ANKARA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

YAZILIMDA İSTATİSTİKSEL SÜREÇ KONTROLÜ VE GÜVENİRLİK KESTİRİM MODELLERİ

Özgül SUGÜNEŞ

İSTATİSTİK ANABİLİM DALI

ANKARA
2010

TEZ ONAYI

Özgül SUGÜNEŞ tarafından hazırlanan "**Yazılımda İstatistiksel Süreç Kontrolü ve Güvenirlik Kestirim Modelleri"** adlı tez çalışması 28/04/2010 tarihinde aşağıdaki jüri tarafından oy birliği ile Ankara Üniversitesi Fen Bilimleri Enstitüsü İstatistik Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Yard. Doç. Dr. Mehmet YILMAZ

Jüri Üyeleri:

Başkan: Prof. Dr. Semra ERBAŞ
Gazi Üniversitesi, İstatistik Bölümü

Üye : Doç. Dr. Cemal ATAKAN
Ankara Üniversitesi, İstatistik Bölümü

Üye : Yard. Doç. Dr. Mehmet YILMAZ
Ankara Üniversitesi, İstatistik Bölümü

Yukarıdaki sonucu onaylarım.

Prof.Dr.Orhan ATAKOL

Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

YAZILIMDA İSTATİSTİKSEL SÜREÇ KONTROLÜ VE

GÜVENİRLİK KESTİRİM MODELLERİ

Özgül SUGÜNEŞ

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
İstatistik Anabilim Dalı

Danışman: Yard. Doç. Dr. Mehmet YILMAZ

Bu çalışma kapsamında, yazılım sistemlerinde kalite güvence ve kalite kontrolünün önemi tanımlar ve örnekler ile açıklanmıştır. Bununla birlikte, yazılım sisteminde ürününü etkileyebilecek süreçlerin istatistiksel süreç kontrolü yöntemi ile nasıl izlenip takip edileceği anlatılarak, sürecin yeterli ve kontrolde olduğu durumlar ve sürecin işleyişindeki farklılaşmayı azaltma yönelik yöntemler tartışılmıştır.

İstatistiksel süreç kontrolü, sürecin takibi için dünyada en etkin kullanılan yöntemlerden biridir ancak, istatistiksel bir tahmin modeli değildir. Bu nedenle, sürecin n zaman sonra davranışı ve kesin tahmin değerleri kontrol şemaları kullanılarak bilinemez. Sadece, sürecin farklılaşması dikkate alınarak belirlenmiş kontrol limitlerine göre pozitif veya negatif eğilimi tahmin edilebilir. Bu nedenle, çalışma kapsamında olasılıksal yazılım güvenirlik modelleri endüstri uygulamaları ile birlikte ayrıca incelenmiştir.

Nisan 2010, 105 sayfa

Anahtar Kelimeler: İstatistiksel Kalite Kontrol, 6 sigma, Yazılım Güvenilirliği, homojen olmayan poisson süreci güvenirlik modelleri

ABSTRACT

Master Thesis

STATISTICAL PROCESS CONTROL AND

RELIABILITY PREDICTION MODELS ON SOFTWARE

Özgül SUGÜNEŞ

Ankara University
Graduate School of Natural and Applied Sciences
Department of Statistics

Supervisor: Yard. Doç. Dr. Mehmet YILMAZ

The scope of this study, it is explained that why quality assurance and quality control are important for software systems with giving definitions and examples. Besides, statistical process control method is explained how to track and monitor processes which can affect product of software system. Process stability and capability conditions and methods to reduce process performance variation are discussed.

Statistical process control is the most effective method used in world to track process but it is not a statistical prediction model. Therefore, process behavior cannot be estimated n times later and exact predicted numbers cannot be known with using control charts. Only, positive or negative trend of process can be estimated according to determined control limits with considering process variation. So, as a part of this study, probabilistic software reliability models are examined separately with applications of industry.

April 2010, 105 pages

Key Words: Statistical Process Control, Six sigma, Software reliability, NHPP reliability models

TEŞEKKÜR

Bana araştırma olanağı sağlayan ve çalışmamın her safhasında yakın ilgi ve önerileri ile beni yönlendiren danışman hocam, Ankara Üniversitesi Fen Fakültesi İstatistik bölümünde görev alan Sayın Yard. Doç. Dr. Mehmet Yılmaz'a teşekkürlerimi sunarım.

Çalışmamı, benden sevgilerini hiçbir zaman esirgemeyen ilk öğretmenlerim canım babam ve annem Bekir SUGÜNEŞ, Döne SUGÜNEŞ'e ve sevgili ablalarım Funda, Almula ve Güldane SUGÜNEŞ'e adayarak, kendilerine sonsuz sevgilerimi sunarım.

Özgül SUGÜNEŞ

Ankara, Nisan 2010

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

1. GİRİŞ

Günümüzde hemen hemen herkes doğrudan veya dolaylı olarak bilgisayar sistemlerinden yararlanmaktadır. Bilgisayar sistemleri farklı alanlarda milyonlarca insanın hayatını etkileyen farklı uygulamalarda kullanılır. Bunlar, hava trafik kontrolü, nükleer reaktörler, hava araçları, gerçek zamanlı algılayıcı ağlar, endüstriyel süreç kontrolü, otomativ makine , emniyet kontrolü ve sağlık hizmetleri vb. dir.

Hayatımızı bu kadar çok etkileyen bilgisayar sistemlerinde meydana gelebilecek hatalar, büyük oranda maddi zarara ve can kaybına yol açabilmektedir. Örneğin, 31 Mart 1986 yılında Mexicana havayolları Boeing 727 uçağı dağda düşmüştür. Buna neden olarak, yazılım sisteminin dağın pozisyonunu doğru değerlendirememesi gösterilmiştir. 1986 Mart ile Haziran ayları arasında Massive Therac-25 isimli radyoterapi makinası, otomasyondaki bir yazılım hatası nedeni ile Georgia, Massachusetts ve Texas eyaletlerinde kanser hastalarına yüksek dozda radyasyon vermilmesine yol açmıştır. 26 Haziran 1988 yılında Air France hava yollarına iki gün erken teslim edilen yeni model A320 tip yolcu uçağı, Fransa'da Mulhouse yakınlarında alçak uçuş yaparken bilgisayar yazılımındaki bir hata nedeni ile düşmüş ve üç yolcunun öldüğü bildirilmiştir.

Amerikan Patriot (Phased-Array Tracking and Intercept Of Target) füze savunma sistemi, savaş esnasında düşman birlik tarafından gelen füzeleri havada etkisiz hale getirmesi ve hedefine ulaşmasını engelleme amacı ile geliştirilen bir üründür. 1991 yılında meydana gelen 1. Körfez savaşında, yazılım hatası nedeni ile Irak'tan gelen Scud füzelerinin hedefine ulaşması Patriot füze savunma sistemi tarafından engellenememiş ve yirmisekiz Amerikan askeri ölmüştür. Sistemin arızalanma nedeni olarak, yazılımın sistem saatini sık sık sıfırlandığı varsayımı ile geliştirildiği, ancak füze bir yerde yüz saatten fazla kaldığında yazılımın arızalandığı ve Patroit Irak füzelerinin izini yakalayamadığı açıklanmıştır.

4 Haziran 1996 yılında, Avrupa Uzay Şubesi (ESA) tarafından Ariane 5 füzesi uzaya fırlatılmış ancak fırlatıldıktan kırk saniye sonra parçalanmıştır. ESA bu başarısızlığın teknik nedeni olarak, Ariane 4 roketinin çalışma kodunu (source code) Ariane 5'te yeniden kullandıklarını, fakat Ariane 5'in daha hızlı olan motorunun, roketin uçuş kontrol sistemi yazılımında bir hatanın ortaya çıkmasına yol açıtığını belirtmiştir. ESA bu başarısızlığın 362 milyon dolar zararla sonuçlandığını bildirmiştir.

Örneklerde de anlatıldığı gibi; özellikle insan hayatını ilgilendiren kritik yazılımların bulunduğu sistemlerde, yazılım geliştirme süreci boyunca tespit edilen hatalarının izlenmesi, bunların yok edilmesi ve güvenirliklerinin tahmin edilmesi oldukça önemlidir.

Bu çalışma kapsamında; yazılım geliştirme sürecinde çıkan hataların izlenebilmesi ve yazılım sistemindeki mevcut hata sayısının tahmin edilebilmesi için, istatistiksel süreç kontrolü yöntemi ve güvenirlik tahmin metotları incelenmiştir.

2. KURAMSAL TEMELLER

Bu bölümde literatürde yaygın olarak kullanılan; kalite güvence kavramları, güvenirlik ölçümleri, dağılım fonksiyonları ve yazılım geliştirme süreçleri anlatılacaktır.

2.1 Kalite ve Kalite Kontrolü İle İlgili Temel Kavramlar

Tanım 2.1.1 Üretim; insan gereksinimlerinin doğa tarafından tam olarak karşılanamaması sonucu ortaya çıkan bir faaliyettir. Üretimin amacı, bir ürün veya hizmet yaratmaktır (Baskan 1997).

Tanım 2.1.2 Kalite; "Kalite" latince "qualitas" demektir ve nasıl oluştuğu anlamına gelen "qualis" sözcüğünden gelir. Sözcük anlamı ile hangi nesne için kullanılıyorsa onun gerçekte ne olduğunu belli etmek amacını taşır. Bu anlamda bir nesnenin kalitesi onun doğasındadır ve o nesne başka bir nesne ile değiştirilmeden kalitesi değiştirilemez. Bu nedenle kalite sözcüğünün tanımında önemli derecede karışıklık vardır. Çünkü kalite zaman içinde değişen koşullara uygun olarak şekil değiştirir. Günümüzde kalite için yapılan tanımlar:

i) Kalite planlamadır. Sorunlar ortaya çıkmadan önce çözümlerini oluşturarak, ürün ve hizmetlerin yapısına tasarım yoluyla üstünlük ve kusursuzluk katar.

ii) Kalite, müşteri tatminidir. Ürün ve hizmetlerin ne kadar iyi olduğu konusunda son kararın verdiği memnunluktur.

iii) Kalite verimliliktir. İşlerini yapabilmek için gerekli eğitimden geçen, ihtiyaç duyduğu gerekli araç-gereç ve talimatlarla desteklenen personelden elde edilir.

iv) Kalite esnekliktir. Talepleri karşılamak için değişmeyi göze almak ve bu konuda istekli olmaktır.

v) Kalite etkili olmaktır. İşleri çabuk ve her defasında doğru olarak yapmaktır.

vi) Kalite stratejik bir araçtır. Temel hedef, müşteri taleplerini karşılamak ve faaliyetlerini iş akışlarını sürekli olarak iyileştirerek bu yeteneği geliştirmektir.

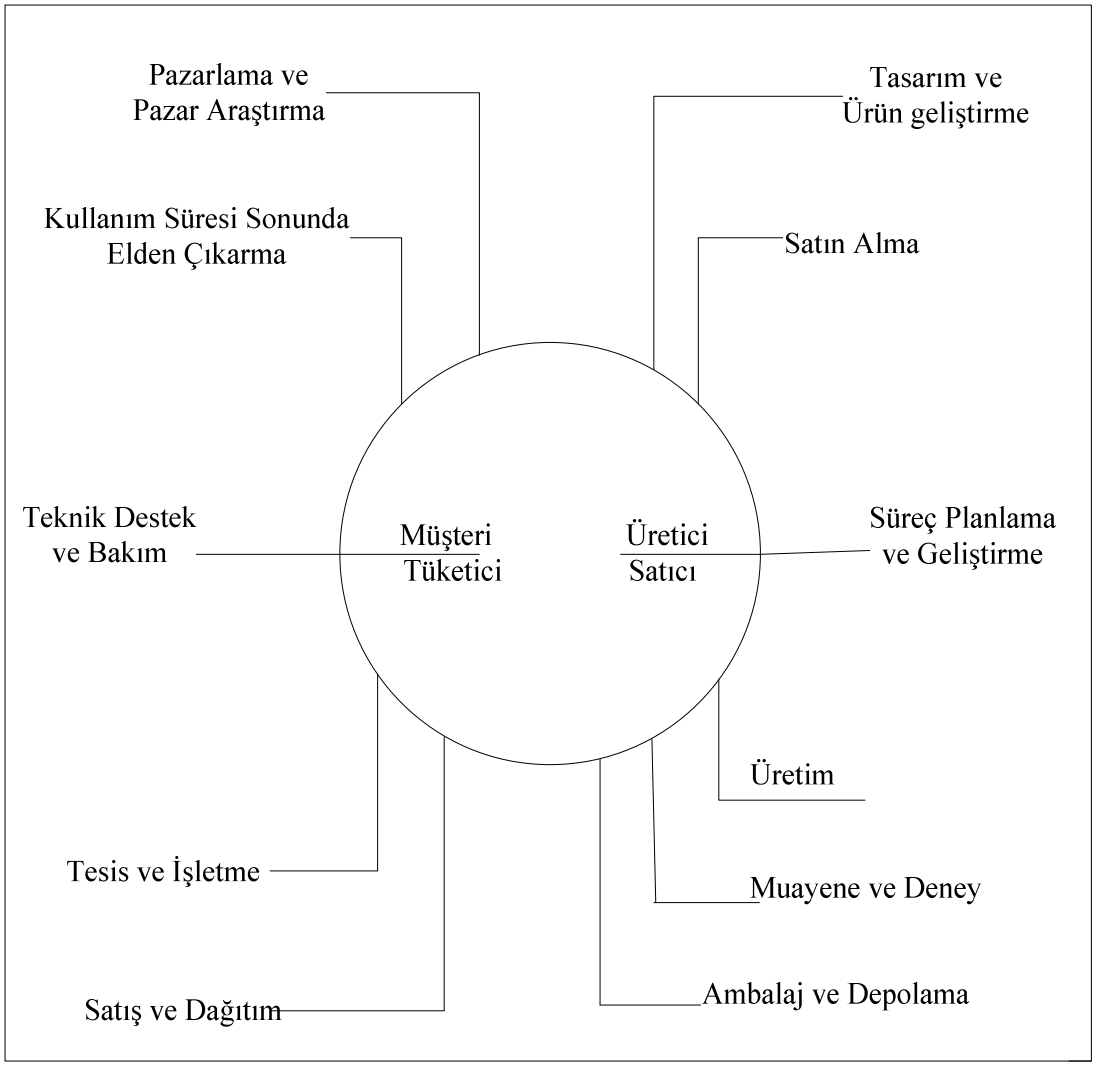
vii) Kalite bir yatırımdır. Uzun dönemde bir işi ilk defada doğru olarak yapmak, o hatayı sonradan düzeltmekten daima daha ucuzdur.

viii) Kalite bir süreçtir. Süregelen bir değişmeyi kapsar.

ix) Kalite bir programa uymaktır. İşleri zamanında yapmaktır.

biçimindedir. Tanımlarda da görüldüğü gibi kalite mutlak anlamda “en iyi” değildir. En genel anlamı ise, kalite bir ürün ya da hizmetin özelliklerinin, insan topluluklarının istek potansiyelini karşılayabilme derecesidir denilebilir (Baskan 1997).

Tanım 2.1.3 Kalite kontrol; üretim satın alma ve başka alanlarda kalitenin sağlanması, sürdürülmesi ve yükseltilmesi çalışmalarını planlama ve geliştirme yolu ile üretimin, tüketici açısından en ekonomik düzeyde ve en yüksek kalitede yapılmasına olanak sağlar. Bu yöntem ve faaliyetler şekil 2.1’deki kalite halkasının farklı aşamalarında işlerin/sürecin izlenmesi, hataların belirlenmesi, genel eğilimin izlenmesi ve uygunsuzluk nedenlerinin ortadan kaldırılmasını amaçlar.

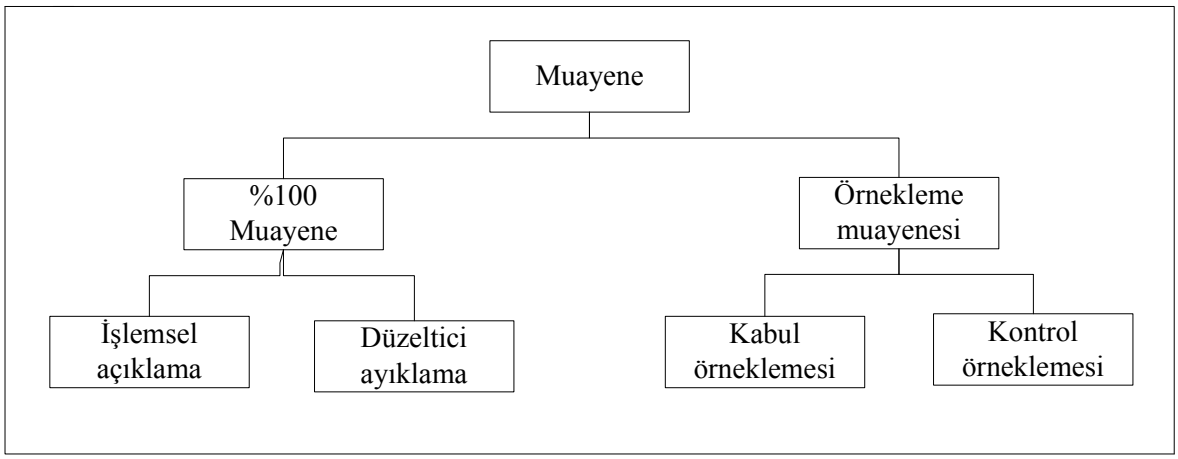


Şekil 2.1 Kalite çemberi (Baskan 1997)

Şekil 2.1'de görülen kalite halkası, herhangi bir ürün veya hizmet kalitesini etkileyen gereksinimlerin belirlenmesinden, bunların yerine getirilip getirilmediğinin araştırılmasına kadar geçen aşamaları kapsayan birbirine bağımlı faaliyetleri içerir.

Tanım 2.1.4 Muayene; tasarlanan ve gerçekleştirilen kalite düzeyleri arasındaki farkın belirlenmesini sağlayan bir faaliyettir (Baskan 1997).

Şekil 2.2'de görüldüğü gibi muayene iki sınıfta incelenir. %100 muayenede üretilen ya da belirli bir işletmeden geçen tüm ürünler incelemeye alınır. Örnekleme yönteminde ise kitleden belirli istatistiksel ölçütlere göre belirlenen ve rastgele seçilen "n" birimlik bir örneklem muayene edilir. Muayene, sonucunda basit anlamda kabul veya red şeklinde karar verilmesini sağlayan bir eleme işlemidir.



Şekil 2.2 Muayenenin sınıflandırılması (Baskan 1997)

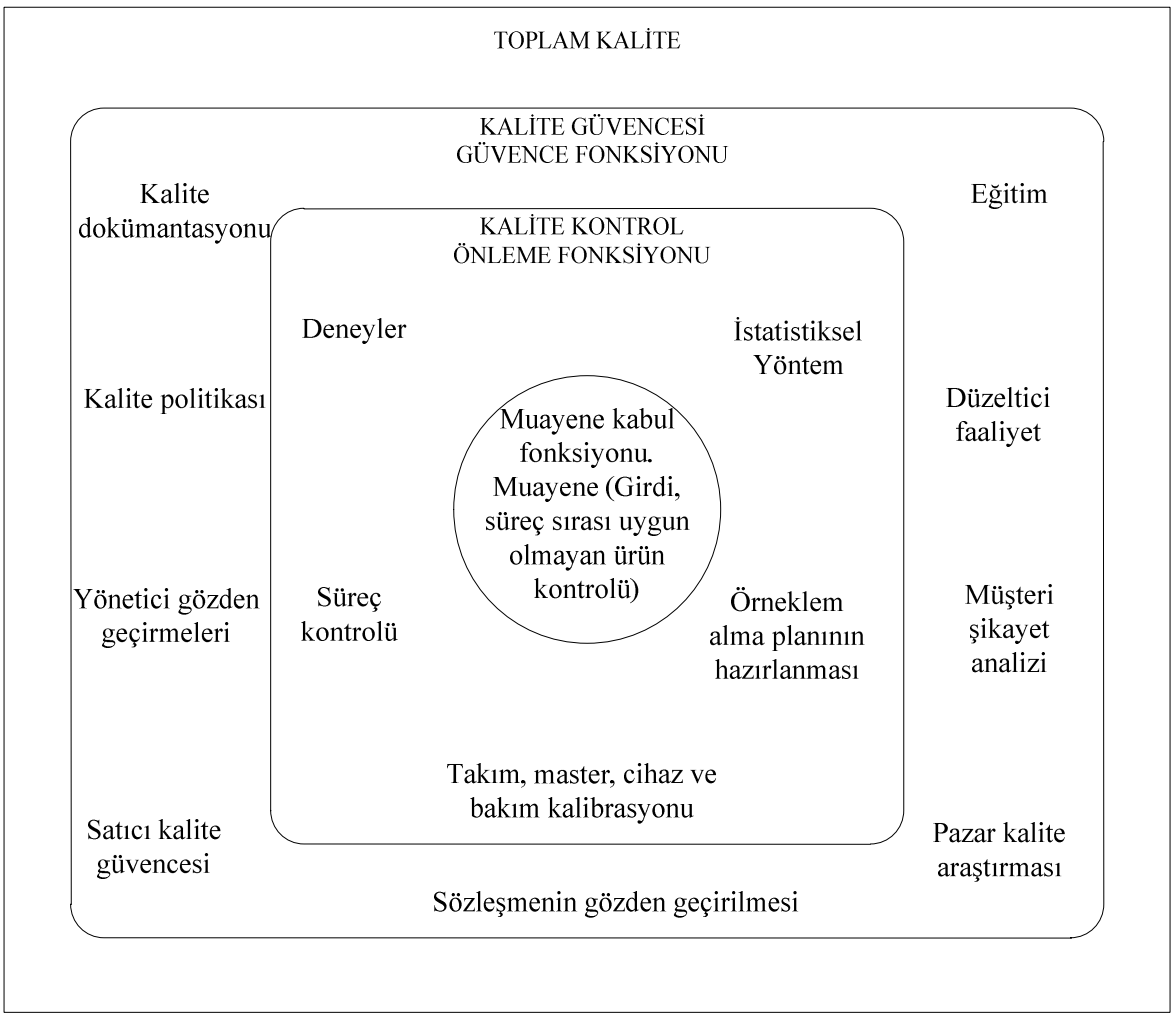
Tanım 2.1.5 Kalite güvencesi; ürün veya hizmetin kalite için belirlenen istekleri karşılamak amacıyla yeterli güveni sağlaması için gereken planlı ve sistematik faaliyetler bütünüdür.

Kaliteye yönelik faaliyetlerin planlı ve sistematik olarak yürütülmesi ile gerek firmanın çalışanlarına, gerekse müşterilere güven sağlanır. Kalite güvencesi şirket içi ve şirket dışı olmak üzere ikiye ayrılır (Baskan 1997):

a) Şirket içi kalite güvencesi; istenilen kaliteye ulaşması için şirket yönetimine güvence sağlamayı amaçlayan faaliyetlerdir.

Kalite güvencesinin tam olarak sağlanabilmesi için yönetim yapılacak işin ayrıntılı analizini, isteklerin belirlenmesini, uygun personelin seçimi ve eğitimini, uygun ekipman kullanımını, yapılacak işe uygun çevre koşullarının oluşturulmasını ve işi yapacak kişilerin sorumluluklarının belirlenmesi gibi temel konuları kapsayan çalışmalar yapılmalıdır.

b) Şirket dışı kalite güvencesi; kalite sisteminin alıcının belirlediği kalite taleplerine göre ürün veya hizmeti sağladığı konusunda alıcıya güven vermeyi amaçlayan faaliyetlerdir. Kalite güvencesi kısaca iyi yönetimdir, sistematik yaklaşımdır, ilk defada, zamanında ve doğru üretmektir. Kalite güvence fonksiyonu şekil 2.3'de verilmiştir.



Şekil 2.3 Kalite güvencesi kavramı (Baskan 1997)

Tanım 2.1.6 Kalite yönetimi; belirlenmiş ve tüm ilgililerce kabul gören bir kalite politikasının varlığı, gerekli ortamın oluşturulması, olanakların sağlanması, bu politikanın uygulanması ve etkinliğinin garanti altına alınması amacıyla gerekli faaliyetlerin gerçekleştirilmesidir (Baskan 1997).

Tanım 2.1.7 Kalite sistemi; kalite yönetiminin uygulanması için gerekli olan araçlardır ve iki kısımdan oluşur. Bunlardan ilki şirketin sorumluluk ve yapılarıdır, diğeri ise belirlenmiş olan sorumlulukları ve aktiviteleri şirket içerisindeki bireylere ulaştırmak için oluşturulan dokümantasyondur.

Öte yandan, kalite ile ilgili faaliyetlerin ve sonuçlarının planlanan düzenlemelere uygun olup olmadığının, bu düzenlemelerin etkili olarak uygulanıp uygulanmadığının ve amaca ulaşmak için uygun olup olmadığının sistematik ve tarafsız olarak incelenmesi ise kalite incelemesi olarak tanımlanır. Kalite incelemesi kalite iyileştirmelerinin etkin araçlarından biridir (Baskan 1997).

