Őekilde d zeltebilirler.  nceliklerini,  r n karmasını ve odak noktasını deęiŐtirebilirler. Ve en  nemlisi, fikirlerini deęiŐtirebilirler.

# 11: Değişim Maliyetinizi Düşürün

## Değişimin önündeki engelleri azaltarak esnek kalın

Değişim senin en iyi arkadaşın. Bir değişiklik yapmak ne kadar pahalıysa, o kadar az olasıdır. Rakipleriniz sizden daha hızlı değişebiliyorsa, büyük bir dezavantajdasınız demektir. Değişim çok pahalı hale gelirse, ölürsünüz.

İşte burada yalın kalmanın size gerçekten yardımcı olduğu yer. Bir kuruşta değiştirme yeteneği, küçük takımların varsayılan olarak sahip olduğu, büyük takımların asla sahip olamayacağı bir şeydir. Büyük adamların küçük adamları kıskandığı yer burasıdır. Büyük bir organizasyonda büyük bir ekibin değişmesi haftalar süren bir şeyin değişmesi, küçük, yalın bir organizasyonda sadece bir gün sürebilir. Bu avantaj paha biçilemez. Ucuz ve hızlı değişiklikler küçüklerin gizli silahıdır.

Ve unutmayın: Tüm para, tüm pazarlama, dünyadaki tüm insanlar küçük olmanın verdiği çevikliği satın alamaz.

Web teknolojisi söz konusu olduğunda, değişim kolay ve ucuz olmalıdır. Anında değişemezseniz, değiştirebilecek birine zemin kaybedersiniz. Bu yüzden daha az kütle için ateş etmeniz gerekiyor.

## ortaya çıkış

Ortaya çıkış, çevikliğin temel ilkelerinden biridir ve saf büyüye en yakın olanıdır.

Ortaya çıkan özellikler tasarlanmamıştır veya yerleşik değildir, yalnızca sistemin geri kalanının dinamik bir sonucu olarak ortaya çıkarlar. "Ortaya Çıkma", "öngörülemeyen bir olay" anlamında 17. yüzyıl ortası Latincesinden gelir. Bunu planlayamaz veya programlayamazsınız, ancak olmasına izin verebileceğiniz ve bundan faydalanabileceğiniz bir ortam geliştirebilirsiniz.

Klasik bir ortaya çıkış örneği, kuşların sürü davranışında yatmaktadır. Bir bilgisayar simülasyonu, en az üç basit kural kullanabilir ("birbirinize rastlamayın" çizgisinde) ve sürü, engellerin etrafında yeniden şekillenerek, gökyüzünde zarif bir şekilde ilerlerken aniden çok karmaşık davranışlar elde edersiniz. ve benzeri. Bu gelişmiş davranışların hiçbirisi (bir engelin etrafında aynı şekli yeniden şekillendirmek gibi) kurallarla belirtilmez; sistemin dinamiklerinden doğar.

Kuş simülasyonunda olduğu gibi basit kurallar karmaşık davranışlara yol açar. Çoğu ülkedeki vergi yasasında olduğu gibi karmaşık kurallar aptalca davranışlara yol açar.

Pek çok yaygın yazılım geliştirme uygulamasının talihsiz yan etkisi, herhangi bir acil davranış olasılığını ortadan kaldırmadır. Optimizasyon girişimlerinin çoğu - bir şeyi çok açık bir şekilde bağlamak - etkileşimlerin ve ilişkilerin genişliğini ve kapsamını azaltır;

ortaya çıkış kaynağı. Sürü kuşları örneğinde, iyi tasarlanmış bir sistemde olduğu gibi, ilginç davranışı yaratan etkileşimler ve ilişkilerdir.

İşleri ne kadar sıkılaştırırsak, yaratıcı, yeni ortaya çıkan bir çözüm için o kadar az yer kalır.

Gereksinimleri iyi anlaşılmadan önce kilitlemek veya kodu zamanından önce optimize etmek veya son kullanıcıların sistemle oynamasına izin vermeden karmaşık gezinme ve iş akışı senaryoları icat etmek olsun, sonuç aynıdır: temiz, zarif bir sistem yerine aşırı karmaşık, aptal bir sistem ortaya çıkmasını sağlar.

Küçük tutun. Basit tutun. Hadi olsun.

—Andrew Hunt, Pragmatik Programcılar \_\_\_\_\_

## 12: Üç Silahşörler

### 1.0 sürümü için üç kişilik bir ekip kullanın

Uygulamanızın ilk sürümü için yalnızca üç kişiyle başlayın. Bu, size yeterli insan gücü verecek, aynı zamanda modern ve çevik kalmanıza izin verecek sihirli sayıdır. Bir geliştirici, tasarımcı ve bir süpürücü ile başlayın (her iki dünya arasında dolaşabilen biri).

Elbette, yalnızca birkaç kişiyle bir uygulama oluşturmak zor. Ama doğru takıma sahipseniz, buna değer. Yetenekli insanlar sonsuz kaynaklara ihtiyaç duymazlar. Kısıtlamalar içinde çalışmanın ve sorunları çözmek için yaratıcılıklarını kullanmanın zorluğundan kurtulurlar. İnsan gücü eksikliğinizi, sürecin başlarında takaslarla uğraşmak zorunda kalacağınız anlamına gelir - ve sorun değil. Önceliklerinizi daha sonra değil, daha erken belirlemenizi sağlayacaktır. Ve sürekli olarak insanları işin dışında bırakma konusunda endişelenmenize gerek kalmadan iletişim kurabileceksiniz.

Sürümünüzü üç kişiyle oluşturamıyorsanız, o zaman ya farklı kişilere ihtiyacınız var ya da ilk sürümünüzü küçültmeniz gerekiyor. Unutmayın, ilk sürümünüzü küçük ve sıkı tutmanız sorun değil. Fikrinizin kanatları olup olmadığını çabucak göreceksiniz ve varsa, üzerine inşa etmek için temiz, basit bir temeliniz olacak.

### Metcalf Yasası ve proje ekipleri

Ekibi mümkün olduğunca küçük tutun. Proje ekipleri söz konusu olduğunda, "bir iletişim sisteminin değeri, sistemin kullanıcı sayısının yaklaşık karesi kadar büyür" diyen Metcalfe Yasası'nın bir sonucu vardır: Ekibin verimliliği, yaklaşık olarak sistemin karesinin tersidir. takımındaki üye sayısı. 1.0 ürün sürümü için üç kişinin ideal olduğunu düşünmeye başlıyorum...Ekibe eklemeyi planladığınız kişi sayısını azaltarak başlayın ve ardından biraz daha azaltın.

—Marc Hedlund, O'Reilly Media'da ikamet eden girişimci

## iletiřim akıřı

İletiřim, küçük ekiplerde büyük ekiplerden daha kolay akar. Bir projedeki tek kiři sizseniz, iletiřim basittir. Tek iletiřim yolu sizinle müşteri arasındadır. Ancak bir projedeki insan sayısı arttıkça iletiřim yollarının sayısı da artar. Toplamsal olarak artmaz, kiři sayısı arttıkça çarpımsal olarak artar, kiři sayısının karesiyle orantılıdır.

—Steve McConnell, Construx Software Builders Inc.'de Bař Yazılım Mühendisi (Less is More'dan : Küçük Ekiplerle Verimlilięi Hızla Bařlatma) \_\_\_\_\_

# 13: Kısıtlamaları Kucaklayın

## Sınırlamaların sizi yaratıcı çözümlere yönlendirmesine izin verin

Etrafta dolaşmak için asla yeterli değil. Yetersiz zaman. Yetersiz para. Yeterli insan yok.

Bu iyi birşey.

Bu kısıtlamalar hakkında çıldırmak yerine, onları kucaklayın. Size rehberlik etmelerine izin verin. Kısıtlamalar yeniliği yönlendirir ve odaklanmayı zorlar. Onları kaldırmaya çalışmak yerine, onları kendi yararınıza kullanın.

Basecamp'ı inşa ederken birçok sınırlamamız vardı. Sahibiz:

- Çalıştırılacak bir tasarım firması
- Mevcut müşteri çalışması
- 7 saatlik bir zaman farkı (David, Danimarka'da programlamayı yapıyordu, geri kalanımız Amerika'daydık)
- Küçük bir ekip
- Dış finansman yok

"Yeterli değil" mavilerini hissettik. Bu yüzden tabağımızı küçük tuttuk. Bu şekilde ancak bu kadarını koyabildik. Büyük görevler aldık ve bunları küçük parçalara böldük ve her seferinde bir tane ele aldık. Adım adım ilerledik ve ilerledikçe öncelik verdik.

Bu da bizi yaratıcı çözümler bulmaya zorladı. Her zaman daha az yazılım geliştirerek değişim maliyetimizi düşürdük. İnsanlara kendi sorunlarını kendi yöntemleriyle çözmelerine yetecek kadar özellik verdik - ve sonra yoldan çekildik. Aramızdaki saat farkı ve mesafe, iletişimimizde daha verimli olmamızı sağladı. Yüz yüze görüşmek yerine, neredeyse sadece im ve e-posta yoluyla iletişim kurduk ve bu da bizi hızlı bir şekilde konuya girmeye zorladı.

Kısıtlamalar genellikle kılık değiştirmiş avantajlardır. Risk sermayesi, uzun serbest bırakma döngüleri ve hızlı işe alımları unutun. Bunun yerine, sahip olduklarınızla çalışın.

## Kötülükle savaş

"Sürünen zarafet" olarak tanımlanan şey, muhtemelen "özellik yanıklığı" olarak daha iyi tanımlanır, çünkü bir bitki üzerindeki bir mantar gibi, yavaş yavaş detaylandırır ve özsuynunu boşaltırken ürünün gerçek ana hatlarını bulanıklaştırır. Yanıklığın panzehiri, elbette, "daraltıcı son tarih"tir. Bu, özelliklerin uygulanması için gereken süre ile orantılı olarak atılmasına neden olur. Çoğu zaman, en kullanışlı özelliklerin uygulanması en uzun sürer.

Bu nedenle, felaket ve son teslim tarihinin birleşimi, bildiğimiz ve sevdiğimiz gibi, bol miktarda yararsız özellikten oluşan bir yazılım sağlar.

—Jef Raskin, yazar ( “Yazılım Neden Olduğu Gibidir” kitabından)



## 14: Kendiniz Olun

### Kişisel ve arkadaş canlısı davranarak kendinizi büyük şirketlerden farklılaştırın

Birçok küçük şirket, büyük davranmaya çalışmak gibi bir hata yapar. Sanki büyüklüklerini örtbas edilmesi gereken bir zayıflık olarak algılıyorlar. Çok kötü. Küçük olmak, özellikle iletişim söz konusu olduğunda, aslında büyük bir avantaj olabilir.

Küçük şirketler daha az formalite, daha az bürokrasi ve daha fazla özgürlüğe sahiptir. Daha küçük şirketler varsayılan olarak müşteriye daha yakındır. Bu, müşterilerle daha doğrudan ve kişisel bir şekilde iletişim kurabilecekleri anlamına gelir. Küçükseniz, jargon yerine tanıdık bir dil kullanabilirsiniz. Siteniz ve ürününüz kurumsal bir drone gibi ses çıkarmak yerine insan sesine sahip olabilir. Küçük olmak, müşterilerinizle konuşabileceğiniz anlamına gelir, onlara bağlı değil.

Küçük şirketlerde de iç iletişimin avantajları vardır. Formalitelerden kurtulabilirsiniz. Her şeyde zorlu süreçlere ve çoklu imzalara gerek yok.

Bu süreçteki herkes açık ve dürüstçe konuşabilir. Bu sınırsız fikir akışı, küçük kalmanın en büyük avantajlarından biridir.

Gururla, meydan okurcasına dürüst ol

Bir müşterinin, şirketinizdeki çalışan sayısı veya tekliflerinizin genişliği konusunda abartılarla kandırılabilirliğini düşünüyor olsanız da, akıllı olanlar, gerçekten istediğiniz kişiler, ister sezgi yoluyla ister çıkarım yoluyla her zaman gerçeği öğreneceklerdir. Utanç verici bir şekilde, geçmişte bunun gibi beyaz yalanların bir parçası oldum ve bu durumların hiçbirisi bir iş için en önemli olan şeyle sonuçlanmadı: sunulan hizmetlere gerçekten ihtiyacı olan insanlarla anlamlı, kalıcı ve karşılıklı yarar sağlayan ilişkiler. Daha iyi yol, şirketin tam büyüklüğü ve genişliği hakkında gururla, meydan okurcasına dürüst olmak olurdu.

-Khoi Vinh, Subtraksiyon.com ve Behavior LLC'nin kurucu ortağı

herhangi bir zamanda

Hangi işte olursanız olun, iyi bir müşteri hizmeti, herhangi bir müşterinin yapacağı en büyük talep olmalıdır. Kullandığımız hizmetler için talep ediyoruz, neden müşterilerimizin farklı olacağını düşünelim? En başından beri, müşterilerimizin her türlü soru veya soru için bizimle iletişime geçmelerini kolay ve şeffaf hale getirdik. Web sitemizde cep telefonlarımıza yönlendirilen ücretsiz bir numara listeliyoruz ve kartvizitlerimizde her birimiz cep telefon numaralarımızı listeliyoruz. Müşterilerimize iletişim kurabileceklerini vurguluyoruz.

Sorun ne olursa olsun her zaman bizimle. Müşterilerimiz bu güven düzeyini takdir ediyor ve hiç kimse bu hizmeti kötüye kullanmadı.

–Edward Knittel, Satış ve Pazarlama Direktörü, KennelSource

# 15: Büyük Fikir Nedir?

## Uygulamanız için tek nokta vizyonunu açıkça tanımlayın

Uygulamanız ne anlama geliyor? Gerçekten ne hakkında? Herhangi bir şey tasarlamaya veya kodlamaya başlamadan önce, ürününüzün amacını, yani vizyonunu bilmeniz gerekir. Büyük düşün. Neden var? Diğer benzer ürünlerden farkı nedir?

Bu vizyon, kararlarınıza rehberlik edecek ve sizi tutarlı bir yolda tutacaktır. Bir takılma noktası olduğunda, "Vizyona sadık kalıyor muyuz?" diye sorun.

Vizyonunuz da kısa olmalıdır. Fikri aktarmak için bir cümle yeterli olmalıdır. İşte ürünlerimizin her biri için vizyon:

- Basecamp: Proje yönetimi iletişimidir
- Sırt çantası: Hayatın yarım kalan uçlarını bir araya getirin
- Kamp ateşi: IM üzerinden grup sohbeti berbat
- Ta-da Listesi: Bir post-it notuyla rekabet etmek
- Yazı tahtası: Sözcük aşırıya kaçıyor

Örneğin Basecamp ile vizyon "Proje yönetimi iletişimidir" idi. Bir projede etkili iletişimin kolektif sahiplenme, katılım, yatırım ve ivmeye yol açtığını güçlü bir şekilde hissettik. Herkesi ortak bir amaç için aynı sayfada toplar.

Basecamp bunu başarabilirse, diğer her şeyin sıraya gireceğini biliyorduk.

Bu vizyon, Basecamp'ı mümkün olduğunca açık ve şeffaf tutmamızı sağladı. İletişimi bir firma ile sınırlamak yerine, müşterilere de erişim sağladık. İzinler hakkında daha az, tüm katılımcıları katılmaya teşvik etmek hakkında daha fazla düşündük. Vizyon, çizelgeleri, grafikleri, tabloları, raporları, istatistikleri ve elektronik tabloları atlayıp bunun yerine mesajlar, yorumlar, yapılacaklar listeleri ve dosya paylaşımı gibi iletişim önceliklerine odaklanmamızın nedenidir. Vizyonunuzla ilgili büyük kararı önceden verin ve gelecekteki tüm küçük kararlarınız çok daha kolay hale gelsin.

## Beyaz tahta felsefesi

Andy Hunt ve ben bir keresinde bir banka kartı işlem anahtarı yazdık. Önemli bir gereklilik, banka kartı kullanıcısının aynı işlemi hesabına iki kez uygulamaması gerektiği idi. Başka bir deyişle, ne tür bir başarısızlık modu olursa olsun, hata, yinelenen bir işlemi işlemekten ziyade bir işlemi gerçekleştirmeme tarafında olmalıdır.

Bu yüzden, paylaşılan beyaz tahtamıza büyük harflerle yazdık: Err, kullanıcıların lehine.

Yaklaşık yarım düzine diğer özdeyişlere katıldı. Bunlar ortaklaşa olarak, karmaşık bir şey inşa ederken verdiğiniz tüm bu zor kararlara rehberlik etti. Bu yasalar birlikte uygulamamıza güçlü bir iç tutarlılık ve büyük bir dış tutarlılık kazandırdı.

—Dave Thomas, Pragmatik Programcılar

## Mantra yap

Organizasyonların rehberlere ihtiyacı var. Bir taslağa ihtiyaçları var; çalışanların her gün uyandıklarında neden işe gideceklerini bilmeleri gerekir. Bu taslak kısa ve tatlı olmalı ve her şeyi kapsamalıdır: Neden varsın? Seni ne motive eder? Ben buna mantra diyorum - neden var olduğunuzun üç veya dört kelimelik bir açıklaması.

—Guy Kawasaki, yazar ( Make Mantra'dan )

# 16: Ayrıntıları Erkenden Yoksay

## Büyükten küçüğe çalışın

Ayrıntılara deli oluyoruz.

- nesneler arasındaki boşluk
- Mükemmel tip lider
- mükemmel renk
- mükemmel sözler
- Yedi yerine dört satır kod
- %90 vs %89
- 760px vs 750px
- 39\$/ay vs. 49\$/ay

Başarı ve memnuniyet ayrıntılarda gizlidir.

Ancak, ayrıntılarda bulacağınız tek şey başarı değildir. Ayrıca durgunluk, anlaşmazlık, toplantılar ve gecikmeler de bulacaksınız. Bu şeyler moralinizi bozabilir ve şansınızı azaltabilir.

başarı.

Bütün bir gün boyunca kendinizi ne sıklıkla tek bir tasarım veya kod ögesine takılıp kalmış buldunuz?

Bugün kaydettiğiniz ilerlemenin gerçek ilerleme olmadığını ne sıklıkla fark ettiniz? Bu, süreçte çok erken ayrıntılara odaklandığınızda olur. Mükemmeliyetçi olmak için çok zaman var. Daha sonra yap.

Birinci haftadaki başlık yazı tipinizin boyutu hakkında endişelenmeyin. İkinci haftada o mükemmel yeşil tonunu çivilemenize gerek yok. Üçüncü haftada bu "gönder" düğmesini üç piksel sağa taşımanız gerekmez. Şimdilik sayfadakileri alın. O zaman kullan. Çalıştığınızdan emin olun. Daha sonra ayarlayabilir ve mükemmelleştirebilirsiniz.

Siz inşa ettiğiniz şeyi kullandıkça ayrıntılar kendini gösterir. Neye daha çok dikkat edilmesi gerektiğini göreceksiniz. Eksik olanı hissedeceksin. Hangi çukurları açacağınızı bileceksiniz çünkü onlara çarpmaya devam edeceksiniz. İşte o zaman dikkat etmeniz gerekir, daha erken değil.

### Şeytan Ayrıntılarda Gizlidir

Biraz çizim dersi aldıktan sonra "hemen ayrıntıya gir" tavrını gerçekten aştım... Detayları hemen çizmeye başlarsanız, çizimin berbat olacağından emin olabilirsiniz. Aslında, noktayı tamamen kaçıırıyorsunuz.

Tüm sahne için doğru orantıları alarak başlamalısınız. Ardından sahnenizdeki en büyük nesneleri en küçüğüne kadar çizersiniz. Çizim bu noktaya kadar çok gevşek olmalıdır. Ardından hacim kazandırmaktan oluşan gölgelendirme işlemine geçebilirsiniz. Yalnızca üç tonla (açık, orta, koyu) başlarsınız. Bu size bir ton çizimi verir. Ardından çiziminizin her bölümü için üç ton tonunu yeniden değerlendirir ve uygularsınız. Hacimler gelene kadar yapın (birden fazla yineleme gerektirir)...

Büyükten küçüğe çalışın. Hep.

—Patrick Lafleur, Creation Objet Inc. ( Sinyal vs. Gürültüden)

# 17: Sorun Olduğunda Sorun Olur

## Henüz sahip olmadığınız problemler için zaman kaybetmeyin

Bu noktaya gelmeniz iki yılınızı alacaksa, bugün 100.000 kullanıcıya ölçeklendirme konusunda gerçekten endişelenmenize gerek var mı?

Bugün sadece üç programcıya ihtiyacınız varsa, gerçekten sekiz programcı kiralamak zorunda mısınız?

Bir yıl boyunca iki sunucuda çalışabiliyorsanız, şimdi gerçekten 12 üst düzey sunucuya ihtiyacınız var mı?

## Sadece Kanatla

İnsanlar genellikle henüz sahip olmadıkları sorunları çözmeye çalışırken çok fazla zaman harcarlar.

Yapma. Heck, Basecamp'ı müşterileri faturalandırma yeteneği olmadan başlattık! Ürün aylık döngüler halinde faturalandırıldığından, bunu çözmek için 30 günlük bir boşluğumuz olduğunu biliyorduk. Bu süreyi daha acil sorunları çözmek için kullandık ve piyasaya sürüldükten sonra faturalandırmayla uğraştık. İyi çalıştı (ve bizi gereksiz çanlar ve ısıklar olmadan basit bir çözüme zorladı).

Gerçekten zorunda kalana kadar bir şeyleri terletme. Aşırı inşa etmeyin. Gerektiğinde donanım ve sistem yazılımını artırın.

Bir veya iki hafta yavaş olursanız, bu dünyanın sonu değil. Sadece dürüst olun: müşterilerinize bazı büyüyen ağrılar yaşadığınızı açıklayın. Heyecanlanmayabilirler ama samimiyeti takdir edeceklerdir.

Özet: İhtiyacınız olan gerçek bilgilere erişiminiz olduğunda, kararları tam zamanında alın. Bu arada, acilen ihtiyaç duyulan şeylere bolca dikkat edebilirsiniz. bakım.

## 18: Doğru Müşterileri İşe Alın

### Uygulamanız için temel pazarı bulun ve yalnızca onlara odaklanın

Müşteri her zaman haklı değildir. Gerçek şu ki, uygulamanız için kimin haklı kimin haksız olduğunu belirlemeniz gerekiyor. İyi haber şu ki, internet doğru insanları bulmayı hiç olmadığı kadar kolaylaştırıyor.

### Herkesi memnun etmeye çalışırsan kimseyi memnun edemezsin

Basecamp'ı kurduğumuzda pazarlamamızı tasarım firmalarına odakladık. Pazarımızı bu şekilde daraltarak, ürünü jelleştirecek tutkulu müşterileri çekmeyi daha olası hale getirdik. Uygulamanızın gerçekten kime yönelik olduğunu bilin ve onları memnun etmeye odaklanın.

#### Yaptığımız En İyi Çağrı

Campaign Monitor'ü kesinlikle web tasarım pazarına hedefleme kararı, şimdiye kadar yaptığımız en iyi çağrıydı. Hangi özelliklerin gerçekten yararlı olacağını ve daha da önemlisi hangi özelliklerin dışarıda bırakılacağını kolayca belirlememizi sağladı. Daha küçük bir insan grubunu hedefleyerek sadece daha fazla müşteri çekmekle kalmadık, bu müşterilerin hepsinin benzer ihtiyaçları var ve bu da işimizi çok daha kolaylaştırıyor. Campaign Monitor'de bir web tasarımcısı dışında hiç kimseye faydası olmayacak bir sürü özellik vardır.

Bir çekirdek pazara odaklanmak, yazılımınız hakkındaki sözleri yaymayı da çok daha kolay hale getirir. Artık sıkı bir şekilde tanımlanmış bir kitleye sahip olduğumuza göre, çevrimiçi olarak sık sık gittikleri yerlere reklam verebilir, ilginç bulabilecekleri makaleler yayınlayabilir ve genellikle ürünümüz etrafında bir topluluk oluşturabiliriz.

—David Greiner, kurucu, Campaign Monitor



## 19: Daha Sonra Ölçeklendir

# Henüz bir ölçeklendirme sorununuz yok

"Milyonlarca insan kullanmaya başladığında uygulamam ölçeklenecek mi?"

