

Yazılım Gereksinimlerinin Neden-Sonuç Grafikleri (NSG) ile Analizi ve Test Durumlarının Belirlenmesi

Reşat Erhan Yüceer¹

Nevra Yaldız²

^{1,2} Savunma Teknolojileri Mühendislik ve Tic. A.Ş., Ankara

¹e-posta: eyuceer@stm.com.tr

²e-posta: nyaldiz@stm.com.tr

Özetçe

Bu makalenin amacı bir kara-kutu test tekniği olarak Neden-Sonuç Grafikleri tekniğinin anlatılmasıdır. Ayrıca tekniğin kullanılmasının getireceği yararlar ve zorluklar da incelenmektedir. Tekniğin nasıl kullanılacağı örnekler eşliğinde verilmeye çalışılmıştır.

1. Giriş

Bir Neden-Sonuç Grafiği “*Neden*”ler ve “*Sonuç*”lar arasındaki mantık ilişkilerini ve kısıtları ortaya koyan grafiksel bir gösterim yöntemidir [1].

Neden-Sonuç Grafikleri (NSG) kavramı önceleri temel olarak sistemleri modellemekte kullanılmakta iken 1973 yılında Bill Elmendorf tarafından yazılıma uyarlanmıştır [2].

Bu haliyle Neden-Sonuç Grafikleri tekniği sistemlerin davranışlarına uygulanabilecek bir kara-kutu tekniğidir. Sistemi modellemenin yanı sıra, oluşturulan model kullanılarak test durumlarını da tasarlanmasını olanaklı kılar.

Özellikle karmaşık davranışlar içeren yazılımlarda test için kullanılacak doğru ve yeterli girdi kombinasyonunu seçmek çok zor bir iş olabilir. Eğer sistematik bir yaklaşım kullanılmazsa, doğru girdi kombinasyonunu seçmek genellikle pek mümkün olmaz [1]. Neden-Sonuç Grafikleri kullanılarak sistematik olarak doğru ve yeterli test durumları ve girdileri oluşturulabilir.

Bölüm 2’de Neden-Sonuç Grafikleri anlatılmakta ve Neden-Sonuç Grafiklerinde kullanılan tanımlardan ve grafik öğelerinin gösteriminden bahsedilmektedir.

Bölüm 3’te Neden-Sonuç Grafikleri kullanılarak test durumlarının oluşturulması süreci örnek eşliğinde sunulmaktadır.

Bölüm 4’te ise bu çalışmadan elde edilen sonuçlar verilmeye çalışılmıştır.

2. Neden-Sonuç Grafikleri

Neden-Sonuç Grafiği; bir yazılım belirtiminde “*Neden*”ler ve “*Sonuç*”lar arasındaki mantıksal ilişkileri ortaya koyan grafiksel bir gösterim yöntemidir [3]. Bu yöntem ile hem gereksinimlerin düzgün ve anlaşılır bir şekilde ifade edilmesi sağlanır, hem de test adımları daha kolay türetilir.

Neden-Sonuç Grafikleri yöntemi bir kara-kutu test (black-box) metodudur; sistemin sadece istenen çıktısı/sonucu göz önüne alınır. Neden-Sonuç Grafikleri aynı zamanda birden fazla “*Neden*”in kombinasyonunu dikkate alan tek kara-kutu test metodudur.

Neden-Sonuç Grafikleri gereksinimdeki nedenlerde ayrı ayrı ortaya çıkabilecek aksaklıkların belirlenebilmesini sağlar.

Neden-Sonuç Grafikleri temelde doğal dil ile yazılan ifadelerin bazı boolean ifadeler yardımı ile formal bir gösterime dönüştürülmesidir.

Neden-Sonuç Grafikleri;

- Gereksinimin nedeni/nedenlerinin bütünlüğünü sağlamak için,
- Sistemin tamamını etkileyen nedenlerin tanımlanmasına yardımcı olmak için,
- Gereksinimden test adımları üretildiğinde herhangi bir boşluk olmaması için

kullanılır.

