

Basit maliyetle ilgili zorluklar ve risk analizi

Gereksinim maliyetini, riskini, zamanlamasını ve benzerlerini dengelemek için tipik bir yaklaşım, bu faktörleri Tablo 10.1'de gösterildiği gibi bir matrise koymak ve maliyeti, riski veya zamanlamayı en aza indirmek için deneyimsel bir teknik kullanmaktır.

Ancak bu basit gösterimle ilgili birkaç sorun var. Örneğin, gereksinimlerin gerçek maliyetlerini hesaplamak, belirsizlik konisinin önerdiği gibi, erken dönemlerde oldukça zor olabilir. Ayrıca, matris gereksinimler arasındaki bağımlılıkları dikkate almaz. Bunlar, bağımlı gereksinimleri listelemek için matrislere ek bir sütun eklenerek açıklanabilir, ancak daha sonra sorunun formülleştirilmesi farklıdır. Bir diğer sorun da riskin nicelleştirilmesinde ve tanımlanmasında ortaya çıkmaktadır. Farklı müşteriler ve paydaşlar riski farklı şekilde değerlendirecektir. Risk matrisi yaklaşımı, projeler arasındaki derecelendirme risk faktörlerinin tutarlılığını da hesaba katmamaktadır. Tüm bu sorunlarla başa çıkmak için fikir birliği oluşturma tekniklerini kullanırsanız da, bu zaman alıcı ve pahalı bir yaklaşımdır.

Tablo 10.1 Satın Alınabilirlik Optimizasyonuna Risk Matris Yaklaşımı

| <i>Requirement #</i> | <i>Cost Estimate</i> | <i>Performance Metric(s)</i> | <i>Rank</i> | <i>Schedule (Staff Months)</i> | <i>Risk Factor 1 = low 5 = high</i> |
|----------------------|----------------------|------------------------------|-------------|--------------------------------|---|
| ... | | | | | |
| 3.1.2 | 2.5K | Response time | High | .1 | 5 |
| 3.1.3 | 5K | Accuracy | Medium | .3 | 3 |
| 3.1.4 | 7.5K | Accuracy | Low | .4 | 1 |
| ... | | | | | |

Yazılımın veya sistemin yaşam döngüsünün başlarında gereksinimlerin maliyetini tahmin etmeye çalışırken ek sorunlar vardır.

Farklı müşterilerin ve paydaşların gereksinim riski farklı şekilde derecelendireceği zaten belirtilmiştir. Ancak işlevselliği farklı şekilde ayrıştırırlar. Örneğin, bir müşteri belirli bir stil kullanıcı ekranının gereksinimini küçük bir görev olarak görebilirken, bir diğeri bu özelliğin uygulanmasını büyük bir girişim olarak görebilir.

Müşteriler ve diğer paydaşlar bir projeye farklı bağlılık seviyelerine sahiptir ve bu nedenle girdilerine az ya da çok düşünce yatırır. Herhangi bir gereksinim kararıyla yaşaması gereken bir çalışan, gereksinimler analizi sırasında alınan kararlarla başa çıkması gerekmeyen bir dış danışmandan kesinlikle farklı davranacaktır.

Gereksinimlerin ortaya çıkarma sürecine katılanlar, nihai proje sonucu açısından farklı kişisel katılım veya "oyundaki cilt" olacaktır ve bu gerçek, gereksinim ve anlaşmaya katılımlarının kalitesini etkileyecektir.

Katılımcıların farklı kişisel gündemlerinin etkilerinin de farkında olun. Örneğin, yakında şirketten ayrılacak olan bir müşteri temsilcisi, bir özellik açıklamasının doğruluğunu, projenin potansiyel yöneticisinden daha az önemseyecektir.

Son olarak, maliyete dayalı gereksinim mühendisliği, bireyleri bazı 'makul inkar edilebilirlik', yani proje planlandığı gibi gitmezse bir mazeret verme yeteneği oluşturmaya teşvik edebilir. Bu bireyleri genellikle tanımlayabilirsiniz, çünkü adlarını belirli belgelere eklemek istemezler veya gereksinim müzakereleri sırasında riskten korunma kelimeleri kullanırlar

Bölümün geri kalanında, gereksinim mühendisleri için değer mühendisliği faaliyetine yardımcı olacak bazı basit yaklaşımlara bakacağız. Bu bölümler, proje yönetimi, maliyet muhasebesi ve sistem mühendisliğini birleştiren bu çok karmaşık faaliyetlere yalnızca bir giriş sağlar.

Cocomo ve Türevlerini Kullanarak Tahmin Etmek

En yaygın kullanılan yazılım maliyet ve çaba tahmin araçlarından biri, ilk olarak 1981'de tanıtılan Boehm'in COCOMO modelidir. COCOMO, yapısal maliyet modelinin kısaltmasıdır; bu, tahminlerin projeyi karakterize eden bir dizi parametreden belirlendiği anlamına gelir. Orijinal (temel), orta ve gelişmiş modeller de dahil olmak üzere COCOMO'nun çeşitli sürümleri vardır, her biri tahminleri ayarlamak için artan sayıda değişkene sahiptir. En yeni COCOMO modelleri, daha etkileyici modern dillerin yanı sıra temelde aynı çabayla daha fazla kod üretme eğiliminde olan yazılım oluşturma araçlarını daha iyi barındırır. Web tabanlı uygulamalar ve yazılım yoğun (ancak saf yazılım değil) sistemler için geçerli olan COCOMO türevleri de vardır.