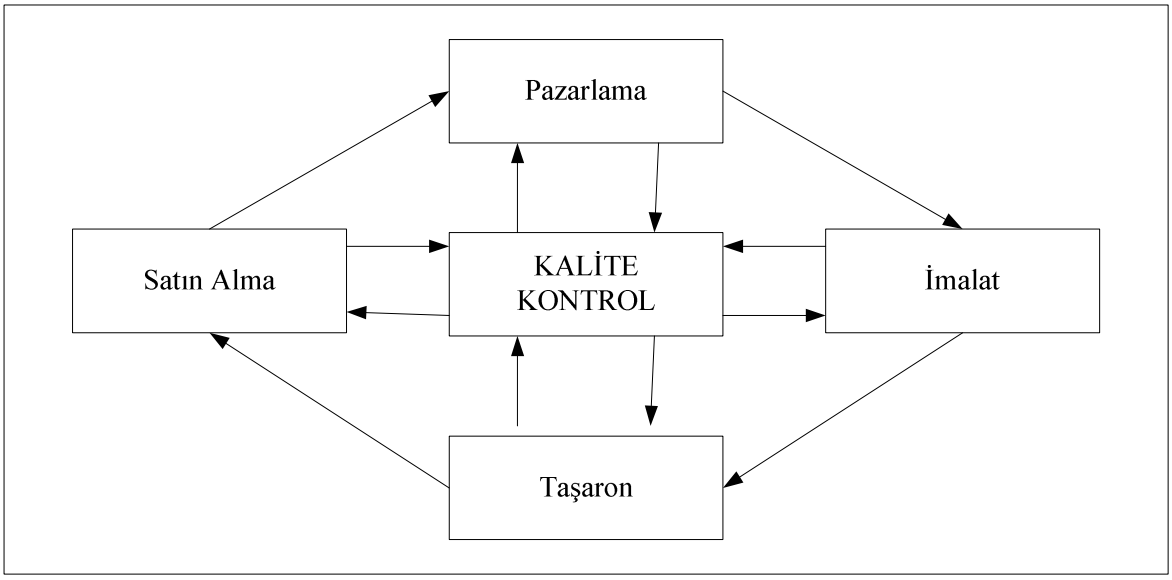
Tanım 2.1.8 Toplam kalite; yönetim, çalışanlar ve iş vericinin müşteri isteklerinin tatminine odaklanmasıdır. Toplam kalite programının etkenleri;

i) Üst yönetimin yönlendirmesi, önderliği ve katılımı,
ii) Gerekli alt yapının hazırlanması,
iii) Çalışanların katılımı ve motivasyonu,
iv) İş akışlarının süreçlerin kontrol altında tutulması,
v) Sürekli iyileştirme ve geliştirme ilkesi
biçimde sıralanabilir.

Geçmişte kalite kontrol olarak açıklanan tüm konular artık yerini Kalite Güvence Sistemi olarak değiştirmiştir. Bunun nedeni Avrupa Topluluğu (A.T)'na üye olan ülkelerin 31.12.1992'den itibaren belirli prensipleri yeterince yerine getirmeyen firmalarla çalışmayacağını açıklamış olmasıdır. ISO 9000 kalite standartına uyulması

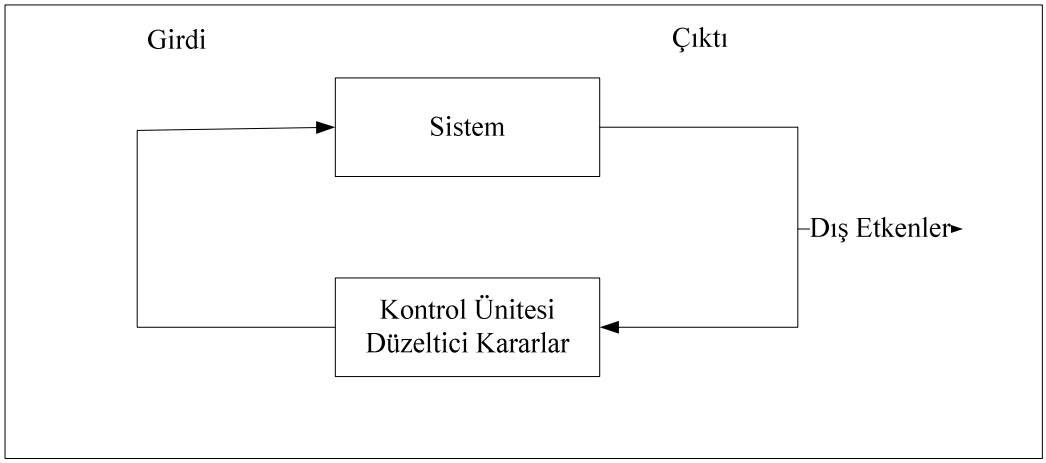
zorunlu kılınmıştır. Kalite güvencesi ürün veya hizmetin kalite için belirlenen istekleri karşılamak için gereken planlı ve sistematik faaliyetlerin tümüdür ve kalite kontrol birimi bu sistem içinde önemli yer tutar (Şekil 2.4). Kalite Kontrol Yöneticisi'nin amacı, işletmenin öngörülen amaç ve hedefine ulaşması için gerekli sistemi kurmak ve sistemin parçaları arasındaki ilişkileri saptayarak bunların bütünleşmesini sağlamak ve sistemi çalışır tutmaktır.

Sözü edilen sistem, aralarında ilişki ve bağımlılık bulunan elemanlardan oluşan bir yapı veya organik bütün, birbirleri ile olan ilişki göz önüne alınarak mantıksal bir plana göre düzenlenmiş bir olaylar, prensipler, kurallar, düşünceler, fiziksel varlıklar topluluğu olarak tanımlanır ve her eleman kendisinden daha büyük bir sistem içinde yer alır.

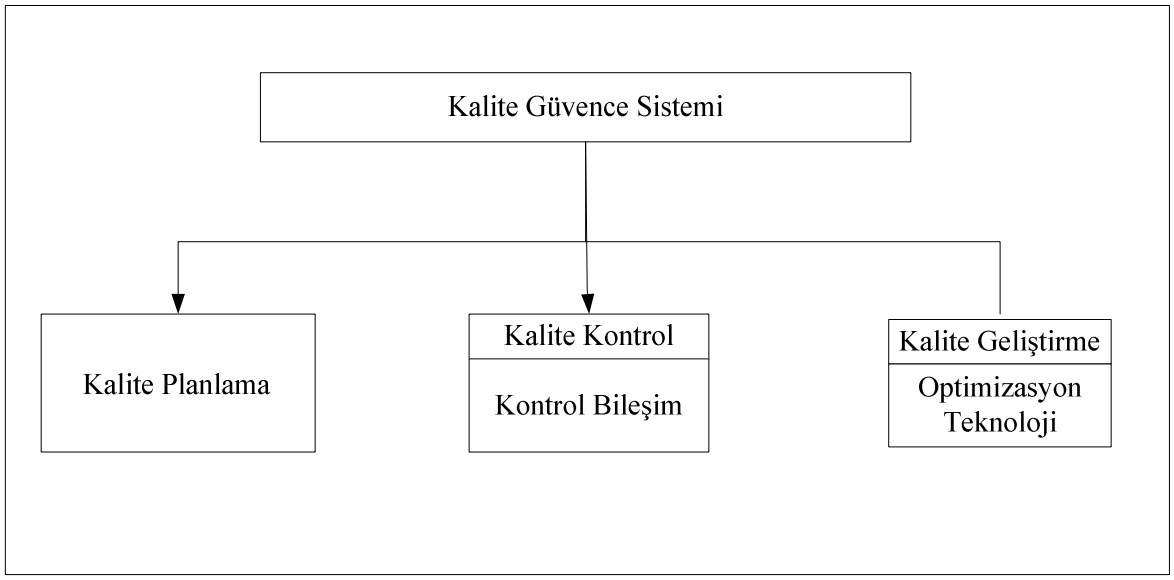


Şekil 2.4 Kalite kontrolün işletmedeki yeri (Baskan 1997)

İşletmedeki kalite hedefinin optimum şekilde gerçekleştirilmesi için, konunun işletme bünyesinde bir sistem dahilinde ele alınıp yürütülmesi gerekir (Şekil 2.5). Bu amaçla, önce kalitenin planlanması, sonra kontrol altına alınması ve geliştirilmesi aşamalarından söz edilebilir.



Şekil 2.5 Kapalı bir sistemin elemanları (Baskan 1997)



Şekil 2.6 Kalite güvence sisteminin ana yapısı (Baskan 1997)

Şekil 2.6'da görülen kalite güvence sisteminin planlama kısmını politikalar ve standart çalışmaları oluşturur ve mamül kalitesinin ne olacağının saptanmasına ilişkin kuralları kapsar. Bu kısım kontrol yöntemleri kullanımı ve bileşim sistemlerinden oluşmaktadır. Kalite geliştirme ise sistemdeki mevcut durumu esas alarak bir yandan maaliyetleri düşürücü, diğer yandan kalite ve üretkenliği artırıcı alternatif kararlar üretmek ve bunların geliştirilmesini sağlamaktır.

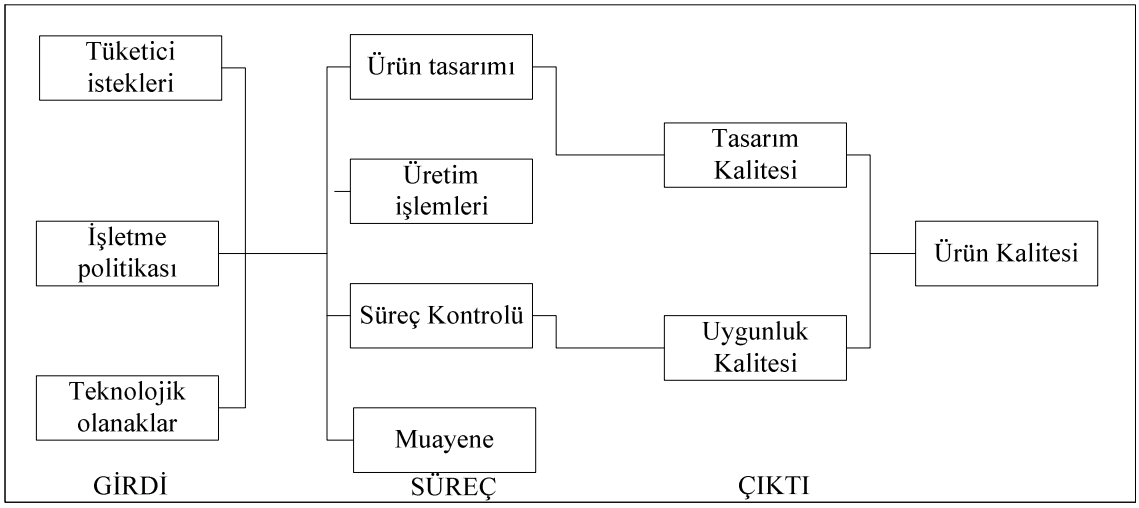
Özet olarak kalite güvence çemberini:

i) Kalite Kontrol
ii) İmalat
iii) Pazarlama
iv) Ambalaj
v) Satış
vi) Montaj
vii) Satın alma
viii) Teknik destek
ix) AR-GE

oluşturur. Görüldüğü gibi kalite güvencesi tüm servislerde oluşturulabilecek bir sistemdir.

Tanım 2.1.9 İstatistiksel Kalite Kontrolü; istatistiksel yöntemleri kullanarak kaliteye ilişkin sorunlu alanları saptayıp gerekli analiz ve incelemeleri yaparak kalite sorunlarına çözüm getirilmesine yardımcı olan sisteme istatistiksel kalite kontrol birimi denir. Kalite kontrol biriminin görevlerinin yoğunlaştığı bu alanlar kısaca giriş ve çıkış, süreç kontrolü olarak özetlenebilir.

Tanım 2.1.10 Süreç, üretim ile ilgili girdilerin, bazı işlemler ve ölçümler sonucunda çıktıya dönüşmesidir (Baskan 1997). Üretim birimlerinin hedeflenen kalitede ürün verebilmeleri için ürünün tasarlanması ve üretilmesi aşamalarında diğer birimlere yardımcı olan kaliteden sorumlu bir kalite kontrol biriminin olması bir zorunluluktur. Kalite kontrol sistemi ya da kalite güvence sistemi içinde yer alan bu birimlerin görevleri şekil 2.7'de görülmektedir.



Şekil 2.7 İstatistiksel kalite kontrolünün ana elemanları (Baskan 1997)

İstatistiksel kalite kontrolde, sürecin kontrol altında olup olmadığını belirlemek için kullanılan en etkin araçlar kontrol şemalarıdır. Bölüm 3.1'de detaylı olarak anlatılacaktır.

2.2 Güvenirlik Ölçümleri ve Dağılım Fonksiyonları

2.2.1 Güvenirlik ölçümleri

Güvenirlik, literatürde uygulamacılar ve araştırmacılar tarafından farklı tanımlanmaktadır. Kabul görmüş genel tanım aşağıdaki gibidir:

Tanım 2.2.1.1 Güvenirlik (Reliability), bir başarma olasılığıdır. Başka bir ifade ile sistemin belirlenen tasarım limitlerinde beklenen işlevini gerçekleştirme olasılığıdır (Kumar vd. 2006). Güvenirlik aynı zamanda müşterinin üründen beklediği kalite karakteristiklerinden biridir.

Matematiksel olarak güvenirlik, [0-t] zaman aralığında sistemin başarılı olması, başka bir ifade ile işlevini başarılı olarak sürdürmesi olasılığıdır. Yani, R(t) güvenirlik fonksiyonunu ifade etmek üzere,

$$R(t) = P(T > t) \ , \ t \geq 0$$

dır (Pham 2006).

Bu eşitlikte T, sistemin hatalılık zamanını (failure time) belirten rastgele değişkendir.

Tanım 2.2.1.2 Güvenilmezlik (Unreliability), F(t), hatalılık ölçümü, t zamanında sistemin hatalı olması, başka bir ifade ile işlevini başarılı olarak sürdürememesi olasılığı,

$$F(t) = P(T \leq t) \ , \ t \geq 0$$

dır.

F(t) hata dağılım fonksiyonudur. Eğer T hatalılık zamanı rastgele değişkeni f(t) yoğunluk fonksiyonuna sahipse R(t) ile f(t) arasındaki ilişki,

$$R(t) = \int_{t}^{\infty} f(s)ds$$

ve

$$f(t) = -\frac{d}{dt}[R(t)]$$

biçimindedir.

Yoğunluk fonksiyonu matematiksel olarak,

$$\lim_{\Delta t \to 0} P(t < T \le t + \Delta t) \cong R(t) - R(t + \Delta t)$$

biçiminde tanımlanabilir (Pham 2006).

Bu eşitlik; T hatalılık zamanının, t ile t+Δt zaman aralığında meydana gelme olasılığı olarak yorumlanabilir.

Test süreci başarılı olarak tamamlanmış yeni bir sistemin, t=0 operasyon zamanında başarılı olma olasılığı yüksektir, ancak operasyon zaman aralığı arttıkça başarılı olma olasılığı azalır. Güvenirlilik, görev zamanın (mission time) bir fonksiyonudur. Örneğin kimileri bir sitemin 24 saat görev zamanındaki güvenirliliği 0.995 diyebilir; ancak kesin zaman aralığı bilinmediği için, sistemin güvenirliliği 0.995'tir ifadesi anlamsızdır. Sistemin hangi zamanlar arasında güvenirliliğinin ölçülmek istendiği net bir şekilde tanımlanmalıdır (Pham 2006).

Tanım 2.2.1.3 Sistem hatalılık ortalama zamanı (System mean time to failure (MTTF)); Sistemin güvenirlik fonksiyonu R(t) olarak verildiğinde beklenen hatalılık zamanı ya da sistem hatalılık ortalama zamanı (MTTF),

$$MTTF = \int_0^\infty t f(t) dt$$

biçiminde tanımlanır.

$$f(t) = -\frac{d}{dt}[R(t)]$$

eşitliği yukarıdaki MTTF eşitliğinde yerine konulduğunda;

$$MTTF = -\int_0^\infty t d[R(t)]$$

$$= \left[-tR(t)\right]_0^{\infty} + \int_0^{\infty} R(t)dt$$

olarak elde edilir (Pham 2006).

Eşitliğin sağ tarafındaki ilk işlemin değeri her iki limit içinde sıfıra eşit olduğundan,

$$MTTF = \int_0^{\infty} R(t)dt$$

biçiminde ifade edilir (Pham 2006).

MTTF fonksiyonu, hatalılık zamanı dağılım fonksiyonunun tanımlı olduğu durumlarda kullanılmalıdır. MTTF değeri; sistemde minimum ne zaman hata çıkacağı bilgisini vermeyip, sistemde çıkmış olan hataların ortalama zamanı olarak tanımlanır (Pham 2006).

Tanım 2.2.1.4 Hatalılık oranı fonksiyonu (Failure rate function); $[t_1, t_2]$ zaman aralığındaki sistemin hatalılığı, güvenirlilik fonksiyonuna göre,

$$\int_{t_1}^{t_2} f(t)dt = \int_{t_1}^{\infty} f(t)dt - \int_{t_2}^{\infty} f(t)dt = R(t_1) - R(t_2)$$

biçimindedir ve hatalılık dağılım fonksiyonuna göre,

$$\int_{t_1}^{t_2} f(t)dt = \int_{-\infty}^{t_2} f(t)dt - \int_{-\infty}^{t_1} f(t)dt = F(t_2) - F(t_1)$$

biçimindedir.

[t_1, t_2] gibi belirlenmiş bir zaman aralığında hatanın meydana gelme oranına hatalılık oranı (failure rate) denir. Başka bir ifade ile; sistemin, belirlenmiş bir zaman aralığında hatalı çıkmasının, t_1 zamanından önce hatalı olmamasına oranıdır. Bu oran,

$$\frac{R(t_1)-R(t_2)}{(t_2-t_1)R(t_1)}$$

dır (Pham 2006).

Hatalılık oranı zamanın bir fonksiyonudur. Zaman aralığı [t, t+Δt] olarak tanımlandığında ifade,

$$\frac{R(t\ )-R(t\ +\Delta t)}{\Delta tR(t\ )}$$

biçiminde değişir.

Hatalılık fonksiyonu, zaman aralığı sıfıra yakınsadığında hatalılık oranının limiti olarak tanımlanır. Hatalılık fonksiyonu ani hatalılık oranıdır ve

$$h(t)=\lim_{\Delta t\to 0}\frac{R(t\ )-R(t\ +\Delta t)}{\Delta tR(t\ )}$$

$$=\frac{1}{R(t)}\left[-\frac{d}{dt}R(t)\right]$$

$$=\frac{f(t)}{R(t)}$$

olarak ifade edilir (Pham 2006).

Hatalılık fonksiyonunun önemi, sistemin yaşamı boyunca hatalılık oranındaki değişimi tek eksende hatalılık fonksiyonlarını çizerek gösterir. Örneğin, iki sistem belirlenmiş bir zaman diliminde aynı güvenirlik değerinde olabilir ancak sonraki zamanda hatalı oranları farklılık gösterebilir (Pham 2006).

2.2.2 Bazı güvenirlik dağılım fonksiyonları

Bu bölümde, güvenirlik analizlerinde yaygın olarak kullanılan dağılımların olasılık yoğunluk, dağılım ve güvenirlik fonksiyonları verilmiştir (Pham 2006).

2.2.2.1 Binom dağılımı

Binom dağılımı kalite kontrolde ve güvenirlikte en yaygın kullanılan kesikli dağılımlardan biridir. Binom dağılımının olasılık yoğunluk fonksiyonu;

$$P(X = x) = \binom{n}{x} p^x (1-p)^{n-x};\ \ x = 0,\ 1,\ 2,\ ...,\ n$$

biçimindedir. Burada,

$$\binom{n}{x} = \frac{n!}{x!(n-x)!}$$

n: deneme sayısı,

x: başarı sayısı,

p: başarı olasılığıdır.

Güvenirlik fonksiyonu ise (n birimden en az k tanesinin başarılı olduğu durumda);

$$R(k) = \sum_{x=k}^{n} \binom{n}{x} p^x (1-p)^{n-x}$$

dır (Pham 2006).

2.2.2.2 Poisson dağılımı

Poisson dağılımı; Binom dağılımının uygulandığı durumlara benzer durumlarda uygulanabilir, ancak örneklem sayısının bilinmediği olaylarda kullanılır (Pham 2006). Poisson dağılımı da kesikli bir rastgele değişkeni karakterize eder ve olasılık yoğunluk fonksiyonu;

$$P(X = x) = \frac{(\lambda t)^x e^{-\lambda t}}{x!}; \quad x = 0,\ 1,\ 2,\ ...$$

biçimindedir ve burada,

λ: sabit hatalılık oranı,

x : olay sayısıdır.

Başka bir ifade ile, $P(X = x)$ t zamanında x tane hatalı çıkma olasılığını verir. Poisson dağılımının güvenirlik fonksiyonu (k tane hatadan az çıkması olasılığı);

$$R(k) = \sum_{x=0}^{k} \frac{(\lambda t)^x e^{-\lambda t}}{x!}$$

biçimindedir (Pham 2006).

Bu dağılım yedek artık sistemlerde (standby redundant system) ihtiyaç duyulan yedek sayısını tespit etmek amacıyla kullanılabilir.

2.2.2.3 Üstel dağılım

Üstel dağılım, güvenirlik mühendisliğinde sabit bir hatalılık oranına sahip olduğundan temel rol oynar. Bu dağılım elektronik, elektronik parçalar ve sistemlerin yaşam zamanlarını modellemek için kullanılmaktadır.

Üstel dağılım yoğunluk fonksiyonu;

$$f(t) = \frac{1}{\theta} e^{\frac{-t}{\theta}} = \lambda e^{-\lambda t} \text{ , t} \geq 0$$

biçimindedir ve güvenirlik fonksiyonu;

$$R(t) = e^{\frac{-t}{\theta}} = e^{-\lambda t} \text{ , t} \geq 0$$

dır.

θ' nın değeri sıfırdan büyük iken, MTTF'in bir parametresidir ve $\lambda \geq 0$ ise hatalı oranı sabitidir. Üstel fonksiyonun hatalılık fonksiyonu veya hatalılık oranı,

$$h(t) = \frac{f(t)}{R(t)} = \frac{\frac{1}{\theta} e^{-\frac{1}{\theta}}}{e^{-\frac{1}{\theta}}} = \frac{1}{\theta} = \lambda$$

biçiminde bir sabittir.

Üstel dağılımın hatalılık oranı sabit olduğundan yaygın olarak kullanılmaktadır. Küvet eğrisi dediğimiz (bathtub curve) eğrinin düz kısmı için üstel dağılım çok iyi bir modeldir. Çünkü sistemler yaşamlarının büyük kısmını küvet eğrisinin bu kısmında geçirirler (Pham 2006).

2.2.2.4 Normal dağılım

Normal dağılım, Merkezi Limit teoreminden dolayı klasik istatistikte önemli rol oynar. Güvenirlik mühendisliğinde normal dağılım öncelikle ürünün hassasiyet ölçümlerinde kullanılır. Normal dağılım ortalama değere göre simetrik olup çan eğrisi modelindedir.

Normal dağılım yoğunluk fonksiyonu;

$$f(t)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2} \; ; -\infty<t<\infty, -\infty<\mu<\infty, \sigma>0$$

biçimindedir. Burada,

μ : ortalama değer;

σ : standart sapma

dır.

Birikimli dağılım fonksiyonu ;

$$F(t)=\int_{-\infty}^{t}\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{s-\mu}{\sigma}\right)^2}ds$$

dır ve güvenirlik fonksiyonu;

$$R(t)=\int_{t}^{\infty}\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{s-\mu}{\sigma}\right)^2}ds$$

biçimindedir (Pham 2006). Eğer,

$$Z=\frac{T-\mu}{\sigma}$$

dönüşümü yapılır ise standart normal yoğunluk fonksiyonu ;

$$f(z)=\frac{1}{\sqrt{2\pi}}e^{-\frac{z^2}{2}}\text{ , - }\infty<z<\infty$$

biçiminde elde edilir. Bu dağılım fonksiyonu ortalaması 0 ve standart sapması 1 olan standart normal dağılım fonksiyonudur.

Birikimli dağılım fonksiyonu ;

$$\Phi(t)=\int_{-\infty}^{t}\frac{1}{\sqrt{2\pi}}e^{-\frac{s^2}{2}}ds$$

dır. Φ standart normal dağılım fonksiyonudur. Böylece μ ortalamalı ve σ standart sapmalı normal dağılımlı rastgele değişken T için,

$$P(T\le t)=P\left(Z\le\frac{t-\mu}{\sigma}\right)=\Phi\left(\frac{t-\mu}{\sigma}\right)$$

eşitliği yazılabilir. Böylece, standart normal tablolar kullanılacaksa Φ gerekli ilişkiyi sağlamaktadır. Normal dağılım için Hatalılık oranı fonksiyonu t zamanında monoton olarak artar. Tüm t zamanında h(t) ≥ 0 olduğu kolaylıkla söylenebilir (Barlow ve Proschan 1975).

2.2.2.5 Log Normal dağılım

Log normal dağılım deneysel olarak birçok tip hata verisine uyumluluk sağlayacak esnek bir modeldir. Bu dağılım sürdürülebilirlik mühendisliği (maintaibility engineering) uygulamalarında yeniden tamir edilebilen sistemlerin hata olaslıklarının modellenmesinde ve hatalılık oranı şüphesinin modellenmesinde kullanılır (Pham 2006).

Log normal yoğunluk fonksiyonu;

$$f(t)=\frac{1}{\sigma t\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{\ln t-\mu}{\sigma}\right)^2}$$; $t\geq 0$, $-\infty<\mu<\infty$, $\sigma>0$

biçimindedir.