2.1. Neden-Sonuç Grafikleri Öğeleri

Bir Neden-Sonuç Grafikleri içerisinde yer alan öğelerin tanımları aşağıdaki gibi verilebilir:

- “*Neden*”; bir gereksinimdeki girdi koşulu veya koşullarıdır. Neden-Sonuç Grafiklerinde nedenler her zaman pozitif ve atomik olmak zorundadırlar. Literatürde; nedenlerin hepsi bir booleana karşılık gelmektedir; gerçekleşmesi durumunda “1” yada “doğru”, gerçekleşmemesi durumunda “0” yada “yanlış” ifadeleri kullanılmaktadır. Bu dokümanda “*Neden*”in gerçekleşmesi durumu için “1”, gerçekleşmemesi durumu için “0” kullanılacaktır.
- “*Sonuç*”; bir gereksinimdeki “*Neden*” yada “*Neden*”lerin kombinasyonu ile ortaya çıkan durum yada mesaj çıktısıdır.

“*Neden*” => Ara düğüm(ler) => “*Sonuç*”

- Bir gereksinimden oluşturulan her “*Neden*” bir “*mantık değişkeni*” ile ilişkilendirilir. İstenilen sonuç ise mantık değişkenlerinden oluşan “*mantık işlemleri*” ile gösterilir. (VE, VEYA)
- Bir gereksinimde; bazı durumlarda “*Neden*”lerin kombinasyonu mümkün değildir. Bu durumda “*kısıt*” notasyonları kullanılmaktadır.

“*E Kısıt*” (Exclusive); “*Neden*”lerin hepsinin aynı anda gerçekleşemeyeceğini ifade eder.

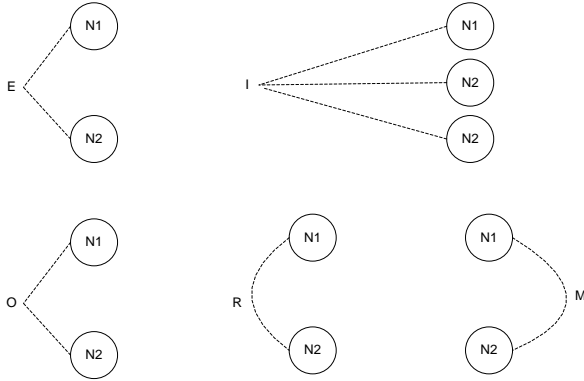
“*I Kısıt*” (Inclusive); her zaman için “*Neden*”lerden en az birinin gerçekleşeceğini ifade eder.

“*O Kısıt*” (One and Only One); “*Neden*”lerden sadece ve sadece birinin gerçekleşeceğini ifade eder.

“*R Kısıt*” (Requires); eğer “*Neden*”lerden biri gerçekleşiyorsa, diğerinin de mutlaka gerçekleşeceğini ifade eder.

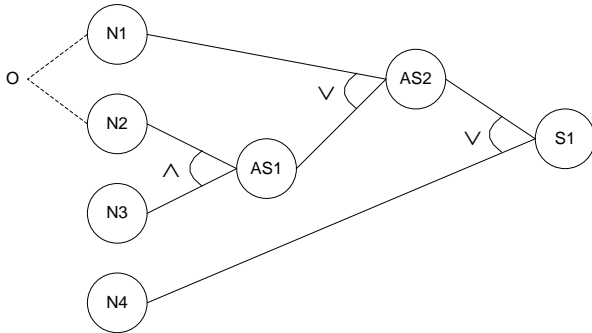
“**M Kısıtı**” (Mask); eğer “**Sonuç**”lardan biri gerçekleşecek ise; diğer sonucun gerçekleşmeyeceğini ifade eder. “**M**” kısıtında dikkat edilmesi gereken konu, diğer kısıtların aksine nedenlerle değil, sonuç ile ilgilidir.

Kısıt sembolleri Şekil-1’de verilmiştir.



Şekil 1: Kısıt Sembolleri

- Neden-Sonuç Grafiklerinde her “**Neden**”-“**Sonuç**”, ilişkilendirildiği bir “**düğüm**”e sahiptir. “**Neden**”lerin ilişkilendirildiği düğümlere “**neden düğümleri**”, “**Sonuç**”ların ilişkilendirildiği düğümlere ise “**sonuç düğümleri**” adı verilir. Birden fazla “**Neden**”in “**Sonuç**”a ulaşmadan ilişkilendirildiği düğümler ise “**ara düğüm**” olarak adlandırılır. Şekil-2’de verilen şekilde N1, N2, N3 ve N4 neden düğümü, S1 sonuç düğümü, AS1 ve AS2 ise ara düğüm olarak adlandırılmaktadır.