Ne biliyor musun? Bu gerçekten olana kadar bekleyin. Sisteminize aşırı yüklenen çok sayıda insan varsa, o zaman huzzah! Bu olması gereken bir sorun. Gerçek şu ki, web uygulamalarının ezici çoğunluğu bu aşamaya asla ulaşamayacak. Ve aşırı yüklenmeye başlasanız bile, bu genellikle ya hiç ya da hiç bir sorun değildir. Sorunu ayarlamak ve yanıtlamak için zamanınız olacak. Ayrıca, lansmandan sonra ele alınması gereken alanları bulmak için kullanabileceğiniz daha fazla gerçek dünya verisine ve kıyaslamasına sahip olacaksınız.

Örneğin, Basecamp'ı ilk yıl tek bir sunucuda çalıştırdık. Bu kadar basit bir kurulumla yola çıktığımız için, uygulanması sadece bir hafta sürdü. 15 kutudan oluşan bir kümeyle başlamadık veya ölçeklendirme konusunda aylarca endişe duymadık.

Herhangi bir sorun yaşamadık mı? Bir kaç. Ancak, kısa bir yavaşlama gibi korktuğumuz sorunların çoğunun müşteriler için o kadar da önemli olmadığını da fark ettik. İnsanları döngüde tuttuğunuz ve durum hakkında dürüst olduğunuz sürece, anlayacaklardır. Geriye dönüp baktığımızda, "mükemmel kurulumu" yaratmak için lansmanı aylarca ertelediğimiz için çok memnunuz.

Başlangıçta, ölçeklenebilirlik ve sunucu çiftlikleri üzerinde kafa yormak yerine sağlam bir çekirdek ürün oluşturmayı önceliğiniz yapın. Harika bir uygulama oluşturun ve ardından çılgınca başarılı olduğunda ne yapacağınız konusunda endişe edin. Aksi takdirde, asla olmayacak bir şeye takılıp enerjinizi, zamanınızı ve paranızı boşa harcayabilirsiniz.

İster inanın ister inanmayın, daha büyük sorun ölçekleme değil, ölçeklemeniz gereken noktaya geliyor. İlk sorun olmadan ikinciye sahip olmayacaksınız.

## Yine de tekrar ziyaret etmelisin

Gerçek şu ki, herkesin ölçeklenebilirlik sorunları var, hiç kimse hizmetlerinin sıfırdan birkaç milyon kullanıcıya ulaşmasıyla tasarım ve mimarisinin neredeyse her yönünü yeniden gözden geçirmeden başa çıkamaz.

— Dare Obasanjo, Microsoft ( [Scaling Up and Startups](#)'tan)

## 20: Görüşlü Yazılım Yapın

### Uygulamanız taraf tutmalı

Bazı insanlar yazılımın agnostik olması gerektiğini savunuyor. Geliştiricilerin özellikleri sınırlamasının veya özellik isteklerini görmezden gelmesinin kibirli olduğunu söylüyorlar. Yazılımın her zaman mümkün olduğunca esnek olması gerektiğini söylüyorlar.

Bunun saçmalık olduğunu düşünüyoruz. En iyi yazılımın bir vizyonu vardır. En iyi yazılım taraf tutar. Ne zaman birisi yazılım kullanıyor, sadece özellik aramıyorlar, bir yaklaşım arıyorlar.

Bir vizyon arıyorlar. Vizyonunuzun ne olduğuna karar verin ve onunla birlikte koşun.

Ve unutmayın, eğer sizin vizyonunuzu beğenmiyorlarsa, insanlar için bir sürü başka vizyonlar var. Asla mutlu edemeyeceğin insanların peşinden koşma.

Harika bir örnek, orijinal wiki tasarımıdır. Ward Cunningham ve arkadaşları, geçmişte işbirliğini belgelemek için ayrılmaz olduğu düşünülen birçok özelliğin wiki'sini kasıtlı olarak çıkardılar.

Belgedeki her değişikliği belirli bir kişiye atfetmek yerine, mülkiyetin görsel temsilinin çoğunu kaldırdılar. İçeriği egosuz ve zamansız hale getirdiler. İçeriği kimin yazdığının veya ne zaman yazıldığının önemli olmadığına karar verdiler. Ve bu bütün farkı yaptı. Bu karar ortak bir topluluk duygusunu teşvik etti ve Wikipedia'nın başarısında kilit bir bileşen oldu.

Uygulamalarımız da benzer bir yol izledi. Tüm insanlar için her şey olmaya çalışmazlar. Bir tavırları var. Gerçekte ortak olan müşteriler ararlar. Vizyonumuzu paylaşan insanlarla konuşuyorlar. Ya otobüstesiniz ya da otobüsten iniyorsunuz.

# 21: Yarım, Yarım Değil

## Yarım ürün değil, yarım ürün oluşturun

Web uygulaması geliştirmeye yönelik "mutfak lavabosu dışında her şey" yaklaşımına dikkat edin. Ortaya çıkan her iyi fikri atın ve ürününüzün yarım yamalak bir versiyonuna sahip olacaksınız. Gerçekten yapmak istediğiniz şey, kıcı tekmeleyen bir ürünün yarısını inşa etmek.

Gerçekten gerekli olana bağlı kalın. İyi fikirler masaya yatırılabilir. Ürününüzün ne olması gerektiğini düşünüyorsanız onu alın ve ikiye bölün. Yalnızca en önemli olanlarla kalıncaya kadar özellikleri azaltın. Sonra tekrar yap.

Basecamp ile sadece mesajlar bölümü ile başladık. Bunun uygulamanın kalbi olduğunu biliyorduk, bu yüzden şimdilik kilometre taşlarını, yapılacaklar listelerini ve diğer öğeleri görmezden geldik. Bu, gelecekteki kararları önseziler yerine gerçek dünya kullanımına dayandırmamıza izin veriyor.

Yalın, akıllı bir uygulamayla başlayın ve ilgi çekmesine izin verin. Ardından, oluşturduğunuz sağlam temele eklemeye başlayabilirsiniz.

## 22: Sadece Önemli Değil

### Yalnızca temel öğeler

"Bunu neden yapmadın veya neden yapmadın?" sorusuna en sevdiğimiz cevap. soru her zaman şudur:

"Çünkü önemli değil." Bu ifade, bir ürünü harika yapan şeyin somutlaşmış halidir.

Neyin önemli olduğunu bulmak ve gerisini dışarıda bırakmak.

Campfire'ı piyasaya sürdüğümüzde, ürünü ilk kez kontrol eden insanlardan şu soruların bazılarını duyduk:

"Neden zaman damgası sadece her 5 dakikada bir? Neden her sohbet hattına zaman damgası atmıyorsunuz?"

Cevap: Sadece önemli değil. Bir konuşmayı ne sıklıkla saniye veya dakika olarak izlemeniz gerekir?

Kesinlikle %95 değil. 5 dakikalık pullar yeterlidir çünkü daha spesifik bir şey önemli değildir.

"Sohbetlerde neden kalın, italik veya renkli biçimlendirmeye izin vermiyorsunuz?" Cevap: Sadece önemli değil.

Bir şeyi vurgulamanız gerekiyorsa, güvenilir büyük harf kilidi tuşunu kullanın veya kelimenin veya ifadenin etrafına birkaç \* işareti koyun. Bu çözümler ek yazılım, teknik destek, işlem gücü veya öğrenme eğrisi gerektirmez. Ayrıca, basit bir metin tabanlı sohbette ağır biçimlendirme önemli değil.

"Belirli bir zamanda odadaki toplam insan sayısını neden göstermiyorsunuz?" Cevap: Sadece önemli değil.

Herkesin adı listeleniyor, yani orada kim olduğunu biliyorsun, ama 12 ya da 16 kişi olsa ne fark eder ki?

Davranışınızı değiştirmiyorsa, o zaman önemli değil.

Bu şeylerin olması güzel olur mu? Emin. Ama onlar vazgeçilmez mi? Gerçekten önemliler mi? Hayır.

Ve bu yüzden onları dışarıda bıraktık. En iyi tasarımcılar ve en iyi programcılar, en iyi becerilere veya en çevik parmaklara sahip olanlar veya Photoshop veya kendi seçtikleri ortamla rock and roll yapabilenler değil, neyin olmadığını belirleyebilenlerdir. Önemli olmak.

İşte asıl kazançlar burada elde edilir.

Harcadığınız zamanın çoğu önemsiz şeylere harcanıyor. İş kesebilir ve önemli olmadığını düşünürseniz, asla hayal etmediğiniz üretkenliğe ulaşabilirsiniz.

## 23: Hayır İle Başlayın

### Özelliklerin uygulanması için sıkı çalışmasını sağlayın

Yarım yamalak bir ürün yerine yarım bir ürün yapmanın sırrı hayır demek.

Bir özelliğe her evet dediğinizde, bir çocuk evlat ediniyorsunuz. Bebeğinizi bir dizi olaydan (örn. tasarım, uygulama, test vb.) geçirmeniz gerekir. Ve bu özellik bir kez orada olduğunda, ona takılıp kalırsınız. Sadece piyasaya sürülen bir özelliği müşterilerden almaya çalışın ve ne kadar sinirlendiklerini görün.

### evet adamı olmayın

Her özelliğin uygulanması için çok çalışmasını sağlayın. Her özelliğin kendini kanıtlamasını sağlayın ve hayatta kalan olduğunu gösterin. "Dövüş Kulübü" gibi. Özellikleri, yalnızca üç gün boyunca verandada durup içeri alınmayı beklemeye razı olduklarında göz önünde bulundurmalısınız.

Bu yüzden hayır ile başlıyorsunuz. Bize veya bizden gelen her yeni özellik talebi, hayırla karşılanır. Dinliyoruz ama hareket etmiyoruz. İlk yanıt "şimdi değil". Bir özellik için istek gelmeye devam ederse, o zaman daha derine bakma zamanının geldiğini anlarız. O zaman ve ancak o zaman, özelliği gerçek olarak düşünmeye başlarız.

Peki onların özellik fikrini benimsemediğinizde şikayet eden insanlara ne diyorsunuz? En başta uygulamayı neden beğendiklerini onlara hatırlatın. "Seviyorsun çünkü hayır diyoruz. Seviyorsun çünkü başka 100 şey yapmıyor. Seviyorsunuz çünkü her zaman herkesi memnun etmeye çalışmıyor."

"Bin Özellik İstemiyoruz"

Steve Jobs, bazı bağımsız plak şirketlerine iTunes Music Store hakkında küçük bir özel sunum yaptı. Günün en sevdiğim repliği, insanların "[x] olur mu?", "[y] eklemeyi düşünüyor musun?" deyip durduklarıydı. Sonunda Jobs, "Bekle, ellerini indir. Dinleyin: iTunes'un sahip olabileceği tüm harika özellikler için binlerce fikriniz olduğunu biliyorum. Ee yapıyor muyuz. Ama biz bin özellik istemiyoruz. Bu çirkin olurdu. İnovasyon her şeye evet demek değildir. En önemli özellikler dışında hepsine HAYIR demekle ilgili."

—Derek Sivers, başkan ve programcı, CD Baby ve HostBaby (varsayılan olarak Say NO'dan) \_\_\_\_\_

## 24: Gizli Maliyetler

### Yeni özelliklerin fiyatını açıklayın

Bir özellik "hayır" aşamasını geçse bile, yine de gizli maliyetlerini ortaya çıkarmanız gerekir.

Örneğin, özellik döngüleri (yani daha fazla özelliğe yol açan özellikler) arayışında olun. Basecamp'a bir toplantı sekmesi eklemek için istekler aldık. Yakından inceleyene kadar yeterince basit görünüyor. Bir toplantı sekmesinin gerektirebileceği tüm farklı öğeleri düşünün: konum, zaman, oda, kişiler, e-posta davetleri, takvim entegrasyonu, destek belgeleri vb. Bu, promosyon ekran görüntülerini, tur sayfalarını, sss'yi değiştirmemiz gerektiği anlamına gelmez. yardım sayfaları, hizmet şartları ve daha fazlası. Siz farkına bile varmadan, basit bir fikir kartopu gibi büyük bir baş ağrısına dönüşebilir.

İhtiyacınız olan her yeni özellik için...

- 1. Hayır deyin.
- 2. Özelliği değerini kanıtlamaya zorlayın.
- 3. Yine "hayır" ise, burada bitirin. "Evet" ise devam edin...
- 4. Ekran(lar)ı/ui'yi çizin.
- 5. Ekranları/ui'yi tasarlayın.
- 6. Kodlayın.
- 7-15. Test et, ince ayar yap, test et, ince ayar yap, test et, ince ayar yap, test et, ince ayar yap...
- 16. Yardım metninin değiştirilmesi gerekip gerekmediğini kontrol edin.
- 17. Ürün turunu güncelleyin (gerekirse).
- 18. Pazarlama kopyasını güncelleyin (gerekirse).
- 19. Hizmet şartlarını güncelleyin (gerekirse).
- 20. Verilen sözlerin yerine getirilip getirilmediğini kontrol edin.
- 21. Fiyatlandırma yapısının etkilenip etkilenmediğini kontrol edin.
- 22. Başlat.
- 23. Nefesini tut.

## 25: Başa Çıkabilir misin?

### Yönetebileceğiniz bir şey oluşturun

Bir ortaklık programı başlatırsanız, muhasebe ve ödemeleri idare edecek sistemleriniz var mı? Belki de her ay bir çek yazmak, imzalamak ve postalamak yerine, insanların üyelik ücretlerine karşılık kredi kazanmasına izin vermelisiniz.

Sırf Google yaptı diye 1 gb'lık alanı ücretsiz olarak vermeyi göze alabilir misiniz? Belki 100 mb'de küçük başlamalı veya yalnızca ödeme yapan hesaplarda yer açmalısınız.

Sonuç: Yönetebileceğiniz ürünler oluşturun ve hizmetler sunun. Söz vermek kolaydır. Onları tutmak çok daha zor. Yaptığınız her ne ise, gerçekten sürdürebileceğiniz bir şey olduğundan emin olun - organizasyonel, stratejik ve finansal olarak.

## 26: İnsan Çözümleri

### Genel kavramlar için yazılım oluşturun ve insanları kendi çözümlerini yaratmaya teşvik edin

İnsanlara sözleşmeler dayatmayın. Bunun yerine herkesin kendi çözümünü bulabilmesi için yazılımınızı genel hale getirin. İnsanlara kendi sorunlarını kendi yöntemleriyle çözmelerine yetecek kadar verin. Ve sonra yoldan çekil.

Ta-da List'i oluşturduğumuzda, birçok şeyi kasıtlı olarak atlادık. Birine yapılacak iş atamanın bir yolu yoktur, bir son tarih işaretlemenin bir yolu yoktur, öğeleri kategorilere ayırmanın bir yolu yoktur, vb.

İnsanların yaratıcı olmasına izin vererek aracı temiz ve düzenli tuttuk. İnsanlar sorunları kendi başlarına nasıl çözeceklerini buldular. Yapılacaklar ögesine bir tarih eklemek isterlerse, öğenin önüne (son tarih: 7 Nisan 2006) ekleyebilirler. Bir kategori eklemek isterlerse, öğenin önüne [Kitaplar] ekleyebilirler. İdeal? Hayır. Sonsuz esnek mi? Evet.

Bu senaryoları özellikle ele almak için yazılım oluşturmaya çalışsaydık, bu endişelerin geçerli olmadığı tüm durumlar için onu daha az kullanışlı hale getirirdik.

Sorunun köküyle ilgili yapabileceğiniz en iyi işi yapın ve sonra kenara çekilin. İnsanlar sizin genel çerçeveniz içinde kendi çözümlerini ve uzlaşımlarını bulacaktır.



## 27: Özellik İsteklerini Unutun

Müşterilerinizin size neyin önemli olduğunu hatırlatmasına izin verin

Müşteriler her şeyi güneşin altında ister. Özellik istekleriyle sizi çığ gibi büyütürler. Sadece ürün forumlarımıza göz atın; Özellik isteği kategorisi, diğerlerini her zaman geniş bir farkla geride bırakır.

"Bu küçük ekstra özellik" veya "bu zor olamaz" veya "bunu eklemek kolay olmaz mıydı" veya "yerleştirmesi sadece birkaç saniye sürer" veya "eğer istersen" gibi şeyler duyacağız. bunu ekledim iki katı kadar ederim" vb.

Elbette insanları istekte bulundukları için suçlamıyoruz. Bunu teşvik ediyoruz ve söyleyeceklerini duymak istiyoruz. Ürünlerimize eklediğimiz çoğu şey müşteri talebi olarak başlar. Ancak, daha önce de belirttiğimiz gibi, ilk cevabınız hayır olmalıdır. Peki, akan tüm bu isteklerle ne yapıyorsunuz? Onları nerede saklıyorsunuz? Onları nasıl yönetiyorsunuz? yapmazsın. Sadece onları okuyun ve sonra atın.

Evet, okuyun, atın ve unutun. Kulağa küfür gibi geliyor ama önemli olanlar yine de köpürmeye devam edecek. Hatırlamanız gereken tek şey bunlar. Bunlar gerçekten gerekli olanlar. Gelen her isteği izleme ve kaydetme konusunda endişelenmeyin.

Müşterilerinizin hafızanız olmasına izin verin. Gerçekten hatırlamaya değerse, sen unutana kadar sana hatırlatacaklar.

Bu sonuca nasıl ulaştık? Basecamp'ı ilk başlattığımızda, bir Basecamp yapılacaklar listesindeki her önemli özellik talebini izledik. Bir istek başka biri tarafından tekrarlandığında, listeyi fazladan bir kare işaretiyle (II veya III veya IIII, vb.) güncelleriz. Bir gün bu listeyi gözden geçireceğimizi ve en çok istenen özelliklerden aşağıya doğru çalışmaya başlayacağımızı düşündük.

Ama gerçek şu ki, ona bir daha hiç bakmadık. Bundan sonra ne yapılması gerektiğini zaten biliyorduk çünkü müşterilerimiz sürekli aynı istekleri tekrar tekrar bize hatırlattı. Her şey gerçek zamanlı olarak gerçekleştiği için bir listeye veya çok sayıda analize gerek yoktu.

Her gün hatırlatıldığında neyin önemli olduğunu unutamazsın.

Ve bir şey daha: Sadece x sayıda insan bir şey talep ediyor diye, onu dahil etmeniz gerektiği anlamına gelmez. Bazen hayır demek ve ürünle ilgili vizyonunuzu korumak daha iyidir.

## 28: Mayo'yu tutun

### İnsanlara ne istemediklerini sorun

Çoğu yazılım anketi ve araştırma sorusu, insanların bir üründe ne istediklerine odaklanır. "Sence hangi özellik eksik?" "Tek bir şey ekleydin, bu ne olurdu?" "Bu ürünü sizin için daha kullanışlı yapan ne olabilir?"

Madalyonun diğer yüzü ne olacak? Neden insanlara ne istemediklerini sormuyorsunuz? "Bir özelliği kaldırabilseydin, bu ne olurdu?" "Ne kullanmıyorsun?" "Yolunuza en çok ne çıkar?"

Cevap daha fazlası değil. Bazen müşteriler için yapabileceğiniz en büyük iyilik, bir şeyi dışarıda bırakmaktır.

### Yenilik Hayır Demekten Doğar

[İnovasyon], yanlış yola girmediğimizden veya çok fazla şey yapmaya çalışmadığımızdan emin olmak için 1000 şeye hayır demekten gelir. Her zaman girebileceğimiz yeni pazarlar hakkında düşünüyoruz, ancak yalnızca hayır diyerek gerçekten önemli olan şeylere konsantre olabilirsiniz.

—Steve Jobs, CEO, Apple ( The Seed of Apple's Innovation'dan)

## 29: Yazılımı Çalıştırma Yarışı

### Gerçek bir şeyi hızlı bir şekilde kurun ve çalıştırın

Yazılım çalıştırmak, ivme kazanmanın, ekibinizi harekete geçirmenin ve işe yaramayan fikirleri temizlemenin en iyi yoludur. İlk günden itibaren bir numaralı önceliğiniz olmalıdır.

Yazılımın daha hızlı çalışmasına yol açacaksa, daha az şey yapmanız, ayrıntıları atlamanız ve işleminizde kısayollar kullanmanız sorun değildir. Oraya vardığınızda, nasıl ilerleyeceğiniz konusunda çok daha doğru bir bakış açısıyla ödüllendirileceksiniz. Hikayeler, tel kafesler, hatta html maketleri sadece yaklaşık değerlerdir. Çalışan yazılım gerçektir.

Gerçek, çalışan yazılımla herkes gerçek anlayışa ve anlaşmaya yaklaşır. Zaten önemli olmadığı ortaya çıkan eskizler ve paragraflar üzerinde hararetli tartışmalardan kaçınırsınız. Önemsiz olduğunu düşündüğünüz kısımların aslında oldukça önemli olduğunu farkındasınız.

Gerçek şeyler gerçek tepkilere yol açar. Ve böylece gerçeğe ulaşırsınız.

### Gerçek Şey Anlaşmaya Yol Açar

Bir grup farklı insan neyin uyumlu olduğunu bulmak için yola çıktığında... tam ölçekli, gerçek şeylerle dalga geçiyorlarsa, bu konudaki görüşleri birbirine yaklaşma eğiliminde olacaktır. Tabii ki, eskiz yapıyorlarsa ya da fikir veriyorlarsa, aynı fikirde olmayacaklar. Ancak, gerçek olanı yapmaya başlarsanız, kişi anlaşmaya varma eğilimindedir.

—Christopher Alexander, Mimarlık Profesörü  
Mimari)

En kısa sürede çalışmasını sağlayın

Uzun bir planlama ve tartışma dönemiyle başlayan ve eş zamanlı geliştirme yapılmayan, zamanlama, maliyet veya işlevsellik açısından başarılı olan büyük ya da küçük bir yazılım projesinde yer aldığımı hiç sanmıyorum. Gereksiz veya uygulanamaz olduğu ortaya çıkan özellikler icat ederek değerli zamanınızı boşa harcamak çok kolay ve bazen eğlencelidir.

Bu, gelişimin tüm seviyeleri için geçerlidir ve “gerçek bir şeyi çalışır duruma getirin” fraktal bir mantradır. Sadece bir bütün olarak proje için geçerli değildir, en azından uygulamanın oluşturulduğu bileşenlerin daha küçük ölçekli geliştirilmesine eşit derecede uygulanabilir.

Mevcut bir anahtar bileşenin çalışan bir uygulaması olduğunda, geliştiriciler kendi uygulama parçasıyla nasıl çalışıp çalışmayacağını anlamak isterler ve genellikle bunu yapmaya çalışırlar.

mümkün olan en kısa sürede kullanın. Uygulama ilk başta mükemmel veya eksiksiz olmasa bile, bu erken işbirliği genellikle tam olarak ihtiyaç duydukları şeyi yapan iyi tanımlanmış arayüzlere ve özelliklere yol açar.

—Matt Hamer, geliştirici ve ürün müdürü, Kinja

— —

## 30: Durulayın ve Tekrarlayın

### Yinelemelerde çalışın

İlk seferde doğru yapmayı beklemeyin. Uygulamanın büyümesine ve sizinle konuşmasına izin verin. Değişmesine ve gelişmesine izin verin. Web tabanlı yazılımla mükemmelliği göndermeye gerek yok. Ekranlar tasarlayın, kullanın, analiz edin ve yeniden başlayın.

Her şeyi önceden almaya odaklanmak yerine, yinelemeli süreç, ilerledikçe bilinçli kararlar vermeye devam etmenizi sağlar. Ayrıca, hemen mükemmellik için çabalamadığınız için etkin bir uygulamayı daha hızlı çalışır hale getirirsiniz. Sonuç, nelere dikkat etmeniz gerektiğine dair gerçek geri bildirim ve gerçek rehberliktir.

### Yinelemeler kurtuluşa yol açar

Zaten daha sonra tekrar yapılacağını biliyorsanız, ilk denemede mükemmelliği hedeflemenize gerek yoktur. Sorunları tekrar gözden geçireceğinizi bilmek, uçup uçmayacaklarını görmek için fikir edinmek için harika bir motive edicidir.

### belki benden daha zekisin

Belki benden çok daha akıllısın.

Tamamen mümkün. Aslında, muhtemeldir. Ancak, çoğu insan gibiyseniz, o zaman benim gibi, göremediğiniz, hissedemediğiniz ve dokunamadığınız şeyleri hayal etmekte zorlanıyorsunuz.

