

**KURUMSAL BİR DEĞİŐIKLİK YÖNETİM SİSTEMİ TASARIMI VE
GERÇEKLEŐTİRİMİ**

SERTEL ASIM ÜNSAL

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĐİ BÖLÜMÜ**

**TOBB EKONOMİ VE TEKNOLOĐİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ARALIK 2009

ANKARA

Fen Bilimleri Enstitü onayı

Prof.Dr. Ünver KAYNAK

Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

Doç.Dr. Erdoğan DOĞDU

Anabilim Dalı Başkanı

Sertel Asım ÜNSAL tarafından hazırlanan KURUMSAL BİR DEĞİŞİKLİK YÖNETİM SİSTEMİ TASARIMI VE GERÇEKLEŞTİRİMİ adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Yrd. Doç.Dr. Osman ABUL

Tez Danışmanı

Tez Jüri Üyeleri

Başkan : Yrd.Doç.Dr. A. Murat ÖZBAYOĞLU

Üye : Doç.Dr. Ali AKGÜNDÜZ

Üye : Yrd.Doç.Dr. Osman ABUL

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Sertel Asım ÜNSAL

Üniversitesi : TOBB Ekonomi ve Teknoloji Üniversitesi
Enstitüsü : Fen Bilimleri
Anabilim Dalı : Bilgisayar Mühendisliği
Tez Danışmanı : Yrd.Doç.Dr. Osman ABUL
Tez Türü ve Tarihi : Yüksek Lisans – Aralık 2009

Sertel Asım ÜNSAL

KURUMSAL BİR DEĞİŞİKLİK YÖNETİM SİSTEMİ TASARIMI VE GERÇEKLEŞTİRİMİ

ÖZET

Üretim, imalat ve tasarım süreçleri içerisindeki karmaşık işlemlerde ortaya çıkabilecek çok ufak değişiklikler süreçte büyük etkilere yol açar. Kurumsal Değişiklik Yönetim Sistemi bu değişikliklere bilimsel bir yaklaşımla, modern anlamda bir mühendislik yaklaşımı ile ele alarak yapılan işin kalitesini, güvencesini, hızını, organizasyonunu, yönetilmesini, mevcut yapıya uyumluluğunu, gelişmeye açık olması ve uygun değer maliyetle işlemin yapılmasını amaçlar. Bu tez süreçlerin hızını, çalışanlarının performansını ve karar mekanizmalarının güvenilirliğini artırarak zaman ve kaynak israfını azaltarak kağıtsız bilgi sistemine geçmeyi hedefler.

Anahtar Kelimeler: Değişiklik yönetim sistemi, tasarım, üretim, gerçekleştirim, yazılım mühendisliği, bilgi sistemleri

University : TOBB Economics and Technology University
Institute : Institute of Natural and Applied Sciences
Science Programme : Computer Engineering
Supervisor : Asst. Prof. Dr. Osman ABUL
Degree Awarded and Date : M.Sc. – December 2009

Sertel Asım ÜNSAL

DESIGN AND IMPLEMENTATION OF AN INSTITUTIONAL CHANGE MANAGEMENT SYSTEM

ABSTRACT

Minor changes in the complex operations of the design and production phases can affect the whole process overall. Institutional Change Management System aims to implement such changes with a scientific approach, handling the job's quality with a post modern engineering approach, assuring, integrating them with the current system, managing them, making it suitable for upgrading the system and operate with a suitable value cost. This thesis aims to pass on to the e-IT System by decreasing the loss of source where increasing the speed of test processes, employers performances and the reliability of decision-making mechanisms at the same time.

Keywords: Change management system, design, production, implementation, software engineering, information systems

TEŐEKKÜR

Çalıőmalarım boyunca deęerli yardım ve katkılarıyla beni yönlendiren hocam Yrd.Doç.Dr. Osman ABUL'a yine kıymetli tecrübelerinden faydalandığım TOBB Ekonomi ve Teknoloji Üniversitesi Bilgisayar Mühendislięi Bölümü öğretim üyelerine teşekkürü bir borç bilirim.

Her zaman bana destek verip beni cesaretlendiren sevgili Eőim Gülőah ÜNSAL, canım Oęlum Mehmet Demir'e çok teşekkür ederim.

İÇİNDEKİLER	Sayfa
ÖZET	iv
ABSTRACT	v
TEŞEKKÜR	vi
İÇİNDEKİLER	vii
ŞEKİLLERİN LİSTESİ	x
TABLolarIN LİSTESİ	xi
KISALTMALAR	xii
1. GİRİŞ	1
1.1 Değişiklik Yönetim Sistemlerinin Tarihi	2
1.2 Değişiklik Yönetim Sistemlerinin Önemi	4
1.3 Dokümana Bakış	6
2. DEĞİŞİKLİK YÖNETİM SİSTEMLERİ	7
2.1 Tasarım ve Üretim Süreçlerindeki Değişiklik Nedenleri	13
2.2 Mühendislik Yönetim Sistemleri Açısından Değişiklik Yönetim Sistemleri	14
3. KURUMSAL DEĞİŞİKLİK YÖNETİM SİSTEMİ	18
3.1 Kurumsal Değişiklik Yönetim Sistemine Genel Bir Bakış	18
3.2 Oluşan Bir Değişikliğin İzlediği Fazlar	19
3.2.1 Başlatma Fazı	19
3.2.2 Değerlendirme Fazı	21
3.2.3 İnceleme Fazı	22
3.2.4 Uygulama Fazı	23
3.3 Sistem Tarafından Oluşturulan Raporlar	23

4.	TASARLANAN SİSTEM HAKKINDA BİLGİ	24
4.1	Sistemde Kullanılan Bileşenler	24
4.2	Sistemde Tanımlı Roller	25
4.2.1	Sistemde Tanımlı Olan Rollerin Görevleri	26
4.3	Kullanılan Yazılım Geliştirme Teknolojileri	27
4.3.1	Yahoo User Interface	27
4.3.2	AJAX Teknolojisi	29
4.3.3	E-POSTA Sunucu	30
4.3.4	UML Modelleme Dili	31
4.3.5	DB2 Veritabanı Yönetim Sistemi	33
4.4	Yazılım Veritabanı Şeması	34
5.	KURUMSAL DEĞİŞİKLİK YÖNETİM SİSTEMİNDE ÖRNEK BİR OLAYIN İNCELENMESİ	36
5.1	Sisteme Giriş	36
5.2	Değişiklik Tanımlama	38
5.3	Değişiklik İnceleme	40
5.4	Ajanda Oluşturma	43
5.5	Etki Analizi	46
5.6	Atama İşlemleri	47
5.7	CIN Oluşturma İşlemleri	49
6.	SONUÇ	50
	KAYNAKLAR	53
	ÖZGEÇMİŞ	55