Şekil 2: Örnek bir Neden-Sonuç Grafiği

2.2. Neden-Sonuç Grafikleri'nin Faydaları

Neden-Sonuç Grafikleri bir gereksinimdeki nedenlerin ve aralarındaki ilişkilerin görsel olarak ortaya koyarak anlaşılabilirliği ve fark edilebilirliğini artırmaktadır. Böylelikle başlangıç aşamasında ortaya çıkabilecek sorunlar, eksiklikler belirlenmekte, belirsizlikler ortaya konmakta ve test adımları kolaylıkla yazılabilmektedir.

Neden-Sonuç Grafiklerinin Sınırlı Girdi Seçim Tablosuna çevrilmesi ile “**Neden**”-“**Sonuç**” lar arasındaki mantıksal ilişki kolaylıkla kurulabilir, bu tablodaki her kolonun bir test adımı olması dolayısıyla test adımları ortaya çıkar. Her test adımı ayrı bir “**Neden**” kombinasyonu ile ilişkilendirilmekte ve gereksiz kombinasyonlar elenmektedir. Bu sayede, oluşturulması gereken test adımlarının sayısı azalmaktadır.

Ayrıca test aşamasında ilgili kişi test adımlarının değişikliklerini kolaylıkla yapabilmekte ve karmaşıklığı kontrol edebilmektedir.

3. Neden-Sonuç Grafikleri ile Test Durumlarının Tanımlanması

Neden-Sonuç Grafikleri ile test durumlarının tanımlanması süreci aşağıdaki aşamalar ile gerçekleştirilir:

1. Gereksinimlerin gruplanması
2. “**Neden**”lerin, “**Sonuç**” ların ve “**Kısıt**”ların belirlenmesi
3. Neden-Sonuç Grafiklerinin oluşturulması
4. Test durumlarının tanımlanması

3.1. Gereksinimlerin Gruplanması

Bu aşamada doğal dil kullanılarak yazılmış olan fonksiyonel gereksinimler kendi içlerinde mantıksal gruplara ayrılarak gruplanır. Daha sonra da bu gruplar kendi içlerinde daha küçük gruplara bölünerek yönetilmesi ve analiz edilmesi daha kolay gruplar elde edilir.

Sistematik olarak yazılım geliştirme metodolojilerinin kullanıldığı ortamlarda yazılım gereksinimleri tanımlanırken gereksinimler mantıksal gruplara bölünmüş olabilirler. Bu durumda bu mantıksal gruplar direk olarak kullanılabilir.

3.2. “Neden”lerin, “Sonuç”ların ve Kısıtların Belirlenmesi

3.2.1. “Neden”lerin Belirlenmesi

Gereksinimlerde bulunan ve uygulamanın dışından gelecek uygulamada bir iş yapılmasına yol açan girdiler belirlenir. Daha sonra bu girdiler kendi koşullarında incelenirler ve Bölüm 2’de verilen “**Neden**” tanımına uygun olarak ayrı girdiler ve/veya denklik sınıfına göre “**Neden**”ler tanımlanır. Belirlenen her bir “**Neden**” için ayrı bir tanımlama yapılmalıdır. Bu tanımlamalar sonucunda bir girdiden bir veya daha fazla “**Neden**” oluşturmak olasıdır.

Aşağıda bu durumları daha iyi açıklamak için iki örnek verilmektedir:

Örnek 1:

Analiz sırasında belirlenen girdilerden bir tanesi “Pilot_Gösterge_Modu” dur ve alabileceği değerler (NAV, PFD, EICAS) olarak belirlenmiştir. Bu bilgiler ışığında birbirinden bağımsız üç adet “**Neden**” tanımlanır:

1. Pilot_Gösterge_Modu = NAV
2. Pilot_Gösterge_Modu = PFD
3. Pilot_Gösterge_Modu = EICAS

Örnek 2:

Bu örnekte ise gereksinimde var olan numerik bir karşılaştırmanın “**Neden**”lerinin belirlenmesi gösterilmiştir.

“Hava Hızı < 300 knot” şeklinde geçen bir ifade için birbirinden bağımsız üç adet “**Neden**” tanımlanır:

1. Hava Hızı < 300 knot
2. Hava Hızı ≥ 300 knot

3.2.2. “Sonuç”ların Belirlenmesi

Gereksinimlerde bulunan ve uygulamanın dışından gelecek uygulamada bir iş yapılmasına yol açan girdilerin oluşturduğu

çıktılar belirlenirç Belirlenen her bir “Sonuç” için ayrı bir tanımlama yapılmalıdır.

