

Thought

NSAN '16

Gelece i ekillendiren
e ilimler ve teknoloji üzerine
görü lerimiz

thoughtworks.com/radar

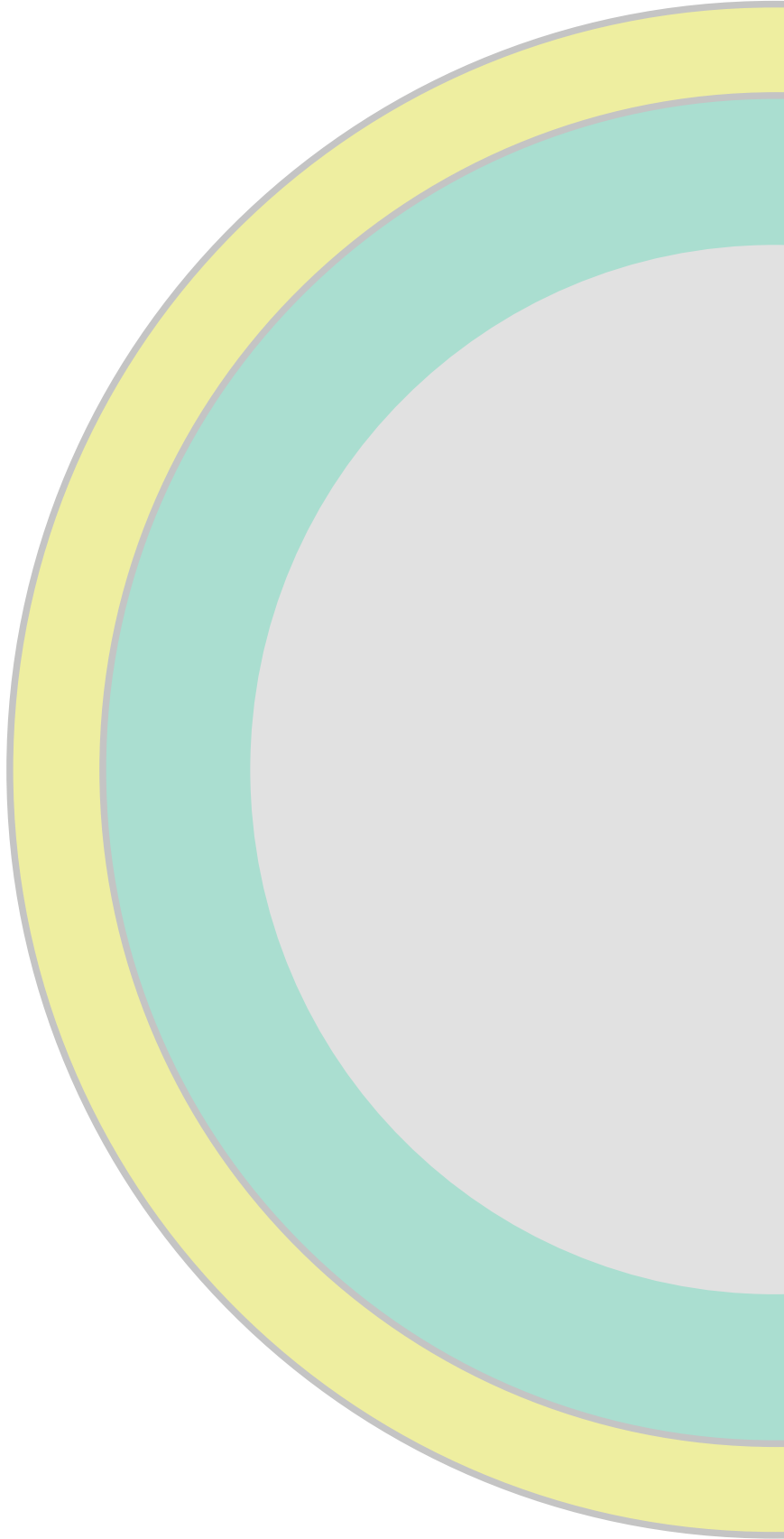
M9B @? @9F

Bu sayıda öne çıkan başlıklar şunlardır:

J 9F A @ '6 F 'M5B ' , F , B 'C@5F5? '5E ? '5MB5?

H9? B C @C > 'F 585F ' < 5?? B 85

ThoughtWorks alı anları iin teknoloji bir tutkudur. Teknoloji retiyoruz, ara tırmalar ve deneyler yapıyoruz, teknolojiyi aık kaynak olarak sunuyoruz, teknoloji hakkında yazılar yazıyoruz ve herkes iin teknolojiyi geli tirmek amacıyla srekli alı ıyoruz. Hedefimiz, yazılımda mkemmeliyeti savunmak ve BT alanında devrim yapmaktır. Bu misyon do rultusunda ThoughtWorks Teknoloji Radarı'nı ıkarıp payla ıyoruz. Radarı, ThoughtWorks'teki st dzey teknoloji liderlerinden olu an ThoughtWorks Teknoloji Danı ma Kurulu hazırlıyor. Kurul sık sık toplanarak, ThoughtWorks'un kresel teknoloji stratejisini ve sektrmz ciddi olarak etkileyen teknoloji trendlerini tartı ıyor. Radar, Teknoloji Danı ma Kurulunun bu tartı malarını CIO





nsanların bizimle nasıl etkileşime girdiklerini anlamak için Büyük Verinin önemini uzun zamandır anlamı olmamıza rağmen, Büyük Veri özentisiyle verilen bir trendi fark etmi bulunmaktayız: «aslında çok da büyük olmayan» Verileri yönetmek için karmaşık araçlar