İnsanlar, çevredeki şeylere tepki vermede son derece iyidir. Bir kaplan odaya girdiğinde nasıl paniğe kapılacağımızı ve yıkıcı bir selden sonra nasıl temizleneceğimizi biliyoruz.

Ne yazık ki, önceden plan yapmakta, eylemlerimizin sonuçlarını anlamakta ve gerçekten önemli olan şeylere öncelik vermekte berbatız.

Belki de hepsini kafasında tutabilen birkaç kişiden birisin. gerçekten değil

Önemli olmak.

Web 2.0, herkesin zaten web kullandığını varsayarak başladığımız dünya, akıllı geliştiricilerin bu insani zayıflığı kendileri için çalıştırmalarına izin veriyor. Nasıl? Kullanıcılarınızın bu konuda bir şeyler yapmak için hala zamanları varken ne düşündüklerini size söylemelerine izin vererek.

Ve bu son cümle neden bu şekilde geliştirmeniz gerektiğini ve nasıl tanıtım/lansman yapmak isteyebileceğinizi açıklıyor.

Hikayeni doğru anla. Parçaların çalıştığından emin olun. Ardından başlatın ve gözden geçirin. Hiç kimse hepimiz kadar akıllı değil.

—Seth Godin, yazar/girişimci

# 31: Fikirden Uygulamaya

## Beyin fırtınasından eskizlere, HTML'den kodlamaya geçin

Gerçeği almak için kullandığımız süreç:

### beyin fırtınası

Fikirlerle gel. Bu ürün ne yapacak? Basecamp için kendi ihtiyaçlarımıza baktık. Proje güncellemelerini yayınlamak istedik. Müşterilerin katılmasını istedik. Projelerin kilometre taşları olduğunu biliyorduk. İnsanların eski şeyleri kolayca gözden geçirebilmesi için arşivleri merkezileştirmek istedik. Tüm projelerimizde neler olup bittiğine dair büyük bir resim, kuş bakışı görmek istedik. Birlikte, bu varsayımlar ve diğer birkaçı, temelimiz olarak hizmet etti.

Bu aşama önemsiz ayrıntılarla ilgili değildir. Bu büyük sorularla ilgili. Uygulamanın ne yapması gerekiyor? Faydalı olduğunu nasıl anlayacağız? Tam olarak ne yapacağız? Bu, piksel düzeyindeki tartışmalarla değil, üst düzey fikirlerle ilgilidir. Bu aşamada, bu tür ayrıntılar sadece anlamlı değildir.

### Kağıt eskizler

Eskizler hızlı, kirli ve ucuzdur ve tam olarak böyle başlamak istersiniz. Bir şeyler çizin. Karalama şeyler. Kutular, daireler, çizgiler. Fikirlerinizi kafanızdan çıkarın ve kağıda dökün. Bu noktada amaç kavramları kaba arayüz tasarımlarına dönüştürmek olmalıdır. Bu adım tamamen deneme ile ilgilidir. Yanlış cevaplar yoktur.

### HTML ekranları oluşturun

Bu özelliğin (veya daha uygunsa bölüm veya akış) html sürümünü yapın. Gerçek bir şey yayınlayın, böylece herkes ekranda nasıl görüldüğünü görebilir.

Basecamp için önce “mesaj gönder” ekranını, ardından “mesaj düzenle” ekranını yaptık ve oradan devam etti.

Henüz herhangi bir programlama kodu yazmayın. Sadece html ve css'de bir maket oluşturun. Uygulama daha sonra gelir.

# Kodla

Model iyi görüldüğünde ve gerekli işlevselliği yeterince gösterdiğinde, devam edin ve programlama kodunu girin.

Tüm bu süreç boyunca esnek kalmayı ve birden fazla yineleme beklemeyi unutmayın. Herhangi bir belirli adımın çıktısını atmaktan çekinmemelisiniz ve işler berbat hale gelirse yeniden başlamalısınız. Bu döngüden defalarca geçmek doğaldır.



## 32: Tercihlerden Kaçının

### Küçük ayrıntılara karar verin, böylece müşterileriniz

Zor bir kararla karşı karşıyasınız: Her sayfaya kaç mesaj ekliyoruz? İlk eğiliminiz, “İnsanların 25, 50 veya 100'ü seçebileceği bir tercih yapalım” demek olabilir. Gerçi bu kolay çıkış yolu. Sadece bir karar ver.

### Tercihler, zor kararlar vermekten kaçınmanın bir yoludur

En iyi yolu seçmek için uzmanlığınızı kullanmak yerine, onu müşterilerin eline bırakıyorsunuz.

Onlara bir iyilik yapıyormuş gibi görünebilirsin ama onlar için sadece meşgul bir iş yapıyorsun (ve muhtemelen yeterince meşguller). Müşteriler için sonsuz sayıda seçeneğe sahip tercih ekranları bir lütuf değil, baş ağrısıdır. Müşteriler her küçük ayrıntıyı düşünmek zorunda kalmamalı - sizin sorumluluğunuz olması gerektiğinde bu yükü onlara yüklemeyin.

Tercihler de kötüdür çünkü daha fazla yazılım oluştururlar. Daha fazla seçenek daha fazla kod gerektirir. Ayrıca yapmanız gereken tüm ekstra testler ve tasarımlar da var. Ayrıca, hiç görmediğiniz tercih permütasyonları ve arayüz ekranları ile karşılaşacaksınız. Bu, bilmediğiniz hatalar anlamına gelir: bozuk düzenler, bozuk tablolar, garip sayfalandırma sorunları vb.

### Arama yap

Müşterileriniz adına basit kararlar alın. Basecamp'ta yaptığımız buydu. Sayfa başına mesaj sayısı 25'tir. Genel bakış sayfasında son 25 öge gösterilir. Mesajlar ters kronolojik sırada sıralanır. En son beş proje gösterge tablosunda gösterilir. Herhangi bir seçenek yok. Tek yolu bu.

Evet, kötü bir arama yapabilirsiniz. Ama ne olmuş yani. Bunu yaparsanız, insanlar şikayet eder ve size bunu söyler. Her zaman olduğu gibi, ayarlayabilirsiniz. Gerçekleşmek, anında değişebilmekle ilgilidir.

### Tercihlerin Bir Maliyeti Vardır

Tercihlerin bir maliyeti olduğu ortaya çıktı. Elbette bazı tercihlerin de önemli faydaları vardır ve bunlar çok önemli arayüz özellikleri olabilir. Ancak her birinin bir fiyatı vardır ve değerini dikkatlice düşünmeniz gerekir. Birçok kullanıcı ve geliştirici bunu anlamıyor ve tercihleri için çok fazla maliyet ve çok az değerle sonuçlanıyor...

Tembelce tercihler eklemek yerine Just Work'ün iyi varsayılanlara sahip olmasıyla ilgili, bu da genel kullanıcı arayüzünü doğal olarak doğru yöne yönlendiriyor.

—Havoc Pennington, teknoloji lideri, Red Hat ( Özgür yazılım ve iyi kullanıcı arayüzlerinden)

# 33: “Bitti!”

## Kararlar geçicidir, bu yüzden aramayı yapın ve devam edin

Tamamlandı. Bunu sihirli bir kelime olarak düşünmeye başlayın. Bitirdiğinizde, bir şeylerin başarıldığı anlamına gelir. Bir karar verildi ve devam edebilirsiniz. Bitti, inşa ediyorsunuz demektir itme.

Ama bekle, ya işleri batırırsan ve yanlış arama yaparsan? Önemli değil. Bu beyin ameliyatı değil, bir web uygulaması. Söylemeye devam ettiğimiz gibi, yine de süreç boyunca özellikleri ve fikirleri birçok kez tekrar gözden geçirmeniz gerekecek. Ne kadar plan yaparsan yap, yine de yarı yarıya ihtimalin var. Bu yüzden “analiz yoluyla felç” şeyini yapmayın. Bu sadece ilerlemeyi yavaşlatır ve morali bozar.

Bunun yerine, ilerlemenin ve ilerlemenin önemine değer verin. Karar verme ritmine girin. Hızlı, basit bir arama yapın ve ardından geri dönün ve işe yaramazsa bu kararı değiştirin dışarı.

Kararların geçici olduğunu kabul edin. Hataların olacağını kabul edin ve bunları çabucak düzelttebildiğiniz sürece önemli olmadığını anlayın. Yürütün, momentum oluşturun ve devam edin.

## Cellat Ol

İnsanların fikirlere karşı bu kadar koruyucu olduğunu duymak çok komik. (Bana en basit fikri anlatmak için bir anda imzalamamı isteyenler.)

Bana göre fikirlerin, uygulanmadıkça hiçbir değeri yoktur. Onlar sadece bir çarpan. İnfaz milyonlar değerinde.

Açıklama:

- Kötü fikir = -1
- Zayıf fikir = 1
- Yani fikir = 5
- İyi fikir = 10
- Harika fikir = 15
- Harika fikir = 20
- Yürütme yok = \$1
- Zayıf yürütme = 1000 \$
- Şöyle böyle yürütme = 10.000 \$

- İyi uygulama = 100.000\$
- Harika uygulama = 1.000.000\$
- Mükemmel uygulama = 10.000.000\$

Bir iş yapmak için ikisini çarpmanız gerekir.

İcrası olmayan en parlak fikir 20 dolar değerindedir. En parlak fikir, 20.000.000 \$ değerinde harika bir uygulama gerektirir.

Bu yüzden insanların fikirlerini duymak istemiyorum. İnfazlarını görene kadar ilgilenmiyorum.

—Derek Sivers, başkan ve programcı, CD Baby ve HostBaby . \_\_\_\_\_ .

## 34: Vahşı Doğada Test Edin

### Uygulamanızı gerçek dünya kullanımıyla test edin

Uygulamanızı gerçek yollarla kullanan gerçek kişilerin yerini hiçbir şey tutamaz. Gerçek verileri alın.

Gerçek geri bildirim alın. Sonra bu bilgilere göre geliştirin.

Resmi kullanılabilirlik testi çok katı. Laboratuvar ayarları gerçeği yansıtmaz. Birinin omzunun üstünde durursanız, neyin işe yarayıp yaramadığına dair bir fikir edirsiniz, ancak insanlar genellikle kamera önünde iyi performans göstermezler. Bir başkası izlediğinde, insanlar hata yapmamaya özellikle dikkat ederler - yine de tam olarak aradığınız şey hatalardır.

Bunun yerine, beta özelliklerini gerçek uygulamanın içinde birkaç seçilmiş kişiye yayınlayın. Yayınlanan özelliklerin yanı sıra beta özelliklerini kullanmalarını sağlayın. Bu, bu özellikleri insanların gerçek verilerine ve gerçek iş akışına sunacaktır. Ve işte gerçek sonuçları alacağınız yer burasıdır.

Ayrıca, bir yayın sürümüne ve bir beta sürümüne sahip değilsiniz. Her zaman aynı şey olmalıdır.

Ayrı bir beta sürümü yalnızca yüzeysel bir inceleme yapacaktır. Bazı beta özellikleri serpiştirilmiş gerçek sürüm, tam antrenmanı alacak.

## Beta Kitabı

Geliştiriciler kodu serbest bırakmak konusunda gerginlerse, yayıncılar ve yazarlar kitap yayınlamaktan korkarlar. Bir kitap kağıda döküldüğünde, onu değiştirmek çok zor bir iş olarak görülür. (Gerçekten değil, ancak eski teknolojilerle ilgili sorunların algılanması ve hatıraları hala sektörde oyalanıyor.)

Bu nedenle, yayıncılar, kitapları yayınlanmadan önce "doğru" hale getirmek için çok zahmete (ve masrafa) girerler.

Rails ile Çevik Web Geliştirme kitabını yazdığımda, geliştiriciler arasında çok fazla bastırılmış talep vardı: kitabı bize şimdi verin — Rails hakkında bilgi edinmek istiyoruz. Ama bir yayıncının zihniyetine düşmüştüm. "Henüz hazır değil," derdim. Ancak topluluktan gelen baskı ve David Heinemeier Hansson'dan gelen bazı sözler fikrimi değiştirdi. Kitabı tamamlanmadan yaklaşık 2 ay önce pdf formatında yayınladık. Sonuçlar muhteşemdi. Sadece çok kitap satmakla kalmadık, aynı zamanda geri bildirim aldık - çok fazla geri bildirim. Okuyucuların yorumlarını yakalamak için otomatik bir sistem kurdum ve sonunda neredeyse 850 rapor veya yazım hatası, teknik hata ve yeni içerik önerileri aldım. Neredeyse hepsi son kitaba doğru yol aldı.

Bir kazan-kazandı: Çok daha gelişmiş bir basılı kitap teslim etmem gerekiyordu ve topluluk istedikleri bir şeye erken erişim elde etti. Ve rekabetçi bir yarıştıysanız, bir şeyi daha erken ortaya çıkarmak, insanların rekabetinize değil, size bağlı kalmasına yardımcı olur.

—Dave Thomas, Pragmatik Programcılar

## çabuk yap

1. Yapmaya değer olup olmadığına karar verin ve eğer

öyleyse: 2. Çabuk yapın - mükemmel değil. sadece yap.

3. Kaydet. yükle. 4. İnsanların ne

düşündüğünü görün

Bir şeye yeni özellikler eklemek konusunda her zaman isteksiz olsam da, bir kez "evet!" Bir şeyin yapmaya değer olduğuna karar verme anı, genellikle birkaç saat sonra web sitesinde yayınlanır, kusurlu ancak başlatılır, geri bildirimin gelecekteki iyileştirmeyi yönlendirmesine izin verir.

—Derek Sivers, başkan ve programcı, CD Baby ve HostBaby .

## 35: Zamanınızı Küçültün

### Yerle bir etmek

Haftalara veya aylara yayılan tahminler fantezilerdir. Gerçek şu ki, o kadar önceden ne olacağını bilmiyorsunuz.

O yüzden zamanınızı küçült. Zaman dilimlerini daha küçük parçalara ayırmaya devam edin. 12 haftalık bir proje yerine 12 haftalık bir proje olarak düşünün. 30 saatten fazla süren görevleri tahmin etmek yerine, onları daha gerçekçi 6-10 saatlik parçalara ayırın. Ardından adım adım ilerleyin.

Aynı teori diğer problemler için de geçerlidir. Aklınızı sarmayacak kadar büyük bir sorunla mı karşılaşıyorsunuz? Yerle bir etmek. Onları sindirebilecek duruma gelene kadar problemleri daha küçük parçalara ayırmaya devam edin.

#### Daha Küçük Görevler ve Daha Küçük Zaman Çizelgeleri

Yazılım geliştiriciler özel bir tür iyimserdir: bir programlama görevi sunulduğunda, "Bu kolay olacak! Hiç de fazla zaman almayacak."

Bu nedenle, bir programcıya büyük bir görevi tamamlaması için üç hafta verin, o iki buçuk erteleyerek ve ardından bir programlama yaparak geçirecektir. Görevin görüldüğünden daha karmaşık olduğu ortaya çıktığından, program dışı sonuç muhtemelen yanlış gereksinimleri karşılayacaktır. Artı, ekibin üç hafta önce ne üzerinde anlaştığını kim hatırlayabilir?

Bir programcıya küçük, özel bir modülü kodlaması için bir öğleden sonra verin ve o, bir sonrakine geçmeye hazır olarak onu çalıştıracaktır.

Daha küçük görevler ve daha küçük zaman çizelgeleri daha kolay yönetilebilir, daha az olası gereksinim yanlış anlamalarını gizler ve fikrinizi değiştirmek veya yeniden yapmak için daha az maliyetlidir. Daha küçük zaman çizelgeleri, geliştiricilerin katılımını sağlar ve onlara bir başarı hissinin tadını çıkarmak için daha fazla fırsat verir ve "Ah, bunu yapmak için çok zamanım var. Şimdilik iTunes arşivimdeki şarkıları derecelendirmeyi bitirmeme izin verin."

—Gina Trapani, web geliştiricisi ve Lifehacker editörü, üretkenlik ve yazılım kılavuzu

#### Gerçek Faktörler

Bir dahaki sefere, birisi sizi bilinmeyen bir soruya kesin bir cevap için sıkıştırmaya çalıştığında - ister son tarih, ister nihai proje maliyeti veya Büyük Kanyon'a sığacak süt miktarı olsun - sadece havayı alarak başlayın. Oda: "Bilmiyorum" deyin.

Bu, güvenilirliğinize zarar vermek şöyle dursun, karar verirken gösterdiğiniz özeni gösterir. Sadece akıllı görünmek için kelimeler söylemeyeceksin. Ayrıca, soruyu işbirlikçi bir konuşma olarak yeniden çerçeveleyerek oyun alanını düzleştirir. Tahmininizin ne kadar kesin olması gerektiğini (ve nedenini) öğrenerek, sayıların arkasındaki gerçek faktörler hakkında ortak bir anlayış geliştirmek için birlikte çalışabilirsiniz.

—Merlin Mann, 43folders.com'un yaratıcısı ve editörü

## Yüzünüze Bakarken Tek Sorunu Çözün

Son zamanlarda web'de gerçekleşmesi en sevdiğim şey, "nofollow" özelliğinin serbest bırakılması ve benimsenmesidir. Daha önce kimse bundan bahsetmedi. Bir grup yahoo'nun semantik veya dilbilgisel doğasını tartışabileceği hiçbir konferans veya komite yoktu. Basit bir fikri 20 satırlık bir xml snippet'ine dönüştürebilecek hiçbir rfc yok, sadece nasıl kullanılacağını bulmak için okumam ve sonra kullanmamam gerekiyordu çünkü .3 sürümü için mi biçimlendirdiğimden emin değildim veya 3.3b.

Basittir, etkilidir, bir seçenek isteyen insanlar için bir seçenek sağladı - ve bu, spesifikasyonları veya saygıyı umursamayan bir web popülasyonu ile uğraşırken çok daha önemlidir.

Bazen sonraki yirmi sorunu çözmek, yüzümüze bakan birini çözmek kadar yararlı ya da ihtiyatlı değildir. Bu sadece spam'e karşı küçük bir zafer değildi (spam'a karşı tüm zaferler küçüktür), aynı zamanda bir web geliştiricisi olmanın basit ve hızlı sonuçlarından hoşlanan bizler için bir zaferdi.

—Andre Torrez, programcı ve Federated Media Publishing'de Mühendislik Başkan Yardımcısı



## 36: Birlik

### Silolara bölünmeyin

Çok fazla şirket tasarım, geliştirme, metin yazarlığı, destek ve pazarlamayı farklı silolara ayırıyor. Uzmanlığın avantajları olsa da, çalışanların web uygulamasının tüm bağlamı yerine sadece kendi küçük dünyalarını gördükleri bir durum yaratır.

Süreç boyunca sağlıklı bir karşılıklı diyalog olması için ekibinizi mümkün olduğunca entegre edin. Bir kontrol ve denge sistemi kurun. Çeviride işlerin kaybolmasına izin vermeyin. Metin yazarlarının tasarımcılarla çalışmasını sağlayın. Destek sorgularının geliştiriciler tarafından görüldüğünden emin olun.

Daha da iyisi, geliştirme sırasında farklı şapkalar takabilecek birden fazla yeteneğe sahip kişileri işe alın. Sonuç daha uyumlu bir ürün olacaktır.

## 37: Yalnız Zaman

### İnsanların işleri halletmek için kesintisiz zamana ihtiyacı var

Basecamp, dört şehir ve sekiz zaman dilimine yayılmıştır. Provo, Utah'tan Kopenhag, Danimarka'ya, beşimiz sekiz saat ayırırız. Bu sekiz saatlik farkın olumlu bir yan etkisi yalnız geçirilen zamandır.

Gün içinde hep birlikte çalıştığımız sadece 4-5 saat var. Diğer zamanlarda, Danimarka'da bulunan David çalışırken us takımı uyuyor. Zamanın geri kalanında David uyurken çalışıyoruz. Bu bize günün yaklaşık yarısını birlikte ve diğer yarısını yalnız verir.

Bilin bakalım günün hangi bölümünde en çok işi bitiriyoruz? Yalnız kısım. Gerçekten o kadar şaşırtıcı değil. Birçok insan, rahatsız edilmedikleri zamanlarda sabah erken veya gece geç saatlerde çalışmayı tercih eder.

Rahatsız olmadığınızda uzun bir esneme yaptığınızda bölgeye girebilirsiniz. Bölge, en üretken olduğunuz zamandır. Çeşitli görevler arasında zihin değiştirmek zorunda olmadığınız zamandır. Bu, bir soruyu yanıtlamak, bir şey aramak, e-posta göndermek veya anlık ileti yanıtlamak için kesintiye uğramadığınız zamandır. Yalnız bölge, gerçek ilerlemenin kaydedildiği yerdir.

Bölgeye girmek zaman alır. İşte bu yüzden kesinti sizin düşmanınızdır. Bu, rem uykusu gibidir - sadece rem uykusuna gitmezsiniz, önce siz uyursunuz ve rem uykusuna gidersiniz. Herhangi bir kesinti sizi baştan başlamaya zorlar. rem, gerçek uyku büyüünün gerçekleştiği yerdir. Tek başına zaman dilimi, gerçek geliştirme büyüünün gerçekleştiği yerdir.

İşyerinde bir kural belirleyin: Günün yarısını yalnız geçirin. 10:00 - 14:00 saatleri arasında kimse birbiriyle konuşamaz (öğle yemeği hariç). Veya günün ilk veya son yarısını tek başına zaman dilimi yapın. Verimliliği öldüren kesintileri önlemek için bu dönemin bitişik olduğundan emin olun.

Başarılı bir yalnız zaman periyodu, iletişim bağımlılığından kurtulmak anlamına gelir. Yalnız olduğunuz zamanlarda anlık mesajlaşma, telefon görüşmeleri ve toplantılardan vazgeçin. Anında yanıt gerektirecek herhangi bir e-posta dizisinden kaçının. Kapa çeneni ve işe koyul.

### Groove'a Girin

Hepimiz, bilgi çalışanlarının en iyi şekilde, tamamen işlerine konsantre oldukları ve çevrelerinden tamamen uzak oldukları "bölgede" olarak da bilinen "akışa" girerek çalıştıklarını biliyoruz. Zaman kavramını kaybederler ve mutlak konsantrasyon yoluyla harika şeyler üretirler... sorun şu ki, alanın dışına çıkmak çok kolay. Gürültü, telefon görüşmeleri, öğle yemeğine çıkmak,

Kahve içmek için Starbucks'a arabayla 5 dakika gitmek zorunda kalmanız ve iş arkadaşlarınız tarafından yapılan kesintiler - özellikle iş arkadaşlarınız tarafından yapılan kesintiler - hepsi sizi alanın dışına iter. Size bir soru soran bir iş arkadaşınız tarafından 1 dakikalık bir kesintiye uğrarsanız ve bu, yeniden üretken olmanız için yarım saatinizi alacak kadar konsantrasyonunuzu bozarsa, genel üretkenliğiniz ciddi bir sıkıntıya girer.

—Joel Spolsky, yazılım geliştiricisi, Fog Creek Software (Bu İnsanlar (Orijinal Olmayan) Fikirlerini Nereden Alıyorlar?)

## 38: Toplantılar Zehirlidir

### toplantı yapma

Gerçekten bir toplantıya ihtiyacın var mı? Toplantılar genellikle bir kavram yeterince açık olmadığına ortaya çıkar. Bir toplantıya başvurmak yerine, konsepti basitleştirmeye çalışın, böylece e-posta, im veya Kamp ateşi aracılığıyla hızlıca tartışabilirsiniz. Amaç toplantılardan kaçınmaktır. Bir toplantıda harcamaktan kaçındığınız her dakika, bunun yerine gerçek işi halledebileceğiniz bir dakikadır.

Verimlilik için bir toplantıdan daha zehirli bir şey yoktur. İşte bunun birkaç nedeni:

- İş gününüzü, doğal iş akışınızı bozan küçük, tutarsız parçalara bölerler.
- Genellikle kelimeler ve soyut kavramlarla ilgilidirler, gerçek şeylerle değil (bir kod parçası veya bazı arayüz tasarımı gibi)
- Genellikle dakikada son derece az miktarda bilgi iletirler.
- Çoğu zaman, herkesin zamanını saçmalıklarla harcamaya sırasını kaçınılmaz olarak alan en az bir moron içerirler.
- Yoğun karda bir Chicago taksisinden daha kolay konudan uzaklaşıyorlar
- Sık sık gündemleri o kadar belirsiz ki kimse ne hakkında olduğundan emin değil.
- İnsanların nadiren yaptığı kapsamlı bir hazırlık gerektirirler.