3.2.3. “Kısıt”ların Belirlenmesi

Belirlenen “Neden”ler ve “Sonuç”lar kendi içlerinde incelenerek varsa Bölüm 2’de tanımlanan “Kısıt”lar belirlenir.

3.3. Neden-Sonuç Grafikleri’nin oluşturulması

Tüm “Neden”ler, “Sonuç”lar ve “Kısıt”lar belirlendikten sonra, eldeki gereksinimler de kullanılarak “Neden”lerin birbirine bağlanmasıyla sonuçların oluşturulduğu bir grafik oluşturulur. Bu aşamada birbiriyle ilişkili “Neden” ve “Sonuç”ların mümkünse bir grafik içerisinde belirtilmesi oluşturulacak testlerin daha net belirlenmesi açısından büyük önem taşımaktadır. Bu aşamada ayrıca ara düğümler de belirlenmeli ve grafik oluşturulurken aynı seviyede oluşturulan ara düğümlerin grafik üzerinde aynı seviyeye gelmesine dikkat edilmelidir. Aşağıda yer alan Örnek 3 ile Neden-Sonuç Grafiklerinin oluşturulması aşamaları daha detaylı olarak verilmektedir:

Örnek 3:

Aşağıdaki gereksinim için bir Neden-Sonuç Grafiği oluşturulması durumunda Şekil 3’teki Neden-Sonuç Grafikleri ortaya çıkmaktadır:

Gereksinim-0100:

Aşağıdaki koşullardan herhangi biri sağlandığında gösterge otomatik olarak “ARAZİ GÖSTERİMİ” moduna geçer;

Koşul 1:

- Arazi gösterme seçimi aktif olduğunda, veya
- Arazi gösterme seçimi pasif ve yükseklik 1000 metreden az olduğunda

Koşul 2:

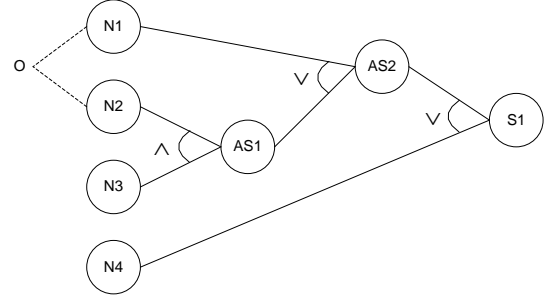
- Arazi tehlike uyarısı varsa

Yukarıdaki gereksinim incelendiğinde Tablo-1’deki “Neden”ler, “Sonuç” ve “Kısıt”lar belirlenmiştir:

Tablo 1: Neden-Sonuç-Kısıt Tablosu

Tanımlama	“Neden”
N1	Arazi Gösterim = Aktif
N2	Arazi_Gösterimi = Pasif
N3	Yükseklik < 1000 metre
N4	Arazi_Tehlike_Uyarısı = VAR
	“Sonuç”
S1	Gösterge = ARAZİ GÖSTERİMİ
	“Kısıt”
K1	N1 ve N2 arasında “O”

Tanımlanan gereksinim ve bu gereksinimin incelenmesi sonucunda belirlenen Tablo-1 kullanılarak Şekil-3’te verilen Neden-Sonuç Grafiği çizilmiştir.



Şekil 3: Gereksinim-0100 için Neden-Sonuç Grafiği

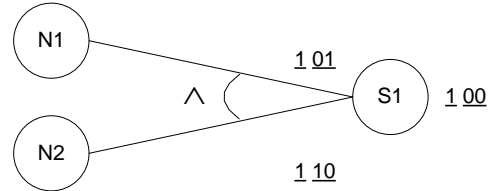
3.4. Test Durumlarının Tanımlanması

Test durumları tanımlarken girdilerin seçimi sırasında aşağıdaki iki temel kural gözönünde tutulmalıdır.

- “Neden”lere atanacak değerler “Sonuç”ların alabileceği tüm değerleri sağlayacak şekilde seçilmelidir. Yani “Neden”lere herhangi bir ilgili “Sonuç” hem “1” hem de “0” değerini en az bir kez alabilecek şekilde veri kombinasyonları sağlanmalıdır
- “Neden”lere değer setleri verilirken herhangi bir değer “Sonuç” değerini bağımsız olarak etkilemesi gerekmektedir.