EKLER	56
EK 1 Deęişiklięin Tanımlandığı Doküman	56
EK 2 Deęişiklięin Deęerlendirildięi Doküman	57
EK 3 Deęişiklięin Modifikasyon Dokümanı	58
EK 4 Deęişiklik İsteęi Raporu	59
EK 5 Deęişiklik İsteęi Tarihçe Raporu	60
EK 6 Yapılacak İşlerin Tarihçe Raporu	61
EK 7 Veritabanına Tablo Oluşturma SQL Örnekleri	62
EK 8 Veritabanında Kullanılan Trigger Örnekleri	70

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 1 TAI Firmasındaki 3 Projenin Yıllara Göre Değişiklik İstek Dağılımı	5
Şekil 2 Değişiklik Yönetim Sistemleri İş Akış Diyagramı	12
Şekil 3 Değişiklik Yönetim Sistemleri Girdi ve Çıktı Diyagramı	13
Şekil 4 KDYS Sistemine Genel Bir Bakış	19
Şekil 5 Başlatma Fazı	20
Şekil 6 Değerlendirme Fazı	21
Şekil 7 İnceleme Fazı	22
Şekil 8 Uygulama Fazı	23
Şekil 9 Sistem Ana Bileşenleri	24
Şekil 10 KDYS Alt Sistemleri	25
Şekil 11 Sistemde Tanımlı Roller	26
Şekil 12 Rollerin Sistemdeki Görevleri	27
Şekil 13 YUI Geliştirme Diyagramı	29
Şekil 14 Kurumsal Değişiklik Yönetimi Veritabanı Diyagramı	34
Şekil 15 Sisteme Giriş Ekranı	36
Şekil 16 Değişiklik Tanımlamaya Yetkili Kullanıcının Menüsü	37
Şekil 17 Değişikliği İncelemeye Yetkili Kullanıcının Menüsü	37
Şekil 18 Sistemde Yönetici Olan Kullanıcının Menüsü	38
Şekil 19 Değişiklik Tanımlama Ekranı	39
Şekil 20 Proje Liderine Gönderilen Bilgi E-Posta	40
Şekil 21 Değişiklik İnceleme Ekranı	41
Şekil 22 Değişikliği Tanımlayan Kullanıcıya Gönderilen İncelendi Bilgisi E-Posta	42
Şekil 23 Değişikliği Tanımlayan Kullanıcıya Gönderilen İptal Bilgisi E-Posta	42
Şekil 24 Değişikliği Tanımlayan Kullanıcıya Gönderilen Revise Bilgisi E-Posta	43
Şekil 25 Ajanda Oluşturma Ekranı	44
Şekil 26 Değişiklik İsteği Raporu	45
Şekil 27 Toplantı Katılımcılarına Davet E-Posta	46
Şekil 28 Etki Analizi Ekranı	46
Şekil 29 Değişikliği Atama Ekranı	47
Şekil 30 Değişikliğin Atandığı Kişiye Gönderilen E-Posta	48
Şekil 31 Değişiklik Kapatma Ekranı	48
Şekil 32 CIN Numarası Oluşturma Ekranı	49

TABLULARIN LİSTESİ

Tablo	Sayfa
Tablo 1 Değişiklik Yönetimini Destekleyen Uygulamaların Kullanım Oranı	3
Tablo 2 Değişikliğin Gerçekleştiği Aşamalar ve Bu Aşamalara Gösterilen Önem	4
Tablo 3 Tez Projesi ve Diğer Sistemlerin Karşılaştırılması	11

KISALTMALAR

Kısaltmalar	Açıklama
TOBB	Türkiye Odalar Borsalar Birliği
TDK	Türk Dil Kurumu
DB2	IBM Veritabanı
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
ECM	Engineering Change Management
TAI	Turkish Aerospace Industries
KDYS	Kurumsal Değişiklik Yönetim Sistemi
CIN	Configuration Identification Number
UML	Unified Modeling Language
RAD	Rational Application Development
J2EE	Java 2 Platform, Enterprise Edition
AJAX	Asynchronous JavaScript and XML
YUI	Yahoo User Interface
AJAX	Asynchronous JavaScript and XML
BSD	Berkeley Software Distribution
CSS	Cascading Style Sheets
HTTP	Hypertext Transfer Protocol
XML	Extensible Markup Language
MDC	Multi-Dimensional Clustering
PLM	Product Lifecycle Management
JDBC	Microsoft'un ODBC (open database connectivity- açık veritabanı bağlantısı) ara yüzünü taban alan SQL API' lerinin bir kümesi
MDY	Mühendislik Değişiklik Yönetimi

BÖLÜM 1

GİRİŞ

Modern işletmeler zaman kaybetmeden ve maksimum verimlilikle çalışabilmeleri için bilgi sistemleri kullanmak zorundadırlar. Bilgi sistemleri profesyonel bilgi teknolojilerini ve kurumlardaki iş süreçlerini bütünleştirir. Bilgi sistemlerinde sadece bilgisayarların rolünü vurgulamak yanlış olur. Pratikte, bilgi sistemlerinin çoğu insan ve bilgisayar temelli aktivitelerin bir karışımıdır. Bu karışımında insan ve bilgisayarın, sistemin başarılı bir şekilde çalışmasını sağlamak için birbirinden farklı fakat tamamlayıcı rolleri bulunmaktadır.

Günümüzde bilgi sistemleri, kurumlarda doğru kararlar alınmasında kullanılan teknolojiyle iç içe yapılandırılmış sistemlerdir. Temeli doğru kararlar alınmaya çalışılmasına dayanır. Kararların ne için alındığını bulmak bize bilgi sistemlerinin yararlarını ortaya çıkaracaktır. Kurumların bilgi sistemlerinden bekledikleri yararlar; daha iyi güvenlik, daha iyi hizmet, rekabet avantajları, daha az hata, daha yüksek doğruluk, daha yüksek ürün kalitesi, daha çok iletişim, artan etkinlik, artan verimlilik, daha etkin yönetim, daha çok fırsat, azalan çalışan ihtiyaçları, azalan maliyetler, finansal kararlarda daha üstün başarılar, operasyonlar üzerinde daha başarılı denetim ve daha başarılı yönetsel karar alma şeklinde sıralanabilir.