Not: μ ve σ parametreleri bu dağılım için ortalama ve standart sapmayı ifade etmez.

Matematiksel olarak X rastgele değişkeni $X=\ln T$ dönüşümü altında,

$$E(X)=E(\ln T)=\mu$$

$$V(X)=V(\ln T)=\sigma^2$$

μ ortalamalı ve σ^2 varyanslı normal dağılıma sahiptir.

$T=e^X$ olduğundan log normal dağılımın ortalaması, normal dağılım kullanılarak

$$E(T)=E\left(e^X\right)=\int_{-\infty}^{\infty}\frac{1}{\sigma\sqrt{2\pi}}e^{\left[x-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]}dx$$

$$E(T)=e^{\mu+\frac{\sigma^2}{2}}\int_{-\infty}^{\infty}\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2\sigma^2}\left[x-\left(\mu+\sigma^2\right)\right]^2}dx$$

bulunabilir. Böylece log normal dağılımın ortalaması;

$$E(T)=e^{\mu+\frac{\sigma^2}{2}}$$

dır. Başka bir ifade ile;

$$E\left(T^2\right)=E\left(e^{2X}\right)=e^{2\left(\mu+\sigma^2\right)}$$

dır (Pham 2006).

Log normal dağılımın varyansı;

$$V(T)=e^{2\mu+\sigma^2}\left(e^{\sigma^2}-1\right)$$

dır ve birikimli dağılım fonksiyonu ;

$$F(t)=\int_0^t \frac{1}{\sigma s\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\ln s-\mu}{\sigma}\right)^2} ds$$

dır (Pham 2006).

Z standart normal dönüşümü yapıldığında,

$$F(t)=P[T\le t]=P(\ln T\le \ln t)=P\left[Z\le \frac{\ln t-\mu}{\sigma}\right]$$

dır.

Güvenirlik fonksiyonu;

$$R(t)=P\left[Z>\frac{\ln t-\mu}{\sigma}\right]$$

dır ve hatalılık oranı fonksiyonu;

$$h(t)=\frac{f(t)}{R(t)}=\frac{\Phi\left(\frac{\ln t-\mu}{\sigma}\right)}{\sigma t R(t)}$$

biçiminde elde edilir. Φ standart normal dağılımın birikimli dağılım fonksiyonudur (Pham 2006).

2.2.2.6 Weibull dağılımı

Weibull dağılımı üstel dağılımın genelleştirilmiş halidir. Weibull dağılımı gayet esnek ve değişik tipte mühendislik uygulamalarının gösteriminde ve değişken hatalılık oranı fonksiyonlarının modellenmesinde uygundur (Pham 2006).

Üç parametreli Weibull yoğunluk fonksiyonu;

$$f(t)=\frac{\beta(t-\gamma)^{\beta-1}}{\theta^{\beta}}e^{-\left(\frac{t-\gamma}{\theta}\right)^{\beta}} \;;\; t\geq\gamma\geq 0$$

biçimindedir. Burada, θ ve β parametreleri ölçek (scale) ve biçim (shape) parametreleri, γ ise konum (location) parametresi olarak bilinir. Bu parametreler her zaman pozitiftir. Farklı parametreleri kullanarak bu dağılımı üstel dağılım veya normal dağılıma dönüştürmek mümkündür. $t \geq \gamma$ olduğu durumda güvenirlik fonksiyonu;

$$R(t)=e^{-\left(\frac{t-\gamma}{\theta}\right)^{\beta}} \;;\; t>\gamma>0,\beta>0,\theta>0$$

ve hatalılık oranı fonksiyonu;

$$h(t)=\frac{\beta(t-\gamma)^{\beta-1}}{\theta^{\beta}};\; t>\gamma>0,\beta>0,\theta>0$$

biçiminde olup, hatalılık oranı fonksiyonu β<1 iken azalır, β>1 iken artar, β=1 iken sabittir (Pham 2006).

2.2.2.7 Gamma dağılımı

Gamma dağılımı çarpık dağılıma sahip örneklemler için hatalılık olasılık fonksiyonu olarak kullanılabilir.

Gamma dağılımının yoğunluk fonksiyonu;

$$f(t)=\frac{t^{\alpha-1}}{\beta^{\alpha}\Gamma(\alpha)}e^{-\frac{t}{\beta}}: t\geq 0,\ \alpha,\ \beta>0$$

biçimindedir. Burada, α biçim parametresi, β ölçek parametresidir.

Gamma dağılımı güvenirlik fonksiyonu,

$$R(t)=\int_{t}^{\infty}\frac{1}{\beta^{\alpha}\Gamma(\alpha)}s^{\alpha-1}e^{-\frac{s}{\beta}}ds$$

biçimindedir. Burada α tam sayı ise, güvenirlik fonksiyonu;

$$R(t)=e^{-\frac{t}{\beta}}\sum_{i=0}^{\alpha-1}\frac{\left(\frac{t}{\beta}\right)^{i}}{i!}$$

biçiminde ifade edilir ve hatalılık oranı fonksiyonu;

$$h(t)=\frac{f(t)}{R(t)}=\frac{\frac{1}{\beta^{\alpha}\Gamma(\alpha)}t^{\alpha-1}e^{-\frac{t}{\beta}}}{e^{-\frac{t}{\beta}}\sum_{i=0}^{\alpha-1}\frac{(\frac{t}{\beta})^{i}}{i!}}$$

biçimindedir.

Gamma dağılım fonksiyonunun şekli Weibull dağılımına çok yakındır. $\alpha = 1$ olduğunda gamma dağılımının hatalılık oranı $1/\beta$ olan üstel dağılıma dönüşür. Gamma dağılımı sistemin n inci hatalılık zamanını modellemek için kullanılabilir. Buradaki hatalılık dağılımı üstel olmalıdır. Bu nedenle, X_i değişkeni $\theta = 1/\beta$ parametresine sahip üstel dağılım ise $T = X_1 + X_2 + ... + X_n$ değişkeni β ve n parametreleri ile gamma dağılımına sahip olur.

Diğer bir yöntem ile gamma yoğunluk fonksiyonu;

$$f(t) = \frac{\beta^{\alpha} t^{\alpha-1}}{\Gamma(\alpha)} e^{-t\beta};\ t > 0$$

biçiminde de yazılabilir. Yoğunluk fonksiyonu α biçim parametresi, β ölçek parametresi olarak iki parametreden oluşur. $0 < \alpha < 1$ olduğunda hatalılık oranı monoton azalır, $\alpha = 1$ olduğunda hatalılık oranı sabittir, $\alpha > 1$ olduğunda hatalılık oranı monoton artar.

Yoğunluk fonksiyonunun ortalama, varyans ve güvenirlik eşitlikleri sırasıyla;

Ortalama (MTTF) = $\frac{\alpha}{\beta}$,

Varyans = $\frac{\alpha}{\beta^2}$,

Güvenirlik = $\int_t^{\infty} \frac{\beta^{\alpha} x^{\alpha-1}}{\Gamma(\alpha)} e^{-x\beta} dx$

biçimindedir (Pham 2006).

2.2.2.8 Pareto dağılımı

Pareto dağılımı; şehrin nüfus yoğunluğu, hisse senedi fiyat dalgalanmalarında ve kişi geliri dağılımlarının sağ kuyrukları uzun olduğundan bu tip verileri modellemek için geliştirilmiştir (Pham 2006).

Pareto dağılımının yoğunluk fonksiyonu ;

$$f(t)=\frac{\alpha k^{\alpha}}{t^{\alpha+1}};\ k<t<\infty,\alpha>0$$

olmak üzere, yoğunluk fonksiyonunun ortalama, varyans, güvenirlik ve hatalılık oranı eşitlikleri sırasıyla;

Ortalama=$\frac{k}{(\alpha-1)};\ \alpha>1$

Varyans=$\frac{\alpha k^{2}}{(\alpha-1)^{2}(\alpha-2)};\ \alpha>2$

Güvenirlik=$\left(\frac{k}{t}\right)^{\alpha}$

Hatalılık oranı fonksiyonu=$\frac{\alpha}{t}$

biçimindedir (Pham 2006).

2.2.2.9 Rayleigh dağılımı

Rayleigh yoğunluk fonksiyonu ;

$$f(t)=\frac{t}{\sigma^{2}}e^{\left(-\frac{t^{2}}{2\sigma^{2}}\right)},\ t>0$$

olmak üzere, yoğunluk fonksiyonunun ortalama, varyans, güvenirlik ve hatalılık oranı eşitlikleri sırasıyla;

Ortalama = $\sigma\left(\frac{\pi}{2}\right)^{\frac{1}{2}}$,

Varyans= $\sigma^2\left(2-\frac{\pi}{2}\right)$

Güvenirlik= $e^{\frac{-\sigma t^2}{2}}$,

Hatalılık oranı fonksiyonu= $\frac{t}{\sigma^2}$

biçimindedir (Pham 2006).

2.3 Yazılım Geliştirme Süreçleri

Yazılım kelimesinin sözlük anlamına bakıldığında; yazılım, "bir bilgisayarda donanıma hayat veren ve bilgi işlemde kullanılan programlar, yordamlar, programlama dilleri ve belgelemelerin tümü" olarak ifade edilmektedir. Yazılım ayrıca, mevcut bir problemi çözmek amacıyla değişik cihazların birbirleriyle haberleşebilmesini sağlayan ve görevlerini ya da kullanılabilirliklerini geliştirmeye yarayan bilgisayar dili kullanılarak oluşturulmuş anlamlı ifadeler bütünü olarak da nitelendirilebilir. Yazılım geliştirme, yazılımın hem üretim hem de kullanım süreci boyunca geçirdiği tüm aşamalar olarak tanımlanabilir. Yazılım geliştirmede beş farklı süreç değerlendirmesinden söz etmek mümkündür. Bununlar analiz, tasarım, kodlama, test ve bakım fazı olmak üzere ele alınmaktadır (Özlü ve Özbilgin 2010).

a) Analiz fazı

Bir problemin çözümü olarak nitelediğimiz yazılımların ne yapacağını ve nasıl yapacağını belirlediğimiz yani problemi tanımladığımız aşama analiz aşamasıdır. Yazdığınız kod ancak isteneni doğru bir biçimde yerine getiriyorsa başarılı bir yazılımdır. Bu nedenle öncelikle yazılımdan ne istendiğinin doğru bir biçimde tanımlanması gerekir. Analiz aşaması personel, donanım ve sistem gereksinimlerinin belirlenmesi, sistemin fizibilite çalışmasının yapılması, kullanıcıların gereksinimlerinin analizi, sistemin ne yapıp ne yapmayacağının kısıtlamalar göz önüne alınarak belirlenmesi, bu bilginin kullanıcılar tarafından doğrulanması ve proje planı oluşturulması adımlarından oluşur (Özlü ve Özbilgin 2010).

b) Tasarım fazı

Analiz aşaması sonucunda belirlenen gereksinimlere yanıt verecek yazılımın temel yapısının oluşturulduğu aşamadır. Yazılım tasarımı, bir bileşen veya sistemin nasıl gerçekleştirileceğini belirlemek için kullanılan teknikler, stratejiler, gösterimler ve desenlerle ilgilidir. Bu aşama yazılım bileşenleri arasındaki içsel ara yüzler, mimari tasarım, veri tasarımı, kullanıcı ara yüzü tasarımı, tasarım araçları ve tasarımın değerlendirilmesi alt süreçlerini de kapsamaktadır. Tasarım aşaması, yazılımın hem kullanıcı ara yüzünü hem de programın omurgasını ortaya koymaktadır. Yapılacak tasarım, yazılımın işlevsel gereksinimlere uygun olmasının yanı sıra kaynaklar, performans ve güvenlik gibi kavramları da göz önüne alınarak gerçekleştirilmelidir (Özlü ve Özbilgin 2010).

c) Kodlama fazı

Kodlama aşaması, tasarım sürecinde ortaya konulan veriler doğrultusunda yazılımın gerçekleştirilmesi aşamasıdır. Bu süreç programlama çalışmalarının yanı sıra yazılımın geliştirilmesi ve kullanıcıya ulaştırılması sürecindeki bütün çalışmaları kapsar. Tasarım

sonucu üretilen süreç ve veri tabanının fiziksel yapısını içeren fiziksel modelin bilgisayar ortamında çalışan yazılım biçimine dönüştürülmesi çalışması olarak da nitelendirilebilir. Yazılım geliştirme ortamı, programlama dili, veri tabanı yönetim sistemi, yazılım geliştirme araçları seçimi kodlama aşamasında gerçekleştirilir (Özlü ve Özbilgin 2010).

d) Test fazı

Test aşaması, yazılım kodlanması sürecinin ardından gerçekleştirilen sınama ve doğrulama aşamasıdır. Elde edilen uygulama yazılımının hem belirlenen gereksinimleri sağlayıp sağlamadığı hem de gerçekleştirimin beklentilere uygun olup olmadığını kontrol etmek için statik ve dinamik sınama tekniklerinden yararlanır. Yazılım üretiminde ilk testler genelde geliştirme sürecinde programcı tarafından yapılır. Bununla birlikte, asıl hata ayıklama ve geribildirim hizmeti test ekipleri tarafından yapılır. Testler ve geribildirim müşteri yazılımı kullandığı sürece devam eder. Test sürecinde en faydalı geribildirimler son kullanıcı test gruplarından gelir (Özlü ve Özbilgin 2010).

e) Bakım fazı

Yazılımın tesliminden sonra hata giderme ve yeni eklentiler yapma aşamasıdır. Yazılımın kullanıma başlanmasından sonra yazılımın desteklenmesi sürecini kapsar. Yazılımın eksiklerinin giderilmesi, iyileştirilmesi gibi alt aşamaları içeren aşamadır (Özlü ve Özbilgin 2010).

3.METERYAL VE YÖNTEM

3.1 İstatistiksel Süreç Kontrolü (İ.S.K)

Süreç kontrolü, üretim sırasında dış etkenlerin neden olduğu kalite sorunlarının geciktirilmeden incelenip giderilmesini böylece doğabilecek zararların önlenerek verimliliğin en üst düzeyde tutulmasını amaçlar. Teorik yapısı 1926'da W.A Shewhart tarafından oluşturulan kontrol şemaları bu sürecin istatistiksel yöntemlerle olmak üzere ekonomik ve güvenilir biçimde kontrol altında tutulmasında en etkili araçlardır (Baskan 1997).

Doğal olmayan nedenlerle ortaya çıkan değişiklikler süreci olumsuz olarak etkilediğinden bu nedenlerin tanımlanmaları, araştırılmaları ve kontrol altında tutulmaları gerekir. Bir kontrol şeması süreçte meydana gelen değişikliklerin doğal veya doğal olmayan nedenlerden oluştuğunu ayırdetmeye yarayan önemli bir araçtır.

İstatistiksel süreç kontrolü (İ.S.K)'nün amaçları ve İ.S.K'dan beklenen yaralar,

i) Ürün kalitesinin gelişmesi
ii) Kalite farklılıklarının en aza indirgenmesi
iii) Kalite maliyetlerinin düşmesi
iv) Hatalı ürün sayısının azalması
v) Muayene ve test masraflarının azalması
vi) Ürün güvenirliliğinin artması
vii) Karın artması
viii) Rekabet gücü ve pazar payının artması
ix) Reklam harcamalarının azalması
biçiminde sıralanabilir (Baskan 1997).

W.A Shewhart, kontrol şemalarının ilk olarak işletmecinin üretimi için bir standartı yani amacı belirlemeye, ikinci olarak bir amaca ulaşmak için bir araç olarak kullanılmaya, üçüncü olarak ise amaca ulaşıp ulaşmadığını ölçmeye hizmet ettiğini belirtir.

Yine Shewhart'a göre herhangi bir süreç kontrolünde doğal değişkenlere ilişkin limitleri belirlemek olanaklıdır. Bu limitler arasındaki değişimler rastgele değişkenin yapısındaki değişimlerden meydana gelmekte ve bu limitlerin dışında kalan durumlar ise üretimdeki diğer önemli değişiklikler sonucu meydana gelmektedir.

Genel olarak, kalitenin üretimin herhangi bir aşamasında hammaddeden ürüne kadar ölçülebilen ya da ölçülemeyen bir karakteristik olduğu bilinir ve kalite karakteristikleri iki gruba ayrılır.

a) Ölçülebilen özellikler

Bu özellikler bir sınıftaki öğrencilerin ağırlığı, bir elektrik ampulünün ömrü, bir makinenin ömrü, lif uzunluğu, bant numarası, kumaş mukavemeti, gram, watt, milimetre vb. gibi özelliklerdir (Baskan 1997).

b) Ölçülemeyen özellikler

Hatalı, hatasız biçiminde sınıflandırmanın yapıldığı özelliklerden olup iki tür özellikten söz edilebilir.

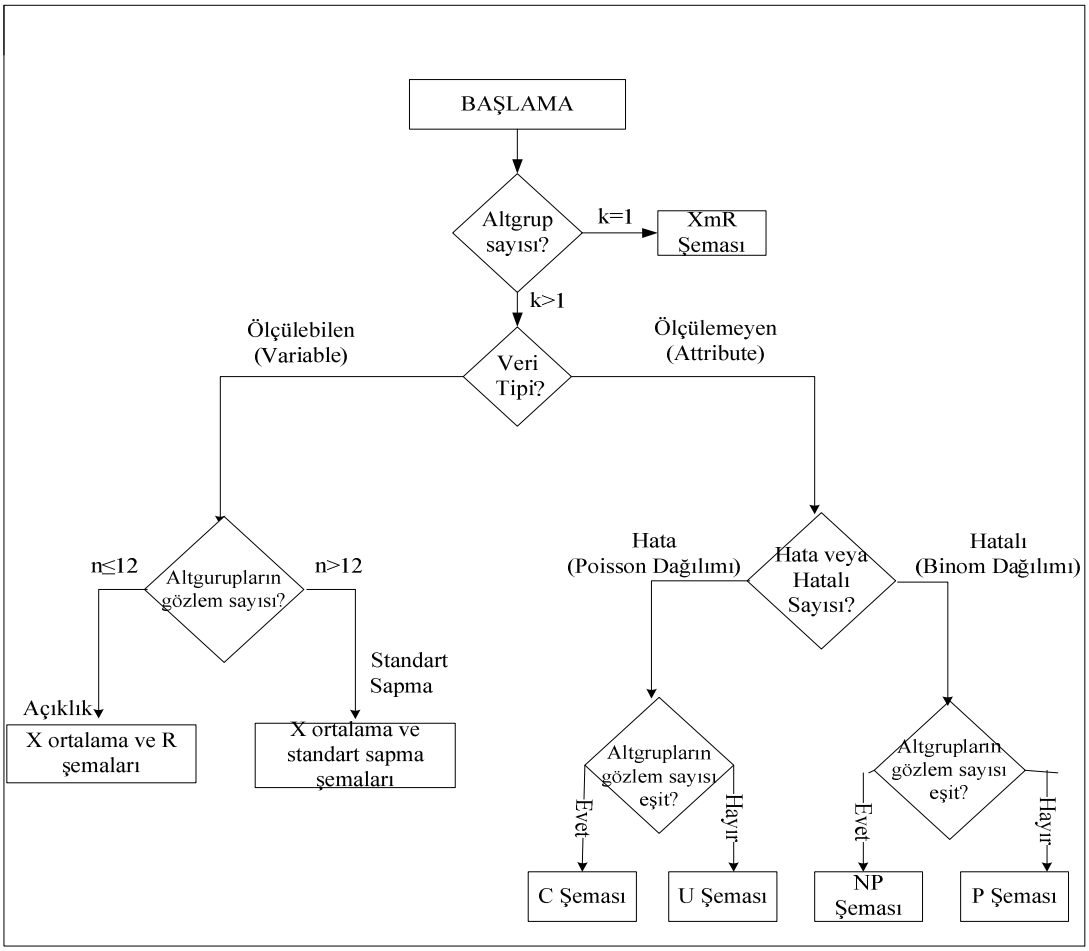
Nitelik gösteren renk, eksik parça, çatlak dokuma hataları gibi gerçekten ölçüme uygun olmayan sadece gözle vaya başka bir araçla muayene edilebilen nesnelerin kalite karakteristikleridir.

Ölçülemeyen ancak sayılabilen bu özellikler, ölçülmesi olanaklı olan ancak zaman ve maaliyet tasarrufu amacıyla ölçülemeyen kalite karakteristikleridir. Örneğin birim uzunluktaki iplik hataları, dokuma hataları bu tür özelliklerdir.

Günlük yaşamda karşılaşılan pek çok değişken gibi üretim ortamı incelendiğinde örneğin makinalar, hammaddeler, yabancı malzemeler, işçiler istenen koşullar vb. değişkenlere ilişkin verilerin normal dağılıma uygun frekans dağılımına sahip oldukları gözlenmiştir. Sürekli bir rastgele değişkenin değerleri olan ölçüm değerlerinin genellikle normal dağılıma uygun olduğu bilinir (Baskan 1997).

Öte yandan hata oranı, hata sayısı gibi ölçülemeyen değişkenlerin dağılımlarının ise Binom, Poisson ya da Hipergeometrik dağılım gibi kesikli dağılımlarından birine uydukları söylenebilir.

Bir üretim süreci, ürün veya herhangi bir çıktıya ait ölçülebilen ya da ölçülemeyen kalite özelliklerinin bütün çeşitlerine uygulanabilir. Bir çok kontrol şeması vardır ancak süreç için en uygun kalite özelliği ve bu özelliğe en uygun kontrol şemaları seçilmelidir. Şekil 3.1’de hangi durumda hangi tip kontrol şemasının seçileceği bir akış olarak verilmiştir.



Şekil 3.1 Süreç kontrol şeması

3.1.1 Altgurup sayısı (k) 1'e eşit olan durumlar için kontrol şemaları

Ölçümlerin zamana genişçe yayıldığı, her ölçümün sürecin kontrolünde ve değerlendirilmesinde kullanıldığı durumlardır. Altgruplar kolaylıkla rastgele olmayan gözlemleri içinde barındırabilirler, rastgele olmayan gözlemlerin standart sapma hesaplamasında etkisini azaltmak için altgrup sayısı olabildiğince azaltılır. En küçük altgrup sayısı 1 dir. Standart sapma iki gözlem arasındaki açıklık hesaplanarak elde edilir (Florac ve Carleton 2001).

Sıralı k tane gözlemin n-1=r tane hareketli açıklık (moving range) değeri olur.

i. hareketli açıklık;

$$mR_i = \left| X_{i+1} - X_i \right|, \; 1 \le \text{i} \le \text{n-1}$$

biçiminde tanımlanır.

Bireysel (Individual) kontrol şemaları için ortalama hareketli açıklık değeri, merkez çizgisi ve kontrol sınırları sırasıyla;

$$\overline{mR} = \frac{1}{r}\sum_{i=1}^{r} mR_i \quad ,$$

Merkez Çizgisi (M.Ç);

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \quad ,$$

Üst Kontrol Sınırı (Ü.K.S);

$$\overline{X} + \frac{3\overline{mR}}{d_2} = \overline{X} + 2.66\overline{mR}$$

ve

Alt Kontrol Sınırı (A.K.S);

$$\overline{X} - \frac{3\overline{mR}}{d_2} = \overline{X} - 2.66\overline{mR}$$

dır.

Hareketli açıklık (moving range) kontrol şeması için Merkez Çizgisi (M.Ç) ve kontrol sınırı;

$$\overline{mR}$$

ve

Üst Kontrol Sınırı (Ü.K.S);

$$D_4\overline{mR}$$

dır. Burada, d_2 ve D_4 sırasıyla orta çizgi ve kontrol limiti çarpanlarıdır ve altgrup gözlem sayısına göre belirlenmiş değerleri çizelge 3.1'de tanımlanmıştır.

3.1.2 Ölçülebilen özellikler için kontrol şemaları

Bu kontrol şemaları bir sürece ilişkin olan belli sayıdaki gözlemin ortalamasında ve yayılmasında meydana gelen değişimleri görebilme aracıdır ve sürecin istatistiksel olarak kontrol altında olduğunu denetler.

3.1.2.1 $\overline{X}$ kontrol şeması

$\overline{X}$ kontrol şeması süreçte işlem gören veya çıkan parçaların merkezi eğilimlerinin incelenmesi amacıyla ya da diğer bir deyişle geçmiş ve gelecekteki üretim kontrolü için kullanılır. Üretimin doğru bir görünüşünü elde edebilmek için üretimden genellikle 4 veya 5 birimden oluşan en az 20 genellikle 25 alt grup örneklem olarak seçilir (Baskan 1997).

Kontrol şemasının çizilebilmesi için kitlenin μ (ortalama) ve σ^2 (varyans) parametrelerinin bilinip bilinmediği durumlarına göre iki farklı yol izlenmelidir. Bunlar:

a) μ ve σ bilindiği durumda $\overline{X}$ kontrol şemaları

Gelecekteki üretimin kontrolü amacıyla oluşturulan bu şemalarda, geçmişteki üretimin kontrol dışında oluşmasının etkileri araştırılıp giderildikten ve bunların gelecekteki üretime etkide bulunmalarının önüne geçildikten sonra, ilerideki üretimin bu gibi

etkilerden korunabilmesi için standart kontrol şemaları oluşturulur. Kontrol şemalarında merkez çizgisi ve kontrol sınırları sırasıyla,

Merkez Çizgisi (M.Ç);

μ,

Üst Kontrol Sınırı (Ü.K.S);

$\mu + \frac{3\sigma}{\sqrt{n}}$ veya $\mu + A\sigma$,

Alt Kontrol Sınırı (A.K.S);

$\mu - \frac{3\sigma}{\sqrt{n}}$ veya $\mu - A\sigma$

biçimindedir ve burada A= $\frac{3}{\sqrt{n}}$ dır (Baskan 1997).

b) μ ve σ parametrelerinin bilinmediği durumda $\overline{X}$ kontrol şemaları

Bu durum özellikle yeni bir malzemenin hizmete girişinde, yeni hammadde kullanıldığında veya yeni bir ürün yapımında ortaya çıkar.