Kesinlikle bir toplantı yapmanız gereken zamanlar için (bu nadir bir olay olmalıdır), şu basit kurallara uyun:

- 30 dakikalık bir zamanlayıcı ayarlayın. Zil çaldığında toplantı biter. Dönem.
- Mümkün olduğunca az kişiyi davet edin.
- Açık bir gündem olmadan asla toplantı yapmayın.

### Daha az toplantı yapın

Çok fazla toplantı var. Mantıklı olmayan veya verimsiz olan toplantıları geri itin. Yalnızca tartışacak önemli bir iş sorunuz olduğunda ve girdi, onay veya anlaşma istediğinizde veya buna ihtiyacınız olduğunda bir toplantı ayırtın. O zaman bile, herkesi ve kardeşlerini davet etme dürtüsüne karşı koy - insanların zamanını gereksiz yere boşa harcamaya.

—Lisa Haneberg, yazar ( Don't Let Meetings Rule!dan)

### Yerle bir etmek

Projeler büyüdükçe, insanların eklenmesi azalan bir getiriye sahiptir. En ilginç nedenlerden biri, artan iletişim kanallarıdır. İki kişi sadece birbirleriyle konuşabilir;

sadece tek bir iletişim yolu var. Üç işçinin üç iletişim yolu vardır; 4 var 6. Aslında, bağlantıların büyümesi üsteldir... Çok geçmeden notlar ve toplantılar tüm iş gününü tüketir.

Çözüm açık: Bu iletişim bağlantılarını azaltmak için ekipleri daha küçük, özerk ve bağımsız birimlere ayırın.

Benzer şekilde, programları daha küçük birimlere bölün. Sorunun büyük bir kısmı bağımlılıklardan (genel değişkenler, işlevler arasında geçen veriler, paylaşılan donanım vb.) kaynaklandığından, birimler arasındaki bağımlılıkları ortadan kaldırmak veya en aza indirmek için programı bölümlendirmenin bir yolunu bulun.

—Ganssle Grubu ( Keep It Small'dan) \_\_\_\_\_

## 39: Küçük Zaferleri Arayın ve Kutlayın

### Bugün bir şey yayınla

Yazılım geliştirmede en önemli şey motivasyondur. Motivasyon yereldir - şu anda üzerinde çalıştığınız şey sizi motive etmiyorsa, muhtemelen olması gerektiği kadar iyi olmayacaktır. Aslında, muhtemelen berbat olacak.

Uzun, uzun süreli serbest bırakma döngüleri, motivasyon katilleridir. Kutlamalar arasına çok fazla zaman ayırırlar. Öte yandan, kutlayabileceğiniz hızlı galibiyetler harika motive edicilerdir. Uzun serbest bırakma döngülerinin hızlı kazançları bozmasına izin verirsiniz, motivasyonu öldürsünüz. Bu da ürününüzü öldürebilir.

Bu nedenle, aylarca süren bir yayın döngüsünün ortasındaysanız, bazı küçük zaferler için haftada bir gün (veya iki haftada bir) ayırın. Kendinize “Ne yapabiliriz ve 4 saat içinde serbest bırakabiliriz?” Diye sorun. Ve sonra yap. Olabilir...

- Yeni basit bir özellik
- Mevcut bir özellik için küçük bir geliştirme
- Destek yükünü azaltmak için bazı yardım metinlerini yeniden yazma
- Gerçekten ihtiyacınız olmayan bazı form alanlarını kaldırma

Bu 4 saatlik hızlı kazanımları bulduğunuzda, kutlama bulacaksınız. Bu moral oluşturur, motivasyonu artırır ve ekibin doğru yönde ilerlediğini bir kez daha teyit eder.

## 40: Daha Az İşe Alın ve Daha Sonra İşe Alın

### Hızlı gitmek için yavaş ekleyin

Erken ya da geç büyümeye gerek yok. En iyi 100 kişiye erişiminiz olsa bile, hepsini bir kerede işe almaya çalışmak yine de kötü bir fikirdir. Bu kadar çok insanı tutarlı bir kültüre hemen asimile etmenin hiçbir yolu yok. Antrenman baş ağrıları, kişilik çatışmaları, iletişim gecikmeleri, farklı yönlere giden insanlar ve daha fazlasını yaşayacaksınız.

Bu yüzden işe almayın. Yok canım. İnsanları işe almayın. Başka bir yol arayın. Sana yük olan iş gerçekten gerekli mi? Ya yapmazsan? Sorunu bunun yerine bir parça yazılım veya bir uygulama değişikliği ile çözebilir misiniz?

GE'nin eski CEO'su Jack Welch ne zaman birini kovsa, hemen yenisini işe almazdı. O kişi ve o pozisyon olmadan ne kadar süre idare edebileceğini görmek istiyordu. Bu teoriyi test etmek için insanları kovmayı kesinlikle savunmuyoruz, ancak Jack'in bir şeyler üzerinde olduğunu düşünüyoruz: Düşündüğünüz kadar çok insana ihtiyacınız yok.

Başka bir yol yoksa, kiralamayı düşünün. Ancak tam olarak kime ulaşacağınızı, onları işle nasıl tanıştıracığınızı ve onlardan tam olarak nasıl dindirmesini beklediğiniz acıyı bilmelisiniz.

### Brooks yasası

İnsanları geç bir yazılım projesine eklemek, onu daha sonra yapar.

—Fred Brooks

### Programlama ve Mozart'ın Ağıtı

Tek bir görev üzerinde çalışan tek bir iyi programcının koordinasyon veya iletişim yükü yoktur. Aynı görev üzerinde çalışan beş programcı koordine ve iletişim kurmalıdır. Bu çok zaman alır... Birkaç iyi programcı yerine çok sayıda vasat programcı kullanmanın asıl sorunu, ne kadar uzun süre çalışırlarsa çalışsınlar asla büyük programcıların üretebilecekleri kadar iyi bir şey üretememeleridir. Beş Antonio Salieris Mozart'ın Ağıtını yapmayacak. Durmadan. 100 yıl çalışırlarsa olmaz.

—Joel Spolsky, yazılım geliştiricisi, Fog Creek Software ( Hitting the High Notes'tan)

## 41: Lastikleri Tekmele

### Önce potansiyel çalışanlarla test bazında çalışın

Bir portföye, özgeçmişe, kod örneğine veya önceki çalışmaya bakmak bir şeydir. Gerçekten biriyle çalışmak başka bir şey. Mümkün olduğunda, potansiyel yeni ekip üyelerini bir "test sürüşü" için dışarı çıkarın.

Birini işe almadan önce onlara üzerinde düşünmeleri için küçük bir proje veriyoruz. Projeyi nasıl ele aldıklarını, nasıl iletişim kurduklarını, nasıl çalıştıklarını vb. görüyoruz. Birkaç ekran tasarlarken veya kodlarken biriyle çalışmak size tonlarca fikir verecektir. Doğru havanın orada olup olmadığını oldukça çabuk öğreneceksiniz.

Planlama bu tür şeyler için zor olabilir ama sadece 20 veya 40 saat olsa bile hiç yoktan iyidir. İyi ya da kötü bir uyum varsa, bariz olacaktır. Ve değilse, her iki taraf da önce durumu test ederek kendilerini birçok sorun ve riskten kurtarır.

#### Küçük başla

Başlamak için küçük bir test ödevi deneyin. Tüm işlerinize aynı anda atlamayın. Yeni [sanal asistanınıza] üzerinde çalışması ve kimyanın nasıl geliştiğini görmesi için bir veya iki test projesi verin. Başlangıçta, gül renkli gözlüklerle olası sorunları örtbas etmek çok kolaydır. Bunun bir deneme çalışması olduğunu açıkça belirtin.

—Suzanne Falter-Barns, yazar/yaratıcılık uzmanı ( Mükemmel VA'yı Nasıl Bulunur ve Saklanır'dan )



## 42: Sözlere Değil Eylemler

### Potansiyel teknoloji çalışanlarını açık kaynak katkılarına göre değerlendirin

Derecelere, özgeçmişlere vb. dayalı teknik pozisyonlar için tipik işe alma yöntemi birçok yönden aptalcadır. Birinin derecesinin veya not ortalamasının nereden geldiği gerçekten önemli mi? Bir özgeçmiş veya referansa gerçekten güvenebilir misiniz?

Açık kaynak, teknik insanları işe alması gerekenler için bir hediyedir. Açık kaynak ile, uzun bir süre boyunca birinin çalışmasını ve katkılarını - iyi ve kötü - takip edebilirsiniz.

Bu, insanları yalnızca sözleriyle değil, eylemleriyle yargılayabileceğiniz anlamına gelir. Gerçekten önemli olan şeylere dayanarak bir karar verebilirsiniz:

- İşin kalitesi Birçok programcı konuşmayı konuşabilir, ancak yürüyüş zamanı geldiğinde takılır. Açık kaynakla, bir kişinin programlama becerilerinin ve uygulamalarının en önemli ayrıntılarını elde edersiniz.
- Kültürel perspektif Programlama tamamen kararlarla ilgilidir. Fazlası ve çok daha fazlası. Kararlar, kültürel bakış açınız, değerleriniz ve idealleriniz tarafından yönlendirilir. Kültürel bir eşleşme olup olmadığını görmek için bir adayın kodlama, test etme ve topluluk tartışmalarında verdiği belirli kararlara bakın. Burada bir uyum yoksa, her karar bir mücadele olacaktır.
- Tutku düzeyi Tanım olarak, açık kaynağa katılım en azından biraz tutku gerektirir. Aksi halde neden bu kişi boş zamanını ekranın karşısında oturarak geçirin ki? Açık kaynak katılımının miktarı genellikle bir adayın programlamaya gerçekten ne kadar önem verdiğini gösterir.
- Tamamlanma yüzdesi Bir kişi bir şeyler yapamıyorsa, tüm zekalar, uygun kültürel eğilimler ve tutku değerli yazılımlar anlamına gelmez. Ne yazık ki, birçok programcı yapamaz. Öyleyse, gemiye vermek için o gayreti arayın. Kapıdan çıkarması gereken ve bunun gerektirebileceği pragmatik takasları yapmaya istekli birini işe alın.
- sosyal maç Hem stres/gevşeme hem de inişler/çıkışlar sırasında uzun bir süre biriyle çalışmak size onun gerçek kişiliğini gösterecektir. Birinin görgü veya sosyal becerileri eksikse, onları filtreleyin.

Programcılar söz konusu olduğunda, yalnızca açık kaynak aracılığıyla tanıdığımız kişileri işe alıyoruz. Başka bir şey yapmanın sorumsuzluk olduğunu düşünüyoruz. Yayınlarını ve Ruby topluluğuna katılımını takip ettiğimiz için Jamis'i işe aldık. Yukarıda belirtilen tüm alanlarda mükemmeldi. değildi

Onu gerçekten önemli olan şeye göre yargılayabileceğimiz için ikincil faktörlere güvenmek gereklidir: işinin kalitesi.

Ve müfredat dışı etkinliklerin, çalışanların günlük işlerinden odağı ve tutkuyu uzaklaştıracağından endişelenmeyin. Eski bir klişenin dediği gibi: Bir şeyin yapılmasını istiyorsanız, tanıdığınız en meşgul kişiye sorun. Jamis ve David, Rails'e en çok katkıda bulunanlardan ikisi ve hala teknik olarak Basecamp'i sürmeyi başarıyorlar. Programlamayı ve işleri halletmeyi seven insanlar, tam olarak ekibinizde olmasını istediğiniz türden insanlardır.

## Açık Kaynak Tutkusu

Yeni bir kişiden en çok beklediğiniz şey, yaptığı işe olan tutkudur ve bunu göstermenin açık kaynak projelerindeki bir bağlılık izinden daha iyi bir yolu yoktur.

—Jarkko Laine, yazılım geliştiricisi

( Reduce the Risk, açık kaynaktan kiralama )

## 43: Çok Yönlü Bireyler Olun

### Yerleşik uzmanlar yerine hızlı öğrenme uzmanlarına gidin

Bilgi mimarı olan birini asla işe almayacağız. Sadece fazla spesifik. Bizimki gibi küçük bir ekiple, bu kadar dar bir şekilde tanımlanmış becerilere sahip insanları işe almak mantıklı değil.

Küçük takımların farklı şapkalar giyebilen insanlara ihtiyacı vardır. Yazabilen tasarımcılara ihtiyacınız var. Tasarımdan anlayan programcılara ihtiyacınız var. Herkesin bilginin nasıl oluşturulacağı hakkında bir fikri olmalıdır (bu ne anlama geliyorsa). Herkesin organize bir zihne sahip olması gerekir. Herkes müşterilerle iletişim kurabilmelidir.

Ve herkesin yolda vites değiştirmeye istekli ve yetenekli olması gerekiyor. Küçük ekiplerin genellikle yön değiştirmesi ve bunu hızlı bir şekilde yapması gerektiğini unutmayın. Tek bir şey yapabilen çamura bulanmış birinin aksine, uyum sağlayabilen, öğrenebilen ve akabilen birini istiyorsunuz.

## 44: Sahte Coşku Yapamazsınız

### Sinirli ve harika yerine mutlu ve ortalamayı seçin

Heves. Bu, taklit edemeyeceğiniz bir özellik. İşe alma zamanı geldiğinde, bir guruya veya bir teknoloji ünlüsüne ihtiyacınız olduğunu düşünmeyin. Çoğu zaman, onlar zaten sadece primadonnalardır. Mutlu ama ortalama bir çalışan, huysuz bir uzmandan daha iyidir.

Hevesli birini bulun. Yalnız bırakıldığında işleri halletmek için güvенеbileceğiniz biri.

Daha büyük, daha yavaş bir şirkette acı çeken ve yeni bir çevre özlemi çeken biri.

İnşa ettiğiniz şeyi inşa etmekten heyecan duyan biri. Senin nefret ettiğin şeylerden nefret eden biri. Treninize binmekten heyecan duyan biri.

### Soru sormak için ekstra puan

Potansiyel bir işe alınan kişinin projeniz hakkında çok soru sorup sormadığını gözlemleyin. Tutkulu programcılar bir sorunu olabildiğince iyi anlamak isterler ve birçok soruya yol açan potansiyel çözümleri ve iyileştirmeleri hızla önerirler. Soruları açıklığa kavuşturmak, projenizin binlerce farklı şekilde uygulanabileceği anlayışını da ortaya koyuyor ve web uygulamanızın nasıl çalıştığını tam olarak nasıl hayal ettiğinizi olabildiğince açık bir şekilde belirlemeniz çok önemli. Ayrıntılara girdikçe, kişinin iyi bir kültürel eşleşme olup olmadığı konusunda bir fikir edineceksiniz.

—Eric Stephens, [BuildV1.com](http://BuildV1.com)

# 45: Sözcüler

## İyi yazarlar işe alın

Bir pozisyonu doldurmak için birkaç kişi arasında karar vermeye çalışıyorsanız, her zaman daha iyi bir yazarı işe alın. Bu kişinin tasarımcı, programcı, pazarlamacı, satış elemanı ya da her ne olursa olsun, önemli değil, yazma becerileri işe yarayacaktır. Etkili, özlü yazma ve düzenleme, etkili, özlü kod, tasarım, e-postalar, anlık mesajlar ve daha fazlasını sağlar.

Çünkü iyi bir yazar olmak kelimelerden daha fazlasıdır. İyi yazarlar nasıl iletişim kurulacağını bilirler. Her şeyin anlaşılmasını kolaylaştırırlar. Kendilerini başkasının yerine koyabilirler. Neyi atlayacaklarını biliyorlar. Açıkça düşünüyorlar. Ve bunlar ihtiyacınız olan niteliklerdir.

## Organize Bir Zihin

İyi yazma becerileri, bilgi ve argümanları sistematik bir şekilde düzenleyebilen ve aynı zamanda diğer insanların bir şeyleri anlamasına yardımcı olan (yapmayan) organize bir zihnin göstergesidir. Kodlara, kişisel iletişimlere, anlık mesajlaşmaya (uzun mesafeli işbirlikleri için) ve hatta profesyonellik ve güvenilirlik gibi ezoterik kavramlara kadar yayılıyor.

—Dustin J. Mitchell, geliştirici ( Signal vs. Noise'dan)

## Net Yazmak, Net Düşünmeye Yol Açar

Net yazma, net düşünmeye yol açar. Ne bildiğini, ifade etmeye çalışmadan bilemezsin. İyi yazmak kısmen bir karakter meselesidir. Sizin için kolay olanı yapmak yerine, okuyucunuz için kolay olanı yapın.

—Michael A. Covington, Georgia Üniversitesi'nde Bilgisayar Bilimleri Profesörü (Nasıl Daha Açık Yazılır, Daha Açık Düşünülür ve Karmaşık Malzemeyi Daha Kolay Öğrenmek kitabından) .

## 46: İlk Arayüz

### Programlamaya başlamadan önce arayüzü tasarlayın

Çok fazla uygulama, program öncelikli bir zihniyetle başlar. Bu kötü bir fikir. Programlama, bir uygulama oluşturmanın en ağır bileşenidir, yani en pahalı ve değiştirilmesi en zor olanıdır.

Bunun yerine, önce tasarlayarak başlayın.

Tasarım nispeten hafiftir. Bir kağıt eskiz ucuzdur ve değiştirilmesi kolaydır. html tasarımlarını değiştirmek (veya atmak) hala nispeten basittir. Bu programlama için doğru değil. Önce tasarlamak sizi esnek tutar. Programlama önce sizi içine alır ve ek maliyetler için sizi hazırlar.

Öncelikle tasarlamamanın bir başka nedeni de arayüzün sizin ürününüz olmasıdır. İnsanların gördüğü şey, sattığınız şeydir. Sonunda bir arayüzü tokatlarsanız, boşluklar görünecektir.

Uygulamanın nasıl göründüğünü ve nasıl hissettirdiğini en baştan görebilmemiz için arayüzle başlıyoruz. Süreç boyunca sürekli revize edilmektedir. Mantıklı geliyor? Kullanımı kolay mı? Eldeki sorunu çözüyor mu? Bunlar, yalnızca gerçek ekranlarla uğraşırken gerçekten yanıtlayabileceğiniz sorulardır. Önce tasarlamak sizi esnek tutar ve bu yanıtlara daha sonra değil, süreçte daha erken ulaşmanızı sağlar.

### Blinksale'i Başlatan Turuncu Kalem

Kullanıma hazır faturalama yazılımıyla ilgili hayal kırıklığımı fark eder etmez, faturalama çözümümün işe yaramasını nasıl tercih edeceğimi çizmeye karar verdim. Turuncu bir kalem çıkardım çünkü o akşam kullanışlı olan tek şey buydu ve birkaç saat içinde kullanıcı arayüzünün yaklaşık yüzde 75'i çizilmişti.

O sırada ütü yapan karım Rachel'a gösterdim ve "Ne düşünüyorsun?" diye sordum.

Ve gülümseyerek cevap verdi, "Bunu yapmalısın. Gerçek için."

Sonraki iki hafta boyunca tasarımları geliştirdim ve Blinksale olacak olanın neredeyse ilk versiyonunun neredeyse tamamı için statik html sayfalarını tamamen taklit ettim. Orangepen eskizlerin ötesinde hiçbir zaman tel çerçeveler yapmadık ve doğrudan html tasarımına girmek, o sırada neye bulaştığımızı gerçekten bilmiyor olsak da, projenin ne kadar "gerçek" hale geldiği konusunda heyecanlanmamıza yardımcı oldu.

Html maketleri tamamlandıktan sonra geliştiricimiz Scott'a Blinksale fikriyle yaklaştık. Kullanıcı arayüzünün çoğunun önceden tasarlanmış olması, birkaç düzeyde son derece faydalıydı.

İlk olarak, Scott'a nereye gittiğimiz konusunda gerçek bir vizyon ve heyecan verdi. Bir fikirden çok daha fazlasıydı, gerçekte.

İkincisi, tasarımı işlevsel bir uygulamaya dönüştürmek için Scott'ın ne kadar çaba ve zaman gerektireceğini doğru bir şekilde ölçmemize yardımcı oldu. Maddi durumunuz varken

bir projeyi önyüklerken, bütçe gereksinimlerini ne kadar erken tahmin edebilirsiniz o kadar iyidir. Kullanıcı arayüzü tasarımı, ilk proje kapsamı için referansımız oldu. Son olarak, kullanıcı arayüzü tasarımı, biz geliştirmeye devam ederken uygulamanın neyle ilgili olduğunu bize hatırlatmak için bir rehber görevi gördü. Yeni özellikler eklemek bizi cezbederken, “Tabii, şunu ekleyelim!” diyemedik. Tasarıma geri dönüp bu yeni özelliğin nereye gideceğini kendimize sormamız gerekiyordu ve eğer yeri olmasaydı eklenmeyecekti.

—Josh Williams, Blinksale'nin kurucusu

## 47: Merkez Üs Tasarımı

### Sayfanın merkezinden başlayın ve dışa doğru inşa edin

Merkez üssü tasarımı, sayfanın gerçek özüne - merkez üssü - odaklanır ve ardından dışa doğru inşa edilir. Bu, başlangıçta uç noktaları göz ardı ettiğiniz anlamına gelir: gezinme/sekmeler, altbilgi, renkler, kenar çubuğu, logo vb. Bunun yerine, merkez üssünden başlar ve içeriğin en önemli parçasını önce tasarlıyorsunuz.

Sayfanın kesinlikle onsuz yaşayamayacağı şey, merkez üssüdür. Örneğin, bir blog gönderisini görüntüleyen bir sayfa tasarlıyorsanız, blog gönderisinin kendisi merkez üssüdür. Kenar çubuğundaki kategoriler değil, üstteki başlık değil, alttaki yorum formu değil, gerçek blog gönderisi birimi. Blog gönderisi birimi olmadan sayfa bir blog gönderisi değildir.

Yalnızca bu ünite tamamlandığında, sayfadaki en kritik ikinci öğeyi düşünmeye başlıyorsunuz. Sonra ikinci en kritik unsurdan sonra üçüncüye geçersiniz ve bu böyle devam eder. Bu merkez üssü tasarımı.

Merkez üssü tasarımı, geleneksel "çerçeveyi oluşturalım, sonra içeriği içeri bırakalım" modelinden kaçınır. Bu süreçte, sayfa şekli oluşturulur, daha sonra gezinme dahil edilir, ardından pazarlama "malzemeleri" eklenir ve son olarak, temel işlevsellik, sayfanın asıl amacı kalan alana dökülür. En öncelikli olması gerekeni alan ve onu sona saklayan geriye dönük bir süreçtir.

Epicenter tasarımı bu süreci tersine çevirir ve ilk gün gerçekten önemli olan şeylere odaklanmanızı sağlar. Önce temeller, sonra ekstralar. Sonuç, müşteriler için daha samimi, odaklanmış, kullanılabilir bir ekrandır. Ayrıca, sayfanın tüm yönlerinin ilk önce sıraya girmesini beklemek yerine tasarımcı ve geliştirici arasındaki diyalogu hemen başlatmanıza olanak tanır.



## 48: Üç Devletli Çözüm

### Normal, boş ve hata durumları için tasarım

Her ekran için üç olası durumu göz önünde bulundurmanız gerekir:

- Normal  
İnsanların her şeyin yolunda gittiğini ve uygulamanızın verilerle dolu olduğunu gördüğü ekran.
- Boşluk  
Uygulamayı ilk kez kullanırken, veri girilmeden önce insanların gördüğü ekran.
- Hata  
İnsanların bir şeyler ters gittiğinde gördüğü ekran.

Normal durum hiç de kolay değil. Bu, zamanınızın çoğunu geçireceğiniz ekrandır. Ancak diğer eyaletlere de zaman ayırmayı unutmayın (bununla ilgili daha fazla bilgi için aşağıdaki makalelere bakın).

## 49: Boş Sayfa

### Düşünceli bir ilk çalıştırma deneyimi ile beklentileri belirleyin

Boş sayfa aşamasını görmezden gelmek, yapabileceğiniz en büyük hatalardan biridir. Boş sayfa, uygulamanızın ilk izlenimidir ve asla bir saniyeniz olmaz... bilirsiniz.

