

Bölüm 2

Gereksinimlerin Ortaya Çıkarılması İçin Hazırlık

Ürün Misyon Beyanı

Yeni bir sistem geliştirmeyi veya eskisinin yeniden tasarlanmasını üstlenirken yapmamız gereken ilk şey ne olduğuna dair kısa bir tanım elde etmek veya geliştirmektir. Böyle bir ifadeye genellikle bir **ürün misyonu beyanı** denir. Sistem görev bildirimini olaraktan adlandırılabilir.

Bazı kuruluşlar, daha uzun olsa da, misyon beyanına benzer bir operasyon kavramı (**Conops**) tercih eder. Bazı ortamlarda Conops, çok yüksek düzeyde gereksinimler belirtimine benzeyen bir belgedir. Ürün misyon beyanı veya Conops, sisteme dahil olan herkes için bir odak noktası görevi görür ve "bu işlevsellik amacına nasıl hizmet ediyor?" Sorusunu sorarak çeşitli özelliklerin önemini tartmamıza izin verir. Çevik metodolojilerde, daha sonra tartışılacağı üzere, misyon ifadesinin veya Conops'un "sistem metaforu" rolünü oynadığını söyleyebiliriz.

Örneğin, Ek A'nın 1.1 Bölümü, Smith Smart Home, için bir misyon beyanı içermektedir. Ek B Bölüm 1, atık su pompalama kontrol sistemi için bir Conops beyanı içerir.

Misyon beyanları yazmak sıkıntılı bir iş olabilir ve birçok insan, küçük ayrıntılarda çıkmaza girme eğilimi olabileceğinden, bunu yapmaktan çekinir veya korkar. Bazen, ürün misyon beyanları çok uzun sürebilir ve aslında bir Conops belgesine dönüşebilir. Bir ürün misyonu ifadesi çok kısa, açıklayıcı, ilgi çekici ve asla ayrıntılı olmamalıdır.

En çok alıntı yapılan "iyi" sistem görev açıklamalarından biri, orijinal Star Trek televizyon dizisinden Starship Enterprise ile ilgili olanıdır. Kabaca ifade edilen bu misyon ifadesi: Garip yeni dünyalar keşfetmek, yeni yaşam biçimleri ve yeni medeniyetler aramak, daha önce hiç kimsenin gitmediği yerlere cesurca gitmek.

Bu ifade açık, ikna edici ve ilham vericidir. Ve bu "faydalı" - bu klasik dizinin hayranları, yıldız gemisi ekibi tarafından yapılacak belirli eylemlerin sistem görevi beyanına göre yapıldığı birkaç bölümü hatırlayacak.

Peki, bagaj taşıma sistemi için bir ürün veya sistem misyonu beyanı nasıl olabilir?

Yolcu kalkış noktasından varış noktasına kadar bagaj taşımanın tüm yönlerini otomatikleştirmeye ne dersiniz?

Evcil hayvan mağazası POS sistemi için, müşteri satın alma etkileşiminin ve envanter kontrolünün tüm yönlerini otomatikleştirmesi diyebiliriz.

Bunlar mutlaka zekice ya da hayranlık uyandırıcı değil ama sistemin özünü aktarıyorlar. Ve daha sonra, spesifikasyonuna dahil olanların beklentilerini kalibre etmemiz gerektiğinde faydalı olabilirler. Özellikle küresel olarak dağıtılmış kalkınmada, bir sistem metaforuna duyulan ihtiyaç büyük önem taşımaktadır.

Bir Müşteriyle Karşılaşma

Karınızın (ya da "koca", "arkadaş", "oda arkadaşı" veya herhangi birinin yerine) pasta yapmak istediği için mağazaya gidip aşağıdaki öğeleri almanızı istediğini varsayalım:

- 5 kilo un

- 12 büyük yumurta
- 5 kilo şeker
- 1 kilo tereyağı

En yakın markete (evinize yakın olan) gidin.

En yakın markete geldiniz. Markette, onun beyaz mı yoksa esmer şeker mi istediğinden emin olmadığınızı fark ediyorsunuz. Yani onu cep telefonunuzdan arayıp hangi şeker istediğini soruyorsunuz; esmer şekere ihtiyacı olduğunu öğrenirsin. Alışverişinizi yapıp eve dönüyorsunuz.

Ama karınız seçimlerinizden memnun değil. Yanlış türde un satın aldınız; beyaz istediğini ve senin buğday unu aldığını bildiriyor. Yanlış türde tereyağı aldınız; tuzsuz istedi. Yanlış türde şeker de getirmişsin, koyu kahverengi; açık kahverengi istedi. Şimdi başın belada.

Markete geri dönüyorsun ve un, şeker ve tereyağını geri veriyorsun. Beyaz unu ve esmer şekeri buluyorsunuz, ancak bir fiçı yalnızca tuzsuz tereyağı (Çubuk Şeklinde Değil) bulabilirsiniz, ancak bir fiçı tereyağın onun için kabul edilebilir olduğunu varsayıyorsunuz. Satın alma işleminizi gerçekleştiriyor ve ürünlerle birlikte iade ediyorsunuz. Ama şimdi yeni hatalar yaptığınızı keşfediyorsunuz. Açık kahverengi şeker alımı iyi, ancak geri getirdiğiniz beyaz un ağartılmış; ağartılmamış istedi. Ve kutudaki tereyağı kabul edilemez - şuna işaret ediyor: tuzsuz tereyağı çubuk şeklinde bulunabilir. Cahilliğinizden dolayı şimdi size çok kızgın.