COCOMO

Cocomo modelleri, bir dizi faktörle değiştirilen bir kod satırının tahminine dayanmaktadır. Proje çabası ve süre için denklemler

$$Effort = A \prod_{i=1}^n cdi (size)^{P_1} \quad (10.1)$$

$$Duration = B(Effort)^{P_2} \quad (10.2)$$

A ve B, oluşturulacak yazılım sistemi türünün bir fonksiyonu olduğunda, boyut kod satırlarının sayısını temsil eder, cd_i sürücülerini temsil eder (bu sayı değişebilir) ve P_1 ve P_2 , uygulamanın belirli özelliklerine alan adı bağlıdır.

Belirtildiği gibi, cd_i , aşağıdakiler de dahil olmak üzere bir dizi faktöre dayanan maliyet faktörleridir:

- Ürün güvenilirliği ve karmaşıklığı
- Platform zorluğu
- Personel yetenekleri

- Personel deneyimi
- Tesisler
- Zamanlama kısıtlamaları
- Planlı yeniden kullanım derecesi
- Süreç verimliliği ve olgunluğu
- Emsallik (yani, projenin yeniliği)
- Geliştirme esnekliği
- Mimari uyum
- Takım uyumu

Bunlar, çok düşükten çok yükseğe kadar değışen bir Likert ölçeğinde nitel derecelendirmelere sahiptir, yani yanıtların her birine sayısal değler atanır.

Son olarak, P_1 ve P_2 , uygulama alanının belirli özelliklerini, örneğin ölçek ekonomilerinin karakterizasyonu, yeniden çalışmayı önleme yeteneği, uygulamanın uygulamanın yeniliği ve takım uyumunu temsil eder. Bu arada, çaba kişi aylarında toplam proje çabasını temsil eder ve süre takvim aylarını temsil eder. Bu rakamlar, COCOMO tahmini proje için gerçek bir maliyete dönüştürmek için gereklidir.

Gelişmiş Cocomo modellerinde, sistemde kullanılacak kod oranı için, yani değıştirilmiş, kod değıştirilmiş ve entegrasyon değıştirilmiş kod oranı için başka bir adaptasyon ayar faktörü yapılır. Adaptasyon faktörü A, Denklem 10.3 ile verilir.

$$A = 0.4 (\% \text{design modified}) + 0.3 (\% \text{code modified}) + 0.3 (\% \text{integration modified}) \quad (10.3)$$

COCOMO yaygın olarak bir yazılım projesi yönetim aracı olarak tanınmakta ve saygı duyulmaktadır. Altta yatan model gerçekten anlaşılmamış olsa bile yararlıdır. COCOMO yazılımı ticari olarak temin edilebilir ve hatta ücretsiz kullanım için Web'de bulunabilir.

Bununla birlikte, gereksinim mühendisi için önemli bir husus, COMOO'nun tahminini, gereksinim mühendisliği sırasında kolayca tahmin edilemeyen kod satırlarına dayandırmasıdır. İşlev noktaları gibi diğer teknikler, gereksinimleri geliştirirken kullanılabilen özellik kümelerine dayalı kod satırı tahminleri sağlamak için gereklidir. Fonksiyon noktalarına birazdan bakacağız.

WEBMO

WEBMO, COMMO'nun her zaman iyi olmadığı Web tabanlı projelerin projektasyonu için özel olarak tasarlanmış bir COCOMO türevidir. WEBMO, COCOMO ile aynı çaba ve süre denklemlerini kullanır, ancak farklı bir öngörücü kümesine dayanır, yani:

- İşlev noktaları sayısı
- XML, HTML ve Sorgu Dili Bağlantıları Sayısı
- Multimedya dosyalarının sayısı
- Komut dosyaları sayısı
- Web sitesi yapı taşları

Bunlar, çok düşük ila çok yüksek arasında değişen bir Likert ölçeğinde nitel derecelendirmelere ve Tablo 10.2'de gösterilen sayısal eşdeğerlere sahiptir (Reifer 2002). Cocomo modelindeki maliyet sürücülerini için benzer tablolar mevcuttur, ancak simülasyon araçlarına gömülmüştür, bu nedenle gereksinimler mühendisi sadece Likert ölçeğinden derecelendirmeleri seçmek zorundadır. Her halükarda, bir WebMO hesaplamasının net sonucu, projeyi sırasıyla kişi aylarında ve takvim aylarında tamamlamak için bir çaba ve süredir.

Tablo 10.2 WEBMO Maliyet Faktörleri ve Değerleri (Reifer 2002)

| | <i>Ratings</i> | | | | |
|--------------------------|-----------------|------------|----------------|-------------|------------------|
| | <i>Very Low</i> | <i>Low</i> | <i>Nominal</i> | <i>High</i> | <i>Very High</i> |
| Cost driver | | | | | |
| Product reliability | 0.63 | 0.85 | 1.00 | 1.30 | 1.67 |
| Platform difficulty | 0.75 | 0.87 | 1.00 | 1.21 | 1.41 |
| Personnel capabilities | 1.55 | 1.35 | 1.00 | 0.75 | 0.58 |
| Personnel experience | 1.35 | 1.19 | 1.00 | 0.87 | 0.71 |
| Facilities | 1.35 | 1.13 | 1.00 | 0.85 | 0.68 |
| Schedule constraints | 1.35 | 1.15 | 1.00 | 1.05 | 1.10 |
| Degree of planned | | | | | |
| Reuse | — | — | 1.00 | 1.25 | 1.48 |
| Teamwork | 1.45 | 1.31 | 1.00 | 0.75 | 0.62 |
| Process efficiency | 1.35 | 1.20 | 1.00 | 0.85 | 0.65 |