Sorun şu ki, bir kullanıcı arayüzü tasarlarken genellikle verilerle aynı hızadadır. Tasarımcılar her zaman şablonları verilerle doldurur. Her liste, her gönderi, her alan, her köşe ve her köşede bir şeyler var. Bu da ekranın harika görüldüğü ve çalıştığı anlamına gelir.

Ancak, uygulamanın doğal durumu, veri içermeyen bir durumdur. Birisi kaydolduğunda, boş bir sayfa ile başlarlar. Bir web günlüğüne çok benzer şekilde, onu doldurmak onlara bağlıdır - genel görünüm ve his, insanlar verilerini girene kadar şekillenmez: gönderiler, bağlantılar, yorumlar, saatler, kenar çubuğu bilgileri veya her neyse.

Ne yazık ki, bir uygulamanın bu boş sayfa aşamasında - uygulamanın genel kullanışlılığının değerlendirileceği en az miktarda bilgi, tasarım ve içeriğin bulunduğu aşama olan bir uygulamanın değerli olup olmadığına müşteri karar verir. Yeterli boş bir sayfa tasarlamayı başaramadığınızda, insanlar neyi kaçırdıklarını bilemezler çünkü her şey eksiktir.

Yine de çoğu tasarımcı ve geliştirici hala bu aşamayı hafife alıyor. Boş sayfa tasarlamak için çok fazla zaman harcamazlar çünkü uygulamayı geliştirdiklerinde/kullandıklarında, test amacıyla girdikleri verilerle doludur. Boş sayfayla bile karşılaşmıyorlar. Elbette, birkaç kez yeni bir kişi olarak giriş yapabilirler, ancak zamanlarının çoğu, verilerle dolu bir uygulamada yüzerek geçer.

Yararlı bir boş sayfaya ne eklemelisiniz?

- Hızlı öğreticiler eklemek ve tanıtım yazılarına yardımcı olmak için bunu bir fırsat olarak kullanın.
- İnsanların ne beklemeleri gerektiğini (ve neden kalmaları gerektiğini) bilmeleri için verilerle doldurulmuş sayfanın örnek bir ekran görüntüsünü verin.
- Nasıl başlayacağınızı, ekranın sonunda nasıl görüneceğini vb. açıklayın.
- İlk kez görüntüleyenlerin soracağı temel soruları yanıtlayın: Bu sayfa nedir? Ben şimdi ne yapacağım? Bu ekran dolduğunda nasıl görünecek?
- Beklentileri belirleyin ve hayal kırıklığı, korkutma ve genel kafa karışıklığını azaltmaya yardımcı olun.

İlk izlenimler çok önemlidir. Düşünceli bir boş sayfa tasarlamayı başaramazsanız, uygulamanız veya hizmetiniz hakkında olumsuz (ve yanlış) bir izlenim yaratırsınız.

Asla İkinci Bir Şans Veremezsiniz...

Mac OS x ui'nin [Steve] Jobs'tan büyük ölçüde etkilendiğini düşündüğüm bir başka yönü, kurulum ve ilk çalıştırma deneyimidir. Jobs'un ilk izlenimlerin öneminin fazlasıyla farkında olduğunu düşünüyorum... Jobs ilk çalıştırma deneyimine bakıyor ve bir kullanıcının makineyle ilgili genel deneyiminin yalnızca binde biri olabileceğini düşünüyor, ancak en önemli binde biri, çünkü birinci binde biri ve onların beklentilerini ve ilk izlenimlerini belirliyor.

—John Gruber, yazar ve web geliştiricisi ( John Gruber ile Röportajdan )

## 50: Savunmaya Geçin

### İşler ters gittiğinde tasarım

Kabul edelim: İnternette işler ters gidecek. Uygulamanızı ne kadar dikkatli tasarlarsanız tasarlayın, ne kadar test yaparsanız yapın, müşteriler yine de sorunlarla karşılaşacaktır. Peki bu kaçınılmaz arızalarla nasıl başa çıkıyorsunuz? Savunma tasarımı ile.

Defansif tasarım, defansif sürüş gibidir. Sürücülerin her zaman kaygan yollar, pervasız sürücüler ve diğer tehlikeli senaryolar için uyanık olması gerektiği gibi, site kurucuları da sürekli olarak ziyaretçilerin kafa karışıklığına ve hayal kırıklığına uğramasına neden olan sorunlu noktaları aramalıdır. İyi bir site savunması, müşteri deneyimini iyileştirebilir veya bozabilir.

Savunma tasarımı hakkında söyleyeceğimiz her şeyi ayrı bir kitapla doldurabiliriz. Aslında, zaten var. Başlık “Web için Savunma Tasarımı”dır ve hata ekranlarını ve diğer kriz noktalarını nasıl iyileştireceğini öğrenmek isteyen herkes için harika bir kaynaktır.

Unutmayın: Uygulamanız zamanın %90'ında harika çalışabilir. Ancak müşterileri ihtiyaç duydukları anda terk ederseniz, bunu unutmaları pek olası değildir.

## 51: Tutarlılık Üzerinden Bağlam

### Burada mantıklı olan orada mantıklı olmayabilir.

Eylemler düğmeler mi yoksa bağlantılar mı olmalı? Eyleme bağlıdır. Bir takvim görünümü liste biçiminde mi yoksa ızgara biçiminde mi olmalı? Nerede gösterildiğine ve sürenin ne kadar uzun olduğuna bağlı. Her global gezinme bağlantısının her sayfada olması gerekiyor mu? Her yerde küresel bir arama çubuğuna mı ihtiyacınız var? Her sayfada aynı alt bilgiye mi ihtiyacınız var? Cevap: "Buna göre değişir."

Bu yüzden bağlam tutarlılıktan daha önemlidir. Tasarımınız bu şekilde daha mantıklıysa, tutarsız olmanızda bir sakınca yoktur. İnsanlara sadece önemli olanı verin. İhtiyaç duyduklarında onlara ihtiyaç duyduklarını verin ve ihtiyaç duymadıklarından kurtulun. Tutarlı olmaktansa haklı olmak daha iyidir.

### Akıllı Tutarsızlık

Tutarlılık gerekli değildir. Yıllardır ui ve ux öğrencilerine arayüzde tutarlılığın arayüz tasarımının temel kurallarından biri olduğu öğretilmiştir. Belki bu yazılımda geçerlidir, ancak Web'de bu doğru değil. Web'de önemli olan, her bir sayfada, kullanıcının süreçteki bir sonraki adımı hızlı ve kolay bir şekilde ilerleyip ilerletmeyeceğidir.

Creative Good'da buna "akıllı tutarsızlık" diyoruz: süreçteki her sayfanın kullanıcılara sürecin o noktasında tam olarak ihtiyaç duydukları şeyi vermesini sağlamak. Sitenin geri kalanıyla tutarlı oldukları için gereksiz gezinme öğeleri eklemek aptalca.

—Mark Hurst, Creative Good'un kurucusu ve Goovite.com'un yaratıcısı ( The Page Paradigm'den) —

## 52: Metin Yazarlığı Arayüz Tasarımıdır

### Her harf önemlidir

Metin Yazarlığı, arayüz tasarımıdır. Harika arayüzler yazılmıştır. Her pikselin, her simgenin, her yazı tipinin önemli olduğunu düşünüyorsanız, her harfin de önemli olduğuna inanmanız gerekir. Arayüzünüzü yazarken, kendinizi daima arayüzünüzü okuyan kişinin yerine koyun. Neyi bilmeleri gerekiyor? Bunu kısa ve net bir şekilde nasıl açıklayabilirsiniz?

Bir düğmeyi Gönder veya Kaydet veya Güncelle veya Yeni veya Oluştur olarak mı etiketlersiniz? Bu kopya yazarlığı. Üç cümle mi yazarsın, beş mi? Genel örneklerle mi yoksa ayrıntılarla mı açıklarsınız? İçeriği Yeni mi, Güncellendi mi, Yakın Zamanda Güncellendi mi veya Değiştirildi mi etiketliyor musunuz? Yeni mesaj mı var: 5 mi yoksa 5 yeni mesaj mı var yoksa 5 mi yoksa beş mi yoksa mesaj mı yoksa mesaj mı? Bunların hepsi önemli.

Kitlenizle de aynı dili konuşmanız gerekir. Bir web uygulaması yazıyor olmanız, teknik jargondan kurtulabileceğiniz anlamına gelmez. Müşterilerinizi düşünün ve bu düğmelerin ve kelimelerin onlar için ne anlama geldiğini düşünün. Çoğu insanın anlamadığı kısaltmalar veya kelimeler kullanmayın. Dahili lingo kullanmayın. Başka bir mühendisle konuşan bir mühendis gibi konuşma. Kısa ve güzel tut. Neye ihtiyacın olduğunu söyle ve daha fazlasını söyleme.

İyi yazı, iyi tasarımıdır. Tasarıma kelimelerin eşlik etmediği nadir bir istisnadır. Adlı simgeler, örneklerle form alanları, etiketli düğmeler, bir süreçte adım adım talimatlar, geri ödeme politikanızın net bir açıklaması. Bunların hepsi arayüz tasarımıdır.

## 53: Bir Arayüz

### Yönetici işlevlerini genel arayüze dahil edin

Yönetici ekranları – tercihleri, kişileri vb. yönetmek için kullanılan ekranlar – bok gibi görünmeye meyillidir. Bunun nedeni, geliştirme süresinin çoğunun bunun yerine halka açık arayüzde harcanmasıdır.

Crappy-admin-screen sendromundan kaçınmak için yönetici işlevleriyle ilgilenmek için ayrı ekranlar oluşturmayın. Bunun yerine, bu işlevleri (yani düzenleme, ekleme, silme) normal uygulama arabiriminde oluşturun.

İki ayrı arayüz (yani, biri normal kişiler için, diğeri yöneticiler için) bulundurmanız gerekiyorsa, her ikisi de zarar görecektir. Aslında, aynı vergiyi iki kez ödersiniz. Kendinizi tekrar etmeye zorlanıyorsunuz ve bu da özensiz olma ihtimalinizi artırdığınız anlamına geliyor. Endişelenmeniz gereken daha az ekran, daha iyi sonuç verir.

### Ayrı Arayüz Yok

Uygulama her şeydir. Değiştirilebilen, eklenebilen veya ayarlanabilen her şey doğrudan uygulamanın yönetim alanı üzerinden yapılabilir. Bu, müşterilerimizin sahip oldukları herhangi bir sorun veya soruda onlara yardımcı olmak için tam olarak ne gördüğünü görmemizi sağlar. Ve müşterilerimizin farklı görevleri yapmak için ayrı bir arayüzde oturum açma konusunda endişelenmelerine gerek yok. Bir dakika müşterileri için randevularla ilgileniyor olabilirler ve bir sonraki dakika yeni bir çalışan eklemek zorunda kalabilirler. Farklı uygulamalar arasında geçiş yapmaktan ve uygulamaya daha da hızlı uyum sağlayabilecekleri tutarlı bir arayüz sağlamaktan rahatsız olamazlar.

—Edward Knittel, Satış ve Pazarlama Direktörü, KennelSource

# 54: Daha Az Yazılım

## Kodunuzu olabildiğince basit tutun

İki kat daha fazla kodun yazılımınızı yalnızca iki kat daha karmaşık hale getireceğini düşünürdünüz. Ama aslında, kod miktarını her artırdığınızda, yazılımınız katlanarak daha karmaşık hale gelir. Her küçük ekleme, her değişiklik, her karşılıklı bağımlılık ve her tercihin basamaklı bir etkisi vardır. Dikkatsizce kod eklemeye devam edin ve farkına bile varmadan, korkunç Big Ball of Mud'u yaratmış olacaksınız.

Bu karmaşıklıkla savaşmanın yolu daha az yazılımla. Daha az yazılım, daha az özellik, daha az kod, daha az atık anlamına gelir.

Anahtar, çok fazla yazılım gerektiren herhangi bir zor sorunu, çok daha az gerektiren basit bir soruna yeniden ifade etmektir. Tam olarak aynı sorunu çözmüyor olabilirsiniz ama sorun değil. Çabanın %20'si için orijinal sorunun %80'ini çözmek büyük bir kazançtır. Orijinal sorun neredeyse hiçbir zaman onu çözmek için harcanan çabanın beş katına geçecek kadar kötü değildir.

Daha az yazılım, kristal küreyi kaldırdığınız anlamına gelir. Gelecekteki sorunları tahmin etmeye çalışmak yerine, yalnızca bugünün sorunlarıyla ilgileniyorsunuz. Neden? Niye? Yarıyla ilgili korkularınız genellikle asla gerçekleşmez. Bu hayali sorunları çözmeye çalışırken kendinizi boğmayın.

En başından beri ürünlerimizi daha az yazılım konsepti etrafında tasarladık.

Mümkün olduğunda, zor sorunları kolay olanlara böleriz. Kolay sorunlara çözümler bulduk, yalnızca uygulanması ve desteklenmesi daha kolay değil, aynı zamanda anlaşılması daha kolay ve kullanımı daha kolay. Hepsi, kendimizi rakiplerden nasıl farklılaştırdığımızın bir parçası; Daha fazlasını yapan ürünler üretmeye çalışmak yerine, daha az yapan ürünler üretiyoruz.

- Daha az yazılımın yönetimi daha kolaydır.
- Daha az yazılım, kod tabanınızı azaltır ve bu da daha az bakım işi
- (ve daha mutlu bir personel) anlamına gelir.
- Daha az yazılım, hızlı bir şekilde uyum sağlamanız için değişim maliyetinizi düşürür. Bir sürü kodu değiştirmek zorunda kalmadan fikrinizi değiştirebilirsiniz.
- Daha az yazılım daha az hataya neden olur.
- Daha az yazılım, daha az destek anlamına gelir.

Dahil etmeyi veya çıkarmayı seçtiğiniz özelliklerin de daha az yazılımla çok ilgisi var. Yapılması zor olan özellik isteklerine hayır demekten korkmayın. Kesinlikle gerekli olmadıkça, onları dışarıda bırakarak zamandan/çabadan/karışıklıktan tasarruf edin.



Sen de yavaşla. Bir hafta boyunca bir fikir üzerinde harekete geçmeyin ve ilk uğultu geçtikten sonra hala harika bir fikir gibi görünüp görünmediğine bakın. Ekstra marine etme süresi genellikle beyninizin daha kolay bir çözüm bulmasına yardımcı olur.

Programcılar karşı teklifler yapmaya teşvik edin.

Şunu duymak istersiniz: "Önerdiğiniz yol 12 saat sürecek. Ama bunu yapabilmemin sadece bir saat sürecek bir yolu var. x yapmayacak ama y yapacak." Yazılımın geri itmesine izin verin. Programcılara en iyi yol olduğunu düşündükleri şey için savaşmalarını söyleyin.

Ayrıca, daha fazla yazılım yazmanın yollarını arayın. Müşterilere yazılım modelinde bir değişiklik gerektirmeyen alternatif bir yol önermesi için ekrandaki kopyayı değiştirebilir misiniz? Örneğin, insanların sunucu tarafında görüntü manipülasyonu yapmak yerine belirli bir boyuttaki görüntüleri yüklemelerini önerebilir misiniz?

Uygulamanıza dahil eden her özellik için kendinize şunu sorun: Bunun, o kadar fazla yazılım gerektirmeyecek şekilde eklenmesinin bir yolu var mı? Sadece ihtiyacınız olan kodu yazın, daha fazlasını değil. Sonuç olarak uygulamanız daha yalın ve sağlıklı olacaktır.

HİÇBİR KOD YOKTUR!

İyi yazılım tasarımının "sırrı", koda ne yazılacağını bilmek değil; DIŞARI neyi bırakacağını bilmektir! Bu, sert noktaların ve yumuşak noktaların nerede olduğunu tanımak ve daha fazla tasarıma tıkmaya çalışmak yerine alanı/odayı nerede bırakacağını bilmektir.

—Brad Appleton, yazılım mühendisi (Kod YOK'tan daha esnek olan KOD Yok!'dan)

## Karmaşıklık Boyutla Doğrusal Olarak Ölçeklenmiyor

Yazılım mühendisliğinin en önemli kuralı da en az bilinenidir: Karmaşıklık boyutla doğrusal olarak ölçeklenmez... 2000 satırlık bir program, boyutun yarısı kadar geliştirme süresi gerektirir.

—Ganssle Grubu ( Keep It Small'dan)

## 55: Mutluluk için Optimize Edin

### Ekibinizi heyecanlandıran ve motive eden araçları seçin

Mutlu bir programcı, üretken bir programcıdır. Bu yüzden mutluluk için optimize ediyoruz ve siz de yapmalısınız. Sadece endüstri standartlarına veya performans ölçütlerine dayalı araçlar ve uygulamalar seçmeyin. Maddi olmayan varlıklara bakın: Burada tutku, gurur ve ustalık var mı? Günde sekiz saat bu ortamda çalışmaktan gerçekten mutlu olur muydunuz?

Bu özellikle bir programlama dili seçmek için önemlidir. Kamuoyunun aksine algısına rağmen, eşit yaratılmamışlardır. Hemen hemen her dil, hemen hemen her uygulamayı yaratabilirken, doğru dil, çabayı yalnızca mümkün veya katlanılabilir kılmakla kalmaz, aynı zamanda hoş ve canlandırıcı kılar. Her şey günlük işlerin küçük ayrıntılarını keyifli hale getirmekle ilgili.

Mutluluğun basamaklı bir etkisi vardır. Mutlu programcılar doğru olanı yaparlar. Basit, okunabilir kod yazarlar. Temiz, etkileyici, okunabilir, zarif yaklaşımlar benimserler. Onlar eğleniyorlar.

Ruby dilinde programlama mutluluğunu bulduk ve çerçevemiz Rails ile diğer geliştiricilere aktardık. Her ikisi de insanlar ve onların mutluluğu için optimize etme misyonunu paylaşır.

Bu kombinasyona bir şans vermenizi öneririz.

Özetle, ekibinizin sevdikleri araçlarla çalışması gerekiyor. Burada programlama dilleri açısından konuştuk, ancak konsept uygulamalar, platformlar ve diğer her şey için geçerlidir. İnsanları heyecanlandıran sigortayı seçin. Sonuç olarak heyecan ve motivasyon ve daha iyi bir ürün üreteceksiniz.

### İstedığınız mühendis türleri

Uygulamamızı Ruby on Rails kullanarak oluşturmak istememin bir numaralı nedeni, çok zarif, üretken ve güzel tasarlanmış olmasıdır. Bu tür şeylerle ilgilenen mühendisleri cezbetme eğilimindedir... bunlar tam olarak ekibinizde olmasını istediğiniz türden mühendislerdir, çünkü pazarı kazanmak için ihtiyacınız olan güzel, zarif ve üretken yazılımı oluştururlar.

—Charles Jolley, Nisus Software Genel Müdürü (Sinyal vs. Gürültüden)

## 56: Kod Konuşur

### Kodunuz geri geldiğinde dinleyin

Kodunuzu dinleyin. Öneriler sunacaktır. Geri itecek. Size tuzakların nerede olduğunu söyleyecektir. Bir şeyler yapmanın yeni yollarını önerecektir. Daha az yazılım içeren bir modele bağlı kalmanıza yardımcı olacaktır.

Haftalarca zaman ve binlerce kod satırı gerektiren yeni bir özellik mi? Bu, muhtemelen daha iyi bir yol olduğunu söyleyen kodunuz. On saat sürecektir karmaşık bir yol yerine bir şeyi bir saatte kodlamanın basit bir yolu var mı? Yine, bu size rehberlik eden kodunuzdur. Dinlemek.

Kodunuz, ucuz ve hafif düzeltmeler için size rehberlik edebilir. Kolay bir yol ortaya çıktığında dikkat edin. Elbette, yapması kolay olan özellik, ilk başta aklınızdaki özellik ile tam olarak aynı olmayabilir ama ne olmuş yani? Yeterince iyi çalışıyorsa ve başka bir şey üzerinde çalışmanız için size daha fazla zaman veriyorsa, o bir kalecidir.

### Dinle

Tasarım konusunda endişelenmeyin, kodunuzu dinlerseniz iyi bir tasarım ortaya çıkar... Teknik adamları dinleyin. Değişiklik yapmanın zorluğundan şikayet ediyorlarsa, bu tür şikayetleri ciddiye alın ve düzeltmeleri için onlara zaman verin.

—Martin Fowler, Baş Bilim İnsanı, ThoughtWorks ( Design Dead?den) \_\_\_\_\_

### Programcılar Kodu Kaldırmak İçin Para Aldıysa...

Programcılara yeni kod yazmak yerine yazılımı yazılımdan kaldırmaları için para ödenseydi, yazılım çok daha iyi olurdu.

—Nicholas Negroponte, MIT'de Medya Teknolojisi Profesörü (And, geri kalanından (AIGA) \_\_\_\_\_  
Konferans) hikaye) .

## 57: Borç Yönetimi

### Kodunuzu ödeyin ve "faturalar" tasarlayın

Borcu genellikle para olarak düşünürüz, ancak başka şekillerde de gelir. Kolayca kod oluşturabilir ve borç tasarlayabilirsiniz.

İşlevsel ama yine de biraz kılıklı olan bazı kötü kodları bir araya getirin ve borç biriktiriyorsunuz.

Yeterince iyi ama gerçekten iyi olmayan bir tasarımı bir araya getirin ve tekrar yaptınız.

Bunu zaman zaman yapmakta fayda var. Aslında, tüm Get-Real-ASAP olayını yapmanıza yardımcı olan genellikle gerekli bir tekniktir. Ancak yine de bunu borç olarak kabul etmeniz ve bir noktada kılıklı kodu temizleyerek veya şu sayfayı yeniden tasarlayarak ödemeniz gerekir.

Aynı şekilde, gelirinizin bir kısmını düzenli olarak vergilere ayırmanız, kodunuzu ve tasarım borcunuzu ödemek için düzenli olarak biraz zaman ayırmanız gerekir. Bunu yapmazsanız, anaparayı ödemek (ve ilerlemek) yerine sadece faiz ödersiniz (hack'leri düzeltirsiniz).

## 58: Açık Kapılar

### Verileri RSS, API'ler vb. aracılığıyla dünyaya aktarın.

Müşterilerinizi kilitlemeye çalışmayın. Bilgileri istedikleri zaman ve istedikleri şekilde almalarına izin verin. Bunu yapmak için, verileri mühürleme fikrinden vazgeçmelisiniz. Bunun yerine, vahşi çalışmasına izin verin. İnsanlara RSS beslemeleri aracılığıyla bilgilerine erişme izni verin. Üçüncü taraf geliştiricilerin aracınızı geliştirmesine olanak tanıyan api'ler sunun. Bunu yaptığınızda, müşteriler için hayatı daha uygun hale getirir ve uygulamanızın yapabileceklerinin olanaklarını genişletirsiniz.

İnsanlar rss beslemelerini sadece blogları veya haber sitelerini takip etmenin iyi bir yolu olarak düşünürdü. Ancak yemler bundan daha fazla güce sahiptir. Ayrıca, müşterilerin tekrar tekrar oturum açmak zorunda kalmadan bir uygulamanın değişen içeriği hakkında güncel kalmaları için harika bir yol sağlarlar. Basecamp beslemeleri ile müşteriler, URL'yi bir haber okuyucusuna açabilir ve sitede sürekli kontrol etmek zorunda kalmadan proje mesajlarına, yapılacaklar listelerine ve kilometre taşlarına göz atabilir.

API'ler, geliştiricilerin uygulamanız için paha biçilmez olabilecek eklenti ürünler oluşturmaya olanak tanır. Örneğin, Backpack, Chipt Productions'ın bir Mac OS X Dashboard widget'ı oluşturmak için kullandığı bir API sağlar. Widget, kişilerin masaüstünden hatırlatıcılar, liste öğeleri ve daha fazlasını eklemesine ve düzenlemesine olanak tanır. Müşteriler bu widget hakkında bize övgüler yağdırdı ve hatta bazıları bunun Sırt Çantası'nı kullanmalarını sağlayan anahtar faktör olduğunu söyledi.