Test durumlarının tanımlanmasına geçmeden önce Neden-Sonuç Grafiklerinde kullanılan iki temel “Neden” – “Sonuç” durumuna ve bu durumlarda testlerin nasıl oluşturulabileceğine bir göz atmakta fayda olacaktır:

- “VE” yapısı
“VE” yapısındaki temel bir Neden-Sonuç Grafiği için en az 3 test durumu gerçekleştirmek gerekir. Şekil 4’de bu temel yapı da gerçekleştirilmesi gereken test durumları gösterilmektedir.



$$N1 \text{ VE } N2 \Rightarrow S1$$

Şekil 4: “VE” yapısı için test durumları

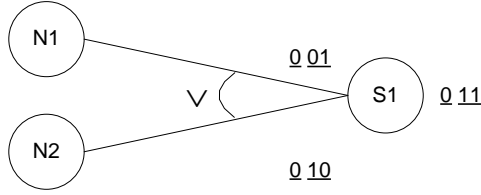
Bu durum bir tablo olarak Tablo-2’de de verilmektedir.

Tablo 2: “VE” Yapısı Test Durumları

Test Durumu	N1	N2	S1
TD1	1	1	1
TD2	1	0	0
TD3	0	1	0

Bölüm 3’ün girişindeki kurallar gözönüne alındığında ikinci kural gereği N1=0 ve N2=0 değerleri ile bir test durumu oluşturulmamaktadır.

- “VEYA” yapısı
“VEYA” yapısındaki temel bir Neden-Sonuç Grafiği için en az 3 test durumu gerçekleştirmek gerekir. Şekil 5’te bu temel yapı da gerçekleştirilmesi gereken test durumları gösterilmektedir.



N1 VEYA N2 => S1

Şekil 5: “VEYA” yapısı için test durumları

Bu durum bir tablo olarak Tablo-3’te de verilmektedir.

Tablo 3: “VE” Yapısı Test Durumları

Test Durumu	N1	N2	S1
TD1	0	0	0
TD2	1	0	1
TD3	0	1	1

Bölüm 3’ün girişindeki kurallar gözönüne alındığında ikinci kural gereği N1=1 ve N2=1 değerleri ile bir test durumu oluşturulmamaktadır.

Test durumlarının oluşturulmasında kullanılan yöntem en sonda yer alan “Sonuç”tan yola çıkarak geriye doğru analiz etmektir. İzlenecek süreç şu şekilde tanımlanabilir:

1. En son “Sonuç” belirlenir.
 - a. İlk olarak en sonda yer alan “Sonuç”a direk olarak bağlı olan “Neden”(ler) ve/veya “AraSonuç”(lar) ele alınarak temel bir Neden-Sonuç Grafiği oluşturulur.
 - b. “Sonuç” = 1 değerini alacak şekilde analiz edilir. Bu analiz sonucunda Şekil 2’ye benzer bir Neden-Sonuç Grafiği veya Tablo2’ye benzer bir tablo veya her ikisi birden oluşturulabilir.

Bundan sonraki basamak hiçbir ara sonuç kalmayana kadar tekrarlanır.

2. Sıradaki “AraSonuç” ele alınır ve kendisine direk olarak bağlı olan “Neden”(ler) ve/veya “AraSonuç”(lar) kullanılarak temel bir Neden-Sonuç Grafiği oluşturulur ve analiz edilir. Burada dikkat edilmesi gereken tek önemli konu “AraSonuç”un çıktıları bir önceki aşamadan bellidir ve kendisine bağlanan “Neden” veya “AraSonuç” kümesinin alabileceği değerler sadece belli olan değerlere göre belirlenmelidir. Bu analiz sonucunda ilk olarak oluşturulan Neden-Sonuç Grafiği geliştirilir ve eğer tablo oluşturma yolu seçildiyse bu durumda ilk oluşturulan tabloya yeni satırlar eklenerek tablo güncelleştirilir.
3. 1.b ve 2 no’lu aşamalar “Sonuç” = 0 değerini alacak şekilde tekrarlanır.
4. 1, 2 ve 3 no’lu aşamada oluşturulan Neden-Sonuç Grafiği ve/veya Tablo birleştirilir.