Bilinmeyen kitle parametrelerinin tahmin edicisi μ nun yansız tahmin edicisi $\overline{\overline{X}}$,

$\overline{\overline{X}} = \frac{\sum \overline{X_i}}{k}$; i=1, 2, 3, … , k

biçimindedir, burada k değeri alt grup sayısını ifade eder.

Kontrol şeması seçiminde önemli olan bir diğer faktör de alt grupların örneklem sayısıdır. Bu durumda;

i) Alt grupların gözlem sayısı n≤12 ise, $\overline{X}-R$ kontrol şeması kullanılmalıdır. Bu durumda $\overline{R}$, n gözlemlik k tane alt grubun açıklığı olmak üzere,

$$\overline{R}=\frac{\sum R_i}{k}\text{; i=1, 2, 3, … , k}$$

biçiminde tanımlanır. Burada σ'nın yansız tahmin edicisi,

$$\frac{\overline{R}}{d_2}=\hat{\sigma}$$

biçimindedir (Baskan 1997). Kontrol şemasının merkez çizgisi ve kontrol sınırları sırasıyla,

Merkez Çizgisi (M.Ç);

$$\overline{\overline{X}}\ ,$$

Üst Kontrol Sınırı (Ü.K.S);

$$\overline{\overline{X}}+\frac{3\overline{R}}{d_2\sqrt{n}}\ \text{veya}\ \overline{\overline{X}}+A_2\overline{R}$$

ve

Alt Kontrol Sınırı (A.K.S);

$$\overline{\overline{X}}-\frac{3\overline{R}}{d_2\sqrt{n}}\ \text{veya}\ \overline{\overline{X}}-A_2\overline{R}$$

dır (Florac ve Carleton 2001).

ii) Altgurupların gözlem sayısı n>12 ise $\overline{X}-S$ kontrol şeması kullanılmalıdır. Burada herbir alt örneklem grubunun standart sapması s_i,

$$s_i = \sqrt{\frac{\sum_{j=1}^{n_i}\left(x_{ij} - \overline{x_i}\right)^2}{n_i - 1}},\ i = 1,\ 2,\ ,...,\ k$$

dır. Bu durumda n gözlemlik k tane alt grubun ortalama standart sapma ise,

$\bar{s} = \frac{\sum_{i=1}^{k} s_i}{k}$; i=1, 2, 3, … , k ile hesaplanır. Bu durumda σ'nın yansız tahmin edicisi,

$$\frac{\bar{s}}{c_2} = \hat{\sigma}$$

biçimindedir (Baskan 1997). $\overline{X} - S$ kontrol şemasının kontrol sınırları,

Üst Kontrol Sınırı (Ü.K.S);

$\overline{\overline{X}} + \frac{3}{c_2\sqrt{n}}\bar{s}$ veya $A_1 = \frac{3}{c_2\sqrt{n}}$ ise $\overline{\overline{X}} + A_1\bar{s}$

ve

Alt Kontrol Sınırı (A.K.S);

$\overline{\overline{X}} - \frac{3}{c_2\sqrt{n}}\bar{s}$ veya $A_1 = \frac{3}{c_2\sqrt{n}}$ ise $\overline{\overline{X}} - A_1\bar{s}$

biçiminde tanımlanır (Florac ve Carleton 2001).

$\overline{X}$ -R ve $\overline{X}$ -S kontrol şemaları için yukarıda verilmiş olan formüllerdeki, A_2 ve A_1 sabitinin çarpan değerleri çizelge 3.1'de, özet formüller ise çizelge 3.2'de verilmiştir.

Çizelge 3.1 Kontrol sınırlarının hesaplanması için çarpanlar (Baskan 1997)

Ölçülebilen veri (Variables Veri) tipi Kontrol Şemaları	X Ortalama ve R Şemaları				X Ortalama ve S Şemaları			
	Ortalama Şemaları	Açıklık (R) Şemaları			Ortalama Şemaları	Standart Sapma (S) Şemaları		
	Kontrol Limitleri Çarpanları	Orta Çizgi Çarpanları	Kontrol Limitleri Çarpanları		Kontrol Limitleri Çarpanları	Orta Çizgi Çarpanları	Kontrol Limitleri Çarpanları	
Altgrup gözlem sayısı n	A_2	d_2	D_3	D_4	A_1	c_2	B_3	B_4
2	1.88	1.128	-	3.267	2.659	0.7979	-	3.267
3	1.023	1.693	-	2.574	1.954	0.8862	-	2.568
4	0.729	2.059	-	2.282	1.628	0.9213	-	2.266
5	0.577	2.326	-	2.114	1.427	0.94	-	2.089
6	0.483	2.534	-	2.004	1.287	0.9515	0.03	1.97
7	0.419	2.704	0.076	1.924	1.182	0.9594	0.118	1.882
8	0.373	2.847	0.136	1.864	1.099	0.965	0.185	1.815
9	0.337	2.97	0.184	1.816	1.032	0.9693	0.239	1.761
10	0.308	3.078	0.223	1.777	0.975	0.9653	0.284	1.716
15	0.223	3.472	0.347	1.653	0.789	0.9727	0.428	1.572
25	0.153	3.931	0.459	1.541	0.606	0.9896	0.565	1.435

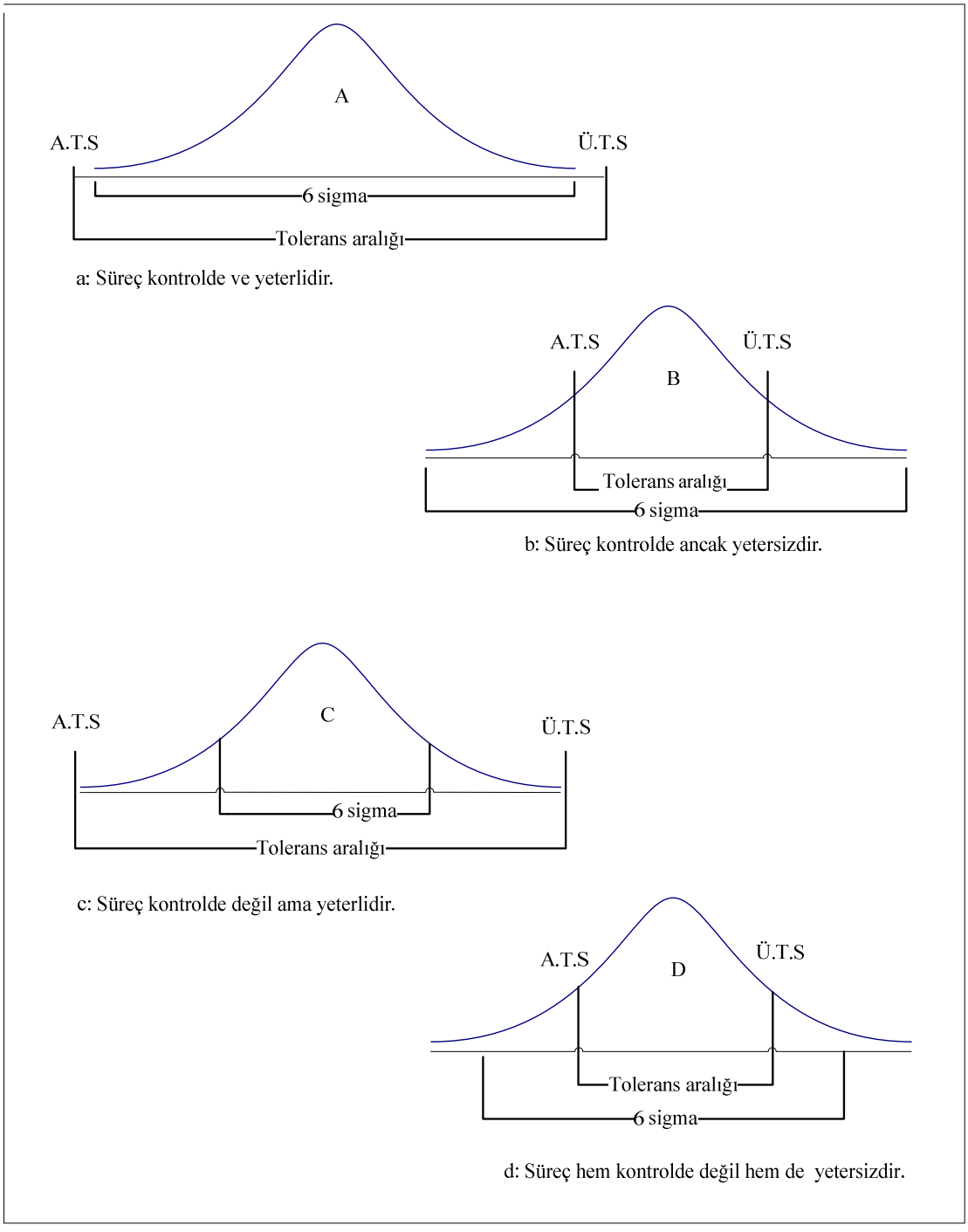
İstatistiksel süreç kontrolü, sürecin kontrolünde ve belirlenmiş spesifikasyonlar arasında devamını belirtir. Kontrolde devam eden süreçlerin otomatik olarak “yeterli” olduğu kabul edilemez. Herhangi bir süreçte şekil 3.2’de tanımlanan 4 değişik durum söz konusu olabilir (Baskan 1997);

a) Süreç kontroldedir ve aynı zamanda yeterlidir.

b) Süreç kontroldedir ama yeterli değildir.

c) Süreç kontrolde değildir ama yeterlidir.

d) Süreç hem kontrolde değildir hem de yeterli değildir.



Şekil 3.2 Sürecin "kontrolde" ve "yeterli" olması durumlarının şematik gösterimi (Baskan 1997)

Bu durumlar detaylı olarak incelendiğinde;

a) $6\sigma >$ (Ü.T.S-A.T.S)

Şekil 3.2 madde b ve d'de belirtilen sürecin yetersiz olduğu durumdur. Sürecin yetersiz olması şirketler taraından istenmez. Çünkü bazı parçaların ölçümleri alt veya üst tolerans sınırları dışında olduğundan hatalı parça üretilmiş olur. Süreç ya da işlem kontrol altında olsa dahi hatalı üretim kaçınılmazdır. Böyle durumlarda çözüm yolları aşağıdaki gibi sıralanabilir (Baskan 1997):

i) Olanaklı ise tolerans sınırlarını genişletmek,

ii) Süreç standart sapmasını azaltacak önlemler almak,

iii) Süreç ortalamasının tüm parçaların üst tolerans sınırı dışına taşacak şekilde yukarı itmek (Bu durumda hatalı parçaları hurdaya ayırma yerine kurtarma işlemine tabi tutmuş olurlar),

iv) Bunların hiçbirisi olanaklı değilse, hatalı parça üretimini kabullenip %100 muayene yapmak.

b) $6\sigma =$(Ü.T.S-A.T.S)

Bu kritik bir durumdur. Süreç dağılımının uç noktaları tolerans sınırları ile çakıştığında ve süreç kontrol altında olduğunda sorun yok denilebilir. Buna karşılık süreç ortalamasında ya da değişiminde olabilecek bir artış sorun yaratabilir. Bu durumda süreç ya da işlemi kontrol altında tutmak gerekir. Bu ek olarak tasarımda bir değişiklik yaparak toleransı artırmak, daha nitelikli işçi, malzeme ya da tezgah kullanarak sürecin dağılım varyansının azaltılması yoluna gidilebilir (Baskan 1997).

c) $6\sigma <$ (Ü.T.S-A.T.S)

Şekil 3.2 madde a ve c'de belirtilen en ideal olan durumdur. Kontrol sınırları, tolerans sınırlarından az olduğu için süreç ortalamasında küçük kaymalar olsa bile bu durum hatalı üretime neden olmayacaktır.

Süreç kontrol altında olduğunda bir noktanın kontrol sınırlarının dışına düşmesi olasılığı 1. tip hata yapma olasılığıdır. Benzer şekilde süreç kontrolden çıkarken bir noktanın kontrol sınırları içine düşme olasılığı sıfırdan büyüktür. Bu 2. tip hata olasılığı olarak bilinir. Bu iki hata birbirleri ile ilişkili olduğundan birindeki bir azalma diğerinde artma ile sonuçlanır (Baskan 1997).

3.1.2.2 R kontrol şeması

Açıklık (R) için kontrol şeması, kitlenin dağılımında zaman içinde bir değişiklik olup olmadığını belirlemek amacıyla kullanılır. Örneğin, mil yatağı aşınması, bir aletin parçalarından birinde gevşeme, operatörün ilgisinin azalması gibi örneklem birimlerinin farklılaşma ölçüsünün belirlenmesi amacıyla kullanılır. R kontrol şemasının merkez çizgisi R, kontrol sınırları ise $3\sigma_R$ genel formülü ile belirlenir. Buna göre,

i) standartlar belli iken kontrol sınırları:

Üst Kontrol Sınırı (Ü.K.S);

$\overline{R} + 3\sigma_R$

ve

Alt Kontrol Sınırı (A.K.S);

$\overline{R} - 3\sigma_R$

dır. Gelecek üretimin kontrolü için ya da standartlar belli iken kontrol sınırları aşağıdaki gibi elde edilir. Merkez çizgi, $E(R) = d_2\sigma$ dır. R nin standart sapması ise $\sigma_R = d_3\sigma$ dır. O halde kontrol sınırları:

$d_2\sigma \mp 3\sigma_R$ veya $\sigma(d_2 \mp 3d_3)$

olarak yazılabilir. Ancak $D_2 = d_2 + 3d_3$, $D_1 = d_2 - 3d_3$ katsayıları çizelge 3.1'deki örneklem ölçümüne göre belirlendiğinde kontrol sınırları;

Merkez Çizgisi (M.Ç);

$d_2\sigma$,

Üst Kontrol Sınırı (Ü.K.S);

$D_2\sigma$

ve

Alt Kontrol Sınırı (A.K.S);

$D_1\sigma$

olur (Baskan 1997).

ii) Standartlar belli olmadığında, geçmiş üretimin kontrolü için kontrol sınırları σ_R nin tahmin edicisi $\overline{R}/d_2$ den elde edilir (Baskan 1997). Böylece R kontrol şemasının kontrol sınırları,

Merkez Çizgisi (M.Ç);

$\overline{R}$,

Üst Kontrol Sınırı (Ü.K.S);

$D_4\overline{R}$

ve

Alt Kontrol Sınırı (A.K.S);

$D_3\overline{R}$

dır. Burada D_3 ve D_4 katsayıları,

$$D_4 = 1 + \frac{3d_3}{d_2}$$

ve

$$D_3 = 1 - \frac{3d_3}{d_2}$$

biçimindedir (Florac ve Carleton 2001).

3.1.2.3 σ kontrol şeması

R-kontrol şeması için merkez çizgi ve kontrol sınırlarının oluşturulmasına benzer şekilde S kontrol şemasının merkez çizgisi $\bar{s}$ ve kontrol sınırları ise $3\sigma_\sigma$ genel formülü ile belirlenir.

i) Standartlar belli iken σ nın beklenen değeri c_2 dir (Baskan 1997). Standart sapması ise;

$$\sigma_\sigma = \left[2(n-1) - 2nc_2^2\right]^{1/2} \times \frac{\sigma}{\sqrt{2n}}$$

dır. O halde kontrol sınırları $c_2\sigma \mp 3\sigma_\sigma$ ya da;

$\sigma\left(c_2 \mp \frac{3}{\sqrt{2n}}\left[2(n-1) - 2nc_2^2\right]^{1/2}\right)$ biçiminde yazılabilir.

Kontrol sınırları için katsayılar,

$$B_2 = c_2 + \frac{3}{\sqrt{2n}}\left[2(n-1) - 2nc_2^2\right]^{1/2} 1$$

ve

$$B_1 = c_2 - \frac{3}{\sqrt{2n}}\left[2(n-1) - 2nc_2^2\right]^{1/2}$$

dir. Böylece,

Üst Kontrol Sınırı (Ü.K.S);

$B_2\sigma$

ve

Alt Kontrol Sınırı (A.K.S);

$B_1\sigma$

olur (Baskan 1997).

ii) σ bilinmediğinde ise $\bar{s}/c_2$ den tahmin edilebilir (Baskan 1997). Böylece tahmin edilmiş kontrol sınırları,

$\bar{s}\left(1 \mp \frac{3}{\sqrt{2n}c_2}\left[2(n-1) - 2nc_2^2\right]^{1/2}\right)$ olarak yazılabilir.

Katsayılar,

$$B_4 = 1 + \frac{3}{\sqrt{2n}c_2}\left[2(n-1) - 2nc_2^2\right]^{1/2}$$

ve

$$B_3 = 1 - \frac{3}{\sqrt{2n}c_2}\left[2(n-1) - 2nc_2^2\right]^{1/2}$$

dir. Böylece,

Üst Kontrol Sınırı (Ü.K.S);

$B_4\bar{s}$

ve

Alt Kontrol Sınırı (A.K.S);

$B_3\bar{s}$

olur (Florac ve Carleton 2001).

Çizelge 3.2 Ölçülebilen veri tipi için kontrol şemaları formül özeti (Florac ve Carleton 2001)

	Merkez Çizgisi	Kontrol Limitleri		σ_x
X Ortalama ve R Kartları	$M.Ç = \bar{\bar{X}}$	$Ü.K.S = \bar{\bar{X}} + A_2\bar{R}$	$A.K.S = \bar{\bar{X}} - A_2\bar{R}$	$\frac{\bar{R}}{d_2}$
	$M.Ç = \bar{R}$	$Ü.K.S = D_4\bar{R}$	$A.K.S = D_3\bar{R}$	
X Ortalama ve S Kartları	$M.Ç = \bar{\bar{X}}$	$Ü.K.S = \bar{\bar{X}} + A_1\bar{s}$	$A.K.S = \bar{\bar{X}} - A_1\bar{s}$	$\frac{\bar{s}}{c_2}$
	$M.Ç = \bar{s}$	$Ü.K.S = B_4\bar{s}$	$A.K.S = B_3\bar{s}$	

3.1.3 Ölçülemeyen özellikler için kontrol şeması

Bazı kalite özellikleri sadece nitelikleri açısından incelenebilir. Yani her gözlenen parça iki sınıfa gerekli kalite koşullarına uyma ve uymama olarak ayrılabilir. Örneğin bir silindirde döküm sırasında olabilecek bir hatadan dolayı meydana gelen çatlak silindirin hatalı olarak ayrılmasına yol açar.

Nitel özelliklerin kontrolünde bireyler sadece hatalı veya hatasız olarak sınıflandırıldığında;

a) Her bir ürün üzerinde saptanan hatalar ürünün niteliğini belirliyor ise hatalı sayısı (np) veya hatalı oranı (p) kontrol şeması,

b) Ürünün hata olması sayılıyor ise c veya u kontrol şeması kullanılır.

Bir ürün ya da parçanın taşıdığı bir ya da daha fazla sayıda hata nedeniyle hatalı olarak nitelendirilir. Hatalı (deficiency) diye nitelendirilen ürünler daha önce belirtilmiş olan kalite spesifikasyonlarına bir kaç konuda uymayabilir. Hata (defect) ise kalite spesifikasyonuna aykırılıktır.

3.1.3.1 P kontrol şeması

P kontrol şemaları nitel olarak incelenen kalite özelliklerine ve geçer geçmez ölçüsü ile incelenen boyutlara uygulanır. Kontrol şemasının oluşturulmasında iki durum sözkonusudur.

a) P oranı bilindiğinde

Süreç kontrol altında olduğunda, süreçteki hatalı oranı bir P_0 değerine eşittir. Bu değer daha önceki gözlemlerden elde edilen deneyim ile elde edilebilir. Daha sonra değişmez n ölçümdeki her örneklem için ortalama bir np_0 sayısı bulunur.

Kontrol şemasının sınırları ardışık örneklemlerden gözlenen hatalı sayısı (h) ile ilişkilidir.

i) $n \leq 50$ olduğunda P kontrol şeması, Binom dağılımına yakınsar ve çizelge 3.3'te hazırlanmış olan değerler kullanılır. Bu durumda üst sınır değeri,

$$\sum_{h=0}^{L_k} p_0^{\,h}\left(1-p_0\right)^{n-h} = 0.999$$

biçimde tanımlanır (Baskan 1997).

ii) $P_0 \leq \%10$ olması durumunda ise Poisson dağılımına göre hazırlanmış çizelgeler kullanılır. Bu çizelgeler P_0'ın değil np_0 'ın fonksiyonu olarak L_k ve L_g değerlerini verir. Çizelge 3.3'deki L_k üst kontrol sınırı değeri ise,

$$\sum_{h=0}^{L_k} \frac{e^{-np_0}(np_0)^h}{h!} = 0.999$$

ilişkisi ile tanımlanır (Baskan 1997).

Çizelge 3.3 Kontrol ve gözetim üst sınırları (P_0, $n \leq 50$) (Baskan 1997)

Hatalı sayısı	n=10		n=15		n=20		n=30		n=40		n=50	
L_K vaya L_G	L_K için $\%p_0$	L_G için $\%p_0$	L_K için $\%p_0$	L_G için $\%p_0$	L_K için $\%p_0$	L_G için $\%p_0$	L_K için $\%p_0$	L_G için $\%p_0$	L_K için $\%p_0$	L_G için $\%p_0$	L_K için $\%p_0$	L_G için $\%p_0$
0	0.01	0.25	0.01	0.17	-	0.13	-	0.09	-	0.06	-	0.05
1	0.5	2.5	0.3	1.9	0.23	1.3	0.15	0.8	0.11	0.6	0.09	0.5
2	2.1	6.7	1.4	4.3	1	3.2	0.66	2.1	0.5	1.7	0.4	1.4
3	4.9	12.2	3.1	7.8	2.2	5.7	1.5	3.7	1.1	2.7	0.9	2.2
4	8.8	18.7	5.5	11.8	4	8.7	2.5	5.6	1.9	4.2	1.3	3.3
5	14.1	26.2	8.5	16.3	6.1	11.9	3.9	7.7	2.9	5.6	2.2	4.5
6	20.4	34.8	12.1	21.3	8.6	15.4	5.4	9.9	4	7.3	3.1	5.8
7	28.1	44.4	16.1	26.6	11.3	19.1	7.2	12.3	5.2	9	4.2	7.1
8	37.6	55.5	20.6	32.3	14.4	23.1	9	14.8	6.5	10.8	5.2	8.5
9	50.1	69.2	25.7	38.4	17.7	27.2	11	17.3	8	12.7	6.2	10
10	-	-	31.3	44.9	21.3	31.5	13.1	19.9	9.5	14.6	7.4	11.5
11	-	-	37.5	51.9	25.1	36.1	15.3	22.6	11.1	16.5	8.6	13.1
12	-	-	44.6	59.5	29.3	40.8	17.7	25.4	12.7	18.5	10	14.6
13	-	-	52.8	68.1	33.7	45.7	20.2	28.3	14.4	20.6	11.3	16.2
14	-	-	63	78.2	38.4	50.9	22.7	31.3	16.2	22.7	12.6	17.8
15	-	-	-	-	43.5	56.3	25.4	34.3	18.1	24.8	14	19.5
16	-	-	-	-	49.1	62.1	28.2	37.4	20	27	15.5	21.2

iii) $np_0 \geq 15$ olduğu durumda, Normal dağılım yaklaşımı kullanılır. N ölçümlü bir örneklemde hatalı sayısı için üst kontrol sınırları yaklaşık olarak;

$$Ü.K.S(L_k) = np_0 + 3np_0(1 - np_0)$$

ve

$$Ü.G.S(L_G) = np_0 + 2np_0(1 - np_0)$$

dır (Baskan 1997).

b) P oranı bilinmediğinde

Geçmiş üretimin hatalı oranı kontrol şeması düzenlenirken yapılacak ilk iş incelenen örneklem sayısını ve bulunan hatalı sayısını belirlemektir. Bundan sonra yapılacak işlem ise ortalama hatalı oranını bulmaktır. Bunun için her alt grubun p değeri hesaplanır. Sonra ortalama hatalı oranı ($\bar{p}$) bulunur. Bunun için ardışık k örneklemdeki toplam hatalı sayısı, bu k örneklemdeki toplam birim sayısına bölünür. Böylece,

$p = \frac{r}{n}$ olmak üzere,

$$\bar{p} = \frac{\sum r_i}{\sum n_i}\text{; i=1, 2, … , k}$$

dır. Burada, $\bar{p}$ değeri kontrol şemasının merkez çizgisidir. Kontrol sınırları ise;

Üst Kontrol Sınırı (Ü.K.S);

$$\bar{p} + 3\sqrt{\bar{p}(1-\bar{p})/n}$$

ve

Alt Kontrol Sınırı (A.K.S);

$$\bar{p} - 3\sqrt{\bar{p}(1-\bar{p})/n}$$

dır (Baskan 1997).