Bilgi sistemleri kabaca kurum için gerekli olan tüm bilgilerin toplandığı bir bilgi havuzudur ve sistem içinde bilgi girişi zamanında ve doğru olarak yapıldığı koşullarda ihtiyaçlara cevap verebilir. Kurulmuş olan otomasyon bilgiye hızlı ve daha az maliyetle ulaşmayı sağlar.

Klasik sistemlerde verilerin kayıtları ve bilgiye ulaşma süreçlerinde kağıt kalem kullanıldığında ya da mekanik aygıtlar ve insan gücü kullanıldığından dolayı bilgilerin istenildiği anda elde edilmesi mümkün olamamaktadır. Ayrıca işlemlerin sayısı arttıkça klasik sistemler buna yanıt verememektedir. Maliyetler ve diğer açılardan sistem zorlanmaktadır.

Bilgisayara dayalı bilgi sistem tipleri aşağıda sıralanmıştır [1];

- Kayıt-Veri İşleme Sistemleri,
- Yönetim Bilgi Sistemleri,
- Karar Destek Sistemleri,
- Ofis Otomasyon Sistemleri,
- Yapay Zeka ve Uzman Sistemleri,
- Üst Yönetim Destek Sistemleri.

Bu tezde anlatılan değişiklik yönetim sistemi aşağıda belirtilen ortak özelliklerinden dolayı *Yönetim Bilgi Sistemleri* sınıfına dahildir.

Değişiklik Yönetim Sistemi,

- Kurumun yönetiminde kullanılan bilgilerin işlenmesi ve iletilmesini sağlar.
- Veri/Kayıt işleme fonksiyonlarını destekler(kayıt saklama vb.) ve bütünleşik bir veritabanına sahiptir.
- Operasyonel, taktik ve stratejik seviye yöneticilerin bilgiye kolay ve zamanında erişimini sağlar.
- Esnektir ve organizasyonun bilgi ihtiyaçlarındaki değişmeye adapte edilebilir.
- Sadece yetkili şahısların erişimine imkan veren sistem güvenliği sağlar.
- Yöneticilere değişik raporlar sunar.
- Standart bir donanım veya yazılım paketi değildir.

1.1 DEĞİŞİKLİK YÖNETİM SİSTEMLERİNİN TARİHİ

Değişiklik, bir bütünün bir bölümünün değişmesiyle ortaya çıkan yeni durum olarak tanımlanır. Bu değişiklikler için standartlaştırılmış metotların, süreçlerin ve prosedürlerin kullanılmasına da değişiklik yönetimi denilmektedir.

Değişikliklerin yönetilmesi ihtiyacı ilk olarak 1980'li yıllarda ortaya çıkmıştır. Bundan önceki yıllara bakıldığı zaman üretim ve tasarım yapan firmaların ürün geliştirme süreci aşamalarında bilgisayar destekli yazılımlar kullanmadıkları görülmektedir. Değişiklik yönetimi, iş akışı yönetimi, veri ve doküman yönetimi gibi bunlara benzer işlemlerin çoğunlukla kağıt üzerinde takip edildiği, onay süreçlerinin posta yoluyla yapılıp, raporların elden çıkarıldığı bir dizi süreçten oluştuğu gözlenmiştir. Süreçlerin bu şekilde takibi, firma içerisindeki tasarım ve üretim

aşamalarında kontrol eksikliği yaratmakta ve beraberinde birçok problemi getirmektedir. Örneğin yanlış bir çizim ürünün fiziksel özellikleri hakkında yanlış bilgi verebilmektedir. Bu gibi yanlışlıklarla birlikte üretimde yapılan iyileştirmeler ve teknolojik nedenlerden dolayı oluşabilecek birçok değişiklik isteği, piyasalardaki rekabetin artmasıyla sürekli değişen ve gelişen şartlar içinde şirketin hayatta kalmak zorunda olması firmalarda değişikliklerin yönetilmesi ihtiyacını doğurmuştur.

1980’li yıllardan sonra bu ihtiyacı karşılamak amaçlı Bilgisayar Destekli Tasarım (*CAD*) ve Bilgisayar Destekli Üretim (*CAM*) yazılımları yaygın olarak ön plana çıkmaya başladı. Fakat zamanla veriler çoğaldı, kontrol edilmesi zorlaştı, disiplin altına alınması gerekti. Tasarım ve imalat süresini kısaltıp oradan tasarruf sağlayarak ürünün ömrünü uzatmaya yönelik sistemlere ihtiyaç duyulmuştur.

Dünya çapındaki lider firmalar oluşan değişikliklerin yönetilmesine, sanayi şirketleri ve diğer ufak şirketlere göre daha çok önem vermektedir. Bu amaçla lider firmaların değişiklik yönetimini destekleyen ürün veri yönetimi, ürün yaşam döngüsü yönetimi gibi uygulamaları da diğer firmalara göre daha çok önemsendiği görülmektedir. Bu uygulamaların farklı firma kategorilerindeki kullanım oranları Tablo 1 de gösterilmiştir [21].

Tablo 1. Değişiklik Yönetimini Destekleyen Uygulamaların Kullanım Oranı

Uygulamalar	Lider Firma	Sanayi Şirketi	Diğerleri
Ürün Veri Yönetimi (<i>PDM</i>)	%71	%39	%39
Ürün Yaşam Döngüsü Yönetimi (<i>PLM</i>)	%41	%19	%17
Tasarım ve Görsel Uygulamalar	%75	%32	%17

Lider firmalar rakiplerine göre iki önemli değişiklik adımını kendi süreçlerinde uygulamaktadır. Bu adımlar etki analizi ve değişiklik uygulama planıdır. Etki analizi adımına hemen hemen beş kat daha fazla önem veren lider firmalar bu sayede daha doğru ve hızlı karar verme yeteneğine sahip olmaktadır.

Lider ve diğer firmalar değişiklikleri yaparken Tablo 2 deki değişikliğin gerçekleştiği aşamalara dikkat etmişlerdir [21].