5. Eğer varsa “Kısa”(lar) göz önüne alınarak Neden-Sonuç Grafiği ve/veya Tablo güncellenir.

3.5. Durum Örneği

Yukarıdaki anlatılan süreç Şekil 1’de verilen Neden-Sonuç Grafiği ile örneklendirilerek test durumları oluşturulmuştur.

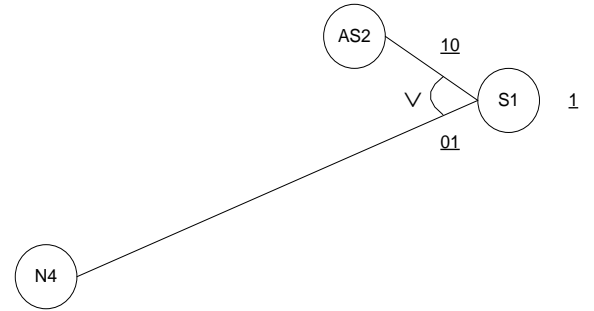
1. En son “Sonuç”tan başlanır.

- a. Şekil 1’e bakıldığında ikili yapı

(AS2 VEYA N4) => S1

olarak görülmektedir. Buradaki temel yapı ikili bir “VEYA” yapısıdır.

- b. “Sonuç” = 1 olarak alındığında sağlayan AS2 ne N4 değerleri de kullanılarak Şekil 6 oluşturulur.



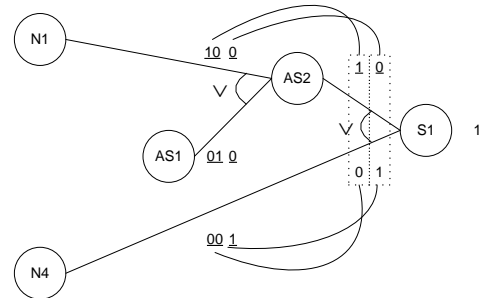
Şekil 6: Aşama 1 Neden-Sonuç Grafiği

Test durumları için Tablo 4 oluşturulmuştur.

Tablo 4: Aşama 1 Test Durumları

Test Durumu	AS2	N4	S1
TD1	1	0	1
TD2	0	1	1

2. En son “Sonuç” incelendikten sonra geriye doğru gidilir.
 - a. Bundan sonraki aşama AS2’nin açılmasıdır. Burada dikkat edilmesi gereken konu, girdilerin değerleri seçilirken “10” setini sağlayacak değerlerin seçilmesidir. Bunun sonucunda Şekil 7 oluşturulur.



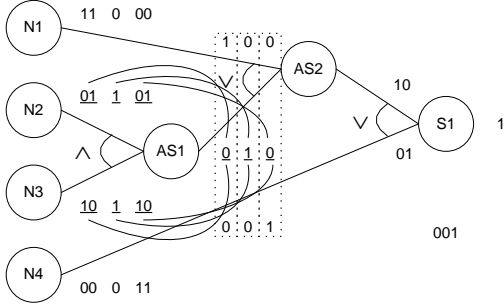
Şekil 7: Aşama 2.a Neden-Sonuç Grafikleri

Test durumları için Tablo 4’te AS2 yerine (N1 VEYA AS1) durumu gözönüne alınmış ve Tablo 4 genişletilerek Tablo 5 oluşturulmuştur.

Tablo 5: Aşama 2.a Test Durumları

Test Durumu		AS2			
		N1	AS1	N4	S1
TD1	TD1.1	1	0	0	1
	TD1.2	0	1	0	1
TD2	TD2.1	0	0	1	1

b. Bir sonraki aşama ise AS1'in açılmasıdır. Burada da dikkat edilmesi gerekli konu, girdilerin değerleri seçilirken AS1 için "010" setini sağlayacak değerlerin seçilmesidir. Bunun sonucunda Şekil 8 oluşturulur.



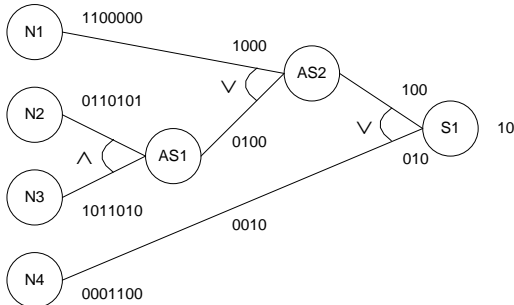
Şekil 8:Aşama 2.b Neden-Sonuç Grafiği

Test durumları için Tablo 6'da AS1 yerine (N1 VE N3) durumu gözönüne alınmış ve Tablo 5 genişletilerek Tablo 6 oluşturulmuştur.