3.1.3.2 NP kontrol şeması

Ölçülemeyen kalite özelliklerinin kontrolü için kullanılan bir diğer kontrol aracı da hatalı sayısı kontrol şemasıdır. Bu şema işletmedeki personel için daha kolay anlaşılabilir olması nedeniyle çoğu kez diğerine yeğlenir. Çünkü kontrol sınırları belirlendikten sonra başka bir işleme gerek duyulmaksızın bir parti muayenesinde öngörülen hatalı parça sayısı şemaya doğrudan işlenebilir. Burada,

Merkez Çizgisi (M.Ç);

$n\bar{p}$,

Üst Kontrol Sınırı (Ü.K.S);

$$n\bar{p}+3\sqrt{n\bar{p}\left(1-\bar{p}\right)}$$

ve

Alt Kontrol Sınırı (A.K.S);

$$n\bar{p}-3\sqrt{n\bar{p}\left(1-\bar{p}\right)}$$

dır (Baskan 1997).

Eğer süreç veya kitle ortalaması hatalı oranı (P_0) önceden biliniyorsa bu değer hesaplamalarda kullanılır. Bilinmediğinde ise hatalı oranı kontrol şemalarında olduğu gibi örneklem verilerinden yararlanılarak tahmin edilir.

3.1.3.3 C kontrol şeması

Bu kontrol şeması birimdeki hatalar için bir kontrol kurmak amacıyla düzenlenir. Birim tek bir parça ürün, parçalar grubu, parçanın bir kısmı v.b sayılabilir. Burada önemli olan örneklem ölçümünün aynı olmasıdır. Bir hatanın ortaya çıkması olasılığı küçük ise C-kontrol şeması Poisson dağılımı üzerine kurulur. Hata sayısı (c), Poisson dağılımına uyduğunda bir parçada h tane hatanın bulunması olasılığı;

$$P(c,\lambda)=\frac{e^{-\lambda}\lambda^{c}}{c!}\text{; c=1,2,…}$$

dir.

i) λ bilindiğinde, birimdeki hataların ortalama değeri λ ve standart sapması $\sqrt{\lambda}$ dır. O halde C kontrol şeması için kontrol sınırları,

Üst Kontrol Sınırı (Ü.K.S);

$\lambda + 3\sqrt{\lambda}$

ve

Alt Kontrol Sınırı (A.K.S);

$\lambda - 3\sqrt{\lambda}$

biçiminde tanımlanabilir. Öte yandan, örneklem ölçümü büyük ve hata olma olasılığının küçük olduğu durumlarda, hata sayılarının dağılımı Normal dağılıma yakınsar.

ii) λ bilinmediğinde ise gözlenen verilerden λ 'nın tahmin edicisi

$\bar{c} = \frac{\sum c}{n}$ biçiminde elde edilir. Böylece kontrol sınırları,

Merkez Çizgisi (M.Ç);

$\bar{c}$,

Üst Kontrol Sınırı (Ü.K.S);

$\bar{c} + 3\sqrt{c}$

ve

Alt Kontrol Sınırı (A.K.S);

$\bar{c} - 3\sqrt{c}$

dır (Baskan 1997).

3.1.3.4 U kontrol şeması

Hata sayısını kontrol amacıyla kullanan C kontrol şeması için her alt grubun tek birimden oluşması (kumaş topu, şişe, uçak kanadı v.b) gerekir. Ancak bazı durumlarda alt gruplar birden fazla ve farklı sayılarda birimlerden oluşabilirler. Bu durumda U Kontrol şemaları kullanılır.

U kontrol şemalarının uygulamaları için birim başına gerçek hata sayısı U_0 bilindiğinde bu değer kontrol sınırlarını hesaplamada kullanılır. Bilinmediğinde ise tüm altgruplarda gözlenen hata sayısı alt grupları oluşturan birim sayısına bölünerek birim başına ortalama hata sayısı $U\left(\bar{c}\right)$ hesaplanır. Bu durumda, ortalama değeri ve kontrol sınırları sırasıyla,

$$\bar{u} = \frac{\sum c}{\sum n} \ ,$$

Üst Kontrol Sınırı (Ü.K.S);

$$\bar{u} + 3\sqrt{\frac{\bar{u}}{n}} \text{ veya } \bar{c} + 3\sqrt{\frac{\bar{c}}{n}}$$

ve

Alt Kontrol Sınırı (A.K.S);

$$\bar{u} - 3\sqrt{\frac{\bar{u}}{n}} \text{ veya } \bar{c} - 3\sqrt{\frac{\bar{c}}{n}}$$

formülleri kullanılır. Öte yandan alt grup büyüklükleri birbirinden farklı olduğuna göre bulunacak kontrol sınırlarının değişen n değerine göre değişeceği hatırlanmalıdır (Baskan 1997).

Çizelge 3.4 Ölçülemeyen veri tipi için kontrol şemaları formül özeti (Florac ve Carleton 2001)

	Merkez Çizgisi	Kontrol Limitleri	
p Kontrol Kartı	$M.Ç = \bar{p}$	Örneklem gözlem sayısının sabit olmadığı durumlarda	
		$Ü.Ç.S = \bar{p} + 3\frac{\sqrt{\bar{p}(1-\bar{p})}}{\sqrt{n_i}}$	$A.Ç.S = \bar{p} - 3\frac{\sqrt{\bar{p}(1-\bar{p})}}{\sqrt{n_i}}$
		Örneklem gözlem sayısı sabit ise (n)	
		$Ü.Ç.S = \bar{p} + 3\frac{\sqrt{\bar{p}(1-\bar{p})}}{\sqrt{n}}$	$A.Ç.S = \bar{p} - 3\frac{\sqrt{\bar{p}(1-\bar{p})}}{\sqrt{n}}$
np Kontrol Kartı	$M.Ç = \overline{np}$	$Ü.Ç.S = \overline{np} + 3\sqrt{\overline{np}(1-\bar{p})}$	$A.Ç.S = \overline{np} - 3\sqrt{\overline{np}(1-\bar{p})}$
c Kontrol Kartı	$M.Ç = \bar{c}$	$Ü.Ç.S = \bar{c} + 3\sqrt{\bar{c}}$	$A.Ç.S = \bar{c} - 3\sqrt{\bar{c}}$
u Kontrol Kartı	$M.Ç = \bar{u}$	Örneklem gözlem sayısının sabit olmadığı durumlarda	
		$Ü.Ç.S = \bar{u} + 3\sqrt{\frac{\bar{u}}{n_i}}$	$A.Ç.S = \bar{u} - 3\sqrt{\frac{\bar{u}}{n_i}}$
		Örneklem gözlem sayılarının ortalaması kullanıldığında	
		$Ü.Ç.S = \bar{u} + 3\sqrt{\frac{\bar{u}}{\bar{n}}}$	$A.Ç.S = \bar{u} - 3\sqrt{\frac{\bar{u}}{\bar{n}}}$
		Örneklem gözlem sayısı sabit ise (n)	
		$Ü.Ç.S = \bar{u} + 3\sqrt{\frac{\bar{u}}{n}}$	$A.Ç.S = \bar{u} - 3\sqrt{\frac{\bar{u}}{n}}$

3.2 Yazılım Sistemleri Güvenirlik Modelleri

Özellikle karmaşık yapılı yazılım sistemlerinde, yazılımın %100 hatasız çalışmasını sağlamak için gerçekleştirilen yazılım test faaliyetleri oldukça maaliyetlidir. Bu nedenle şirketler belirlenmiş zaman aralıklarında kod gözden geçirme ve fonksiyonel test faaliyetlerini optimum seviyede gerçekleştirerek, teslim edecekleri yazılım sisteminin tüm gereksinimleri karşıladığından ve büyük (majör) hatalı ürün teslim etmediklerinden

emin olmak isterler. Bu nedenle müşteriye teslim edilmeden önce yazılım sisteminin güvenirliği ve hata sayısı, yapılan iç gözden geçirme ve test faaliyetlerinde çıkan hatalar girdi olarak değerlendirilerek tahmin edilmekte ve çıkan risk durumlarına göre şirket tarafından önleyici işlem alınmaktadır.

Yazılım sistemleri güvenirlik analizi çalışmaları 30 yılı aşkın süredir devam etmektedir. Bu süre içerisinde yazılımın güvenirliğini ve kalan hata sayısını tahmin eden modeller geliştirilmiştir.

Temel olarak iki çeşit Yazılım Güvenirlik modeli vardır. Bunlar, deterministik ve olasılıksal olarak ikiye ayrılmaktadır.

Deterministik modeller programdaki işletmen (operator) ve işlemci (operands)'leri kullanır. Deterministik tipte performans ölçümleri programın yazılış yapısı analiz edilerek elde edilir, bu nedenle rastgele olayları içermezler.

Deterministik modellerden en iyi bilinen iki tanesi: Halstead's yazılım metriği ve McCabe's cyclomatic karmaşıklık metriğidir.

Halstead's yazılım metriği programda hata sayısını tahmin etmek için kullanılırken McCabe's cyclomatic karmaşıklık metriği, yazılımdaki kalan hata sayısını tahmin etmek için modelin üst sınırını belirler.

Bu çalışma kapsamında olasılıksal yazılım güvenirlik modelleri detaylı olarak incelenecektir.

Olasılıksal yazılım güvenirlik modelleri farklı gruplar halinde sınıflandırılır:

i) Hata kaynağı (Error Seeding) modelleri,

ii) Hatalı oranı (Failure Rate) modelleri,

iii) Güvenirlik geliştirme (Reliability Growth) modelleri,

iv) Homojen olmayan Poisson süreci modelleri

(Pham 2000).

3.2.1. Hata kaynağı (Error Seeding) modelleri

Hata kaynağı modelleri, çok aşamalı örnekleme tekniklerini kullanarak hata sayısını tahmin eder. Hatalar doğal hata (indigenous/inherent errors) ve kaynaklanmış hata (induced errors) olarak ikiye ayrılır. Sayısı bilinmeyen doğal hatalar kaynaklanmış hatalardan tahmin edilir ve bu iki tip hatanın oranı, hataları düzeltilmiş veriden (debugging data) elde edilir.

3.2.1.1 Mill'in Hata Kaynağı Modeli

Mill'in hata kaynağı modeli (Mills 1970), programdaki hata sayısını tahmin etmek için geliştirilmiştir. Düzeltilmiş veri (debugging data), hem doğal hatayı hem de kaynaklanmış hatayı içerir ve bu veriden doğal hata sayısı tahmin edilebilir. Eğer doğal ve kaynaklanmış hatanın tespit edilme olasılığı eşit ise r tane yok edilmiş hatadan k tanesinin doğal hata olma olasılığı Hipergeometrik dağılım gösterir. Hipergeometrik dağılım,

$$P(k; N, n_1, r) = \frac{\binom{n_1}{k}\binom{N}{r-k}}{\binom{N+n_1}{r}} \text{; k=1, 2, … , r}$$

biçimindedir, burada

N : Doğal hataların toplam sayısı

n_1 : Kaynaklanmış hataların toplam sayısı

r : Düzeltme sırasında yok edilen toplam hata sayısı

k : Yok edilen r tane hata içindeki doğal hataların toplam sayısı

r-k : Yok edilen r tane hata içindeki kaynaklanmış hataların toplam sayısıdır.

n_1, r ve k parametreleri bilindiğinde N parametresinin en çok olabilirlik tahmini,

$$\hat{N} = [N_0 + 1]$$

biçimdedir (Huang 1984), burada

$$N_0 = \frac{n_1(r-k)}{k} - 1$$

dır. Eğer N_0 bir tam sayı ise hem N_0 hem de N_0+1, N parametresinin en çok olabilirlik tahmin edicisidir.

Mill'in hata kaynağı modelinin dezavantajları vardır. Bunlar; doğal hataların zorluk derecesinin, konumlarının ve tiplerinin karar verilmesinde yaşanan zorluklar ve tespit edilen doğal hata ile yaklaşık aynı oranda kaynaklanmış hata tespit edilmesinin beklenmesidir.

Yazılım hatalarının tespit edilmesi, test süresini artırmakta ve bu da maliyeti yükseltmektedir.

3.2.1.2 Cai Modeli

Cai modeli (Cai 1998), yazılımı Parça 0 ve Parça 1 olarak ikiye ayırarak Mill'in hata kaynağı modelini değiştirmiştir. Bu model, yazılımda geride kalan hataların sayısını tahmin etmek için kullanılır. Modelin uygulamasında aşağıdaki varsayımlar geçerlidir:

i) Parça 0 olarak ayrılmış yazılımda N_0 ve Parça 1 olan yazılımda N_1 olmak üzere toplam $N=N_0+N_1$ tane hata vardır.

ii) Her hatanın tespit edilme olasılığı aynıdır.

iii) Hata tespit edildiği an yok edilir.

iv) Her bir zamanda sadece bir hata çözülür ve yeni hata tanımlanmaz.

v) Yok edilen hataların sayısı n dir.

t_i, i inci zamanda yok edilen hatanın zamanı göstersin ve Y_i rastgele değişken olmak üzere;

$$Y_i = \begin{cases} 0 \ , \text{i inci zamanda tespit edilen hata Parça 0 `da ise} \\ 1 \ , \text{i inci zamanda tespit edilen hata Parça 1 `de ise} \end{cases}$$

biçimindedir.

$(t_i, t_{i-1}]$ zaman aralığında Parça j de tespit edilen hata sayısı $N_j(i)$ olarak tanımlansın (i=0, 1, 2, … , n ve j=0 veya 1 dir). Böylece,

$$N_0(i) = N_0 - i + \sum_{j=0}^{i} y_j$$

ve

$$N_1(i) = N_1 - \sum_{j=0}^{i} y_j$$

dır.

$(t_i, t_{i-1}]$ zaman aralığında Parça j de tespit edilen hataların olasılığı $p_j(i)$ olarak tanımlansın (i=0, 1, 2, … , n ve j=0 veya 1) dir. Böylece,

$$p_0(i) = \frac{N_0(i)}{N_0(i) + N_1(i)} = \frac{N_0 - i + \sum_{j=0}^{i} y_j}{N_0 + N_1 - i}$$

ve

$$p_1(i) = \frac{N_1(i)}{N_0(i) + N_1(i)} = \frac{N_1 - \sum_{j=0}^{i} y_j}{N_0 + N_1 - i}$$

olarak elde edilir.

N_0 ve N_1 parametrelerinin en çok olabilirlik tahmin yöntemine göre değerleri,

$$\sum_{i=1}^{n} \frac{1 - y_i}{N_0 - i + 1 + \sum_{j=0}^{i-1} y_j} = \frac{1}{N_0 + N_1 - i + 1}$$

ve

$$\sum_{i=1}^{n} \frac{y_i}{N_1 - \sum_{j=0}^{i-1} y_j} = \frac{1}{N_0 + N_1 - i + 1}$$

biçimindedir (Cai 1998).

3.2.1.3 Hipergeometrik dağılım modeli

Tohma modeli (Tohma vd. 1991), teste başlarken veya düzeltme sürecinde (debugging process) var olan hataları Hipergeometrik dağılıma dayanarak tahminin yapılmasını öneren bir model geliştirmiştir.

t_1, t_2 ,…,t_{i-1} test durumuna kadar tespit edilen hataların toplam sayısı C_{i-1} ve t_i inci test durumunda tespit edilen yeni hataların sayısı N_i ise, modelin uygulamasında aşağıdaki varsayımlar geçerlidir:

i) Test fazı başladığında program m tane hatayı içerir.

ii) Bir günde veya bir haftada gerçekleştirilen test durumlarının sayısının toplamı, testin tamamını tanımlar. Test durumları t_i olarak gösterilirler ve i=1, 2, … , n dir.

iii) Test durumları koşulurken bu arada hata çözümlemesi yapılmaz.

iv) Farklı test durumlarında aynı hatalar tespit edilebilir.

t_i test durumunda tespit edilen hataların sayısı W_i olduğunda, W_i tane hatanın t_{i-1} inci test durumunda tespit edilen C_{i-1} tane hata içinde çıkmış olma olasılığı vardır. Bu durumda W_i tane hata içinde şu kategoride olan hatalar vardır:

i) Yeni tespit edilen hatalar,

ii) Tekrar tespit edilen hatalar.

Eğer n_i, N_i örnekleminin gözlem değerleri ise $n_i \leq W_i$ olduğu gözlemlenir. t_i parametresi, test durumunda yeni tespit edilen hataların sayısı (N_i) Hipergeometrik dağılımı gösteriyorsa, W_i hataları arasında yeni tespit edilen n_i tane hatanın tespit edilme olasılığı,

$$P(N_i = n_i) = \frac{\binom{m - C_{i-1}}{n_i}\binom{C_{i-1}}{W_i - n_i}}{\binom{m}{W_i}}$$

biçiminde elde edilir, buradan

$$C_{i-1} = \sum_{k=1}^{i-1} n_k, \quad C_0 = 0, \quad n_0 = 0$$

ve

$$\max\left(0,\ W_i - C_{i-1}\right) \leq n_i \leq \max\left(W_i, m - C_{i-1}\right)$$

dır.

N_i Hipergeometrik dağılıma sahip olduğundan, $[t_{i-1}, t_i]$ zaman aralığında yeni hataların tespit edilmesinin beklenen değeri,

$$E(N_i) = \frac{(m - C_{i-1})W_i}{m}$$

ve C_i nin beklenen değeri,

$$p_i = \frac{W_i}{m};\ \ i = 1, 2, \ldots$$

olmak üzere,

$$E(C_i) = m\left[1 - \prod_{j=1}^{i}(1 - p_j)\right]$$

dir.

3.2.2. Hatalı oranı (Failure Rate) modelleri

Hatalı oranı model grupları temel olarak aşağıdaki gibidir:

a) Jelinski ve Moranda Modeli (Jelinski ve Moranda 1972)
b) Schick ve Wolverton Modeli (Schick ve Wolverton 1978)
c) Jelinski-Moranda geometrik modeli (Moranda 1979)
d) Moranda Geometrik Poisson modeli (Littlewood 1979)
e) Değiştirilmiş Schick ve Wolverton Modeli (Sukert 1977)
f) Goel ve Okumoto modeli (Goel ve Okumoto 1970a)

Bu model grupları, programın hatalı olma zaman aralıklarında hatalı oranının nasıl değiştirdiği ile ilgili yapılan çalışmalardır. Programdaki mevcut hataların sayısı değiştikçe hatalı oranı da değişecektir. Programdaki hata sayısının dağılımı kesikli bir

fonksiyon olduğundan, hatalı oranı da sürekli olmayan zamanlarda (failure times) kesikli dağılımdır.

3.2.2.1 Jelinski-Moranda Modeli

Jelinski-Moranda (J-M) modeli (Jelinski ve Moranda 1972), güvenirlik analizi modellerinin en önce oluşturulanlardan biridir. Diğer oluşturulan modeller bu temel modelden yararlanmışlardır. Modelin uygulamasında aşağıdaki varsayımlar geçerlidir:

i) Program başlangıçta N sayıda bilinmeyen hata içerir.

ii) Programdaki her bir hata bağımsızdır ve test fazı süresince eşit olasılıkta hatalılık durumuna neden olur.

iii) Muhtemel hatanın oluşmasındaki zaman aralıkları birbirinden bağımsızdır.

iv) Hata oluştuğu zaman ilgili hata kesinlikle yok edilir ve bulunan hatanın çözümlenmesi sırasında yeni hata tanımlanmaz.

v) Hataların tespit edilme zaman aralıklarındaki hatalılık oranı sabittir ve bu programdaki mevcut kalan hata sayısı ile orantılıdır.

i inci hata aralığındaki hatalı oranı,

$\lambda(t_i) = \phi(N - (i-1))$, i=1,2,…N

biçimindedir, burada

ϕ : olabilirlik sabiti

N : Programın başındaki hata sayısı

t_i : i inci ve (i-1) inci hata arasındaki zamandır.

Buna göre, ilk hatalı oranı fonksiyonu;

$$\lambda(t_1) = \phi(N)$$

dır. 1. hatadan sonra hata yoğunluğu azalır,

$$\lambda(t_2) = \phi(N-1)$$

ve olasılık yoğunluk fonksiyonu ;

$$f(t_i) = \lambda(t_i) e^{-\int_0^{t_i} \lambda(x_i)dx_i}$$

$$= \phi[N-(i-1)] e^{-\int_0^{t_i} \lambda(x_i)dx_i} ,$$

$$= \phi[N-(i-1)] e^{-\phi(N-(i-1))t_i}$$

dır. Kümülatif dağılım fonksiyonu ve güvenirlik fonksyonu sırasıyla;

$$F(t_i) = 1 - e^{-\phi(N-i+1)t_i} ,$$

$$R(t_i) = e^{-\phi(N-i+1)t_i}$$

biçimindedir.

Hatalı veri setinin $\{t_1, t_2, ..., t_n\}$ olduğu varsayıldığında, N ve ϕ parametrelerinin en çok olabilirlik tahmin edicileri,

$$\hat{\phi} = \frac{n}{\hat{N}\left(\sum_{i=1}^{n} t_i\right) - \sum_{i=1}^{n} (i-1)t_i}$$

ve

$$\sum_{i=1}^{n} \frac{1}{\hat{N} - (i-1)} = \frac{n}{\hat{N} - \left(1 / \sum_{i=1}^{n} t_i\right)\left(\sum_{i=1}^{n} (i-1)t_i\right)}$$

eşitliklerinin çözülmesi ile elde edilir.

3.2.2.2 Schick-Wolverton Modeli

Schick-Wolverton (S-W) (Schick ve Wolverton 1978) modeli, J-M modeli değiştirilerek elde edilmiştir. Jelinski-Moranda (J-M) modelinden tek farkı, i inci zaman aralığındaki hatalı oranı, t_i test fazı arttıkça i inci zaman aralığındaki hatalı oranı da artar varsayımıdır.

(i-1) inci ve i inci zaman aralığı arasındaki program hatalı oranı fonksiyonu,

$$\lambda(t_i) = \phi(N - (i-1))\, t_i$$

biçimde ifade edilir. Eşitlikteki parametre tanımları (ϕ, N), Jelinski-Moranda (J-M) modeli ile aynıdır.

Olasılık yoğunluk fonksiyonu ve güvenirlik fonksiyonu sırasıyla;

$$f(t_i) = \phi[N - (i-1)]t_i e^{-\frac{[N-(i-1)]t_i^2}{2}}; \text{ i=1, 2, …, N}$$

ve

$$R(t_i) = e^{-\frac{\phi(N-i+1)t_i^2}{2}}$$

biçimindedir. N ve ϕ parametrelerinin en çok olabilirlik tahmin edicileri,

$$\hat{\phi} = 2\sum_{i=1}^{n} \frac{1}{\left[\hat{N} - (i-1)\right]T}$$

ve

$$\hat{N} = \frac{2n}{\hat{\phi}T} + \frac{\sum_{i=1}^{n} (i-1)t_i^2}{T}$$

eşitliklerinin çözülmesi ile elde edilir. Burada,

$$T = \sum_{i=1}^{n} t_i^2$$

dır.

3.2.2.3 Jelinski-Moranda Geometrik Modeli

J-M modeli (Moranda 1979), programdaki başlangıç hatalı oranının sabit olduğunu (D) ve zamanla geometrik olarak azaldığı varsayımına göre geliştirilmiştir. i inci hata zaman aralığındaki güvenirlik fonksiyonu ve hata oranı sırasıyla,

$$\lambda(t_i) = Dk^{\,i-1}$$

ve

$$R(t_i) = e^{-Dk^{\,i-1}t_i}$$

biçimindedir. Burada,

D : programdaki başlangıç hatalı oranı

k : geometrik fonksiyonun parametresi (0<k<1)

dır. Eğer zaman aralığında birden fazla hatanın giderilmesine izin verilirse hata oranı ve güvenirlik fonksiyonu,

$$\lambda(t_i) = Dk^{\,n_{i-1}}$$

ve

$$R(t_i) = e^{-Dk^{\,n_{i-1}} t_i}$$

biçiminde elde edilir, burada n_{i-1}: (i-1) inci zaman aralığına kadar tespit edilen toplam hata sayısıdır.

3.2.2.4 Moranda Geometrik Poisson Modeli

Moranda Geometrik Poisson modeli (Moranda 1975), hata aralıklarını T ve 2T olarak eşit uzunlukta sabitlemiş ve i zaman aralığında meydana gelen hata sayısı N_i, Dk^{i-1} ortalamasıyla Poisson dağılımı gösterdiğini varsaymıştır. i inci zaman aralığında m tane hatanın çıkma olasılığı,

$$P(N_i = m) = \frac{e^{-Dk^{i-1}} (Dk^{i-1})^m}{m!}$$

biçiminde tanımlanır.

3.2.2.5 Değiştirilmiş Schick-Wolverton Modeli

Sukert modeli (Sukert 1977), her bir zaman aralığında birden fazla hatanın tespit edilmesi durumuna göre S-W modelini farklılaştırmıştır. Bu durumda hatalı oranı fonksiyonu ve güvenirlik fonksiyonu sırasıyla;

$$\lambda(t_i) = \phi(N - n_{i-1})\ t_i$$

ve

$$R(t_i) = e^{-\phi[N-n_{i-1}]\frac{t_i^2}{2}}$$

biçiminde elde edilir, burada n_{i-1}: (i-1) inci hatalı zaman aralığında tespit edilen toplam hata sayısıdır.