Tablo 2. Değişikliğin Gerçekleştiği Aşamalar ve Bu Aşamalara Gösterilen Önem

Değişikliğin Gerçekleştiği Aşama	Lider Firma	Sanayi Şirketi	Diğerleri
Gereksinim aşamasında olması	%82	%61	%45
İmalat aşamasında olması	%94	%70	%60
Sipariş ve taleplerde olması	%65	%54	%35
Üretim aşamasında olması	%94	%78	%70
Dokümanlarda olması	%65	%50	%45
İlişkili parçalarda olması	%94	%78	%70
Montaj aşamasında olması	%94	%78	%70
Eskimiş parçalarda olması	%59	%63	%55

Tablo 2 de görüldüğü gibi lider firmalar, sanayi şirketi ve diğer firmalara göre değişikliğin gerçekleştiği tüm aşamalara daha fazla önem göstermişlerdir. Örneğin bir değişikliğin imalat aşamasında ortaya çıkmasını lider firma %94 önemserken, sanayi şirketleri %70 diğer firmalar ise ancak %60 önemsemektedir. Değişiklik ürün yaşam döngüsünün başlarında meydana geliyorsa firmalar için bu değişikliğin önemsenmesi azdır fakat üretimin son aşamalarında meydana gelmiş ise firmalar bu değişikliğe çok daha fazla önem göstermektedirler. Bunun en büyük sebebi ise ürün yaşam döngüsünün sonlarına gelindikçe oluşabilecek değişikliğin etkilerinin artmasıdır.

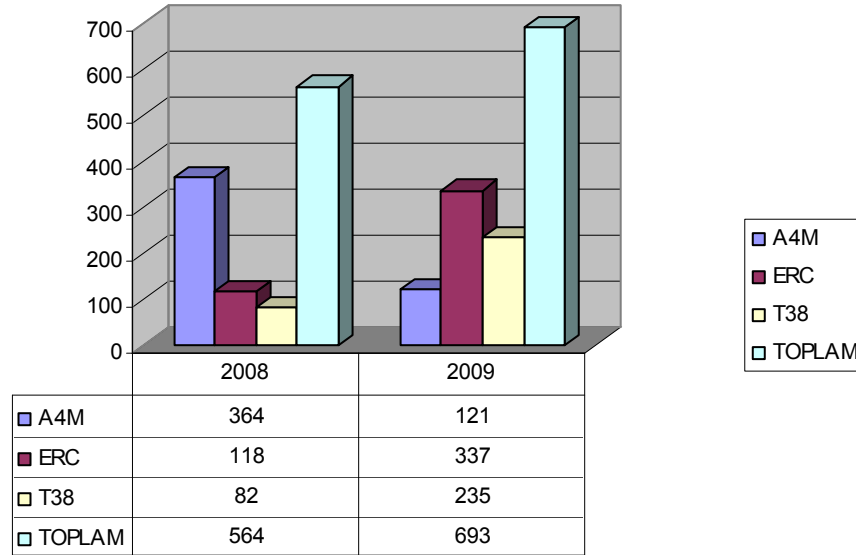
1.2 DEĞİŞİKLİK YÖNETİM SİSTEMLERİNİN ÖNEMİ

Karmaşık tasarlanan süreçlerde değişikliklerin hayata geçirilmesi büyük ölçüde zamana ve maliyete yol açmaktadır. İşte bu aşamada proje çalışma ekipleri için Değişiklik Yönetim Sistemleri değişikliklerin yönetilebilmesi için bir çözüm olarak ortaya çıkmaktadır.

Tasarım veya üretim yapan herhangi bir firma için her gün yaklaşık olarak proje başına en az 1 tane değişiklik geldiğini varsayalım. 5 farklı proje yöneten bu firmada günlük gelen değişiklik sayısı 5 olacaktır. Bir ayda bu rakam 150 değişikliğe kadar ulaşacaktır, her değişikliğin etkilediği başka parçalar ve bölgelerinde olacağı hesaba katılırsa bu firmanın her ay yaklaşık olarak 300-500 tane değişikliği doğru, güvenilir ve hızlı bir şekilde yönetmesi gerekir. Her değişiklik için gerekli olan tüm adımların sırasıyla uygulanmasının gerekliliği düşünüldüğünde, bu süreç zaman ve emek

kaybına sebep olacaktır. İşte tamda bu noktada Değişiklik Yönetim Sistemlerinin önemi ortaya çıkar.

Türkiye’de bulunan ve havacılık sanayisine öncülük eden TUSAŞ-Türk Havacılık ve Uzay Sanayi A.Ş. (TAI) firmasından bir örnek verelim. Firmanın şu anda aktif olarak yürüttüğü yaklaşık 130 projesi olduğu bilinmektedir, buna göre yukarıdaki istatistiklerin 20-30 kat artacağı yani ayda ortalama 5000-10000 değişikliğin oluşacağı ve maliyet açısından şirketi çok yüksek zararlara götürebileceği ortadadır. Şekil 1 deki grafikten de görüldüğü gibi TAI’den örnek olarak alınmış üç farklı projenin 2008 ve 2009 yıllarında meydana gelen değişiklik rakamları verilmiştir. Sadece bu örnekte bulunan üç proje için bile yıllık rakamların ne kadar yüksek olduğu görülmektedir [3].



Şekil 1. TAI Firmasındaki 3 Projenin Yıllara Göre Değişiklik İstek Dağılımı

Şu anda birçok firmada değişiklik yönetimi kağıt ortamda yapılmaktadır ve bu kağıtlar elden ele bir imza sürecinden geçtikten sonra gerekli görüşmeler yapıp ilgili kişilere yazılan e-postalarla atama işlemleri gerçekleştirilmektedir. Fakat bu değişiklikler hangi süreçlerden geçmiş, hangi aşamada bekliyor, ne gibi çözümler üretilmiş, kimlere atanmış ve atanmış kişiler ne kadar zamanda bu değişiklik isteklerini gerçekleştirmiş gibi önemli sorulara verilen yanıtlarda hep bir eksiklik ve güvensizlik ortamı ile karşı karşıya gelinmektedir. Değişiklik yönetim işlemlerini elektronik ortama taşımayan firmalar Ek 1, Ek 2 ve Ek 3 de verilen formlara benzer formları kullanmak zorundadırlar ve bahsedilen tüm riskleri kabullenmişlerdir [3].