NoSQL veri depolarının kullanımını artması ve süreklilik konusunda çok dilli yaklaşımların popülerliğinin yükselmesi sonucunda ekiplerimizin de önünde artık verilerini depolama alanında çok sayıda seçenek yer alıyor. Bu durum birçok avantajı beraberinde getirirken, kırılğan a larla ürün davranışı bazı durumlarda ürün geli tiricileri tarafından bile çok iyi anlaşılmayan gizli (ve bazen çok da gizli olmayan) sorunlar çıkarabiliyor. [Jepsen](#) araç takımı ve beraberinde sunulan [blögu](#), farklı veri tabanlarının ve kuyruk teknolojilerinin olumsuz koşullar altında nasıl tepki verdiklerini anlamak isteyenlere f ili bir referans haline dönü tü. Daha da önemlisi, i lemlerde mü teriyi de dahil eden test yaklaşımı, mikroservis geli tiren birçok ekip için olası hata modlarını mercek altına alıyor.

[LambdaCD](#), ekiplere Clojure'da Sürekli Da ıtım ardı ık düzenlerinin tanımlanması için bir yöntem sunuyor. 'Kod halinde(veya olarak) Altyapı'nin faydalarını 'Sürekli Dağıtım' sunucularına getirmis oluyor. Bu faydaları kaynak kontrol yönetimi, birim testi, refactoring ve kodun yeniden kullanımı olarak söyleyebiliriz. "Kod olarak ardı ık düzenler" alanında, LambdaCD, haf f, ba ımsız ve tamamen programlanabilir olmasıyla dikkat çekiyor ve ekiplerin aynı kodlarıyla yaptıkları gibi kendi ardı ık düzenleriyle çalışmalarına olanak tanıyor.

Anka Sunucu veya [Anka Ortam](#) teknikleri kullanan ekipler, Uygulama Performans Yönetimi (APM) araçlarından

Çı gibi gelen ön kullanıcı JavaScript framework'leri arasında [React.js](#) reaktif veri akışı etrafında geliştirilen tasarımıyla ön plana çıkıyor. Tek yönlü veri bağlamasına izin vermesi, oluşturma mantığını büyük oranda hafifletiyor ve diğer framework'ler ile yazılmış olan uygulamalarda sıklıkla ortaya çıkan sorunların önemli bir kısmını bertaraf ediyor. React.js'nin sağladığı faydaları sayısı giderek artan küçük ya da büyük birçok projede görebilirken, aynı zamanda [AngularJS](#) gibi diğer sevilen framework'lerin durumu ve geleceği hakkında endişemiz sürüyor. Bu da React.js'nin JavaScript

Ekiplerimiz, JQuery ya da ham XHR'den uzaklaarak, uzaktan JavaScript çağrılarına ve özellikle de Fetch polifilli tercih etmeye başladık. Anlamsallıklar benzer olsa da taahhütlere daha açık destek veriyor ve CORS desteğini de sağlıyor. Bunu yeni filiyaklaşım olarak

"2". :&'@. &#D" ile hızlı çoklu platform mobil geli tirm alanındaki başarıların devam etti ini gözlemliyoruz. Sürekli geli tirmelerden geçti i için ortaya çıkan bazı sorunlara rağmen, yerli ve yerli olmayan kod ve görüntüleme, hızlı geli tirme döngüsü (anlık yeniden yükleme, Chrome'da sorunları giderme, Flexbox tasarımı) arasındaki önemsiz entegrasyonun avantajları ve React tarzındaki genel büyüme görüşümüzü olumlu etkiliyor. Diğer birçok framework'te olduğu gibi, kodu iyi yapılandırmak için özenli kullanılması bu noktada oldukça yardımcı oluyor.

"2" \$BE ekiplerimizin önemli bir kısmının mü teri tarafı uygulamalarındaki durumu yönetmek için geli tirdikleri dü ünçe tarzlarını yeni bir çerçeveye oturtmasına yardımcı olan mükemmel ve olgun bir araç. Flux tarzında bir yaklaşım benimseyerek, mantı a oturtması kolay olan gev ek ba lantılı durum makinesi yapısına izin veriyor. Ember ve _____ gibi tercih etti imiz JavaScript

ThoughtWorks, yazılım danışmanlığı, üretimi ve ürünleri konusunda uzmanlaşmış tutkulu ve hedef sahibi bireylerin oluşturduğu bir topluluk ve yazılım şirkettir. Müşterilerimize yaşadıkları en büyük zorluk ve mücadelelerde yardımcı olmak üzere alımlarını dünyada dününün IT sektöründe bir devrim yapmanın ve olumlu bir sosyal değişim yaratmanın yolunu arıyoruz. Harika olmayı çok isteyen yazılım ekipleri için öncü araçlar yapıyoruz. Ürünlerimiz, kuruluşların

sürekli gelişmelerine ve en kritik ihtiyaçları için kaliteli yazılımlar üretmelerine yardımcı oluyor. 20 Yıl önce kurulan ThoughtWorks, Chicago'daki küçük bir şirketten 12 ülkedeki (Avustralya, Brezilya, Kanada, Çin, Ekvador, Almanya, Hindistan, Singapur, Güney Afrika, Türkiye, Birleşik Krallık ve ABD) 35 ofiste 3500'den fazla çalışanı olan bir şirkete dönüşmüştür.

Thought®