3.2.2.6 Goel-Okumoto Modeli

Goel-Okumoto modeli (Goel ve Okumoto 1979b), J-M modelinin farklılaştırılması ile elde edilmiştir. Hata meydana geldiğinde, hatanın p olasılığı ile yok edildiğini varsayar. Düzeltme sürecinin tamamlanmadığı durumda (imperfect debugging), i inci hata zaman aralığındaki hatalı oran fonksiyonu;

$$\lambda(t_i) = \phi(N - p(i-1))$$

$$= p\phi\left(\frac{N}{p} - (i-1)\right) = \phi'[N' - (i-1)]$$

biçimindedir, burada

$$\phi' = p\phi,\ \ N' = \frac{N}{p}$$

dır ve güvenirlik fonksiyonu;

$$R(t_i) = e^{-\phi[N-p(i-1)]t_i}$$

dır.

3.2.3. Güvenirlik geliştirme (Reliability Growth) modelleri

Güvenirlik geliştirme modeli programın, test fazı sürecindeki test durumu veya fonksiyon zamanı başına düşen hatalı oranını veya güvenirliliğini tahmin eder.

3.2.3.1 Wall ve Ferguson Modeli

Wall ve Ferguson modeli (Wall ve Ferguson 1977), test fazı süresince hatalı oranı tahmininde Weibull modeline benzer bir model önermiştir. t zamanında toplam hata sayısı $m(t)$ ise,

$$m(t) = \alpha_0 [b(t)]^{\beta}$$

dır, burada α_0 ve β parametreleri bilinmeyen parametredir. b(t) fonksiyonu, test durumu sayısından veya toplam test süresindir. t zamanında hatalı oranı fonksiyonu:

$$\lambda(t) = m'(t) = \alpha_0 \beta b'(t)[b(t)]^{\beta-1}$$

biçimindedir.

Wall ve Ferguson (1977) bu modeli farklı programların hata verileri üzerinde uygulanmış ve tahmin değerleri ile gerçek değerlerin tutarlı olduğunu gözlemlemişlerdir.

3.2.4. Homojen olmayan Poisson süreci modelleri

Yazılım güvenirlik mühendisliği alanında yapılmış olan araştırmalar sonucu, yazılımın güvenirliliği değerlendirmelerinde, homojen olmayan Poisson süreci modelleri önerilmiştir. Bu modellerin, devam eden süreç olan hata düzeltme sürecini de

(debugging process) dikkate alarak yazılım güvenirliliğinin yorumlanmasında faydalı ve gerçekçi modeller olduğu sonucuna varılmıştır. Homojen olmayan Poisson süreci modelleri, bellirli bir zamana kadar kümülatif hata sayısı gözlemlenmiş bir programın sonraki zaman sürecinde çıkacak hatanın ortalama değer fonksiyonunu tahmin eder. Yazılım güvenirliliği tahmini, ortalama değer fonksiyonu (mean value function) ve modellerdeki parametre değerleri tahmini en çok olabilirlik yöntemi ile yapılır. Homojen olmayan Poisson sürecinde hem hata miktarı fonksiyonu, hem de hata tespit etme oranı zamana bağlıdır ve matematiksel olarak ortalama değer fonksiyonu olarak ifade edilir (Xie 1991).

Notasyon:

m(t) : t zamanında tespit edilmesi tahmin edilen hata sayısı (mean value function)

a(t) : hata miktarı fonksiyonu, yazılımdaki t zamanında tespit edilmiş hem doğal hata hem de kaynaklanmış hata sayısını içeren toplam hata sayısı

b(t) : t zamanında hata tespit etme oranı

N(t) : t zamanında tespit edilen yazılım hatasının kümülatif sayısını gösteren rastgele değişkendir

y(t) : N(t)'nin gerçek değerleridir. (y_i:=y(t_i))

s_j : j inci hatanın tespit edildiği gerçek zamandır

R(s/t) : son hatanın tespit edildiği t zamanından sonraki, (t, t+s) zaman aralığındaki güvenirlik değeridir.

Yazılım güvenirlik tahmininde parametre tahmini önemlidir. Bilinmeyen parametrelerin tahmini en yaygın olarak kullanılan en çok olabilirlik tahmin yöntemi ile yapılacaktır. Hata verisinin Tip 1 ve Tip 2 olmak üzere farklı iki yöntemde toplanması durumunda, bunların en çok olabilirlik fonksiyonları aşağıdaki biçimde tanımlanır.

i) Tip 1 Veri: Aralık tanımında toplanmış veridir. Verinin (0, t_i) i=1, 2, ... , n ve $0<t_1<t_2<\ldots<t_n$ zaman aralığındaki tespit edilen kümülatif hata sayısı y_i olduğununda logaritmik olabilirlik fonksiyonu ,

$$LogL = \sum_{i=1}^{n} (y_i - y_{i-1}) \log[m(t_i) - m(t_{i-1})] - m(t_n)$$

biçimindedir. En çok olabilirlik fonksiyonunun, $\theta = (\theta_1, \theta_2, ..., \theta_n)$ bilinmeyen parametreleri,

$$0 = \sum_{i=1}^{n} \frac{\frac{\partial}{\partial\theta} m(t_i) - \frac{\partial}{\partial\theta} m(t_{i-1})}{m(t_i) - m(t_{i-1})} (y_i - y_{i-1}) - \frac{\partial}{\partial\theta} m(t_n)$$

ve

$$\lambda(t) = \frac{\partial}{\partial t} m(t)$$

eşitliklerinin çözülmesi ile elde edilir.

Θ bilinmeyen her parametrenin temsilcisi olarak belirlenmiştir ve gözlemlenen (t_i, y_i) i=1, 2, … , n verisi kullanılarak, $m(t_i)$ ortalama değer fonksiyonunu t_i zamanında i=n+1, n+2,… tahmin etmek için kullanılır.

ii) Tip 2 Veri: Belirli zaman tanımında toplanmış veridir. Bu veri, hatanın meydana geldiği tanımlanmış olan zamanlarda toplanmıştır. j inci hatanın meydana geldiği zaman s_j (j=1, 2, … , n) ise ve toplamda n tane hata zamanı kayıtlı ise ($0 \leq s_1 \leq s_2 \leq ... \leq s_n$), verinin logaritmik olabilirlik fonksiyonu;

$$LogL = \sum_{i=1}^{n} \log[\lambda(s_i)] - m(s_n)$$

biçimindedir. En çok olabilirlik fonksiyonunun, $\theta = (\theta_1, \theta_2, ..., \theta_n)$ bilinmeyen parametreleri,

$$0 = \sum_{i=1}^{n} \frac{\frac{\partial}{\partial \theta} \lambda(s_i)}{\lambda(s_i)} - \frac{\partial}{\partial \theta} m(s_n)$$

ve

$$\lambda(t) = \frac{\partial}{\partial t} m(t)$$

eşitliklerinin çözülmesi ile elde edilir. Θ bilinmeyen her parametrenin temsilcisi olarak belirlenmiştir.

Bu çalışma kapsamında tanımlanan modellerin hangisinin daha iyi tahmin sonucu verdiğini anlamak için, yaygın olarak kullanılan dört tane kriter vardır. Bunlar sırasıyla,

i) Hata kareler toplamı (HKT)

$$HKT = \sum_{j=1}^{k} \sum_{i=1}^{n} \left[y_{ij} - \hat{m}_j(t_i) \right]^2$$

dır. Burada, y_{ij}; t_i zamanında gözlemlenen j tip hatanın toplam sayısıdır ve $m_j(t_i)$; t_i zamanında tahmin edilmiş j tip hatanın kümülatif sayısıdır (i=1, 2, … , n ve j=1, 2, … , k).

ii) Hata kareler ortalaması (HKO)

$$HKO = \frac{\sum_{j=1}^{k} \sum_{i=1}^{n} \left(m_j(t_i) - y_{ij} \right)^2}{kn - N} \quad ,$$

iii) Akaike bilgi kriteri (ABK)

ABK= - 2× log(en yüksek seviyedeki olabilirlik fonksiyonu) + 2N

dır. Burada N modeldeki parametre sayısını gösterir.

iv) Öngörü oranı riski (Predictive Ratio Risk-PRR)

$$PRR = \sum_{j=1}^{k} \sum_{i=1}^{n} \left(\frac{m_j(\hat{t}_i) - y_{ij}}{m_j(\hat{t}_i)} \right)^2$$

dır. Burada, y_{ij}; t_i zamanında gözlemlenen j tip hatanın toplam sayısıdır ve $m_j(t_i)$; t_i zamanında tahmin edilmiş j tip hatanın kümülatif sayısıdır (i=1, 2, … , n ve j=1, 2, … , k).

Bu kriterlerden hata kareler ortalaması (HKO); modeldeki parametre sayısı (N) ve gözlem sayısını (n) dikkate alarak tahmin edilen model ile gerçek verinin uzaklığını ölçer. Akaike bilgi kriteri (ABK), modelin olabilirlik fonksiyonunu maksimize etme kabiliyetini ölçer ve modelde parametre sayısı arttıkça model daha iyi sonuç verir. Öngörü oranı riski (PRR) gerçek veri ile tahmin edilen model arasındaki uzaklığını ölçer.

Aynı veri setini girdi olarak alan modelerde PRR, HKO, HKT ve ABK kriterler değerinin küçük olduğu model daha iyi tahmin sonucu verir.

3.2.4.1 Üstel homojen olmayan Poisson süreci Goel-Okumoto modeli

Goel Okumoto modeli aşağıdaki varsayımlara dayanır:

i) Programdaki tüm hatalar, hatayı tespit etme yönteminden/bakış açısından bağımsızdır.
ii) Hataların meydana gelme veya tespit edilme olasılığı b(t) sabittir.
iii) Tespit edilmiş hatalar, bir sonraki test durumundan önce yok edilir.
iv) Hata her tespit edilmesinde hemen çözülür ve yeni hata tanımlanmaz. Yani a(t) fonksiyonu sabittir.

Goel- Okumoto modeli (Goel ve Okumoto 1979a) ortalama değer fonksiyonu;

$$m(t) = a\left(1 - e^{-bt}\right)$$

biçimindedir.

Tip 1 veri için modeldeki a ve b parametrelerinin en çok olabilirlik tahmini (EÇOT) yöntemine göre elde edilmiş eşitlikleri,

$$\hat{a} = \frac{y_n}{\left(1 - e^{-\hat{b}t_n}\right)}$$

ve

$$\frac{y_n t_n e^{-\hat{b}t_n}}{1 - e^{-\hat{b}t_n}} = \sum_{k=1}^{n} \frac{\left(y_k - y_{k-1}\right)\left(t_k e^{-\hat{b}t_k} - t_{k-1} e^{-\hat{b}t_{k-1}}\right)}{\left(e^{-\hat{b}t_{k-1}} - e^{-\hat{b}t_k}\right)}$$

biçimindedir. Eşitlikler aynı anda çözüldüğünde, parametrelerin tahmin değerleri elde edilir.

Tip 2 veri için a ve b parametrelerinin en çok olabilirlik tahmin edicisine göre elde edilen eşitlikleri,

$$\hat{a} = \frac{n}{\left(1 - e^{-\hat{b}s_n}\right)}$$

ve

$$\frac{n}{\hat{b}} = \sum_{i=1}^{n} s_i + \frac{n s_n e^{-\hat{b}s_n}}{1 - e^{-\hat{b}s_n}}$$

biçimindedir. Eşitlikler aynı anda çözüldüğünde, parametrelerin tahmin değerleri elde edilir.

EÇOT metodu ile tahmin edilen $\hat{a}$ *ve* $\hat{b}$ ile gösterilirse ortalama değer fonksiyonu ve güvenirlik fonksiyonu tahmin edicisi;

$$\hat{m}(t) = \hat{a}\left(1 - e^{-\hat{b}t}\right)$$

ve

$$\hat{R}(x \mid t) = e^{-\hat{a}\left[e^{-\hat{b}t} - e^{-\hat{b}(t+x)}\right]}$$

biçimindedir.

3.2.4.2 Homojen olmayan S şekilli Poisson süreci modelleri

Homojen olmayan S şekilli modellerde hatanın tespit edilme oranı zaman ile değişir. Test sürecinin başlaması ile en yüksek değerlerine ulaşır ve sonrasında üstel olarak azalır. Başka bir ifade ile, test fazının başında bazı hataların ortaya çıkması diğer hatalar tarafından gizlenir ve örtülmüş olan hatalar diğer hatalar yok edilmeden tespit edilemez. Yamada; yazılım test sürecinin, testçilerin ürünü, çevreyi ve yazılımın özelliklerini öğrendikleri bir süreç olduğunu belirtmiştir (Yamada vd. 1984).

Homojen olmayan S şekilli Poisson süreci aşağıdaki varsayımlara dayanır:

i) Hatalar arasında tespit edilme oranı farklılık gösterir.

ii) Yazılımda hata her zaman meydana gelir, tespit edilen hatalar hemen yok edilir ve yeni hata tanımlanmaz.

Ortalama değer fonksiyonu (Ohba 1984);

$$m(t) = a\left[1 - e^{-\int_0^t b(u)du}\right]$$

dır, burada,

a : test sürecinden önce tahmin edilen toplam hata sayısı,

b(t) : hata başına düşen hata tespit etme oranı,

m(t) : t zamanında tespit edilmesi beklenen hata sayısıdır.

S şekilli modeller, bükümlü S şekilli modeller ve Geciktirilmiş S şekilli modeller olarak incelenir.

a) Bükümlü S şekilli modeller (Inflection S shaped models)

Bükümlü S şekilli modellerde aşağıdaki varsayımlar geçerlidir:

i) Bazı hatalar diğer hatalar yok edilmeden tespit edilmezler.
ii) Herhangi bir zamanda hata tespit etme olasılığı, programdaki mevcut olan tespit edilmesi muhtemel hata sayısı ile orantılıdır.
iii) Tespit edilmesi muhtemel her hatanın hatalı oranı sabit ve tanımlıdır.
iv) Ayrılmış hatalar tamamen yok edilebilirler.

Bükümlü S şekilli model ortalama değer fonksiyonu;

$$m(t)=\frac{a}{1+\beta e^{-bt}}\left(1-e^{-bt}\right)$$

biçimindedir. Burada hata tespit etme oranı,

$$b(t)=\frac{b}{1+\beta e^{-bt}}$$

biçiminde verilir. Bükümlü S şekilli modelin hata yoğunluk fonksiyonu,

$$\lambda(t)=\frac{ab(1+\beta)e^{-bt}}{\left(1+\beta e^{-bt}\right)^2}$$

dır ve t zamanında kalan hataların tahmini sayısı,

$$\hat{m}(\infty)-\hat{m}(t)=\frac{\hat{a}(1+\beta)e^{-\hat{b}t}}{\left(1+\beta e^{-\hat{b}t}\right)}$$

biçimindedir.

Tip 1 veride; tanımlanmış β parametresi için, modeldeki a ve b parametrelerinin en çok olabilirlik tahmini yöntemine göre elde edilmiş eşitlikler,

$$\hat{a} = \frac{y_n\left(1+\beta e^{-\hat{b}t_n}\right)}{\left(1-e^{-\hat{b}t_n}\right)}$$

ve

$$\sum_{i=1}^{n}\left(y_i - y_{i-1}\right)\left(\frac{\left(t_i e^{-\hat{b}t_i} - t_{i-1}e^{-\hat{b}t_{i-1}}\right)}{\left(e^{-\hat{b}t_{i-1}} - e^{-\hat{b}t_{ii}}\right)} + \frac{\beta t_i e^{-\hat{b}t_i}}{\left(1+\beta e^{-\hat{b}t_i}\right)} + \frac{\beta t_{i-1}e^{-\hat{b}t_{i-1}}}{\left(1+\beta e^{-\hat{b}t_{i-1}}\right)}\right) = \frac{y_n t_n e^{-\hat{b}t_n}\left(1-\beta+2\beta e^{-\hat{b}t_n}\right)}{\left(1-e^{-\hat{b}t_n}\right)\left(1+\beta e^{-\hat{b}t_n}\right)}$$

biçimindedir. Eşitlikler aynı anda çözüldüğünde, parametrelerin tahmin değerleri elde edilir.

Tip 2 veri için a ve b parametrelerinin en çok olabilirlik tahmin edicisine göre elde edilen eşitlikler,

$$\hat{a} = \frac{n\left(1+\beta e^{-\hat{b}s_n}\right)}{\left(1-e^{-\hat{b}s_n}\right)}$$

ve

$$\frac{ns_n e^{-\hat{b}s_n}\left(1+\beta\right)}{\left(1-e^{-\hat{b}s_n}\right)\left(1+\beta e^{-\hat{b}s_n}\right)} = \frac{n}{\beta} - \sum_{i=1}^{n} s_i + 2\sum_{i=1}^{n}\frac{\beta s_i e^{-\hat{b}s_i}}{\left(1+\beta e^{-\hat{b}s_i}\right)}$$

biçimindedir. Eşitlikler aynı anda çözüldüğünde, parametrelerin tahmin değerleri elde edilir.

b) Geciktirilmiş S şekilli homojen olmayan Poisson süreci modelleri (Delayed S shaped Models)

Homojen olmayan geciktirilmiş S şekilli modeller aşağıdaki varsayımlara dayanır:

i) Programdaki tüm hatalar, hatayı tespit etme yönteminden/bakış açısından bağımsızdır.
ii) Herhangi bir zamanda hata tespit etme olasılığı, programda mevcut olan tespit edilmesi muhtemel hata sayısı ile orantılıdır.
iii) Tespit edimesi muhtemel olan her hatanın hatalı oranı sabittir.
iv) Programdaki başlangıç hata içeriği ve hatanın meydana geldiği zaman rastgele değişkendir.
v) (i-1) inci ve (i) inci zaman arasında meydana gelmiş olan hatalar (i-1) inci zamanda çıkan hataya bağlıdır.
vi) Yazılımda hata her zaman meydana gelir, tespit edilen hatalar hemen yok edilir ve yeni hata tanımlanmaz.

Ortalama değer fonksiyonu (Yamada vd. 1984),

$$m(t) = a\left[1 - (1 + bt)e^{-bt}\right]$$

biçimindedir. Burada hata tespit etme oranı,

$$b(t) = \frac{b^2 t}{bt + 1}$$

dır. Geciktirilmiş S şekilli modellerin hata yoğunluk fonksiyonu,

$$\lambda(t) = ab^2 te^{-bt}$$

ve güvenirlik fonksiyonu,

$$\hat{R}(s \setminus t) = e^{-[\hat{m}(t+s)-\hat{m}(t)]} = e^{-\alpha\left[\left(1+\hat{b}t\right)e^{-\hat{b}t}-\left(1+\left(\hat{b}+s\right)\right)e^{-\hat{b}(t+s)}\right]}$$

dır. t zamanında sistemdeki kalan hataların tahmini sayısı n(t) ise,

$$\hat{n}(t) = \hat{m}(\infty) - \hat{m}(t) = \hat{a}\left(1 + \hat{b}t\right)e^{-\hat{b}t}$$

dır.

Tip 1 veri modeldeki a ve b parametrelerinin en çok olabilirlik tahmin yöntemine göre elde edilmiş eşitlikler,

$$\hat{a} = \left[\frac{y_n}{1-\left((1+\hat{b}t_n)e^{-\hat{b}t_n}\right)}\right]$$

ve

$$\left[\frac{y_n t_n^2 e^{-\hat{b}t_n}}{1-\left((1+\hat{b}t_n)e^{-\hat{b}t_n}\right)}\right] = \sum_{i=1}^{n}\left[\frac{(y_i - y_{i-1})\left(t_i^2 e^{-\hat{b}t_i} - t_{i-1}^2 e^{-\hat{b}t_{i-1}}\right)}{\left(1+\hat{b}t_{i-1}\right)e^{-\hat{b}t_{i-1}} - \left(1+\hat{b}t_i\right)e^{-\hat{b}t_i}}\right]$$

biçimindedir. Eşitlikler aynı anda çözüldüklerinde parametre tahmin değerleri elde edilebilir.

Tip 2 veri modeldeki a ve b parametrelerinin en çok olabilirlik tahmini yöntemine göre elde edilmiş eşitlikler,

$$\hat{a} = \left[\frac{n}{1-\left(1+\hat{b}s_n e^{-\hat{b}s_n}\right)}\right]$$

$$\frac{2n}{\hat{b}} = \sum_{i=1}^{n} s_i + \left[\frac{n\hat{b}s_n^2 e^{-\hat{b}s_n}}{1-\left(1+\hat{b}s_n e^{-\hat{b}s_n}\right)}\right]$$

biçimindedir. Eşitlikler aynı anda çözüldüklerinde parametre tahmin değerleri elde edilebilir.

3.2.4.3 Düzeltme sürecinin tamamlanmadığı durumlarda homojen olmayan Poisson süreç modelleri

Hata düzeltmesinin tamamlanmadığı durumlarda hata sayısı tahmini için geliştirilmiş modellerdir. Bu modeller aşağıdaki varsayımlar altında çalışırlar:

i) Tespit edilmiş hatalar yok edilirken, yeni hata tanımlanabilir.

ii) Yazılımda hata bulma olasılığı programda kalan hata sayısı ile orantılıdır.

Genel olarak yazılım güvenirlik modelleri üç sürece ayrılırlar. İlk süreç, sayma sürecidir. {N(t), t≥0} ifadesi t zamanında tespit edilen toplam yazılım hata sayısıdır. Bu süreçte kabul edilmiş yaklaşım N(t)'nin zamana bağlı m(t) Poisson parametresi ile birlikte Poisson dağılıma sahip olmasıdır. m(t) fonksiyonu t zamanında tespit edilmesi tahmin edilen yazılım hata sayısını ifade eder (Mean Value Function, MVF).

İkinci süreç yaklaşımı, ortalama değer fonksiyonu olan m(t)'nin analitik olarak tanımlanmasıdır. Bu fonksiyon hata miktarı fonksiyonu olan a(t) ve hata tespit oranı olan b(t) fonksiyonundan etkilenir. Bu parametreler zamana bağımlı olup zamanla artan fonksiyonlardır. Zaman ile paralel olarak a(t) fonksiyonunun artması, hata düzeltme süreci boyunca yeni hataların tanımlanması, hata tespit oran fonksiyonu olan b(t)'nin artması da test mühendislerinin zamanla yazılımı öğrenmeleri nedeni ile daha fazla hata bulmaları olarak değerlendirilmiştir. Bu fonksiyonlar istatistiksel yöntemler ile tahmin edilmelidir.

Üçüncü süreç, gerçek test verisinin birinci ve ikinci adımdaki yaklaşımlar uyumluluğunda istatistiksel olarak analiz edilmesidir.

a) Yamada Modeli 1 (Yamada vd. 1984)

Hata miktarı fonksiyonunun aşağıdaki gibi olduğu varsayıldığında;

$$a(t) = ae^{\alpha t}$$

dır. Ortalama değer fonksiyonu;

$$m(t) = \frac{ab}{b+\alpha}\left(e^{\alpha t} - e^{-bt}\right)$$

biçimindedir. Modelde yer alan a, b ve α parametrelerinin en çok olabilirlik tahmin yöntemine göre elde edilmiş eşitlikler,

$$\hat{a} = \frac{\hat{b}+\hat{\alpha}}{\hat{b}} \frac{y_n}{\left(e^{\hat{\alpha}t_n} - e^{-\hat{b}t_n}\right)} ,$$

$$\sum_{i=1}^{n}\left[\left(y_i - y_{i-1}\right)\frac{\hat{a}}{B_i}\left(\frac{\hat{\alpha}A_i}{\left(\hat{b}+\hat{\alpha}\right)^2} + \frac{\hat{b}C_i}{\hat{b}+\hat{\alpha}} \right) - \hat{\alpha}\left(\frac{\hat{\alpha}A_i}{\left(\hat{b}+\hat{\alpha}\right)^2} + \frac{\hat{b}C_i}{\hat{b}+\hat{\alpha}} \right)\right] = 0$$

ve

$$\sum_{i=1}^{n}\left[\left(y_i - y_{i-1}\right)\frac{\hat{a}}{B_i}\left(\frac{-\hat{b}A_i}{\left(\hat{b}+\hat{\alpha}\right)^2} + \frac{\hat{b}C_i}{\hat{b}+\hat{\alpha}} \right) - \hat{\alpha}\left(\frac{-\hat{b}A_i}{\left(\hat{b}+\hat{\alpha}\right)^2} + \frac{\hat{b}C_i}{\hat{b}+\hat{\alpha}} \right)\right] = 0 ,$$

burada,

$$\hat{A}_i = \left(e^{\hat{\alpha}t_i} - e^{-\hat{\alpha}t_{i-1}}\right) - \left(e^{\hat{b}t_i} - e^{-\hat{b}t_{i-1}}\right)$$

$$\hat{B}_i = \frac{\hat{a}\hat{b}\hat{A}_i}{\hat{b}+\hat{\alpha}}$$

$$\hat{C}_i = t_i e^{\hat{\alpha}t_i} - t_{i-1}e^{-\hat{\alpha}t_{i-1}}$$

biçimindedir. Eşitlikler aynı anda çözüldüklerinde parametre tahmin değerleri elde edilebilir.

b) Yamada Modeli 2 (Yamada vd. 1984)

a(t) fonksiyonunun aşağıdaki gibi olduğu varsayılırsa;

$$a(t) = a(1+\alpha t)$$

Ortalama değer fonksiyonu;

$$m(t) = a\left(1 - e^{-bt}\right)\left(1 - \frac{\alpha}{\beta}\right) + a\alpha t$$

biçimindedir.