Bumerang etkisi elde etmek için verilerin ücretsiz çalışmasına izin veren diğer iyi şirket örnekleri:

- Google Haritalar API'si, insanların başka bir kaynaktan (ör. apartman listeleri) bilgi toplamasına ve bu verileri bir harita üzerinde çizmesine izin veren ilginç karmalar oluşturdu.
- Linkroll'lar, insanlara en son del.icio.us yer imlerini kendi sitelerinde göstermeleri için bir yol sunar.
- Flickr, müşterilerin fotoğraf kitapları, posterler, dvd yedekleri ve pullar satın alabilmesi için diğer işletmelerin ticari API'lere erişmesine izin verir. Flickr'dan Stewart Butterfield, "Amaç, onu tamamen açmak ve fotoğraflarınızla bir şeyler yapmak söz konusu olduğunda size en geniş seçenek yelpazesini sunmak" diyor.

### Bir Widget Fark Yaratır

Basecamp, Backpack'ı bir süre önce piyasaya sürdüğünde, ilk izlenimim... eh.

Bu yüzden, Chipt Productions'ın Tiger için bir Sırt Çantası widget'ını yayınladığı sıralarda - ki bu indirip denemek için çok güzeldi - Sırt Çantası'na ikinci bir bakış attım. Sonuç? Oldukça fark var.

Şimdi ne zaman bir fikir aklıma gelse, widget'ı açar, yazar ve gönderirim - tamamdır. E-posta, kontrol etmek istediğim bir şeyle mi geliyor? Widget'ı açın, yazın ve gönderin - bitti. Widget, kullandığım her Mac'te hazır bulunan anında bir beyin dökümü. Ve her şey web tabanlı olduğundan, herhangi bir sürüm kontrolü veya senkronizasyon yoktur - yalnızca içeriğin nereye gittiği veya daha sonra nasıl erişeceğim konusunda endişelenmenize gerek kalmadan akışkan girişi.

—Todd Dominey, Dominey Design'in kurucusu ( Trying on Backpack'ten)

## 59: Bir hakkında İşlevsel Hiçbir Şey Yok Fonksiyonel Özellikler

### İşlevsel özellikler belgesi yazmayın

Bu plan belgeleri genellikle bitmiş ürünle neredeyse hiçbir ilgisi olmayan bir şekilde sonuçlanır.

İşte nedeni:

### İşlevsel özellikler fantezidir

Gerçeği yansıtmazlar. Bir uygulama, inşaatçılar onu inşa edene, tasarımcılar onu tasarlayana ve insanlar onu kullanana kadar gerçek değildir. İşlevsel özellikler sadece kağıt üzerindeki kelimelerdir.

### İşlevsel özellikler yatıştırma ile ilgilidir

Herkesi dahil ve mutlu hissettirmekle ilgilidirler ki bu sıcak ve bulanık olsa da o kadar da yardımcı olmuyor. Asla zor seçimler yapmak ve maliyetleri ifşa etmekle, harika bir uygulama oluşturmak için olması gereken şeylerle ilgili değiller.

### İşlevsel özellikler yalnızca bir anlaşma yanılısamasına yol açar

Metnin paragrafları üzerinde anlaşmaya varan bir grup insan gerçek bir anlaşma değildir. Herkes aynı şeyi okuyor olabilir ama farklı bir şey düşünüyor. Bu kaçınılmaz olarak daha sonra ortaya çıkıyor: "Bir dakika, aklımdaki bu değildi." "Ha? Biz onu böyle tarif etmedik." "Evet öyleydi ve hepimiz üzerinde anlaştık - hatta imzaladınız bile." Matkabı biliyorsun.

### İşlevsel özellikler, en az bilgiye sahip olduğunuzda sizi en önemli kararları almaya zorlar.

Bir şey hakkında en az şeyi, onu inşa etmeye başladığınızda bilirsiniz. Ne kadar çok inşa ederseniz, o kadar çok kullanırsanız, o kadar çok bilirsiniz. İşte o zaman kararlar vermelisiniz - daha fazla bilgiye sahip olduğunuzda, daha az değil.

## İşlevsel özellikler, özellik aşırı yüklenmesine yol açar

Spesifikasyon aşamasında herhangi bir geri bildirim yoktur. Bir şeyler yazmanın ve başka bir madde işareti eklemenin hiçbir maliyeti yoktur. Acı çeken birini evcil hayvan özelliğini ekleyerek memnun edebilirsiniz. Ve sonra insanlara değil, bu kurşun noktalarına göre tasarım yaparsınız. Ve böylece ekranın üst kısmında 30 sekme bulunan aşırı yüklenmiş bir siteyle karşılaşıyorsunuz.

## İşlevsel özellikler gelişmenize, değiştirmenize ve

yeniden değerlendirmek

Bir özellik kapatılır ve üzerinde anlaşmaya varılır. Geliştirme sırasında bunun kötü bir fikir olduğunu anlasanız bile, buna takılıp kalırsınız. Spesifikasyonlar, bir şey inşa etmeye başladığınızda her şeyin değiştiği gerçeğiyle ilgilenmez.

Peki spec yerine ne yapmalısınız? Sizi gerçek bir şeye yönlendiren daha kısa bir alternatifle gidin. Uygulamanın yapması gerekenler hakkında tek sayfalık bir hikaye yazın. Sade bir dil kullanın ve hızlı olun. Bunu açıklamak bir sayfadan fazla sürerse, o zaman çok karmaşıktır. Bu işlem bir günden fazla sürmemelidir.

Ardından arayüzü oluşturmaya başlayın - arayüz, işlevsel spesifikasyonun alternatifi olacaktır. Bazı hızlı ve basit kağıt eskizleri çizin. Ardından html olarak kodlamaya başlayın. Arayüz tasarımları alternatif yorumlara açık metin paragraflarından farklı olarak ortak paydalardır.

üzerinde herkes hemfikir olabilir.

Herkes aynı ekranları kullanmaya başladığında kafa karışıklığı ortadan kalkar. Arka uç kodu hakkında endişelenmeye başlamadan önce herkesin bakmaya, kullanmaya, tıklamaya ve "hissetmeye" başlayabileceği bir arayüz oluşturun. Kendinizi mümkün olduğunca müşteri deneyiminin önüne getirin.

Kilitli özellikleri unutun. Sizi süreçte çok erken büyük, önemli kararlar almaya zorlarlar. Spesifikasyon aşamasını atlarsanız, değişimi ucuz tutar ve esnek kalırsınız.

### Yararsız Özellikler

Bir "spec" işe yaramaza yakındır. Hem kullanışlı hem de yeterince büyük olan bir özellik hiç görmedim. kesin.

Ve teknik özelliklere dayalı bir sürü saçma sapan iş gördüm. Yazılım yazmanın en kötü yolu budur, çünkü tanımı gereği, yazılımın gerçeğe değil teoriye göre yazıldığı anlamına gelir.

—Linus Torvalds, Linux'un yaratıcısı ( [Linux: Linus On Spesifikasyonlar](#) )



## Engelleyicilerle savařın

Herhangi bir tasarımı bařlamadan önce kapsamlı gereksinim belgeleri üzerinde ısrar eden kiřilerin, yalnızca süreci yavařlatmaya çalıřan gerçekten 'engelleyiciler' (ve genellikle tasarım veya yenilikçi düşünceye katkıda bulunacak hiçbir řeyi olmayan kiřiler) olduđunu gördüm.

En iyi çalıřmalarımız, bir siteyi geliřtirmek, hızlı bir prototip yapmak (statik), tasarımı biraz deđiřtirmek ve ardından gerçekte verilerle canlı bir prototip oluřturmak hakkında kafamızda birkaç konseptle yapıldı. Bu prototipte lastikleri tekmeledikten sonra, genellikle hareket halindeki gerçekte bir proje ve iyi sonuç elde ettik.

—Mark Gallagher, kurumsal intranet geliřtiricisi (Signal vs. Noise'dan)

## 60: Ölü Belgeleri Yapmayın

### Gereksiz evrak işlerini ortadan kaldırın

İşlevsel özelliklerden kaçınmak iyi bir başlangıçtır, ancak burada durmayın; Fazla evrak işlerini her yerde önleyin. Bir belge gerçekten gerçek bir şeye dönüşmeyecekse, onu üretmeyin.

Yapın, yazmayın. Bir şeyi açıklamanız gerekiyorsa, uzun soluklu bir belge yazmak yerine onu taklit etmeyi ve prototiplemeyi deneyin. Gerçek bir arayüz veya prototip, gerçek bir ürün olma yolundadır. Öte yandan bir kağıt parçası sadece çöp kutusuna gidiyor.

İşte bir örnek: Bir tel kafes belgesinin kaderinde durma ve asla doğrudan gerçek tasarım olmaması gerekiyorsa, bunu yapmaya zahmet etmeyin. Tel kafes bir tel kafes olarak başlar ve ardından asıl tasarıma dönüşürse, bunun için gidin.

Başvurunuzdan ayrı yaşayan belgeler değersizdir. Seni hiçbir yere götürmezler. Yaptığınız her şey gerçek şeye dönüşmelidir. Bir belge gerçeğe dönüşmeden önce durursa, ölüdür.

### Kimse Okumayacak

Biz kod yazarken, sorunları tartışırken, sorular sorarken ve kullanıcı testleri yaparken geliştirici ekibimin yanında çürüyen, okunmamış, toz toplayan çok sayfalı ürün spesifikasyonları veya iş gereksinimi belgelerinin sayısını bile sayamıyorum. Hatta saatlerce uzun, açıklayıcı e-postalar yazmak veya okunmayan standart belgeleri kodlamak için harcayan geliştiricilerle bile çalıştım.

Webapps, bol belgelerle ilerlemez. Yazılım geliştirme, etkileşim, ani kararlar ve yol boyunca ortaya çıkan tahmin edilmesi imkansız sorunları içeren sürekli değişen, yinelemeli bir süreçtir. Bunların hiçbirini kağıda yazılamaz veya alınmamalıdır.

O uzun vizyoner cildi yazarak zamanınızı boşa harcamayın; kimse okumayacak. Ürününüze kendi kendine büyümesi için yeterli alan vererseniz, sonunda hakkında yazdığınız hiçbir şeye benzemeyeceği gerçeğiyle teselli edin.

—Gina Trapani, web geliştiricisi ve Lifehacker editörü, Üretkenlik ve yazılım kılavuzu

## 61: Bana Kısa Bir Hikaye Anlat

### Detay değil hikaye yazın

Yeni bir özelliği veya kavramı açıklamak için kelimelere ihtiyaç duyduğunuzu fark ederseniz, bunun hakkında kısa bir hikaye yazın. Teknik veya tasarım detaylarına girmeyin, sadece kısa bir hikaye anlatın. Normal konuşmada yaptığınız gibi, insanca yapın.

Makale olmasına gerek yok. Sadece olanın akışını verin. Ve kısa hikayeyi, geliştirmekte olduğunuz ekranlar bağlamında dahil edebilirsiniz, daha iyi.

Ayrıntılara takılmak yerine deneyime bağlı kalın. Taktik değil, strateji düşünün. Uygulamanızın bu bölümünü oluşturmaya başladığınızda taktikler yerine oturacaktır. Şu anda sadece konuşmayı başlatacak ve sizi doğru yola sokacak bir hikaye başlatmak istiyorsunuz.

## 62: Gerçek Kelimeleri Kullanın

### lorem ipsum yerine gerçek metni ekleyin

Lorem ipsum dolor, tasarımcıların güvenilir bir dostudur. Sahte metin, insanların tasarımın ete kemiğe büründükten sonra nasıl görüneceğini anlamalarına yardımcı olur. Ancak sahte metinler de tehlikeli olabilir.

Lorem ipsum, kopyanın görüntülenme şeklini değiştirir. Metin tabanlı içeriği, olması gereken şey yerine görsel bir tasarım öğesine (metnin şekli) indirger: birinin girmesi ve/veya okuması gereken değerli bilgiler. Sahte metin, gerçek bilgiler girildiğinde ortaya çıkan kaçınılmaz varyasyonları görmeyeceğiniz anlamına gelir. Bu, sitenizdeki formları doldurmanın nasıl bir şey olduğunu bilemeyeceğiniz anlamına gelir. Sahte metin, sizinle gerçeklik arasında bir perdedir.

Belirli alanların ne kadar uzun olması gerektiğini bilmek için gerçek kopyaya ihtiyacınız var. Tabloların nasıl genişleyeceğini veya daralacağını görmek için gerçek kopyaya ihtiyacınız var. Uygulamanızın gerçekte nasıl görüldüğünü bilmek için gerçek bir kopyaya ihtiyacınız var.

Mümkün olan en kısa sürede, gerçek ve alakalı kelimeler kullanın. Siteniz veya uygulamanız veri girişi gerektiriyorsa, gerçek anlaşılmayı girin. Ve aslında metni yazın - başka bir kaynaktan yapıştırmayın. Bu bir adsa, gerçek bir ad yazın. Bu bir şehirse, gerçek bir şehir yazın. Bu bir parolaysa ve iki kez tekrarlanıyorsa, iki kez yazın.

Tabii ki, formları hızla gözden geçirmek ve alanları çöple ("asdsadklja" "123usadfasld" "snaxn2q9e7") doldurmak daha kolaydır. Ama bu gerçek değil. Müşterilerinizin yapacağı şey bu değil. Müşteriler uzun bir yola çıkmak zorunda kaldığında kestirmeden gitmek gerçekten akıllıca mı? Hızlı bir şekilde sahte kopyayı girdiğinizde, bu formu doldurmanın gerçekten nasıl bir his olduğunu bilmiyorsunuz.

Müşterilerinizin yaptığını yapın, onları daha iyi anlayacaksınız. Onları daha iyi anladığınızda ve hissettiklerini hissettiğinizde daha iyi bir arayüz oluşturacaksınız.

### Lorem Ipsum Çöpü

İçeriğin "olabileceğini" hayal edecek hayal gücüne sahip olmamakla, tasarım düşüncesi kaybolur. Anlam bulanıklaşıyor çünkü "sadece metin", anlaşılabilirlik tehlikeye giriyor çünkü kimse bu metinlerin aslında okunması gerektiğini anlamadı.

Gerçek içerik yerine kullandığınız lorem ipsum çöpü fırsat önermediği için fırsatlar kaybolur. Metin daha sonra gerçekten küçülür, çünkü kullanılması amaçlanmamıştır, o güzel beyaz boşluktan bir sürü oluşturabiliriz.

—Tom Smith, tasarımcı ve geliştirici ( [Lorem Ipsum](#) ve [Lorem Ipsum](#) [Kullanıcılarından nefret ediyorum](#) )

## 63: Ürününüzü Kişiselleştirin

### Ürününüzün kişilik tipi nedir?

Ürününüzü bir insan olarak düşünün. Ne tür bir insan olmasını istersin? Kibar? Sert mi?

bağışlayıcı mı? Sıkı? Eğlenceli? Sürgü mü? Ciddi? Gevşetmek? Paranoyak mı yoksa güvenen biri olarak mı çıkmak istersin? Her şeyi bilen biri olarak mı? Yoksa mütevazı ve sevimli mi?

Karar verdikten sonra, ürün oluşturulurken bu kişilik özelliklerini daima aklınızda bulundurun. Metin yazarlığını, arayüzü ve özellik setini yönlendirmek için bunları kullanın. Ne zaman bir değişiklik yaparsanız, kendinize bu değişikliğin uygulamanızın kişiliğine uygun olup olmadığını sorun.

Ürününüzün bir sesi var ve müşterilerinizle günün 24 saati konuşuyor.

## 64: Ücretsiz Örnekler

### Ücretsiz bir şey ver

Dışarıda gürültülü bir dünya var. Gürültünün ortasında insanların sizi fark etmesini sağlamak için ücretsiz bir şeyler verin.

Akıllı şirketler, bedava dağıtmanın müşterileri cezbetmek için harika bir yol olduğunu biliyor. Apple'a bakın. iPod ve iTunes müzik mağazasına talep oluşturmak için iTunes yazılımını ücretsiz olarak sunarlar. Çevrimdışı dünyada perakende satış mağazaları da aynı şeyi yapıyor. Starbucks, müşterilere verdikleri her beş içecek numunesi için yeni bir satın almanın teşvik edildiğini söylüyor. O kadar da eski püskü değil.

Bizim için Writeboard ve Ta-da list, insanları diğer ürünlerimizi kullanma yoluna sokmak için kullandığımız tamamen ücretsiz uygulamalardır. Ayrıca, her zaman tüm uygulamalarımızın bir çeşit ücretsiz sürümünü sunuyoruz.

İnsanların ürünü, arayüzü ve ürettiklerimizin kullanışlılığını deneyimlemelerini istiyoruz.

Bağlandıklarında, ödeme planlarından birine yükseltme olasılıkları çok daha yüksektir (bu, daha fazla projeye veya sayfaya izin verir ve insanlara dosya yükleme ve ssl veri şifreleme gibi ek özelliklere erişim sağlar).

Isırık büyüklüğünde parçalar

Isırık büyüklüğünde parçalar yapın: Müşterilerin ısırmasını sağlamak için özel, daha küçük teklifler tasarlayın. En az bir ürünü veya hizmeti ucuz, kolay veya eğlenceli küçük parçalara ayırmaya karar verin.

—Ben McConnell ve Jackie Huba, Church of the Customer Blog'un yazarları ( Müşteri Evangelizmi Nedir?'den)

### Hit Single'ınızı Verin

Şarkılarınızdan birini (albüm başına) dünyaya ücretsiz tanıtım indirmesi olarak vermeyi düşünün - film fragmanı gibi olmak - radyoya gönderilen hit single gibi - insanların müziğinizi satın almak istemesini sağlayan şarkı.

Bu şarkı için korsanlık konusunda endişelenme. İnsanların oynamasına, kopyalamasına, paylaşmasına, vermesine izin verin. Eğer dünya duyarsa, daha fazlasını ödeyeceklerine güvenin.

—Derek Sivers, başkan ve programcı, CD Baby ve HostBaby (Free Promo Track'ten)

## 65: Kolay Açma, Kolay Kapatma

### Kaydolma ve iptal işlemini ağrısız hale getirin

Uygulamanıza girmeyi ve uygulamadan çıkmayı mümkün olduğunca kolaylaştırın.

Uygulamanızı kullanmak isteyen bir müşteriyseniz, zahmetsiz ve zahmetsiz bir süreç olmalıdır. Açılan ve pazarlama sitenizin her sayfasına yerleştiren büyük, net bir kayıt düğmesi sağlayın. İnsanlara bunun ne kadar kolay olduğunu söyleyin:

"Kayıttan girişe sadece 1 dakika içinde!"

Müşterilerin kredi kartı bilgilerini girmeden uygulamanın demosunu yapabilmeleri için her zaman ücretsiz bir seçenek olmalıdır. Rakiplerimizden bazıları, ürünlerini denemek için bir geri arama, randevu veya özel bir şifre gerektirir.

Bununla ne alakası var? Herkesin uygulamalarımızı istediği zaman ücretsiz denemesine izin veriyoruz.

Kayıt formunu mümkün olduğunca kısa tutun. İhtiyacın olmayan şeyleri isteme ve insanlara uzun süre göz korkutucu bir form atma.

Aynı ilkeler iptal işlemi için de geçerlidir. İnsanları ürününüzün içinde asla "tuzağa düşürmek" istemezsiniz. İnsanlar Basecamp hesaplarını iptal etmeye karar verdiğinde üzgün olsak da, bu süreci asla göz korkutucu veya kafa karıştırıcı hale getirmeyiz. "Hesabımı iptal et", bir kişinin hesap sayfasında gün gibi görünen bir bağlantıdır. Gönderilecek herhangi bir e-posta, doldurulacak özel bir form veya cevaplanacak sorular olmamalıdır.

Ayrıca, ayrılmaya karar verilerse insanların verilerini alabileceklerinden emin olun. Müşterilerin tüm mesajları ve yorumları istedikleri zaman xml formatında kolayca dışa aktarabilmelerini sağlıyoruz. Bu onların verileri ve onunla istediklerini yapabilmeliler.

Bu çok önemlidir çünkü insanlara bilgileri üzerinde kontrol vermek güven oluşturur. Onlara veri adalarına bir köprü veriyorsunuz. Daha iyi bir teklif bulurlarsa cezasız ayrılmalarına izin veriyorsunuz. Bu yapılacak doğru şey ve iyi niyet inşa ediyor.

#### Kolayca Çıkın

Kullanıcıları isteklerine karşı tutmayın. Ayrılmak istiyorlarsa, sitenizdeyken oluşturdukları tüm içeriği alıp bıraksınlar ve bedavaya... Ahır kapısının açılmasına izin vermeli ve müşterilerinizin karnını doyurmaya odaklanmalısınız ki onlar gelsinler. sıkışıp kaldıkları için geri gelmek yerine geri dönerler.

—Charlie O'Donnell, analist, Union Square Ventures (10 Adımdan Çok Başarılı Bir Web 2.0 Şirketine)

## 66: Aptal Tavşan, Hileler Çocuklar içindir

Uzun vadeli sözleşmelerden, kayıt ücretlerinden vb. kaçınin.

Kimse uzun vadeli sözleşmelerden, erken fesih ücretlerinden veya tek seferlik kurulum ücretlerinden hoşlanmaz. Bu yüzden onlardan kaçınin.

Ürünlerimiz aylık olarak faturalandırılmaktadır. İmzalanacak bir sözleşme yok ve istediğiniz zaman herhangi bir ceza ödemedi iptal edebilirsiniz. Ve asla herhangi bir kurulum ücreti yoktur.

Daha fazla para kazanmanın "aldatıcı" yollarını bulmaya çalışmayın. Kazanmak.



## 67: Daha Yumuşak Bir Kurşun

### Önceden haber verme ve büyükbaba cümleleri ile kötü haberlerin etkisini yumuşatın

Fiyat artışı gibi kötü haberler mi vermeniz gerekiyor? İnsanlara önceden haber vererek mümkün olduğunca acısız hale getirin. Ayrıca, belirli bir süre için mevcut müşterileri muaf tutan bir büyükbaba dönemi düşünün. Bu insanlar sizin ekmek ve tereyağınızdır ve onlara oyulmuş değil, değerli hissettirmek istersiniz.

## 68: Hollywood'un Lansmanı

### Tanıtımdan önizlemeye ve başlatmaya gidin

Bir uygulama ormanda başlatılırsa ve onu kullanacak kimse yoksa, ses çıkarır mı? Buradaki nokta, uygulamanızı herhangi bir ön heyecan olmadan başlatırsanız, insanların bunu bilmeyecek olmasıdır.

Heyecan ve beklenti oluşturmak için Hollywood tarzı bir lansman yapın: 1) Teaser, 2) Önizleme ve 3) Lansman.

### Tanıtım

Birkaç ay öncesinden ipuçlarını bırakmaya başlayın. İnsanlara ne üzerinde çalıştığınızı bildirin. Bir logo yayınlayın. Geliştirme hakkında blogunuza gönderin. Belirsiz kal ama tohumu ek. Ayrıca, ilgilenen insanlardan e-posta toplayabileceğiniz bir site kurun.

Bu aşamada, mavenleri ve içerdekileri baştan çıkarmaya da başlamalısınız. Bunlar son teknoloji insanlar.

Onlar tadımcılar. Kendi kibirlerine ve eğrilerin ötesindeki statülerine hitap edin.

Onlara özel bir ön izleme aldıklarını söyleyin. Boing Boing, Slashdot veya Digg gibi bir site uygulamanızı bağlarsa, çok sayıda trafik ve takipçi alırsınız. Ayrıca, Google'daki sayfa sıralamanız da yükselecektir.

### Ön izleme

Lansmandan birkaç hafta önce özellikleri önizlemeye başlayın. İnsanlara sahne arkası erişimi verin.

Ürünün temasını açıklayın. Basecamp için ekran görüntüleri ve vurgulanan hatırlatıcılar, kilometre taşları ve diğer özellikleri yayınladık.

Ayrıca insanlara uygulamanın arkasındaki fikir ve ilkeleri anlatın. Backpack için lansmandan önce manifestomuzu yayınladık. Bu, insanların uygulama hakkında düşünmesini ve konuşmasını sağladı.

Uygulamayı erken kullanmaya başlamaları için birkaç kişiye özel "altın biletler" de sunabilirsiniz. İnsanların erken benimseyenlerden aldıkları o özel parıltıyı hissedecekken, bazı beta test kullanıcılarına sahip olmanın avantajını elde edeceksiniz.

Ve bir kez daha, insanları kaydolmaya teşvik edin, böylece lansman yaptığınızda e-postalardan oluşan bir temele sahip olursunuz. Uygulamalarımızı başlattığımızda, ping atılacak binlerce e-postamız var ve bu da ilgi çekmede büyük yardımcı oluyor.