1.3 DOKÜMANA BAKIŞ

Dokümanın ikinci kısmında, deęişiklik yönetim sistemleri hakkında genel bir bilgi verilecektir. Őu anda dünyada ve Türkiye’de kullanılan benzer sistemlere örnekler verilip tez kapsamında tasarlanan sistem bu sistemlerle karşılaştırılacaktır. Ayrıca mühendislik yönetimi açısından deęişiklik yönetim sistemlerine bakılıp, endüstriyel alanlardaki kullanımlarından bahsedilecektir. Bölümün sonunda ise tasarım ve üretim süreçlerindeki deęişiklik nedenlerine değinilecektir.

Üçüncü kısımda, tezde anlatılan kurumsal deęişiklik yönetim sistemi ile ilgili olarak detaylı bilgi verilecektir ve izlemiş olduęu tüm fazlar gösterilecektir.

Dördüncü kısımda, hazırlanan yazılımın mimarisi hakkında bilgi verilecek ve yazılımın teknik detaylarından bahsedilecektir.

Beşinci kısımda, konunun daha iyi anlaşılabilmesi için örnek bir uygulama alanı üzerinde, ekran görüntüleri ile birlikte önemli bazı senaryolar detaylı bir biçimde anlatılacaktır.

Son kısımda ise sonuç ve gelecek çalışmalar anlatılacaktır.

BÖLÜM 2

DEĞİŞİKLİK YÖNETİM SİSTEMLERİ

Değişiklik Yönetim Sistemleri Türkiye’de ve dünyada aşamalı olarak hayata geçmiştir. Kronolojik olarak ilk sırada, tasarım ve üretim süreçlerinde kullanılan verileri kontrol etmek ve oluşan değişiklikleri takip etmek amaçlı Ürün Veri Yönetimi (*PDM*) sistemleri geliştirilmeye başlandı.

1990’lı yılların sonuna kadar PDM sistemlerin kullanılmasına devam edildi fakat zamanla verilerin çoğalması ve kontrol edilmesi zorlaşınca ilk kez IBM firması tarafından Ürün Yaşam Döngüsü Yönetimi (*PLM*) sistemleri geliştirildi. PLM bir ürünün fikir olarak ortaya çıkmasından itibaren planlanması, geliştirilmesi, üretilmesi, pazara sunulması, satış sonrası destek hizmetlerinin yönetilmesi gibi ürünün tüm ömrü boyunca geçtiği her aşamanın yönetilmesini sağlayan bir iş ve bilgi stratejisi olarak kabul gördü [5].

PLM o zamandan beri hem IBM, hem kavramı kabul eden diğer satıcı firmalar, hem de imalat uygulamaları pazarında faaliyet gösteren danışmanlık şirketleri tarafından geliştirilmeye devam edildi. Şu anda kabul görmüş bir iş stratejisi ve vizyonu olduğu görülmektedir.

Günümüzde ise PLM çözümleri bu kabiliyetlerine ek olarak bir yenisini ekledi. PLM çözümleri tasarım ve üretim aşamalarında oluşabilecek değişiklikleri takip altına alarak tasarımın zamanını kısaltıp, ürünün ömrünü uzatmaktadır. Dolayısıyla, zamandan ve malzemeden tasarruf edip maliyeti azaltıyor, en önemlisi de işin kalitesini ve güvencesini arttırmaktadır.

Dünyada ve Türkiye’de PLM projesi kullanan birçok firma bulunmaktadır. En çok kullanılan PLM yazılımlarına örnek olarak SmarTeam firmasının Enovia yazılımı, SoftTech firmasının ProductCenter yazılımı ve Siemens firmasının TeamCenter yazılımı verilebilir.

ENOVIA:

SmarTeam firmasının Enovia yazılımı, tasarım ve üretim yapan firmalara ürün yaşam sürecinde oluşan tüm değişiklikleri yönetebilmesi ve bir akış diyagramı haline getirebilmesi olanaklarını sunar. Kullanıcılar tüm ürün bilgisine ve ilgili dokümantasyona ulaşabilirler. Dokümanlar güvenli bir yapı içinde saklanır ve esnek bir veritabanı ortamı içinde ulaşılabilirler. Veri bütünlüğü ve güvenliği sağlanmış olduğundan yanlış erişimler önlenir. SmarTeam sayesinde tasarımcılar zamanlarını veri yönetimine harcamaksızın yalnızca tasarımla uğraşırlar.

Dünyada ve Türkiye de bu programı kullanan şirketler [6] arasında

- Lockheed Martin; F35 Joint Strike Fighter Program,
- Thales Avionics,
- Gulfstream Aerospace,
- Israel Military Industries,
- Eads Airbus, ve
- Tübitak sayılabilir.

PRODUCTCENTER:

SoffTech yazılım firmasının geliştirmiş olduğu ProductCenter bileşeninde kullanıcılara PLM çözümler sunulmaktadır. Bu çözümler içerisinde tezde tasarlanan sistemin yapmış olduğu Süreç Yönetimi ve Değişiklik Yönetimi işlemleri ile birlikte e-posta bildirimleri de bulunmaktadır.

Dünyada bu programı kullanan bazı şirketler [7] şunlardır.

- AgustaWestland
- General Electric Company
- NASA Glenn Research
- Siemens Energy & Automation
- Sikorsky Aircraft
- U.S. Army
- United Technologies
- Volkswagen AG

TEAMCENTER:

Siemens Industry Automation Division'ın bir birimi olarak faaliyet gösteren Siemens PLM yazılımı, ürün yaşam-döngüsü yönetimi konusunda dünya genelinde yazılım ve hizmetler sunan lider bir firmadır. Etkili bir mühendislik değişikliği yönetimi sayesinde, süreçler çok daha esnek bir şekilde yönetilir. Pazardan gelebilecek talep değişiklikleri, gerçekleştirilecek üretim hataları çabuk ve etkili bir şekilde ele alınarak giderilir ve değişiklik ürüne çok kısa sürede yansıtılarak üretim maliyetlerinde oluşacak kayıplar elimine edilmiş olur. Bu özelliği sayesinde de Siemens PLM yazılımının TeamCenter paketi tezde tasarlanan sistemin yapmış olduğu işlemlere benzer özellikler içermektedir.

Dünya genelinde bakıldığında, uçak, uzay ve otomotiv sanayinde Boeing, Daimler-Chrysler ve General Motors gibi 50.000 kullanıcıya kadar çıkan birçok referansı vardır [2].