Modelde yer alan a, b, β ve α parametrelerinin en çok olabilirlik tahmin yöntemine göre elde edilmiş eşitlikler,

$$\hat{a} = \frac{y_n}{\left(1 - e^{-\hat{b}t_n}\right)\left(1 - \frac{\hat{\alpha}}{\hat{\beta}}\right) + \hat{\alpha}t_n}$$

$$\hat{b} = \frac{y_n t_n e^{-\hat{b}t_n}\left(1 - \frac{\hat{\alpha}}{\hat{\beta}}\right)}{\left(1 - e^{-\hat{b}t_n}\right)\left(1 - \frac{\hat{\alpha}}{\hat{\beta}}\right) + \hat{\alpha}t_n}$$

$$\hat{a}\left(t_n - \left(\frac{1 - e^{-\hat{b}t_n}}{\hat{\beta}}\right)\right) = \sum_{i=1}^{n} \frac{\left(e^{-\hat{b}t_i} - e^{-\hat{b}t_{i-1}}\right) + \hat{\beta}(t_i - t_{i-1})}{(\hat{\beta} - \hat{\alpha})\left(e^{-\hat{b}t_{i-1}} - e^{-\hat{b}t_i}\right) + \hat{\beta}\hat{\alpha}(t_i - t_{i-1})}(y_i - y_{i-1})$$

$$\hat{a}\left(t_n - \left(\frac{\hat{\alpha}}{\hat{\beta}^2}(1 - e^{-\hat{b}t_n})\right)\right) = \sum_{i=1}^{n} \frac{\hat{\alpha}\left(e^{-\hat{b}t_{i-1}} - e^{-\hat{b}t_i}\right)}{\hat{\beta}(\hat{\beta} - \hat{\alpha})\left(e^{-\hat{b}t_{i-1}} - e^{-\hat{b}t_i}\right) + \hat{\beta}^2\hat{\alpha}(t_i - t_{i-1})}(y_i - y_{i-1})$$

biçimindedir. Eşitlikler aynı anda çözüldüklerinde parametre tahmin değerleri elde edilebilir.

3.2.4.4 Düzeltme sürecinin tamamlanmadığı durumlarda homojen olmayan S şekilli Poisson süreç modelleri

a) Pham-Nordmann-Zhang (PNZ) Modeli (Pham vd. 1999)

Bu model aşağıdaki varsayımlara dayanır:

i) Hata tanımlama miktarı zamana bağlı lineer fonksiyondur.

ii) Hata tespit oranı fonksiyonu zamana bağlı artan bükümlü S şekilli model fonksiyonudur.

Zamana bağlı a(t) ve b(t) fonksiyonlarının aşağıdaki gibi olduğu varsayıldığında,

$$a(t) = a(1 + \alpha t)$$

ve

$$b(t) = \frac{b}{1 + \beta e^{-bt}}$$

biçiminde verilir. a=a(0) test sürecinden önce yazılımda var olan hataların toplam sayısıdır ve $\frac{b}{1+\beta}$ başlangıç hata oranıdır. Ortalama değer fonksiyonu;

$$m(t) = \frac{a}{1 + \beta e^{-bt}} \left(\left[1 - e^{-bt} \right] \left[1 - \frac{\alpha}{\beta} \right] + \alpha t \right)$$

biçimindedir.

PNZ modeli (Pham vd. 1999), hata düzeltme sürecinde hatalar, hata başına α sabiti ile tespit edilebilirler. Böylece hata miktarı fonksiyonu, a(t), test zamanının lineer fonksiyonudur ve hata tespit oranı fonksiyonu b(t) yazılım test mühendislerinin öğrenme süreci olduğundan artan S şekil eğrili fonksiyondur.

b) Hata düzeltme sürecinin tamamlanmadığı durumlarda Üstel Pham Modeli (Pham 2000)

Bu model aşağıdaki varsayımlara dayanmaktadır:

i) Test süresince hata tanımlama miktarı üstel fonksiyondur.

ii) Hata tespit oranı fonksiyonu artan bükümlü S şekilli modellerdir.

Zamana bağlı a(t) ve b(t) fonksiyonlarının aşağıdaki gibi olduğunu varsaydığımızda:

$$a(t) = \alpha e^{\beta t}$$

ve

$$b(t) = \frac{b}{1 + ce^{-bt}}$$

biçimindedir. Ortalama değer fonksiyonu;

$$m(t) = \frac{\alpha b}{b + \beta}\left(\frac{e^{(\beta+b)t} - 1}{e^{bt} + c}\right)$$

dır. Model parametrelerinden α, β, b, c aşağıdaki eşitliklerin çözümü ile elde edilir.

$$\hat{m}(t) = 0 \ ,$$

$$\left[\frac{-1}{\left(\hat{b} + \hat{\beta}\right)\hat{b}} + \frac{te^{\left(\hat{b}+\hat{\beta}\right)t}}{e^{\left(\hat{b}+\hat{\beta}\right)t} - 1}\right]\hat{m}(t) = 0$$

ve

$$\left[\frac{\hat{\beta}}{\left(\hat{b} + \hat{\beta}\right)\hat{b}} + \frac{te^{\left(\hat{b}+\hat{\beta}\right)t}}{e^{\left(\hat{b}+\hat{\beta}\right)t} - 1} + \frac{-te^{\hat{b}t}}{e^{\hat{b}t} + \hat{c}}\right]\hat{m}(t) = 0$$

c) Pham-Zhang Homojen Olmayan Poisson Süreç Modeli (Pham ve Zhang 1997)

Bu model aşağıdaki varsayımlara dayanmaktadır:

i) Test sürecinde hata tanımlama miktarı üstel fonksiyondur. Başka bir ifade ile test sürecinin başlangıcında hatanın tanımlanma sayısı testin sonuna göre daha hızlı artar. Testin sonunda test mühendisleri program hakkında daha fazla bilgiye sahip olduklarından daha fazla hata bulunur.

ii) Hata tespit oranı fonksiyonu artan bükümlü S şekilli modellerdir.

Zamana bağlı a(t) ve b(t) fonksiyonlarının aşağıdaki gibi olduğunu varsaydığımızda:

$$a(t) = c + a\left(1 - e^{-\alpha t}\right) ,$$

$$b(t) = \frac{b}{1 + \beta e^{-bt}}$$

biçiminde verilir. Ortalama değer fonksiyonu ise;

$$m(t) = \frac{1}{\left(1 + \beta e^{-bt}\right)}\left((c + a)(1 - e^{-bt}) - \frac{ab}{b - \alpha}(e^{-\alpha t} - e^{-bt} \right)$$

dır. Genel olarak homojen olmayan Poisson süreci yazılım güvenirlik modelleri tahmini hata sayısını öngörme amacı ile kullanılırlar ve farklı modeller farklı yaklaşımlara dayandığından ortalama değer fonksiyonu tahminlerinde farklı matematiksel eşitlikler kullanılmıştır.

Çizelge 3.5 Homojen olmayan Poisson süreci yazılım güvenirlik modelleri özeti

Model	Ortalama Değer Fonksiyonu m(t)
Goel-Okumoto (G-O)	$m(t) = a\left(1 - e^{-bt}\right)$ a(t)=a b(t)=b
Geciktirilmiş S şekilli modeller	$m(t) = a\left[1 - (1 + bt)e^{-bt}\right]$ a(t)=a $b(t) = \frac{b^2 t}{bt + 1}$
Bükümlü S şekilli modeller (Inflection S Shaped)	$m(t) = \frac{a}{1 + \beta e^{-bt}}\left(1 - e^{-bt}\right)$ a(t)=a $b(t) = \frac{b}{1 + \beta e^{-bt}}$
Hata düzeltme sürecinin tamamlanmadığı durumlarda Yamada Modeli (1)	$m(t) = \frac{ab}{b + \alpha}\left(e^{\alpha t} - e^{-bt}\right)$ $a(t) = ae^{\alpha t}$ b(t)=b

Çizelge 3.5 Homojen olmayan Poisson süreci yazılım güvenirlik modelleri özeti (devam)

Hata düzeltme sürecinin tamamlanmadığı durumlarda Yamada Modeli (2) (Yamada imperfect debugging 2)	$m(t) = a\left(1 - e^{-bt}\right)\left(1 - \frac{\alpha}{\beta}\right) + a\alpha t$ $a(t) = a(1 + \alpha t)$ b(t)=b
Hata düzeltme sürecinin tamamlanmadığı durumlarda Üstel Pham modeli (Pham Exponential I mperfect Debugging Model)	$m(t) = \frac{\alpha b}{b + \beta}\left(\frac{e^{(\beta+b)t} - 1}{e^{bt} + c}\right)$ $a(t) = \alpha e^{\beta t}$ $b(t) = \frac{b}{1 + ce^{-bt}}$
Pham-Zhang homojen olmayan poisson süreç modeli	$m(t) = \frac{1}{\left(1 + \beta e^{-bt}\right)}\left((c + a)(1 - e^{-bt}) - \frac{ab}{b - \alpha}(e^{-\alpha t} - e^{-bt}\right)$ $a(t) = c + a\left(1 - e^{-\alpha t}\right)$ $b(t) = \frac{b}{1 + \beta e^{-bt}}$

4. ARAŞTIRMA BULGULARI

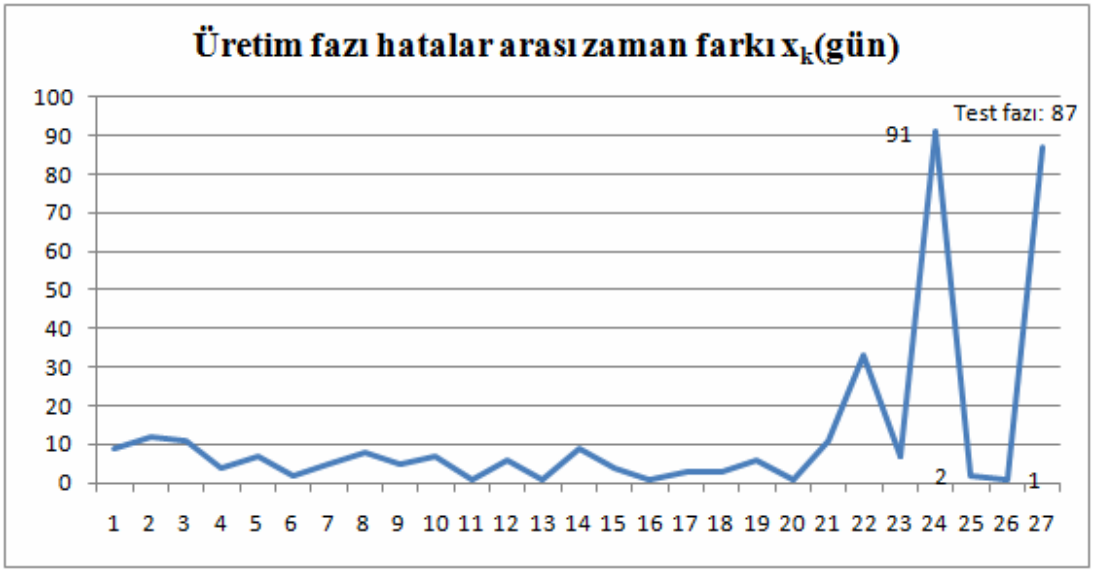
Meteryal ve Yöntem bölmünde anlatılan İstatistiksel Süreç Kontrolü ve Güvenirlilik Analiz yöntemlerinin gerçek hayattaki verilere göre değerlendirilmesi ve yorumlanması bu bölümde incelenecektir.

Örnek 4.1 Amerika Birleşik Devletleri Deniz Kuvvetleri Taktik Veri Sistemleri biriminden gerçek zamanlı ve karmaşık yapılı yazılım geliştirme projesindeki hata sayıları elde edilmiş ve bu veriler çizelge 4.1'de verilmiştir. Yazılım 38 tane farklı modülden oluşmaktadır ve yazılım geliştirme süreci üretim fazı, test fazı, kullanıcı fazı ve kullanım sonrası test fazı olarak 4 farklı faza bölünerek değerlendirilmiştir. Toplamda 26 tane hata üretim fazında, 5 tane hata test fazında, bir tane hata müşterinin kullanım süresince ve iki tane hata da daha sonraki test fazında tespit edilmiştir (Pham 2006).

Çizelge 4.1 Deniz Kuvvetleri Taktik Veri (D.K.T.D) sistemleri yazılım geliştirme hata seti

Geliştirme Fazı	Hata Numarası	Hatalar arası zaman farkı x_k(gün)	Kümülatif zaman $S_n=\Sigma x_k$(gün)	Geliştirme Fazı	Hata Numarası	Hatalar arası zaman farkı x_k(gün)	Kümülatif zaman $S_n=\Sigma x_k$(gün)
Üretim Fazı	1	9	9	Üretim Fazı	18	3	98
Üretim Fazı	2	12	21	Üretim Fazı	19	6	104
Üretim Fazı	3	11	32	Üretim Fazı	20	1	105
Üretim Fazı	4	4	36	Üretim Fazı	21	11	116
Üretim Fazı	5	7	43	Üretim Fazı	22	33	149
Üretim Fazı	6	2	45	Üretim Fazı	23	7	156
Üretim Fazı	7	5	50	Üretim Fazı	24	91	247
Üretim Fazı	8	8	58	Üretim Fazı	25	2	249
Üretim Fazı	9	5	63	Üretim Fazı	26	1	250
Üretim Fazı	10	7	70	Test Fazı	27	87	337
Üretim Fazı	11	1	71	Test Fazı	28	47	384
Üretim Fazı	12	6	77	Test Fazı	29	12	396
Üretim Fazı	13	1	78	Test Fazı	30	9	405
Üretim Fazı	14	9	87	Test Fazı	31	135	540
Üretim Fazı	15	4	91	Bakım Fazı	32	258	798
Üretim Fazı	16	1	92	Bakım Fazı	33	16	814
Üretim Fazı	17	3	95	Bakım Fazı	34	35	849

Şekil 4.1'daki veri yapısı incelendiğinde, üretim fazında tespit edilen hataların ilk 26 tanesinin son üçünün farklı grup oluşturduğu değerlendirilmiştir. Bu grubun kendinden önceki ve sonraki gözlemler ile arasında uzaklık farkı çok olduğundan modeli oluştururken ilk 25 gözlem ya da ilk 26 gözlem dikkate alınmalıdır. Bu yaklaşıma göre, çizelge 4.1'de yer alan ilk 27 gözlem tahmin değerlerinin hesaplanması için kullanılacak ve sonrasındaki bakım/test fazınında meydana gelen 7 gözlem değeri ise tahminlerin değerlendirilmesi amacıyla kullanılacaktır.



Şekil 4.1 Üretim fazında çıkan hatalar arasındaki zaman farkı takibi

Taktik veri sistemleri örneğinde, üretim ve test fazında kodlama faaliyetleri tamamen bitmediğinden sisteme hata dahil etme oranının a(t) ve sürekli devam eden test aktivitesinden dolayı hatanın tespit edilme oranının b(t), zamanla artacağı varsayılmıştır.

Bu nedenle bölüm 3.2.4'te tanımlanmış Homojen olamayan Poisson süreci modeller'inin varsayımları dikkate alındığında, Yamada 1,2 veya PNZ Modelinin daha iyi sonuç vereceği düşünülmüştür. Kullanıcı ve sonrasındaki test fazları değerlendirildiğinde, ürün müşteriye teslim edilmesiyle geliştirme faaliyetleri bitmiştir. Böylece sisteme yeni hatanın tanımlanması olasılığı oldukça azdır. Başka bir ifadeyle, a(t) fonksiyonu sabittir denilebilir. Bu varsayıma göre ise en iyi sonucu Goel Okumoto veya Bükümlü S şekilli model verecektir. Modellerin ortalama değer fonksiyonları,

Model 1 Goel-Okumoto

$$\hat{m}(t) = \hat{a}\left(1 - e^{-\hat{b}t}\right)$$

Model 2 Bükümlü S şekilli modeller

$$\hat{m}(t) = \frac{\hat{a}}{1+\hat{\beta}e^{-\hat{b}t}}\left(1-e^{-\hat{b}t}\right)$$

Model 3 Hata düzeltme sürecinin tamamlanmadığı durumlarda Yamada Modeli (1)
(Yamada imperfect debugging 1)

$$\hat{m}(t) = \frac{\hat{a}\hat{b}}{\hat{b}+\hat{\alpha}}\left(e^{\hat{\alpha}t}-e^{-\hat{b}t}\right)$$

Model 4 Hata düzeltme sürecinin tamamlanmadığı durumlarda Yamada Modeli (2)
(Yamada imperfect debugging 2)

$$\hat{m}(t) = \hat{a}\left(1-e^{-\hat{b}t}\right)\left(1-\frac{\hat{\alpha}}{\beta}\right)+\hat{a}\hat{\alpha}t$$

Model 5 (Pham-Nordmann-Zhang)

$$\hat{m}(t) = \frac{\hat{a}}{1+\beta e^{-\hat{b}t}}\left(\left[1-e^{-\hat{b}t}\right]\left[1-\frac{\hat{\alpha}}{\beta}\right]+\hat{\alpha}t\right)$$

biçimindedir. Bütün modellerin güvenirlik fonksiyonları genel olarak,

$$\hat{R}(s \setminus t) = e^{-[\hat{m}(t+s)-\hat{m}(t)]}$$

formülü ile hesaplanır.

En çok olabilirlik tahmin yöntemine göre, verinin ilk 27 gözlem değeri dikkate alınarak modellerdeki parametre değerleri Matlab paket programı kullanılarak tahmin edilmiş ve bu değerler,

$Model\ \ 1: \hat{a} = 29.42827\ ,\ \ \hat{b} = 0.007402$

$Model\ \ 2: \hat{a} = 27.44246\ ,\ \ \hat{b} = 0.015517\ ,\ \ \hat{\beta} = 2.042596$

$Model\ \ 3: \hat{a} = 29.42827\ ,\ \ \hat{b} = 0.015517\ ,\ \ \hat{\alpha} = 0$

$Model\ \ 4: \hat{a} = 29.42827\ ,\ \ \hat{b} = 0.007402\ ,\ \ \hat{\alpha} = 0$

$Model\ \ 5: \hat{a} = 19.32827\ ,\ \ \hat{b} = 0.047526\ ,\ \ \hat{\alpha} = 0.001256\ ,\ \ \hat{\beta} = 24.33569$

biçiminde elde edilmiştir.

Veri yapısına göre aşağıdaki 5 modelde HKO ve ABK ölçüm değerleri hesaplanarak en iyi hangi modelin tahmin değeri vereceği hesaplanacak ve sonrasında modele göre t zamanında tespit edilmesi tahmin edilen hata sayısı ve güvenirlik değerleri elde edilecektir.

Çizelge 4.2 Modellerin uyum iyiliği ve tahmin gücünün karşılaştırılması

Model	HKO (kestirim) $HKO=\frac{\sum_{k=1}^{27}(\hat{m}(t_k)-y_k)^2}{27-N}$	HKO (öngörü) $HKO=\frac{\sum_{k=28}^{31}(\hat{m}(t_k)-y_k)^2}{31-N}$	ABK
G-O Model	5.46	2.48	88.98
Bükümlü S şekilli model	2.40	4.64	88.44
Yamada Modeli (1)	4.65	32.94	90.36
Yamada Modeli (2)	4.57	79.11	90.36
PNZ Model	0.59	2.14	81.82

Çizelge 4.2'de gözlem değerlerinin HKO ve ABK değerlerini özetlemektedir. Hesaplanan değerlere göre en küçük HKO ve ABK sonucunu veren Model 5 PNZ modelidir. Buna göre çizelgedeki değerlerden PNZ modeli diğer modellere göre daha iyi tahmin sonucu verir şeklinde yorumlanır.

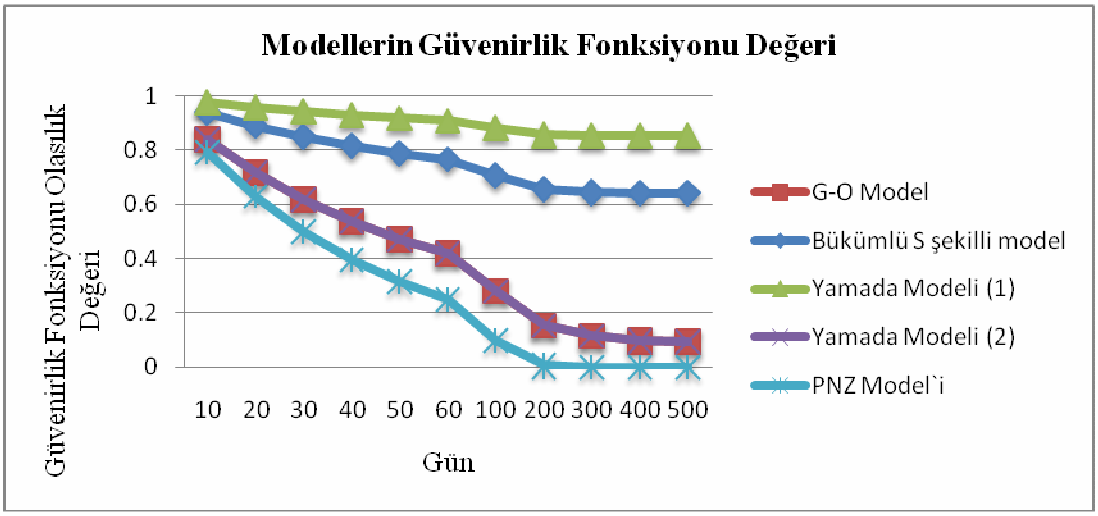
27 hatanın gözlemlendiği t zamandan s gün sonra, sistemde çıkması olası beklenen hata sayısı ve güvenirlik öngörü değerleri çizelge 4.3-4.4'te yer almaktadır. Hesaplanan değerler grafiklendirilmiş olarak sırasıyla şekil 4.2-4.3'tedir.