## Başlatmak

Yayın zamanı. Artık insanlar gerçekten "tiyatroya" gidebilir ve uygulamanızı görebilir. Kaydolanlara e-posta gönderin. Tam pazarlama sitenizi başlatın. Müjdeyi mümkün olduğunca yayın.

Size bağlantı verecek bloglar edinin. İlerlemeniz hakkında mesaj gönderin: Kaç kişi kaydoldu? Hangi güncellemeleri/incelemleri yaptınız? Momentumu göster ve devam et.

## Lansman Gününe Giden Yol

Blinksale'in gerçekten olacağını öğrenir öğrenmez, posta listemize bazı tanıtımlar göndermeye başladık. Bunlar, projelerimiz hakkında bizden bilgi almak isteyen kişilerdir. Dilerseniz bunlar bizim hayranlarımız. Zaten bir grup insanla konuşma izniniz varsa, başlamak için en iyi yer onlar.

Yaptığımız ikinci şey, ürünümüz hakkında daha fazla insanla konuşmak için izin almak. Blinksale lansmanından yaklaşık altı hafta önce, web sitemize çevrimiçi fatura göndermenin daha kolay bir yolunun geldiğini ilan eden bir tanıtım sayfası koyduk. Sayfa, gizli kalması gereken hassas ayrıntıları vermeden, heyecan ve gerilim oluşturmaya yetecek kadar bilgi verdi. Sayfada göze çarpan bir şekilde, ürün piyasaya sürüldüğünde ilgilenenlerin bilgilendirilebilmesi için bir e-postadan (basit tutun) başka bir şey gerektirmeyen bir haber bülteni abonelik formu vardı.

Ürünle de ilgileneceğini düşündüğümüz bir düzine kadar arkadaş ve meslektaşımıza haberi yaydık ve onlar da blogları ve web siteleri aracılığıyla teaser sayfası hakkındaki haberleri yaymaya başladılar. Birkaç gün içinde posta listemizde binlerce kişi vardı. Bunlar son derece önemli insanlardı - ürünümüz hakkında daha fazla bilgi edinmek isteyen ve piyasaya sürdüğümüzde bilmek isteyen insanlar.

Son olarak, lansmandan yaklaşık iki hafta önce, Blinksale beta testi yapmamıza yardımcı olmaları için bir avuç arkadaşımızı, meslektaşımızı ve endüstri liderini davet ettik. Bu, ürünü, kullanımından yararlanabileceğini düşündüğümüz ve piyasaya sürdüğümüzde ürünle ilgili kelimeyi yaymamıza yardımcı olabilecek kişilerin önüne çıkarmamızı sağladı. Kimseyi ürünü kullanmaya veya ürün hakkında yazmaya zorlamadığımızı belirtmek önemlidir. Biz sadece onun görülmesini ve piyasaya çıktığında insanların onun hakkında konuşmasını istedik. Sonunda, bu şekilde ses getirekseniz, ürününüzün teslim edebileceğinden emin olsanız iyi olur. Aksi takdirde, yağmursuz bulutlar gibidir.

Lansman günü geldiğinde, e-posta listemize bir e-posta gönderdik, blog yazan arkadaşlarımızı bilgilendirdik ve beta testçilerimizi fikirlerini söylemeye teşvik ettik. Ve bizim için büyük bir zevk olan çaba, büyük temettüler ödedi. Lansmandan kısa bir süre sonra on binlerce kişi sitemizi ziyaret etti ve binlerce kişi ürünü kullanmak için kaydoldu.

## 69: Güçlü Bir Promosyon Sitesi

### Tanıtımdan önizlemeye ve başlatmaya gidin

En iyi tanıtım aracı harika bir üründür. İnsanların gerçekten yararlı bulduğu bir uygulamanız varsa, Word dışarı çıkar.

Yine de, bir as promosyon sitesine de ihtiyacınız var. Bu siteye ne eklemelisiniz? Bazı fikirler:

- Genel Bakış: Uygulamanızı ve faydalarını açıklayın.
- Tur: İnsanları çeşitli özelliklerle yönlendirin.
- Ekran görüntüleri ve videolar: İnsanlara uygulamanın gerçekte nasıl görüldüğünü ve nasıl kullanılacağını gösterin.
- Manifesto: Arkasındaki felsefeyi ve fikirleri açıklayın.
- Vaka Çalışmaları: Nelerin mümkün olduğunu gösteren gerçek yaşam örnekleri sağlayın.
- Buzz: Müşterilerden, incelemelerden, basından vb. gelen referans alıntıları.
- Forum: Topluluk üyelerinin birbirlerine yardım edebilecekleri bir yer sunun.
- Fiyatlandırma ve Kayıt: İnsanları mümkün olduğunca çabuk uygulamanıza çekin.
- Web günlüğü: Bloglar, sitenizi haberler, ipuçları vb. ile güncel tutar.

## 70: Blog Dalgasına Binin

### Blog yazmak reklamcılıktan daha etkili olabilir (ve çok daha ucuzdur)

Reklam pahalıdır. Ve çeşitli reklam türlerinin etkinliğini değerlendirmek, reklamın kendisinden bile daha pahalı olabilir. Geleneksel reklamcılık rotasına gitmek için zamanınız veya paranız yoksa, bunun yerine blog aracılığıyla tanıtım yolunu düşünün.

Yalnızca ürününüzü tanıtmakla kalmayan, aynı zamanda yararlı tavsiyeler, ipuçları, püf noktaları, bağlantılar vb. sunan bir blog oluşturarak başlayın. Signal vs. Noise blogumuz, yararlı, bilgilendirici ve ilginç bitler ve anekdotlar sayesinde her hafta binlerce benzersiz okuyucu alıyor. günlük olarak yayınlıyoruz.

Böylece ilk ürünümüz Basecamp'ı tanıtmaya zamanı geldiğinde, oradan başladık. Haberi SvN'den aldık ve yayılmaya başladı. Jason Kottke, BoingBoingers, Jim Coudal ve popüler blogları olan diğer çeşitli kişiler, görünürlüğü artırmaya yardımcı oldu ve işler hız kazandı.

Ta-da Lists, blog tabanlı pazarlamanın gücünün bir başka harika örneğidir. Ta-da'yı Signal vs. Noise üzerine tek bir gönderiyle başlattık ve birkaç hafta sonra 200'den fazla blogda bundan bahsedildi ve 12.000'den fazla kişi kendi Ta-da hesabına kaydoldu. Sırt Çantası hakkındaki haberler daha da hızlı yayıldı. Lansmandan sonraki 24 saat içinde 10.000'den fazla kayıt oldu.

## 71: Erken Talep Etme

### En kısa sürede önceden haber alın ve kaydolun

Buna zaten değindik ama tekrar etmekte fayda var: Bir tür site kurun ve mümkün olan en kısa sürede e-posta toplamaya başlayın. Alan adınızı seçin ve uygulamanızın ne yapacağını açıklayan veya en azından ipuçları veren bir logo ve belki bir veya iki cümle koyun. O zaman insanların size e-posta adreslerini vermelerine izin verin. Şimdi, lansmanınızdan haberdar edilmeyi bekleyen ve hazır bir insan temeline sahip olma yolundasınız.

# 72: Eğitim Yoluyla Tanıtım

## Bilginizi dünyayla paylaşın

Bir öğretmen Jeopardy'de yarışmacı olarak görüldüğünde, Alex Trebek sık sık bunun “asil bir meslek” olduğunu söyler. O haklı. Bilginizi başkalarıyla paylaşmak konusunda kesinlikle harika ve ödüllendirici bir şey var. Ve öğrettiğiniz konu sizin uygulamanız olduğunda, iki amaca hizmet eder: Topluluğa sizi destekleyen bir şeyi geri verebilir ve aynı zamanda güzel bir tanıtım teşhiri elde edebilirsiniz.

Bir tanıtım tekniği olarak eğitim, adınızı ve ürününüzün adını daha fazla insanın önüne çıkarmanın yumuşak bir yoludur. Ve sıkı satış “bu ürünü satın al” yaklaşımı yerine, değerli bir hizmet sunarak dikkat çekiyorsunuz. Bu, geleneksel pazarlama taktiklerinin eşleşemeyeceği olumlu bir vızıltı yaratır. Yetiştirdiğiniz insanlar, evangelistleriniz olacak.

Eğitim birçok biçimde gelebilir. İnsanların başkalarıyla paylaşmak isteyecekleri ipuçlarını ve püf noktalarını sitenizde yayınlayın. Konferanslarda konuşun ve daha sonra katılımcılarla tanışmak ve selamlaşmak için kalın. Meraklı hayranların daha fazla bilgi edinebilmesi ve sizinle canlı konuşabilmesi için atölye çalışmaları düzenleyin. Yayınlara röportaj verin. Yararlı bilgileri paylaşan makaleler yazın. Ve kitaplar yaz. ;)

Kendi tarihimizden bir örnek, bir sayfada yakın zamanda değiştirilen bir alanı incelikte vurgulamak için icat ettiğimiz bir yöntem olan Sarı Solma Tekniğidir. Signal vs. Noise'da bununla ilgili bir yazı yazdık. Bu gönderi tur attı ve binlerce ve binlerce sayfa görüntüleme aldı (bu güne kadar büyük trafik yapıyor).

Gönderi hem eğitim hem de tanıtım düzeyinde çalıştı. Bir ders alındı ve ürünlerimizden asla haberdar olmayacak birçok insan onlara maruz kaldı. Başka bir örnek: Ruby on Rails'i geliştirmemiz sırasında altyapıyı açık kaynak yapmaya karar verdik. Akıllıca bir hareket olduğu ortaya çıktı. Topluluğa bir şeyler geri verdik, iyi niyet oluşturduk, ekibimiz için takdir topladık, faydalı geri bildirimler aldık ve dünyanın her yerinden programcılardan yamalar ve katkılar almaya başladık.

Öğretim tamamen iyi karma ile ilgilidir. Peşin ödüyorsun. Başkalarına yardım ediyorsun. Sağlıklı bir terfi alırsın. Ve hatta biraz asaletin tadını çıkarabilirsiniz. Peki, dünyanın duymak istediği ne biliyorsun?

## İleri Öde

Blogumuzun makaleler ve ipuçları bölümü sitemizin en popüler bölümlerinden biridir. E-posta pazarlama konusundaki bilgilerimizi aktarmak, müşterilerimizin hizmetlerimizden en iyi şekilde yararlanmalarını sağlar.

yazılım. Müşterilerine daha iyi hizmet verebilirlerse, daha fazla iş elde etmeleri muhtemeldir, bu da bizim için daha fazla iş yaratır - herkes kazanır.

Bilgimizi özgürce paylaşmak, sektörde uzman olarak konumlanmamıza da yardımcı oldu ve mevcut müşterilerle ilişkilerimizi güçlendirdi. İşlerinin kalitesine önem verdiğimizizi biliyorlar. Son olarak, makalelerimizi okuyucularıyla paylaşan arama motorlarından ve blog yazarlarından çok sayıda hedefli gelen trafik alıyoruz. Bunlar, biz o makaleyi yazmamış olsaydık, yazılımımızı asla duyamayacaklardı.

—David Greiner, kurucu, Campaign Monitor



# 73: Özel Yemek

## Bunun için açlar, bu yüzden servis yapın

Yeni veya ilginç özellikler, uygulamanız için vızıltı oluşturma harika bir yoldur. Özel ilgi grupları, “özellikli yiyecekleri” çığnemeyi ve onu topluluğa geri tükürmeyi sever. Pekala, bu hoş olmayan bir benzetme ama sen anladın.

Örneğin, yeni bir geliştirme platformu olan Ruby on Rails'i kullanarak, geliştirici topluluğu içinde Basecamp'a büyük ilgi topladık.

Uygulamalarımızda kullandığımız Ajax öğeleri çok ses getirdi ve hatta Business 2.0 dergisinin Basecamp'ı Google, Yahoo, Microsoft ve Amazon gibi büyük isimlerin yanı sıra “Ajax'ta önemli bir oyuncu” olarak adlandırmasına yol açtı.

Başka bir örnek: Blogcular, RSS'nin ilk ticari örneklerinden biri olduğu için Basecamp'ın RSS desteğini fark ettiler.

Görünüşte önemsiz bir özellik olan iCal entegrasyonu, başka türlü uygulamadan asla bahsetmeyecek olan bir ton Mac ile ilgili siteye basmamızı sağladı.

Küçük ekipler, yeni fikirleri yazılıma entegre etme konusunda avantajlıdır. Daha büyük şirketler bürokratik darboğazlarla uğraşmak zorunda kalırken, yeni fikirleri hızla hayata geçirebilir ve bunları kullanmak için dikkat çekebilirsiniz.

Teknoloji du jour'un hype paltolarına binmek, vızıltı oluşturma etkili ve ucuz bir yoldur. Bununla birlikte, sadece biraz dikkat çekmek için en son belirsiz teknolojiyi eklemeyin. Ancak yeni veya kayda değer bir şey kullanıyorsanız, devam edin ve özel ilgi için öne çıkarın. gruplar.

## 74: Günlüklerinizi Takip Edin

### Buzz'ı izlemek için günlüklerinizi inceleyin

Senin hakkında kimin konuştuğunu bilmelisin. Günlüklerinizi kontrol edin ve vızıltıların nereden geldiğini öğrenin. Sana kim bağlanıyor? Kim senin hakkında gıcıklık yapıyor? Technorati, Blogdex, Feedster, Del.icio.us ve Daypop'ta listelenen hangi bloglar iz üzerinde sıcak?

Keşfedin ve sonra varlığını hissettirin. Bu bloglara yorum bırakın. Bağlantı gönderen insanlara teşekkür edin. Özel ilerleme listenize dahil edilmek isteyip istemediklerini sorun, böylece gelecekteki sürümler, güncellemeler vb. hakkında ilk öğrenenler arasında olurlar. Olumlu övgü toplayın ve sitenizde bir “vızıltı” sayfası oluşturun. Üçüncü taraf övgüleri çoğu insan için daha güvenilir olduğundan, referanslar uygulamanızı tanıtmanın harika bir yoludur.

Yorumlar olumsuzsa, yine de dikkat edin. Dinlediğinizi gösterin. Eleştirilere düşünceli bir şekilde yanıt verin. Şuna benzer bir şey: "Geri bildirim için teşekkür ederiz ama bunu bu şekilde yaptık çünkü..." Veya "İyi bir noktaya değindiniz ve bunun üzerinde çalışıyoruz." Eleştirmenlerinizi yumuşatacak ve ürününüze bir insan yüzü koyacaksınız. Bir bloga yapılan düşünceli bir yorumun, karşı çıkanları bu kadar dağıtabilmesi ve hatta şikayetçileri evangelistlere dönüştürebilmesi şaşırtıcı.

## 75: Satır İçi Satış

### Uygulama içinde yükseltme fırsatlarını tanıtın

Herkes pazarlama sitesinde adım atmayı bilir. Ama satış burada bitmemeli. Katmanlı bir fiyatlandırma planınız (veya uygulamanızın ücretsiz bir sürümü) varsa, ürünün içinden yükseltme fırsatlarını aramayı unutmayın.

İnsanlara, yükseltmeleri halinde engelleri kaldıracağınızı söyleyin. Örneğin, ücretsiz bir hesabınız varsa Basecamp'ta dosya yükleyemezsiniz. Birisi bir dosya yüklemeye çalıştığında, onları sadece geri çevirmiyoruz. Dosya yüklemenin neden kullanılamadığını açıklıyoruz ve onları ücretli sürüme yükseltmeye teşvik ediyoruz ve bunun neden iyi bir fikir olduğunu açıklıyoruz. Aynı yaklaşım, mevcut müşterileri mevcut planlarını maksimuma çıkardıklarında daha yüksek bir hesaba yükseltmeye teşvik etmek için de kullanılır.

Mevcut müşteriler, satışlar için en iyi seçeneğinizdir. Ürününüzü zaten bilen ve kullanan kişilerden tekrar iş almaya çalışmaktan çekinmeyin.

## 76: İsim Kancası

### Uygulamanıza hatırlaması kolay bir ad verin

Pek çok insanın yaptığı büyük bir hata, uygulama adının aşırı açıklayıcı olması gerektiğini düşünmektir. Aracınızın amacını canlı bir şekilde açıklayan bir ad seçme konusunda endişelenmeyin; Bu genellikle genel, unutulabilir bir isme yol açar. Basecamp, Proje Yönetim Merkezi veya ProjectExpress gibi bir şeyden daha iyi bir isimdir. Writeboard CollaborEdit'ten daha iyidir.

Ayrıca, adlandırma sürecini çok fazla grup veya komiteye odaklamayın. Kısa, akılda kalıcı ve akılda kalıcı bir isim seçin ve ardından onunla birlikte koşun.

Ve tam olarak istediğiniz alan adını alamıyorsanız üzülmeyin. Her zaman yaratıcı olabilir ve birkaç ekstra harfle yakınlaşabilirsiniz (örneğin, backpackit.com veya campfirenow.com).

### Kolay Yapar

Teknoloji endüstrisi, akılda kalıcı, kendi kendini açıklayan isimler düşünmenin nihayetinde ona aynı şekilde fayda sağlayacağını farkında değil mi? Her ne ise ondan daha çok satarlardı çünkü bir grup kibirli mühendis tarafından ileri teknoloji kulübünün dışında tutulduklarını düşünen tüketicileri korkutmazlardı. Teknoloji de daha hızlı yakalanacaktı. Yeni ürünü tanımlaması, kullanması ve satın alması daha kolay olacak - bu da şirketler için satmanın daha kolay olduğu anlamına geliyor.

—David Pogue, köşe yazarı, New York Times ( Ürün Adında Ne Var?'dan)

## 77: Acıyı Hisset

### Destek ve geliştirme arasındaki duvarları yıkın

Restoran işinde, mutfakta çalışanlarla ön tarafta müşterilerle ilgilenenler arasında dünyalar kadar fark vardır. Her iki tarafın da birbirini anlaması ve empati kurması önemlidir. Bu nedenle, aşçılık okulları ve restoranlar, mutfak personelinin müşterilerle etkileşime geçebilmesi ve ne olduğunu görebilmesi için şeflerin genellikle garson olarak öne çıkmasını sağlar.

aslında ön saflarda olduğu gibi.

Birçok yazılım geliştiricisi benzer bir bölünmeye sahiptir. Tasarımcılar ve programcılar “mutfakta” çalışırken, müşteri destekle ilgilenir. Ne yazık ki bu, yazılım şeflerinin müşterilerin gerçekte ne söylediğini asla duyamayacağı anlamına geliyor. Bu sorunlu çünkü müşterileri dinlemek, ürününüzün güçlü ve zayıf yönleriyle uyum sağlamanın en iyi yoludur.

Çözüm? Müşterileriniz ile geliştirme/tasarım ekibi arasına duvarlar inşa etmekten kaçının.

Müşteri desteğini bir çağrı merkezine veya üçüncü tarafa yaptırmayın. Kendin Yap. Siz ve tüm ekibiniz müşterilerinizin ne dediğini bilmelisiniz. Müşterileriniz rahatsız olduğunda, bunu bilmeniz gerekir.

Şikayetlerini duymanız gerekiyor. Senin de sinirlenmen gerekiyor.

Basecamp'ta tüm destek e-postalarımız, ürünü gerçekten yapan kişiler tarafından kişisel olarak yanıtlanır. Neden? Niye? Öncelikle, müşteriler için daha iyi destek sağlar. Doğrudan uygulamayı oluşturan birinin beyninden yanıt alıyorlar. Ayrıca ürünlerimizi kullanan kişilerle ve karşılaştıkları sorunlarla iletişim halinde olmamızı sağlar. Onlar üzıldüklerinde biz de hüsrana uğruyoruz. “Acını hissediyorum” diyebilir ve aslında bunu kastedebiliriz.

Sorunlu noktalarınızı ortaya çıkarmak için istatistiksel analizlere güvenmek cazip gelebilir. Ancak istatistikler seslerle aynı şey değildir. Siz ve müşterilerinizin gerçek sesleri arasındaki arabellekleri mümkün olduğunca ortadan kaldırmanız gerekir.

Ön saflar aksiyonun olduğu yerdir. Oraya çık. Şeflerinizin garson olarak çalışmasını sağlayın. Müşteri e-postalarını okuyun, hayal kırıklıklarını duyun, önerilerini dinleyin ve onlardan bir şeyler öğrenin.

### Orta Adamı Kes

Hemen hemen tüm Campaign Monitor geliştirme, destek ve pazarlama iki kişi tarafından gerçekleştirilir. Ekibi genişletmek zorunda kalsak bile, desteği asla geliştirmeden ayırmayacağız.

Her talebe bizzat cevap vererek kendimizi müşterilerimizin yerine oturmaya ve olaylara onların bakış açısından bakmaya zorluyoruz.

Müşterinizin sadece neye ihtiyaç duyduğunu değil, neden bir şeye ihtiyacı olduğunu anlamak önemlidir. Bu bağlam genellikle bir şeyi nasıl tasarladığımız üzerinde doğrudan bir etkiye sahiptir. Ortadaki adamı kesin. Kulaklarınız yere bu kadar yakınken müşterilerinize istediklerini vermek çok daha kolay.

Bu düzeni bir sürü insanla tartıştım ve ilk yanıt genellikle “desteklerinizi halletmek için bir genç kiralamanız gerekmez mi?” olur. Kendinizi müşterinizin yerine koyun. Bifteğinizin tam istediğiniz gibi pişmesini istiyorsanız, otobüsçüyle mi yoksa gerçekten pişiren şefle mi konuşmayı tercih edersiniz?

—David Greiner, kurucu, Campaign Monitor

## 78: Sıfır Eğitim

### Ürününüzün bir kılavuz veya eğitim gerektirmemesi için satır içi yardım ve SSS'leri kullanın

Yahoo, Google veya Amazon'u kullanmak için bir kılavuza ihtiyacınız yok. Öyleyse neden kılavuz gerektirmeyen bir ürün yapamıyorsunuz? Sıfır eğitim gerektiren bir araç oluşturmaya çalışın.

Bunu nasıl yapıyorsun? Pekala, daha önce de belirttiğimiz gibi, her şeyi basit tutarak başlayın.

Uygulamanız ne kadar az karmaşıksa, insanlara yabani otlardan kurtulmanız için o kadar az ihtiyaç duyacaksınız. Bundan sonra, desteği önlemenin harika bir yolu, olası karışıklık noktalarında satır içi yardım ve sss kullanmaktır.

Örneğin, insanların logolarını Basecamp'a yüklemelerine olanak tanıyan ekranda önleyici destek sunuyoruz. Bazı kişiler, tarayıcı önbelleğe alma sorunu nedeniyle eski bir logoyu görmeye devam etme sorunu yaşadı. Bu nedenle, "logonuzu gönderin" alanının yanına, müşterilerin yeni logoyu görmek için tarayıcılarını zorla yeniden yüklemelerini isteyen bir sss'ye bağlantı ekledik. Bunu yapmadan önce, bu sorunla ilgili günde 5 e-posta alırdık. Şimdi hiçbirini alamıyoruz.

## 79: Hızlı Cevap

### Destek sorgularında hızlı geri dönüş süresi en önemli öncelik olmalıdır

Sorularını hızlı bir şekilde yanıtladığınızda müşteriler aydınlanır. Günler sonra (eğer varsa) ortaya çıkan hazır yanıtlara o kadar alışmışlar ki, hemen düşünceli bir yanıt sunarak kendinizi rakiplerinizden gerçekten ayırabilirsiniz. Mesai saatleri içinde, tüm e-posta destek taleplerinin %90'ını 90 dakika içinde ve genellikle yarım saat içinde yanıtıyoruz. Ve insanlar onu seviyor.

Mükemmel bir cevabınız olmasa bile, bir şeyler söyleyin. Açık ve dürüst bir şekilde hızlı bir şekilde verilen bir yanıtla iyi niyet satın alabilirsiniz. Birisi hemen çözülemeyecek bir sorundan şikayet ediyorsa, ona "Söylediklerini duyduk ve gelecekte bunun üzerinde çalışacağız" gibi bir şey söyleyin. Potansiyel olarak olumsuz bir durumu yaymanın harika bir yolu.

Müşteriler doğrudanlığı takdir eder ve hızlı ve net bir şekilde yanıt verirsiniz, genellikle kızgından kibara geçer.