Türkiye’de bu programı kullanan şirketler arasında;

- Arçelik,
- Tofaş,
- Ford,
- Beko,
- Tübitak Sage,
- Standart Profil, ve
- Tübitak Uzay vardır.

Yukarıda bahsedilen yazılımlar dünya çapında kabul görmüş çok büyük maliyete sahip sistemlerdir. Bu sistemlerin şirket yapılarına uyarlanması çok zor olmaktadır, yazılımların büyük oranda özelleştirilmesi, organizasyonel değişikliklerin uyarlanması gerekir ve bu da çoğu zaman ayları ya da yılları bulabilir. Bu tezde anlatılan sistem yukarıda örnek verilen sistemlere bir alternatif olarak geliştirilmiş ve bu sistemlerin temel özelliklerini kendi bünyesinde barındırıp bu sistemlere karşı bir takım avantajlar kullanıcıya sunmuştur. Bu avantajlar;

Tam Uyum ve Konfigürasyon: Tezde tasarlanan sistem en ince detayına kadar şirket için ayarlama yapmaya olanak sağlar, şirketin ihtiyaçlarına göre özelleştirilebilir işe yaramayan kısımları çıkartıp yenilerini anında eklenebilir, bu konuda sınır yoktur. Şirkette kullanılan tüm diğer projelerle tam bir uyumluluk içinde çalışır. Var olan diğer yatırım ve süreçlerden yararlanılabilir.

Geniř Destek: Kullanıcıların her zaman destek, eğitim alabileceđi kişiler sadece bir telefon uzađındadır ve her zaman görüşme, toplantı yapma olanađı vardır.

Ücretsiz Kullanım: Kurumlara özel tasarlanan bu yazılım yüz binlerce dolar yerine tamamen şirketin kendi bünyesinde geliştirilmeye veya dış kaynaklar yardımı ile çok daha küçük maliyetlere geliştirilmeye müsaittir ve bu sayede şirkete çok büyük kar sağlar.

Sürekli Güncel: Herhangi bir sorunla karşı karşıya kalındığında güncelleme paketlerinin yıllarca beklenilmesine ya da ücret ödenmesine gerek yoktur. Sürekli güncel olarak kullanılabilir.

Tam Kontrol: Sistem üzerinde tam anlamıyla kontrol yazılımcıdadır. Sistem ve yazılımların tüm bileşenlerini ayrı ayrı inceleyebilir, kontrol edebilir, bu uygulamalardan maksimum faydayı sağlamak için her zaman altın kuraldır.

Kullanım Kolaylığı: Yahoo'nun kullanmış olduđu ve kendini dünya üzerinde kanıtlamış olan Yahoo User Interface (YUI) ara yüzleri ve kütüphaneleri kullanılarak kullanıcıya estetik görünüm ve kullanım kolaylığı sunulmuştur.

Kurulum: Tamamen web temelli olması nedeniyle kurulum ve bakım maliyetlerini azaltır. Son kullanıcının bilgisayarına kurulması gerekli bir program yoktur. Kullanım için sadece internet tarayıcısı yeterlidir.

Tablo 3. Tez Projesi ve Diğer Sistemlerin Karşılaştırılması

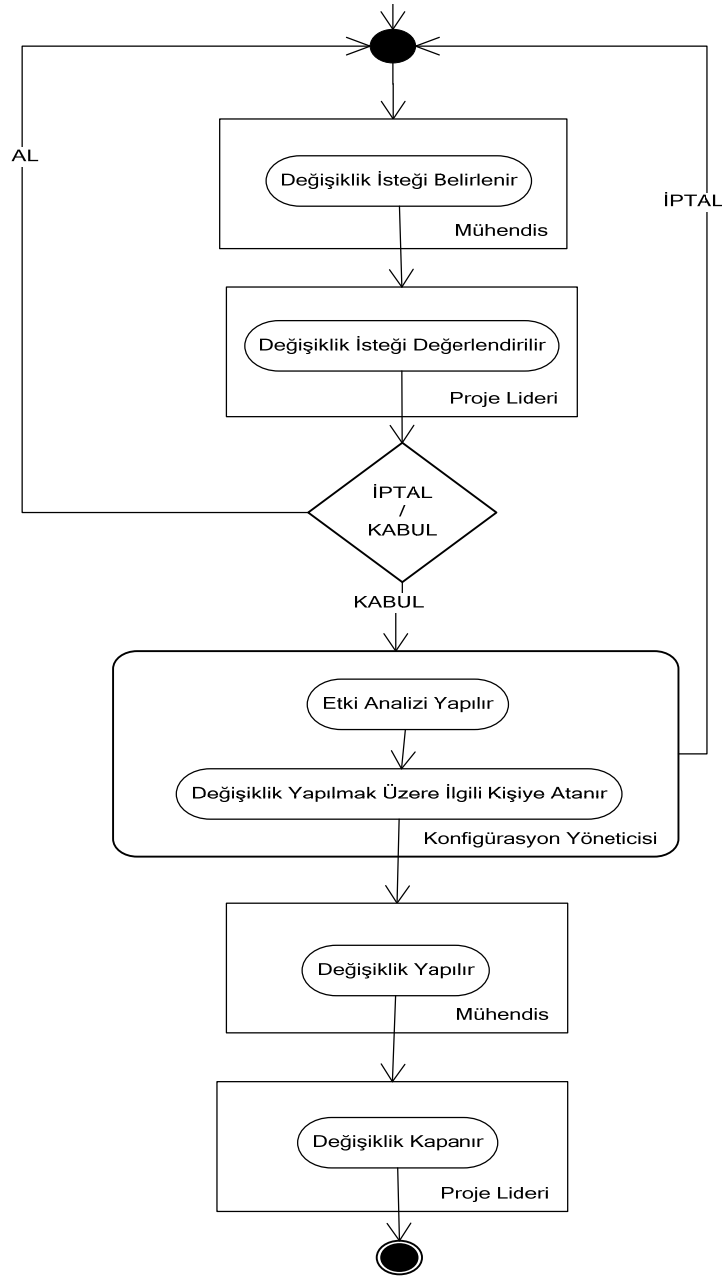
YAZILIM ÖZELLİKLER	TEZ PROJESİ (KDYS)	ENOVIA (SMARTEAM)	PRODUCTCENTER (SOFTECH)	TEAMCENTER (SIEMENS PLM)
Farklı Sektörlerde Kullanılabilirlik	√	√	√	√
Firmaya uyum ve Tam Konfigürasyon	√	X	X	X
Firmaya çabuk uyarlanabilirlik	√	X	X	X
İlave bileşenlerle desteklenme	X	√	√	√
Revizyon takibinin yapılması	√	√	√	√
Düşük Maliyet	√	X	X	X
Anında Güncellik	√	X	X	X
Raporlama	√	√	√	√
Simülasyon	X	X	X	√
MultiCAD özelliği	X	√	√	√
Yetkilendirme	√	√	√	√
Kullanım Kolaylığı	√	X	X	X
Web Tabanlı Olması	√	X	X	X

Tablo 3'den de görüldüğü gibi tez projesi firmaya uyum, firmaya uygulanabilirlik süresi, maliyet, güncellemeler, kullanıcı isteklerinin anında yapılması, kullanım kolaylığı ve web tabanlı bir uygulama olması açısından diğer sistemlere avantaj sağlamaktadır.