Çizelge 4.3 D.K.T.D sistemleri güvenirlik fonksiyonu öngörü değerleri

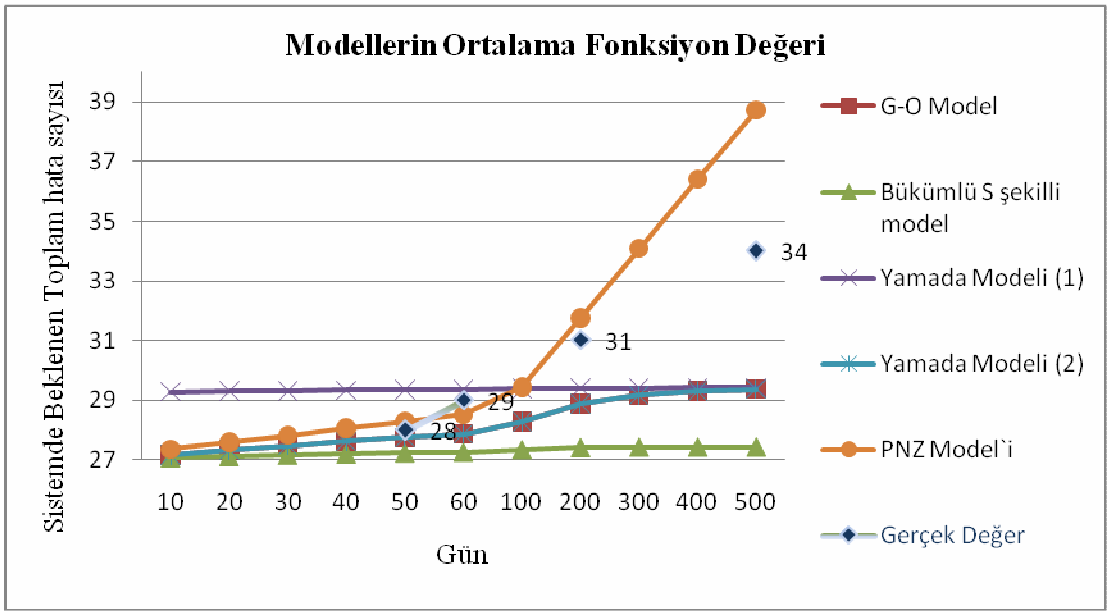
Güvenirlik Fonksiyon Değeri $\hat{R}(t)$					
Gün (s)	G-O Model	Bükümlü S şekilli modeller	Yamada Modeli (1)	Yamada Modeli (2)	PNZ Model
10	0.841	0.939	0.977	0.841	0.793
20	0.716	0.889	0.959	0.716	0.629
30	0.617	0.849	0.943	0.617	0.499
40	0.537	0.816	0.929	0.537	0.396
50	0.472	0.789	0.918	0.472	0.314
60	0.419	0.766	0.908	0.419	0.249
100	0.281	0.706	0.883	0.281	0.098
200	0.153	0.655	0.860	0.153	0.010
300	0.115	0.645	0.855	0.115	0.001
400	0.100	0.643	0.854	0.100	0.000
500	0.094	0.643	0.853	0.094	0.000

Çizelge 4.4 D.K.T.D sistemleri ortalama değer fonksiyonu değerleri

Ortalama Değer Fonksiyonu $\hat{m}(t+s)$						
Gün (s)	G-O Model	Bükümlü S şekilli modeller	Yamada Modeli (1)	Yamada Modeli (2)	PNZ Modeli	Gerçek Değer
10	27	27	29	27	27	
20	27	27	29	27	28	
30	27	27	29	27	28	
40	28	27	29	28	28	
50	28	27	29	28	28	28
60	28	27	29	28	29	29
100	28	27	29	28	29	
200	29	27	29	29	32	31
300	29	27	29	29	34	
400	29	27	29	29	36	
500	29	27	29	29	39	34



Şekil 4.2 D.K.T.D sisteminde modellerin güvenirlik fonksiyonu tahmin değerinin trend analizi

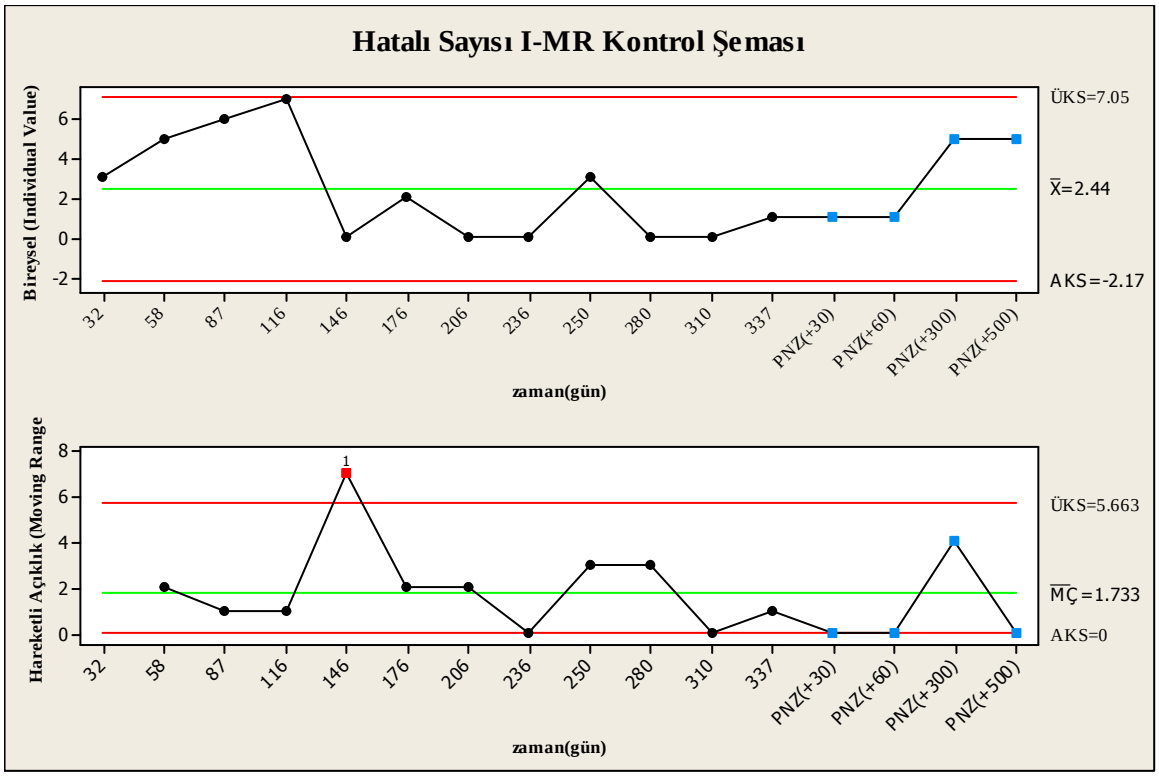


Şekil 4.3 D.K.T.D sisteminde modellerin ortalama değer fonksiyonu trend analizi

Çizelge 4.3'e göre son hatanın tespit edilmesinden sonra ilk 50 gün içinde yeni bir hatanın tespit edilmeme olasılığı G-O modeline göre 0.47, bükümlü S şekilli modele göre 0.79, Yamada 1 modeline göre 0.91, Yamada 2 modeline göre 0.47 ve PNZ modeline göre 0.31 dir. Gerçekleşmiş veri kontrol edildiğinde yaklaşık elli gün sonra bir tane hata gözlemlenmiştir. Çizelge 4.4'e bakıldığında PNZ modelinin tahmin ettiği sistemdeki kümülatif ortalama hata değeri de 28 olmuştur. Gerçek gözlenen değerler ile kıyaslandığında 100 günden sonra PNZ modeli, diğer modellere göre daha anlamlı sonuçlar vermiştir.

Jelinski Moranda modeline göre programın başındaki bilinmeyen hata sayısı (N) tahmin edildiğinde ve 27. hatadan sonra ilk hatanın ortalama ne zaman çıkacağı (MTTF değeri) kestirildiğinde, toplamda 200 hatanın olduğu, 28. hatanın da 13 gün sonra tespit edileceği söylenir. Ancak çizelge 4.4'ten görüleceği üzere, homojen olmayan Poisson süreci modelleri gerçek değerlerle karşılaştırıldığında, bu modellerin daha anlamlı sonuçlar verdiği görülmüştür.

Deniz Kuvvetleri Taktik Veri sistemlerinde geliştirilen çizelge 4.1'de verilen hatalı sayısı veri setine göre süreç kontrol şeması Minitab paket programı kullanılarak çizildiğinde şekil 4.4 elde edilir.



Şekil 4.4 D.K.T.D sistemleri hata sayısı i-mr kontrol şeması

Deniz Kuvvetleri Taktik Veri Sistemleri birimi tarafından geliştirilen örnekteki yazılımda hatanın tespit edilme süreci incelendiğinde, kontrol limitlerini aşan herhangi bir gözlem olmamıştır ve süreç kontroldedir. PNZ modeline göre hatanın son tespit edildiği zamandan otuz, altmış, üçyüz ve beşyüz gün sonrasında çıkması muhtemel hata sayısı şekil 4.4'te mavi renk ile ifade edilmiştir. Tahmin değerlerinde kontrol sınırını aşan bir değere rastlanmamıştır. Hareketli açıklık kontrol şemasındaki kontrol limitini geçen nokta bireysel (individual) kontrol şemasındaki kontrol sınırları geçebileceğine dair uyarı verir.

Eğer Deniz Kuvvetleri Taktik Sistemleri birimi tarafından test aşamasından sonra müşteri alanında sıfır hata çıkması hedefleniyor ise yani spesifikasyon limiti sıfır ise PNZ modeli tahmin değerleri sıfırdan büyük çıktığından test aşamasındaki sürecin kontrolde ancak yeterli olmayacağı tahmin edilir. Bu durumda şirketin olası bu durumu engelleyecek önlemler alması beklenmelidir.

Örnek 4.2 Moa yazılım şirketi tarafından yayımlanan üretim fazı süresince saat başına tespit edilen hata sayıları çizelge 4.5'de verilmiştir. Hata tespit edildiği zaman hemen çözüldüğü ve bu süre içerisinde yeni hatanın tanımlanmadığı varsayılmıştır (Jusu 2008).

Çizelge 4.5 Moa yazılım şirketine ait üretimde tespit edilen hata seti

Zaman	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hata sayısı	27	16	11	10	11	7	2	5	3	1	4	7	2	5	5

Belirtilmiş olan varsayımlar dikkate alındığında, bölüm 3.2.4'te tanımlanmış Homojen olamayan Poisson süreci modellerinde en iyi sonucu Goel Okumoto veya Bükümlü S şekilli model'in vereceği varsayılmıştır. Buna göre verilerin 15 gözlem değeri kullanılarak en çok olabilirlik tahmin yöntemine göre parametre tahmin değerleri Matlab paket programı kullanılarak elde edilmiştir. Bu değerler,

$Model\ \ 1: \hat{a} = 128.5347\ ,\ \hat{b} = 0.1552$

$Model\ \ 2: \hat{a} = 124.5637\ ,\ \hat{b} = 0.1860\ ,\ \hat{\beta} = 0.1283$

biçiminde elde edilmiştir.

Veri yapısına göre modellerin HKO ve ABK ölçüm değerleri hesaplanarak en iyi hangi modelin tahmin değeri vereceği hesaplanacak ve sonrasında modele göre t zamanında tespit edilmesi tahmin edilen hata sayısı ve güvenirlik değerleri elde edilecektir.

Çizelge 4.6 Modellerin uyum iyiliği ve tahmin gücünün karşılaştırılması

Model	HKO (kestirim) $HKO=\frac{\sum_{k=1}^{15}(\hat{m}(t_k)-y_k)^2}{15-N}$	AIC
G-O Model	36.9681	144.252
Bükümlü S şekilli model	42.9387	143.32

Elde deilen bu değerlere göre HKO kriteri Goel Okumoto modelinde daha küçük olduğundan daha iyi tahmin değerleri elde edileceği düşünülür.

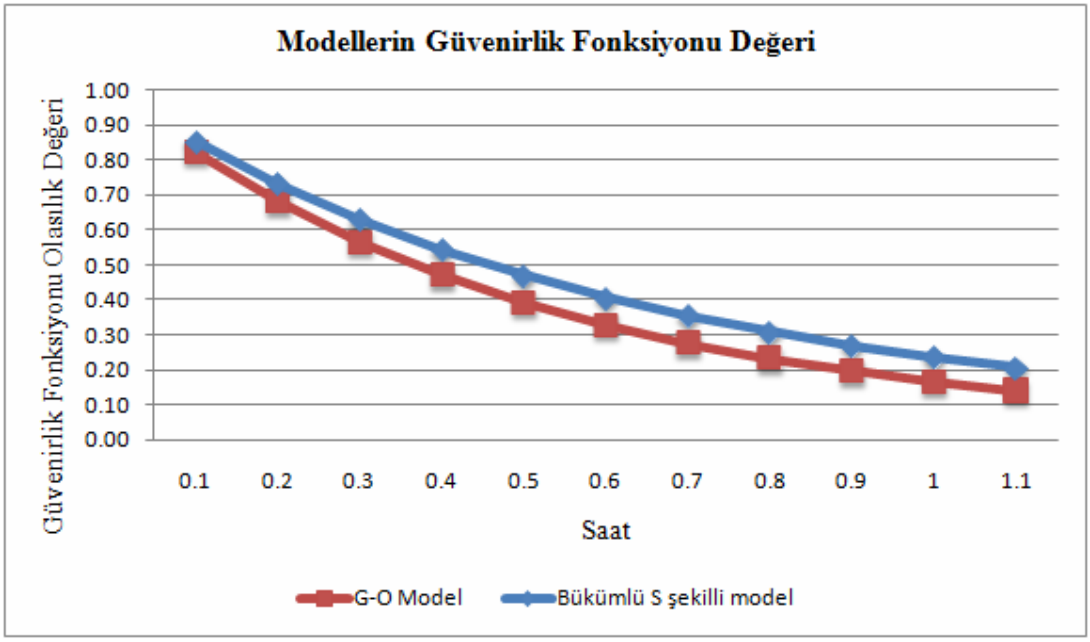
15 hatanın gözlemlendiği t zamandan s gün sonra, sistemde çıkması olası beklenen hata sayısı ve güvenirlik öngörü değerleri çizelge 4.6-4.7'de yer almaktadır. Hesaplanan değerler grafiklendirilmiş olarak sırasıyla şekil 4.5-4.6'dadır.

Çizelge 4.7 Moa yazılım şirketi verisi güvenirlik fonksiyonu öngörü değerleri

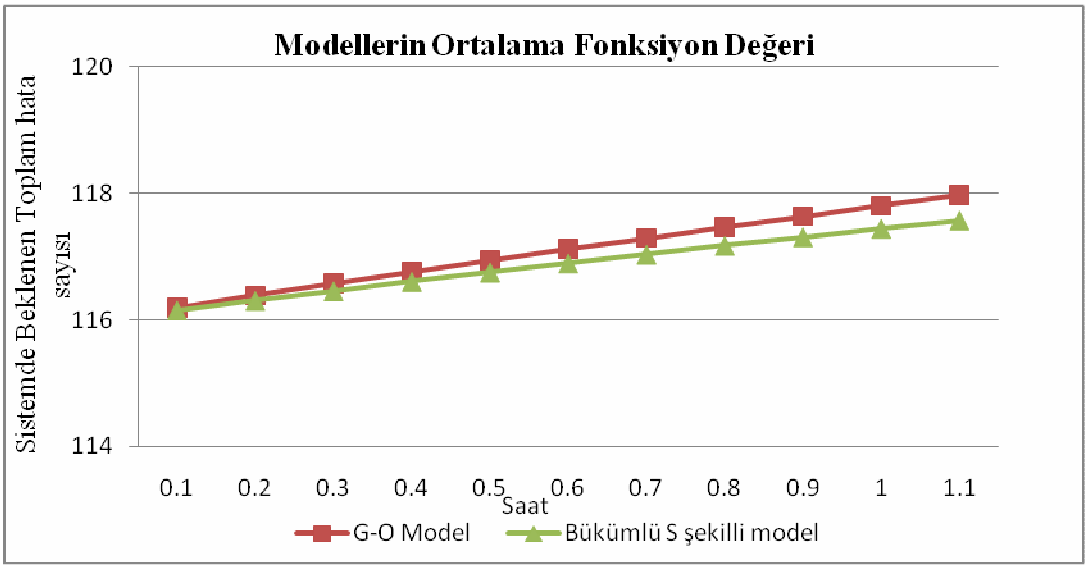
Güvenirlik Fonksiyon Değeri R(t)					
Gün (s)	G-O Model	Bükümlü S şekilli model	Gün (s)	G-O Model	Bükümlü S şekilli model
0.1	0.82	0.85503	0.7	0.28	0.35401
0.2	0.68	0.73315	0.8	0.23	0.30845
0.3	0.57	0.6304	0.9	0.20	0.26943
0.4	0.47	0.54354	1	0.17	0.23592
0.5	0.39	0.46992	1.1	0.14	0.20708
0.6	0.33	0.40734			

Çizelge 4.8 Moa yazılım şirketi verisi ortalama değer fonksiyonu değerleri

Ortalama Değer Fonksiyonu m(t+s)					
Gün (s)	G-O Model	Bükümlü S şekilli model	Gün (s)	G-O Model	Bükümlü S şekilli model
0.1	116	116	0.7	117	117
0.2	116	116	0.8	117	117
0.3	117	116	0.9	118	117
0.4	117	117	1	118	117
0.5	117	117	1.1	118	118
0.6	117	117			



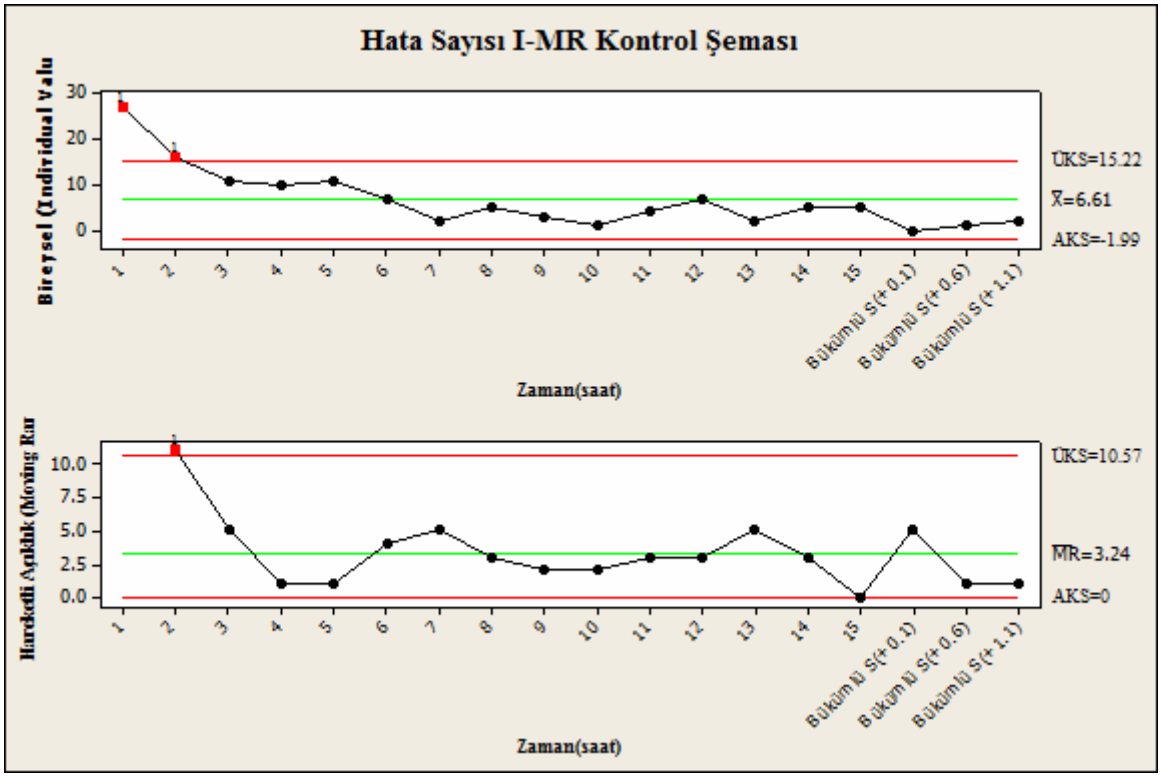
Şekil 4.5 Moa yazılım şirketi verisi için kullanılan modellerin güvenirlik fonksiyonu tahmin değerinin trend analizi



Şekil 4.6 Moa yazılım şirketi verisi için kullanılan modellerin ortalama değer fonksiyonu trend analizi

Çizelge 4.7-4.8 incelendiğinde, Goel Okumoto modelinin güvenirlik değerlerinin Bükümlü S şekilli modele göre daha düşük olduğu ve ortalama değer fonksiyon değerlerinin ise daha yüksek olduğu gözlemlenmiştir.

Moa yazılım şirketinde geliştirilen ürünün, çizelge 4.8'de verilen hatalı sayısı veri setine göre süreç kontrol şeması Minitab paket programı kullanılarak çizildiğinde şekil 4.7 elde edilir.



Şekil 4.7 Moa yazılım şirketi verisi kullanılarak elde edilen hata sayısı i-mr kontrol şeması

Moa yazılım şirketi tarafından geliştirilen örnekteki yazılımda hatanın tespit edilme süreci incelendiğinde, ilk iki gözlemin kontrol limitlerini aştığı gözlemlenmiştir. Bu süre içinde süreç kontrol dışına çıkmıştır, ancak ilerleyen süre içinde süreç tekrar kontrol limitleri içinde kalmıştır. Dolayısı ile ilk iki gözlemden sonra Moa yazılım şirketi tarafından süreci kontrol altında tutmak için düzeltici/ önleyici işlem alındığı gözlemlenmiştir. Hareketli açıklık kontrol şemasındaki kontrol limitini geçen nokta bireysel (individual) kontrol şemasındaki kontrol sınırları geçebileceğine dair uyarı verir.

5. TARTIŞMA VE SONUÇ

Bu çalışma kapsamında, ürün geliştirme sürecinin kontrolde ve yeterli olduğunu anlamak ve olası bir hatalı üretimde sorunun kaynağına giderek tekrar oluşmasını önlemek için istatistiksel süreç kontrol şemalarının kullanılması önerilmiştir. Hangi tip kontrol şemasının kullanılacağını; verinin tipi, altgrup ve altgrup gözlem sayısı gibi parametreler etkiler. Süreci uygulayan şirket tarafından o sürece ait hedef değerleri (üst tolerans sınırı veya alt tolerans sınırı) verilerek, hedeften sapılması durumunda düzeltici ve önleyici işlemlerin alınması beklenmelidir. Tez kapsamında şekil 3.2'de sürecin göstereceği muhtemel durumlar ve genel olarak alınabilecek önleyici işlemler anlatılmıştır.

İstatistiksel kontrol şemaları sürecin performansını izleme amacı ile kullanılan etkin yöntemlerden biridir, ancak dezavantajı süreçten n adım sonraki tahmin değerini vermez. Başka bir ifade ile, istenmeyen durum meydana geldikten sonra düzeltici işlem alınmasını sağlar. Yazılım sistemlerinde geliştirilmiş olan hata tahmin yöntemleri, en iyi modeli seçmek için kullanılacak kriterler tanımlanmış ve bir örnek üzerinde uygulamanın nasıl olduğu açıklanmıştır.

Olasılıksal güvenirlik tahmin modelleri genel olarak; hata kaynağı modelleri, hatalı oranı modelleri, güvenirlik geliştirme modelleri ve homojen olmayan Poisson süreci modelleri olarak ayrıştırılır. Her model için varsayılan ön koşullar tanımlanmış olup, tahmin yapacak kişinin bu koşullara en uygun model grubu üzerine yoğunlaşması önerilir. Varsayımlar dikkate alınarak seçilen modellerden en iyi hangisinin sonuç vereceği, HKO,HKT ve ABK gibi kriterlerinin hesaplanması ile tespit edilir.

Bu çalışmada yer alan güvenilirlik tahmin modelleri, n adım sonra çevre etkenlerin aynı olduğu varsayımı altında tahmin değerlerini verir. Başka bir ifade ile; çevre etkenlerinden bağımsız, her model kendi parametresinin nasıl değişeceğini öngörerek tahmin yapar. Değişen parametreler arasında kullanıcının ve yazılımı test eden veya

kodlayan kişi arasındaki bakış açısı farklılığı veya test koşullarındaki farklılık gibi faktörlerin etkisi yok sayılmıştır. Bu nedenle tahmin sonuçları dış faktörlerin göstermesi muhtemel olumlu veya olumsuz etkileri gözönünde bulundurularak değerlendirilmelidir.

KAYNAKLAR

Baskan, Ş. 1997. İstatistiksel kalite kontrolü. Bilim, 1-131, İzmir.

Barlow, R.E. and Proschan, F. 1975. Statistical theory of reliability and life testing: probability models. Defence technical Information center, 76-77, USA.

Cai, K. 1998. On estimating the number of defects remaining in software. Journal of systems and software, 40(1).

Florac, W.A. and Carleton, A.D. 2001. Measuring the software process. Addison wesley, 85-123, USA.

Goel, A. and Okumoto, K. 1979a. Time dependent error detection rate model for software and other performance measures. . IEEE transactions on reliability, 28; 206-211.

Goel, A. and Okumoto, K. 1979b. A markovian model for reliability and other performance measures of software systems. IEEE computer society press, Los Angeles.

Huang, X. 1984. The hypergeometric distribution model for predicting the relıabılıty of software. Microelectronics and reliability, 24(1).

Jelinski, Z. and Moranda, R. 1972. Software reliability research. Acedemic press, 465-484, New York.

Jusu, J. 2008. Introduction to software reliability models. www.moasoftware.co.kr.

Kumar, U. D., Crooker, J., Chitra, T. and Saranga, H. 2006. Reliability and six sigma. Springer, 57-101, USA.

Littlewood, B. 1979. Software reliability model for modular program structure. IEEE transactions reliability, R-28(3).

Mills, H. 1970. On the statistical validation of computer programs. IBM FSD (basılmamış).

Moranda, P. 1975. A comparison of software error rate models. IEEE computer society press, Los Angeles.

Moranda, P. 1979. An error detection model for application during software development. IEEE transactions on software engineering, R-28(5).

Ohba, M. 1984. Software reliability analysis models. IBM journal of research development, 28; 428-443.

Özlü, M. ve Özbilgin, G.İ. 2010. Yazılım geliştirme süreçleri ve ISO 27001 bilgi güvenliği yönetim sistemi. Tubitak Uekae, http://www.bilgiguvenligi.gov.tr/yazilim-guvenligi/yazilim-gelistirme-surecleri-ve-iso-27001-bilgi-guvenligi-yonetim-sistemi.html.

Pham, H. 2000. Software reliability. Springer, USA.

Pham, H. 2006. System software reliability. Springer, 1-216, USA.

Pham, H. and Zhang, X. 1997. An NHPP software reliability model and its comparison. Quality and safety engineering, 4(3); 269-282.

Pham, H., Nordmann, L. and Zhang, X. 1999. A general impact software debugging model with s shaped fault detection rate. IEEE transactions on reliability, 48(2); 169-175.

Schick, G. and Wolverton, R. 1978. An analysis of competing software reliability models. IEEE transactions on software engineering, SE-4(2).

Sukert, A. N. 1977. An investigation on software reliability models. IEEE reliability society, Piscataway.

Tohma, Y., Yamano, H.and Ohba, M. and Jacoby, R. 1991. The estimation of parameters of the hypergeometric distribution and its application to the software reliability growth model. IEEE transactions on software engineering, SE-17(5).

Wall, J. and Ferguson, P. 1977. Pragmatic software reliability prediction. IEEE reliability society, Piscataway.

Xie, M. 1991. Software reliability modelling. World scientific publishing, 85-111, Singapore.

Yamada, S., Ohba and M., Osaki, S. 1984. S shaped software reliability growth models and their applications. IEEE transactions on reliability, R-33.

ÖZGEÇMİŞ

Adı Soyadı: Özgül SUGÜNEŞ

Doğum Yeri: Nevşehir

Doğum Tarihi: 10.09.1981

Medeni Hali: Bekar

Yabancı Dili: İngilizce

Eğitim Durumu:

Lise :Mimar Sinan Süper Lisesi, Ankara 1999

Lisans: Hacettepe Üniversitesi İstatistik Bölümü, Ankara 1999_2003

Yüksek Lisans: Ankara Üniversitesi İstatistik Bölümü, Ankara 2010

Çalıştığı Kurum/Kurumlar ve Yıl:

MilSOFT Yazılım Teknolojileri A.Ş.`de Kalite Güvence Uzmanı ünvanı ile 2004 yılından bu yana görevine devam etmektedir.

Yayınları (SCI ve diğer) : Yoktur.