Bu durumda, mağazaya geri dönersiniz ve eşyaları mahcup bir şekilde iade eder ve uygun ikamelerini alırsınız. Karınızın öfkesini yatıştırmak için, onun en sevdiği çikolatalı şekerlerden de almaya karar veriyorsunuz.

Eve dönüyorsun ve o hala mutsuz. Sonunda tereyağı, şeker ve unu hazırlarken, karınız akşam yemeği için omlet yaptığını ve omlet ve kek için bir düzine yumurtanın yeterli olmayacağını hatırlıyor - 18 yumurtaya ihtiyacı var. Ayrıca çikolatadan da memnun değil - diyetle olduğunu ve etrafta çikolatanın cazibesine ihtiyacı olmadığını söylüyor.

Bir kez daha marketi ziyaret edip çikolatayı ve bir düzine yumurtayı iade edin. 18 yumurta alıp eve dönüyorsun. “Bu şeyleri nereden aldın?” diye sorduğunda alışverişi doğru yaptığınızı düşünüyorsunuz. Eşyaları marketten aldığınızı fark ettiğinde, kıpkırmızı kesildi - orada fiyatların çok yüksek olduğunu düşünüyor - yolun birkaç mil ilerisindeki süpermarkete gitmeliydiniz.

Bu örnekle devam edebiliriz - karınız her satın alımlarınızda yeni bir kusur keşfettiğinde, miktar veya marka hakkında fikrini değiştirdiğinde, yeni ürünler eklediğinde, diğerlerini çıkardığında, vb.

Ancak bu durumun gereksinim mühendisliği ve paydaşlarla ne ilgisi var? Durum, gereksinim mühendisliği ile ilgili birçok noktayı göstermektedir. İlk olarak, uygulama alanını anlamamız gerekir. Bu durumda, fırıncılık bilgisine sahip olmak, tereyağı, un ve şekerin farklı türleri olduğu konusunda sizi önceden bilgilendirirdi ve muhtemelen alışveriş yolculuğunuza başlamadan önce odak soruları sormuş olurdunuz. Bu senaryodan bir başka nokta da—müşteriler her zaman ne istediklerini bilmiyor.

Karınız, siz mağazaya üç kez gidene kadar daha fazla yumurtaya ihtiyacı olduğunun farkında değildi. Ve hikayede bir ders daha var: Müşterilerin ne istediği konusunda asla varsayımlarda bulunmayın – fiçı yağının kabul edilebilir olduğunu düşündünüz; değildi. Sonunda, müşterilere istediklerinden fazlasını (bu durumda en sevdiği çikolatayı) sağlamanın bile bazen yanlış bir şey olabileceğini öğrendiniz.

Ancak daha geniş anlamda, bir müşteriyle bu karşılaşmadan çıkarılacak en önemli ders, onların sorun çıkarabileceğidir. Ne istediklerini her zaman bilemezler ve bildiklerinde bile isteklerini etkisiz bir şekilde iletebilirler. Müşteriler fikirlerini değiştirebilir ve bildikleriniz ve ne sağlayacağınız konusunda yüksek beklentileri olabilir.

Paydaş etkileşimi çok önemli olduğu için, bu bölümün geri kalanını paydaşların ve özellikle de sistemin kendileri için inşa edildiği paydaşların, yani müşterilerin doğasını belirlemeye ve anlamaya çalışacağız.

Sistem Sınırlarının Belirlenmesi

Paydaşları belirlemede gerekli ilk adım, üst düzey bir sistem modeli oluşturmaktır. Bu model, sistemle ilgili kişi ve kuruluşlar kümesini tanımlayacak ve sistem sınırını (diğer varlıklarla doğrudan ve dolaylı etkileşimleri) tanımlayacaktır (Laplante ve diğerleri, 2016). Sistem sınırlarını doğru bir şekilde tanımlamadan, önemli paydaşların gözden kaçırılması mümkündür - potansiyel olarak feci bir durum.

Bağlam Diyagramları

Bağlam diyagramı, sistem sınırlarını ve diğer sistemlerin, insanlar ve çevre ile etkileşimleri gösteren görsel bir temsil diyagramıdır.

Bağlam diyagramı iki nedenden dolayı önemlidir. İlk olarak, proje yaşam döngüsü boyunca mühendisler ve paydaşlar için sistem hakkında diyalog temeli sağlar. İkincisi, bağlam diyagramı, kapsam kaymasıyla, yani işlevsel gereksinimlerin sistemin amacının ötesinde kontrolsüz büyümesiyle mücadeleye yardımcı olur.

Bağlam diyagramı, belirsiz, karmaşık ve eksik gereksinimler tartışılırken gereksinimlerin anlaşılmasına ve analizine de yardımcı olabilir. Bağlam diyagramı birçok biçim alabilir. Bir blok diyagram yeterli olabilir,

Örneğin, Şekil B.1, ıslak kuyu pompalama sisteminin bağlamını gösteren çok basit bir blok diyagramdır. Kullanım senaryosu diyagramı gibi UML/SysML diyagram ailesinden biri de kullanılabilir. UML ve SysML, Bölüm 4 ve Ek C (UML) ve E'de (UML modellerinden biri olan kullanım durumları) tartışılmaktadır. Kısaca açıklanacak olan soft sistem metodolojisinden (SSM) zengin resimler, bağlam diyagramları için de kullanılabilir.