Dünyada çeşitli açık kaynak kodlu değişiklik yönetimi yazılımları mevcuttur. Fakat bu yazılımları kullanmak kurumsal firmalar için hem bir güvensizlik ortamı oluşturmakta hemde bu yazılımların kodlarının anlaşılıp kuruma uyarlanması çok

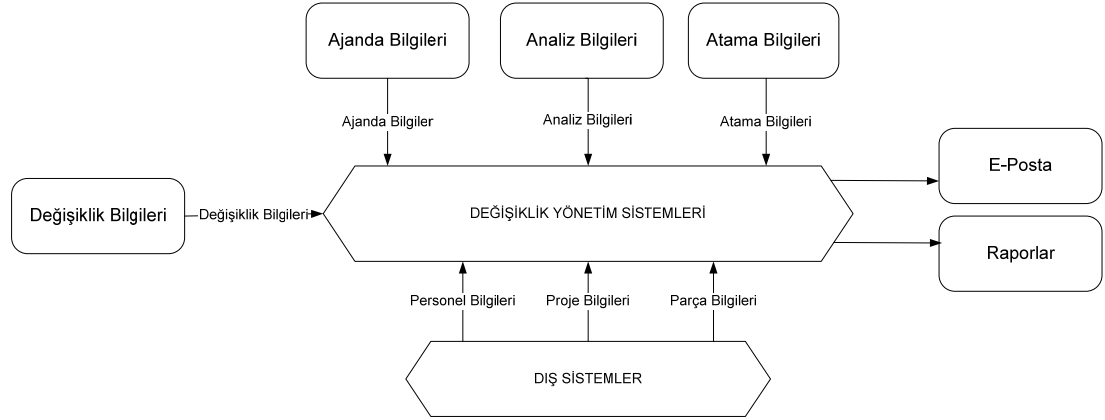
büyük zaman ve emek kaybına yol açmaktadır. Bunun yerine kurumun kendi yazılım ekibi veya dış kaynaklar tarafından bu yazılımı geliştirmesi kurum için büyük kar sağlayacaktır.

Bahsedilen bu sistemler ile diğer değişiklik yönetimi sistemlerinin ortak iş süreç yapısı Şekil 2 deki iş akış diyagramında gösterilmiştir. Bu diyagramda bir değişikliğin tanımlanmasından kapanmasına kadar geçen adımlar gösterilmiş ve bu adımlardan sorumlu kişiler belirtilmiştir [8].



Şekil 2. Değişiklik Yönetim Sistemleri İş Akış Diyagramı

Değişiklik Yönetim Sistemlerinin girdi ve çıktı bilgileri Şekil 3 de gösterilmiştir. Bu şekle göre sisteme girdi olarak; değişiklik bilgileri, ajanda bilgileri, analiz bilgileri, değişikliğin yapılacağı atama işlem bilgileri, sistemde tanımlı kullanıcı, proje ve parça bilgileri tanımlanır. Çıktı olarak ise; çeşitli raporlar ile kullanıcıları bilgilendirici e-posta'lar vardır.



Şekil 3. Değişiklik Yönetim Sistemleri Girdi ve Çıktı Diyagramı

2.1 TASARIM VE ÜRETİM SÜREÇLERİNDEKİ DEĞİŞİKLİK NEDENLERİ

Tasarım ve üretim süreçlerindeki değişiklikler çeşitli nedenlerden kaynaklanabilmektedir. Bunlar arasında aşağıdaki nedenler başlıcalarıdır.

- Bir ürünün tasarımı sırasında modeller ve resimler üzerinde meydana gelen değişimlerden kaynaklı olabilir.
- İyileştirmeler sonucu yeni değişiklikler ortaya çıkmış olabilir.
- Bir parçada yapılan bir değişiklik diğer parçaları etkileyebilir.
- Üretimde meydana gelen sorunlardan kaynaklı değişiklikler oluşabilir.
- Teknolojik nedenlerden dolayı değişiklik meydana gelmiş olabilir.
- Tasarımı dışarıdan sağlanan bir ürün için tasarımı yapan firmanın yapmış olduğu değişiklikler üretim hattına yansiyabilir.

Değişiklikler nitelik ve nicelik olarak farklılık arz ederler. Bazı örnek değişiklik durumları aşağıda listelenmiştir.

- Tasarımda hazırlanmış olan bir dokümanın başlığının değişmesi.
- Teknik resimden kaynaklanan bir hata sonrası oluşan değişiklik.
- Kullanılan parçaların uzatılıp, kısaltılması sonucu oluşan değişimler.
- Üretimde uygun kalıp bulunamadığı için büküm çapı değişikliği.
- Üretim sürecinde montaj aşamasında bir deliğin çapının değişmesi.
- Uçağın iniş takımlarında oluşan bir değişiklik.
- Uçak tasarımının da kullanılan motorun modelinin değişmesi.

2.2 MÜHENDİSLİK YÖNETİM SİSTEMLERİ AÇISINDAN DEĞİŞİKLİK YÖNETİM SİSTEMLERİ

Bugünün yüksek rekabet ortamında şirketler hayatta kalabilmek için tasarım süreçlerine çok önem vermektedirler. Bunu sağlamak için, müşteri isteklerine uygun üretim yapmaları gerekir. Bu da ancak tasarım süreçlerindeki işlemlerin daha kısa sürede gerçekleşmesiyle sağlanabilir.