Tablo 6: Aşama 2.b Test Durumları

Test Durumu		AS2					
		N1	AS1		N4	S1	
			N2	N3			
TD1	TD1.1	1.1.1	1	0	1	0	1
		1.1.2	1	1	0	0	1
	TD1.2	1.2.1	0	1	1	0	1
TD2	TD2.1	2.1.1	0	0	1	1	1
		2.1.2	0	1	0	1	1

3. "Sonuç" = 0 olarak alınır ve 1 ve 2 nolu adımlar tekrarlanır.
4. Elde edilen sonuç Şekil 6'daki sonuçla birleştirildiğinde elde edilen Neden-Sonuç Grafiği Şekil 9'da verilmektedir.



Şekil 9:Tamamlanmış Analiz

Tablo 7 ise kısıtlar gözötilmeksizin oluşan son test durumlarını vermektedir.

Tablo 7: Birleştirilmiş Analiz

Test Durumu	N1	N2	N3	N4	S1
TD1.1.1	1	0	1	0	1
TD1.1.2	1	1	0	0	1
TD1.2.1	0	1	1	0	1
TD2.1.1	0	0	1	1	1
TD2.1.2	0	1	0	1	1
TD3.1.1	0	0	0	0	0
TD3.1.2	0	1	0	0	0

5. Var olan "O" kısıtı da göz önüne alınınca (N1=1 ve N2=1) ve (N1=0 ve N2=0) içeren durumların olamayacağı belli olduğundan bu değerleri sağlayan TD.1.1.2, TD2.1.1 ve TD3.1.1 test durumları silinerek nihai Test Durumları Tablosu Tablo 8'deki şekilde oluşturulur.

Tablo 8: Nihai Test Durumları

Test Durumu	N1	N2	N3	N4	S1
TD1.1.1	1	0	1	0	1
TD1.2.1	0	1	1	0	1
TD2.1.2	0	1	0	1	1
TD3.1.2	0	1	0	0	0

4. Sonuçlar

Bu makalede Neden-Sonuç Grafiklerinin tanımı, kullanım alanları, öğeleri ve faydaları anlatılmış, verilen durum örneği ile kullanım şekli hakkında bilgi verilmiştir.

Neden-Sonuç Grafikleri diğer methodlarının aksine, gerek gereksinimlerin geçerlenmesinde gerekse uygun test adımlarının belirlenmesinde uygulayıcıya kolay ve anlaşılır bir yöntem sunmaktadır. Başlangıçta her ne kadar gereksinimlerin neden olarak adlandırılan atomik parçalara ayrılması ve ayrıştırılan bu nedenlerin kombinasyonlarını oluşturarak grafik ile sonuç oluşturulması zamansal açıdan etkin görülmesede, toplam yazılım geliştirme sürecine bakıldığında çok daha az hataya eğilim ve zaman etkin olduğu değerlendirilmektedir.

Yapılan çalışmalar sırasında Neden-Sonuç Grafiklerinin kullanımındaki tek zorluk olarak "Neden" sayısının ve "AraSonuç"ların çok fazla olduğu gereksinim setlerinde uygulama karmaşık bir hal alabilir. Bunun yanısıra, Neden-Sonuç Grafiği tekniğinin ölçeklenebilirliği metodun doğası gereği mümkün olmamaktadır. Bunların üstesinden gelmek ve insandan kaynaklı hata payını en aza indirebilmek için anlatılan tekniğin bilgisayar destekli bir araca dönüştürülmesi ve sürecin otomatikleştirilmesi uygun görülmektedir.

5. Kaynakça

- [1] Myers, G.L., "The Art of Software Testing", John Wiley & Sons, New York, 1979.
- [2] Nursimulu, K. and Probert, R.L., "Cause-Effect Graphing Analysis and Validation of Requirements", Proceedings of the 1995 Conference of the Centre for Advanced Studies on Collaborative Research, p.46, November 07-09, 1995, Toronto, Ontario, Canada.
- [3] Paradkar, A., Tai, K.C. and Vouk, M.A., "Specification-Based Testing Using Cause-Effect Graphs", SpringerLink Journal Article, Annals of Software Engineering, pp.133-157, November 30, 2004, Netherlands.