### Birçok Ordu

Sadece üç geliştiriciden oluşan küçük bir ekip nasıl yenilikçi bir ürün oluşturabilir ve büyük adamlarla başarılı bir şekilde rekabet edebilir? Cevap, birçok kişiden oluşan bir orduyu askere almaktır.

İlk günden itibaren müşterilerinizin en önemli varlığınız olduğunu ve uzun vadeli başarınız için kesinlikle hayati önemde olduklarını unutmayın, bu nedenle kullanıcı topluluğunuza kraliyet ailesi gibi davranın. Büyük adamlarla rekabet etmenin yolu, küçükten başlamak ve her birine dikkat etmek. müşteriler.

Hatalarda sizi ilk uyaracak olan müşterileriniz, karşılanmayan ihtiyaçlarınızı ilk uyaracak olan ve bayrağı taşıyacak ve yayacak olan ilk müşterileriniz olacaktır.

Mesajın.

Bu, lansman yaptığınızda ürününüzün mükemmel olması gerektiği anlamına gelmez. Tam tersine, erken ve sık salıverin. Bununla birlikte, müşterileriniz hatalarla karşılaştığında, girdileri için teşekkür ederek onlara hızlı bir yanıt gönderdiğinizden emin olun.

Müşteriler, ürününüzün mükemmel olmasını beklemezler ve tüm özelliklerinin uygulanacağını da beklemezler. Ancak, müşteriler sizi dinlemenizi ve onaylamanızı beklerler.



umursadığını, umursadığını göster. Bu, çoğu büyük şirketin büyük bir açık gösterdiği bir alandır, bu nedenle erken bir topluluk duygusu geliştirin.

Blinklist'te, her bir müşteri e-postasına, genellikle ilk saat içinde yanıt verilir (uykuda olmadığımız sürece). Ayrıca çevrimiçi bir forumumuz var ve her bir gönderi ve yorumun onaylandığından emin oluyoruz.

Aynı derecede önemli olan, tüm geliştiricilerimiz müşteri geri bildirimimizi alır ve çevrimiçi tartışma forumlarında aktif katılımcılardır. Bu şekilde, yavaş ama emin adımlarla aktif ve sadık bir BlinkList topluluğu oluşturuyoruz.

—Michael Reining, kurucu ortak, MindValley & Blinklist

## 80: Zor Aşk

### Müşterilerinize hayır demeye istekli olun

Özellik istekleri söz konusu olduğunda, müşteri her zaman haklı değildir. Müşterilerimizin istediği her şeyi ekleseydik, kimse ürünlerimizi istemezdi.

Müşterilerimizin her hevesine uysaydık, Basecamp'ın sahip olacağı özellikler: kapsamlı zaman takibi, kapsamlı faturalandırma, kapsamlı toplantı planlaması, kapsamlı takvim, kapsamlı bağımlılık görev sistemleri, kapsamlı anlık mesaj sohbeti, kapsamlı wiki işlevselliği ve başka ne yaparsanız yapın kapsamlı -hayal etmek.

Yine de, müşteri anketlerinde aldığımız 1 numaralı talep, Basecamp'ı basit tutmaktır.

İşte başka bir örnek: Bazı şikayetlere rağmen ie5'i ürünlerimizle desteklememeye karar verdik. Bu, yazdığımız pazarın %7'siydi. Ama diğer %93 için endişelenmenin daha önemli olduğuna karar verdik. Hataları düzeltmek ve ie5 için test etmek zaman ayırmaya değmez. Herkes için daha iyi bir ürün yapmayı tercih ederiz.

Bir yazılım geliştirme şirketi olarak bir filtre gibi davranmalısınız. Herkesin önerdiği her şey doğru cevap değildir. Tüm istekleri değerlendiriyoruz ama müşteri her zaman haklı olmuyor. Bazı insanları kızdırmak zorunda olduğunuz zamanlar olacak. En güzeli.

Bununla ilgili olarak, bir geliştirme şirketi olarak ürününüzü sevmeniz çok önemlidir. Ve aynı fikirde olmadığınız bir sürü şeyle doluysa, ürününüzü sevmeyeceksiniz. Bu, gerekli olduğuna inanmadığınız müşteri isteklerini veto etmek için başka bir gerekçedir.

# 81: Güzel Forumda

Müşterilerin birbirine yardım etmesine izin vermek için forumları veya sohbeti kullanın

Forumlar ve web tabanlı grup sohbeti, müşterilerin soru sormasına ve birbirlerine yardım etmesine izin vermenin harika bir yoludur. Aracıyı - yani siz - ortadan kaldırarak, açık bir iletişim akışı sağlar ve süreçte kendinize zaman kazandırabilirsiniz.

Ürün forumlarımızda müşteriler ipuçları ve püf noktaları, özellik istekleri, hikayeler ve daha fazlasını yayınlar. Zaman zaman yardım sunmak için içeri giriyoruz, ancak forumlar esas olarak topluluğun birbirine yardım ettiği ve ürünle ilgili deneyimlerini paylaştığı bir yer.

İnsanların birbirine ne kadar yardım etmek istediğine şaşıracaksınız.

## 82: Sıkıntılarınızı Yayınlayın

### Dışarıdan kötü haberler alın ve yoldan çekilin

Bir şeyler ters giderse, insanlara söyle. İlk başta görmemiş olsalar bile.

Örneğin, Basecamp gecenin bir yarısında birkaç saatliğine kapalıydı. Müşterilerimizin %99'u bunu hiç bilmiyordu ama biz yine de Everything Basecamp blogumuza bir "beklenmedik kesinti" bildirimi gönderdik. Müşterilerimizin bilmeyi hak ettiğini düşündük.

İşte bir şeyler ters gittiğinde yayınladığımız bir örnek: "Bu sabah yaşanan kesinti için özür dileriz - bazı insanlar için büyük yavaşlamalara ve kesinti sürelerine neden olan bazı veritabanı sorunları yaşadık. Sorunu düzelttik ve bunun bir daha olmamasını sağlamak için adımlar atıyoruz... Sabrınız için teşekkür ederiz ve bir kez daha kesinti için özür dileriz."

Mümkün olduğunca açık, dürüst ve şeffaf olun. Sır saklamayın veya dönüşün arkasına saklanmayın. Bilgili bir müşteri, en iyi müşterinizdir. Ayrıca, hatalarınızın çoğunun müşterilerinizin zihninde o kadar da kötü olmadığını fark edeceksiniz. Müşteriler, onlara karşı dürüst olduğunuzu bildikleri sürece, genellikle size biraz nefes alma odası vermekten mutluluk duyarlar.

İyi ve kötü haberlerin iletilmesiyle ilgili bir yan not: Kötü haberler geldiğinde, hepsini bir anda ortaya çıkarın. Öte yandan, iyi haberler yavaşça damlatılmalıdır. İyi hisleri uzatabiliyorsan, yap.

### Hızlı, Doğrudan ve Dürüst Olun

Garip gelebilir, ancak en iyi senaryo, şirketin kendisinin kötü haberi bildirmesidir. Bu proaktiftir ve şirketinizin zayıf, savunmacı bir konuma getirilmesini engeller.

—Greg Sherwin, Uygulama Teknolojisi, CNET Başkan Yardımcısı ve Emily Avila, Müdür,  
Calypso Communications (A Primer for Crisis PR'den)

## 83: Bir Aylık Ayar

### Lansmandan 30 gün sonra büyük bir güncelleme yayınlayın

Hızlı bir güncelleme ivme gösterir. Dinlediğinizi gösterir. Kolunuzda daha fazla numara olduğunu gösterir. Size ikinci bir vızıltı dalgası verir. İlk iyi hisleri yeniden onaylar. Size hakkında konuşacak bir şeyler ve hakkında blog yazacak başkaları verir.

Hızlı bir yükseltmenin geleceğini bilmek, lansmandan önce en önemli bileşenlere odaklanmanızı da sağlar. Birkaç şeyi daha sıkıştırmaya çalışmak yerine, sadece temel özellik setini mükemmelleştirerek başlayabilirsiniz. Ardından ürünü gerçek dünyada "havalandırabilirsiniz". Ortaya çıktığında müşteri geri bildirimi almaya başlayabilirsiniz ve bir sonraki aşamada hangi alanlara dikkat edilmesi gerektiğini bilirsiniz.

Bu küçük adım yaklaşımı, Sırt Çantası için iyi çalıştı. Önce temel ürünü piyasaya sürdük ve birkaç hafta sonra, müşterilerimizin bize en çok istediklerini söyledikleri şeyler olduğundan, el bilgisayarları için Backpack Mobile ve etiketleme gibi özellikler ekledik.

## 84: Gönderilerin Gelmesine Devam Edin

### Lansmandan sonra devam eden bir ürün geliştirme blogu tutarak ürününüzün canlı olduğunu gösterin

Başladıktan sonra blog yazmayı bırakmayın. Sık sık (en az haftada bir, mümkünse daha sık) güncellediğiniz özel bir blog tutarak ürününüzün yaşayan bir yaratık olduğunu gösterin.

Dahil edilecek şeyler:

- SSS
- nasıl yapılır
- İpuçları & Hileler
- Yeni özellikler, güncellemeler ve düzeltmeler
- Buzz/basın

Bir blog yalnızca uygulamanızın canlı olduğunu göstermekle kalmaz, şirketinizin daha insani görünmesini de sağlar. Yine, tonu samimi ve kişisel tutmaktan korkmayın. Küçük ekipler bazen her zaman büyük ve ultra profesyonel görünmeleri gerektiğini düşünür. Neredeyse Napolyon Kompleksi'nin bir iş versiyonu gibi. Kulağa küçük gelsin. Müşterilerle bir arkadaş gibi konuşabilmenin keyfini çıkarın.

### Yaşıyor

Sık güncellenen bir ürün blogu, bir web uygulamasının aktif olarak geliştirilmekte olduğunu, sevildiğinin ve evde bir ışığın yandığının en iyi göstergesidir. Terk edilmiş bir ürün blogu, terk edilmiş bir ürünün işaretidir ve sorumlu kişilerin direksiyonda uyuduğunu söyler.

Ürün blogunuzda kullanıcılarınızla sohbete devam edin ve paylaştığınız bilgiler konusunda şeffaf ve cömert olun.

Şirketinizin felsefelerinin parlamasına izin verin.

Rakipleri açık bir şekilde bağlayın ve tartışın. Yaklaşan özellikler hakkında ipucu verin ve yorumları geri bildirim için açık tutun.

Canlı bir ürün, konuşan ve kullanıcılarını dinleyen bir üründür. Sık güncellenen bir ürün blogu, şeffaflığı, topluluk duygusunu ve markanıza bağlılığı teşvik eder. Ekstra, ücretsiz tanıtım bir bonus.

Lifehacker'da editör olarak, Google, Flickr, Yahoo, del.icio.us ve Basecamp ürün blogları gibi sürekli olarak sevdiğim web uygulamalarının ürün bloglarını tarıyorum. Onlardan bahsetme olasılığım çok daha yüksek

birdenbire tek taraflı basın bültenleri gönderen ve kullanıcıları ve hayranlarıyla açık bir görüşme yapmayan web uygulamalarından daha iyidir.

—Gina Trapani, web geliştiricisi ve Lifehacker editörü, üretkenlik ve yazılım kılavuzu

## 85: Daha İyi, Beta Değil

### “Beta”yı günah keçisi olarak kullanmayın

Bu günlerde her şey sonsuza dek beta aşamasındaymış gibi geliyor. Dışarı çıkan bir polis. Bitmek bilmeyen bir beta aşaması, müşterilere gerçekten bitmiş bir ürünü piyasaya sürmeye kararlı olmadığınızı söyler. "Bunu kullan ama mükemmel değilse bizim suçumuz değil" diyor.

Beta parayı müşterilerinize aktarır. Serbest bırakılmanızdan yeterince emin değilseniz, o zaman halkın nasıl olmasını bekleyebilirsiniz? Özel betalar iyidir, genel betalar saçmalıktır. Kamu tüketimi için yeterince iyi değilse, tüketmesi için halka vermeyin.

Ürününüzün mükemmele ulaşmasını beklemeyin. Olmayacak. Serbest bıraktığınız şeyin sorumluluğunu alın. Çıkarın ve buna serbest bırakma deyin. Aksi takdirde, sadece mazeret uydurursunuz.

### Beta Anlamsız

Bunun gibi sorunlara neden olduğu için Google ve diğerlerini suçlayın. Şimdilik, kullanıcılar geliştiriciler topluluğu tarafından "beta"nın gerçekten bir şey ifade etmediğini düşünmek için eğitildi.

—Mary Hodder, bilgi mimarı ve etkileşim tasarımcısı ( The Definition of Beta'dan )

### Her zaman

Sadece ben miyim, yoksa hepimiz her zaman betada mıyız?

—Jim Coudal, kurucu, Coudal Partners



## 86: Tüm Hatalar Eşit Oluşturulmamıştır

### Hatalarınıza öncelik verin (hatta bazılarını görmezden gelin)

Ürününüzde bir hata keşfetmeniz, panik yapma zamanının geldiği anlamına gelmez. Tüm yazılımlarda hatalar vardır - bu sadece hayatın bir gerçeğidir.

Her hatayı anında düzeltmeniz gerekmez. Çoğu hata can sıkıcıdır, yok etmez. Sıkıntılar biraz masaya yatırılabilir. “Doğru görünmüyor” hatalarına veya diğer kabahat hatalarına neden olan hatalar bir süreliğine güvenle bir kenara bırakılabilir. Yine de bir hata veritabanınızı yok ederse, onu hemen düzeltmeniz gerektiği açıktır.

Hatalarınıza öncelik verin. Kaç kişi etkileniyor? Sorun ne kadar kötü? Bu hata acil ilgiyi hak ediyor mu yoksa bekleyebilir mi? Şu anda en fazla sayıda insan üzerinde en büyük etkiye sahip olacak ne yapabilirsiniz? Çoğu zaman yeni bir özellik eklemek, uygulamanız için mevcut bir hatayı düzeltmekten daha önemli olabilir.

Ayrıca, böcekleri çevreleyen bir korku kültürü yaratmayın. Hatalar olur. Sürekli suçlayacak birini aramayın. En son isteyeceğiniz şey, böceklerin açıkça tartışılmak yerine halının altına itildiği bir ortamdır.

Ve dürüstlüğün önemi hakkında daha önce söylediklerimizi hatırlayın. Müşteriler bir hatadan şikayet ederse, onlara karşı dürüst olun. Onlara sorunu not ettiğinizi ve bununla ilgilendiğinizi söyleyin.

Hemen ele alınmayacaksa, nedenini açıklayın ve ürünün daha fazla sayıda insanı etkileyen alanlarına odaklandığınızı söyleyin. Dürüstlük en iyi politikadır.

# 87: Fırtınadan Kurtul

Harekete geçmeden önce, değişikliklere verilen anlık tepkilerin azalmasını bekleyin.

Tekneyi salladığınızda dalgalar olacak. Yeni bir özellik ekledikten, bir politikayı değiştirdikten veya bir şeyi kaldırdıktan sonra, genellikle olumsuz olan anlık tepkiler ortaya çıkacaktır.

Panik yapma veya tepki olarak bir şeyleri hızla değiştirme dürtüsüne direnin. Tutkular başlangıçta alevlenir. Ancak bu ilk 24-48 saatlik süreyi atlarsanız, işler genellikle sakinleşir. Çoğu insan, gerçekten eklediğiniz her şeyi (veya çıkardıklarınız ile anlaştıklarını) incelemeyen ve kullanmadan önce yanıt verir. O yüzden arkanıza yaslanın, hepsini alın ve bir süre geçene kadar hareket etmeyin. O zaman daha mantıklı bir yanıt sunabileceksiniz.

Ayrıca, olumsuz tepkilerin neredeyse her zaman olumlu tepkilerden daha gürültülü ve daha tutkulu olduğunu unutmayın. Aslında, tabanınızın çoğunluğu bir değişiklikten memnun olduğunda bile yalnızca olumsuz sesler duyabilirsiniz. Gerekli, ancak tartışmalı bir kararda aptalca geri adım atmadığınızdan emin olun.

## 88: Jones'lara ayak uydurun

### Rakiplerinize ilgili haber akışlarına abone olun

Hem ürününüz hem de rakiplerinize ilgili haber beslemelerine abone olun (düşmanın yollarını bilmek her zaman akıllıcadır). Güncel kalmak için PubSub, Technorati, Feedster ve diğerleri gibi hizmetleri kullanın (anahtar kelimeler için şirket adlarını ve ürün adlarını kullanın). RSS ile, sürekli değişen bu bilgi size doğrudan iletilecektir, böylece her zaman hızlı olursunuz.

## 89: Şişkinlik Canavarı'na Dikkat Edin

### Daha olgun demek daha karmaşık demek zorunda değil

İşler ilerledikçe, şişkinliğe direnmekten korkmayın. Günaha büyütme olacak. Ama böyle olmak zorunda değil. Bir şeyin yaşanması ve olgunlaşması, daha karmaşık hale gelmesi gerektiği anlamına gelmez.

Baş aşağı yazan bir uzay kalemi olmanıza gerek yok. Bazen sadece bir kalem olmak iyidir. İsviçre çakısı olmanıza gerek yok. Sadece bir tornavida olabilirsin. Müşterileriniz sadece saati bilmek isteyen arazi tutkunlarıysa, 5.000 metrede güvenli bir dalış saati yapmanıza gerek yoktur.

Sadece şişirmek için şişirmeyin. Uygulamalar bu şekilde şişirilir.

Yeni her zaman gelişmiş anlamına gelmez. Bazen bir ürünü olduğu gibi bırakmanız gereken bir nokta vardır.

Bu, geleneksel masaüstü tabanlı yazılımlar yerine web tabanlı yazılımlar oluşturmanın en önemli avantajlarından biridir. Adobe, Intuit ve Microsoft gibi masaüstü yazılım üreticilerinin size her yıl yeni sürümler satması gerekiyor. Ve size aynı sürümü satamayacakları için, yeni özellikler ekleyerek masrafları haklı çıkarmak zorundalar. Şişkinlik burada başlıyor.

Abonelik modeline dayalı web tabanlı yazılım ile insanlar hizmeti kullanmak için aylık bir ücret öderler. Gittikçe daha fazlasını ekleyerek onları büyütmeye devam etmenize gerek yok, sadece devam eden değerli bir hizmet sağlamanız gerekiyor.

## 90: Akışla Devam Edin

### Yeni yollara ve yön değişikliklerine açık olun

Bir web uygulamasının güzelliğinin bir parçası da akışkanlığıdır. Bir kutuya sarıp gönderip bir sonraki sürüm için yıllarca beklemezsiniz. İlerledikçe ince ayar yapabilir ve değiştirebilirsiniz. Orijinal fikrinizin en iyisi olmayabileceği gerçeğine açık olun.

Flickr'a bakın. The Game Neverending adlı çok oyunculu bir çevrimiçi oyun olarak başladı. Yaratıcıları, oyunun fotoğraf paylaşım yönünün oyunun kendisinden (sonunda rafa kaldırılan) daha makul bir ürün olduğunu kısa sürede fark ettiler. Hataları kabul etmeye ve rotayı değiştirmeye istekli olun.

Sörfçü ol. Okyanusu izle. Büyük dalgaların nerede kırıldığını bulun ve buna göre ayarlayın.

# 91: Motorlarınızı Çalıştırın

Tamamlandı!

Tamam, başardın! Umarız uygulamanızla Gerçekleşmeye başlamak için heyecanlısınızdır. Asgari kaynaklarla harika bir yazılım yapmak için gerçekten daha iyi bir zaman olmamıştı. Doğru fikir, tutku, zaman ve beceri ile sınır gökyüzüdür.

Birkaç kapanış düşüncesi:

## Uygulamak

Herkes kitap okuyabilir. Herkes bir fikir üretebilir. Herkesin web tasarımcısı olan bir kuzeni vardır. Herkes blog yazabilir. Herkes bazı kodları kırmak için birini kiralayabilir.

Sizinle diğer herkes arasındaki fark, ne kadar iyi yürüttüğünüz olacaktır. Başarı, büyük uygulama ile ilgilidir.

Yazılım için bu, birçok şeyi doğru yapmak anlamına gelir. Sadece iyi bir yazıya sahip olamazsınız, ancak daha sonra düzyazınızdaki vaatleri yerine getiremezsiniz. Kodunuz hack'lerle doluysa, temiz arayüz tasarımı onu kesmez. Kötü tanıtım, kimsenin bundan haberi olmadığı anlamına geliyorsa, harika bir uygulama değersizdir. Büyük puan almak için tüm bu unsurları birleştirmeniz gerekir.

Anahtar dengedir. Bir yöne çok fazla eğilerseniz, başarısızlığa doğru gidiyorsunuz. Zayıf bağlantılarınızı sürekli olarak arayın ve eşit olana kadar onlara odaklanın.

## İnsanlar

Başarılı bir web uygulaması oluşturmak söz konusu olduğunda en önemli bileşen olduğunu düşündüğümüz bir şeyi yeniden vurgulamaya değer: ilgili insanlar. Mantralar, merkez üssü tasarımı, daha az yazılım ve diğer tüm harika fikirler, bunları uygulayacak doğru insanlara sahip değilseniz, gerçekten önemli olmayacaktır.

İşlerini tutkuyla yapan insanlara ihtiyacınız var. Zanaatlarını önemseyen ve aslında onu bir zanaat olarak düşünen insanlar. İlgili parasal ödül ne olursa olsun, işleriyle gurur duyan insanlar. İnsanların %95'i aradaki farkı bilmesee bile ayrıntılara kafa yoran insanlar. Harika bir şey inşa etmek isteyen ve daha azıyla yetinmeyen insanlar. İnsanlara ihtiyacı olan insanlar. Tamam, bu sonuncusu değil ama karışıma biraz Streisand atmaya karşı koyamadık.

Her neyse, o insanları bulduğunuzda onlara tutunun. Sonunda, ekibinizdeki insanlar projenizi ve şirketinizi yapacak veya bozacak.

## Yazılımdan Daha Fazlası

Gerçekleşme kavramının sadece bir web uygulaması oluşturmak için geçerli olmadığını da belirtmekte fayda var. İlgili fikirleri kavramaya başladığınızda, onları her yerde göreceksiniz. Bazı örnekler:

- Yeşil Bereliler veya Donanma Mühürleri gibi özel hareket kuvvetleri, diğer birimlerin çok büyük veya çok yavaş olduğu görevleri yerine getirmek için küçük ekipler ve hızlı konuşlandırma kullanır.
- The White Stripes, basit bir formüle bağlı kalarak kısıtlamaları benimsiyor: iki kişi, akıcı şarkılar, çocuksu davul çalma, stüdyo süresini minimumda tutma vb.
- Apple'ın iPod'u, yerleşik bir fm radyo veya ses kaydedici gibi özellikler sunmayarak rakiplerinden farklılaşıyor.
- Futbolda acele hücumlar, toplanma ve oyun çağrılarının “bürokrasisini” ortadan kaldırarak büyük sahaları toplar.
- Rachael Ray, başarılı yemek kitaplarını ve TV şovunu 30 dakikalık “Gerçek Yemekler Al” konseptine dayandırıyor.
- Ernest Hemingway ve Raymond Carver basit ve anlaşılır bir dil kullanmalarına rağmen yine de maksimum etkiyi sağladılar.
- Shakespeare, sonelerin sınırlamalarından, on dört satırlık iambik beşli ölçülü lirik şiirlerden keyif aldı.
- Ve tekrar tekrar...

Elbette, Gerçekleşmek harika bir yazılım oluşturmakla ilgilidir. Ama orada durması için hiçbir sebep yok. Bu fikirleri alın ve hayatınızın farklı yönlerine uygulamayı deneyin. Bazı düzgün sonuçlara rastlayabilirsiniz.

## İletişimi koparmamak

Gerçek Olmanın sizin için nasıl çalıştığını bize bildirin. [getreal \[at\] basecamp \[dot\] com](mailto:getreal[at]basecamp[dot]com) adresine bir e-posta gönderin.

Ayrıca, kullanılabilirlik, tasarım ve bir sürü başka şeyle ilgili blogumuz olan [Signal vs. Noise](#)'ı ziyaret ederek Basecamp'ın en son tekliflerinden haberdar olun .

Okuduğunuz için teşekkürler ve iyi şanslar!