Şirketler dış kaynak kullanımı ile genişleyen işlemler için çalışanlar arasında etkin bir iletişim ortamına ihtiyaç duyarlar. Bu iletişim sadece bir fikir paylaşımından ibaret olmayıp aynı zamanda mühendislik değişiklikleri de olabilmektedir.

Bir ürünün tasarımı sadece geliştirilmesi ile sınırlı değildir. Ürün, yaşam döngüsü boyunca birçok değişikliğe uğrar. Bu değişiklikler formlardaki modifikasyonlar, düzeltmeler, parametrelerdeki boyutların değişmesi olabilir, işte bu gibi değişikliklere mühendislik değişiklikleri denir ve Mühendislik Değişiklik Yönetimi (*MDY*) ile takip edilir [24].

Mühendislik değişiklik yönetimi işlerin organizasyon, kontrol, iş akış, bilgi akış gibi işlemlerini yönetir.

MDY işlemleri üç temel faza sahiptir;

Değişiklik İstek Fazı: Bu faz değişikliği sisteme tanımlayan kişi tarafından başlar. Bu kişi bir bölümden veya mühendislik takımından bir üye olabilir. Değişikliğin nerede oluştuğunu, hangi özelliklere sahip olduğunu, niçin meydana geldiğini ve eklentilerini içerir.

Değişiklik Onay Fazı: Değişiklikleri çeşitli bölümlerden oluşan bir kurul inceler. Bu kurul değişikliğin diğer alt ve ilişkili sistemleri ne kadar etkilediğinin incelemesini yapıp, MDY döngüsünde kritik açıdan önemini belirler.

Değişiklik Bildirim Fazı: Tüm kurul üyeleri tarafında onaylanan bu değişiklik yapılmak üzere koordine edilip ilgili bölümlere bildirilir.

MDY süreci organizasyondaki karmaşık ve sorunlu süreçlerden birisidir [25]. Oluşan değişiklikler üretim sürecinin %8,5'lik bir kısmını alır [26].

MDY sürecinde meydana gelen büyük zorluklar şunlardır;

- MDY sürecinin karmaşık ve çok boyutlu yapısı karar kurulundaki katılımcılar tarafından değişikliğin detaylı ve derinlemesine bir şekilde incelenmesini gerektirir.
- Kuruldaki katılımcılardan bazıları yeterli düzeyde teknik bilgiye sahip olmadıkları için teknik çizimlerde önemli hatalara sebep olabilir ve konuları anlamakta zorluk çekebilirler.
- MDY sürecindeki grafiksel ve parametrik işler her zaman için ortak iletişim kurularak çözümlenmelidir, aksi takdirde çift kayıt oluşabilir [27].
- Değişikliklerin etkilediği diğer bölgeleri bulmak için karar mekanizmalarının daha detaylı bir inceleme yapması gerekir bu da bizi *Change Propagation*'a götürür ve bu zor bir analizdir. *Change Propagation*; çalışma ürün bilgilerinin sentetik modellerini oluşturmaya ve sistem bileşenleri arasındaki ilişkiyi tanımlamaya yardımcı olur, bu da fonksiyonel bağımlılıkların belirlenmesini sağlar [28].
- Ürünün yaşam döngüsünün uzun sürmesi durumunda değişiklikler artar, bu da önceliklerin belirlenip koordine edilmesini gerektirir.

- Tarih tekerrürden ibarettir, önceden incelenen ve sonuca bağlanan değişiklikler, bir öncekinin varlığından habersiz şekilde tekrar karşımıza çıkabilir. Bu iki şekilde olur; birincisi değişik ürün ailelerinde, ikincisi ret edilmiş değişikliklerde, her iki durumda da geçmiş tecrübeler çok değerli olan zamanı bize kazandırır.

Yukarıda belirtilen tüm bu zorluklar için endüstri alanında, standart değişiklik yönetimi kalıplarına ek olarak farklı kabiliyetler kazandırılıp yeni yaklaşımlar geliştirilmiştir. Bu yaklaşımlardan en önemlisi Akgündüz ve Koçar'ın makalesinde anlatılan “A Virtual Environment for Engineering Change Management (*ADVICE*)” sistemidir [24]. Bu sistemde standart yöntemlere ek olarak; değişiklik süreçlerini kavramsal ve somut tasarım ile desteklemek için “Virtual Collaborative Design Environments (*VCDE*)” ve daha önceden sisteme tanımlı verilerden yola çıkarak öncelendirme yapmak için “Sequential Pattern Mining” teknikleri kullanılmıştır.

MDY süreçlerini modellemek için VCDE'nin kullanılması yönetimi daha önceden kullanılmamış bir metottur ve bu yöntemde kümeleme ve veri analizi gibi veri madenciliği metotları da kullanır [29]. Veri madenciliği metotlarının kullanılmasının asıl amacı depolanan verilerden yola çıkarak değişikliğin önceliğini belirlemektir.

Bu sistem sayesinde kullanıcıların konuyu daha iyi anlaması sağlanıp, dolayısıyla zamandan tasarruf edilmiş olur. Oluşan değişikliğin etkileri daha önceden tahmin edilip, önceliklerin daha rahat belirlenmesini sağlayarak kullanıcının işi kolaylaştırılmıştır.

Bu sistemin temeli geçmiş tecrübeleri irdelemeye dayanır. Veritabanında bulunan detaylı analiz raporları arasındaki benzer değişikliklerin bulunması ile karar verenlere doğru karar vermeleri sağlanır.

Tasarlanan bu sistemin asıl hedefleri;

- Temel değişiklik süreçlerini grafiksel olarak yönetebilmek.
- Sisteme girilmiş önceki verilerden yola çıkarak önceliklendirme yapmak.
- Değişikliğin etkilediği diğer parçalar hakkında tahmin yaparak bağımlılıkları belirlemek.
- Geçmişte aynı değişiklik tanımlanmış ise aynı işlemlerin tekrar yapılmasını engellemektir.

Büyük endüstrilerde oluşan ürünün yaşam döngüsü boyunca oluşan değişiklikler beraberinde de binlercesini tetikler, bunların aralarındaki ilişkileri görmek karar takımları için çok zor olmaktadır. Rouibah ve Caskey [31] bu işlem için parametrik bir yaklaşım geliştirmiştir. Burada tasarım parametreleri ve bunların ilişkileri önceki projelerdeki sonuçlardan oluşan bir parametre ağında tanımlanır ve bu da karar takımlarına büyük kolaylık sağlamış olur.

Stojanovic ve Stojanovic [30] bu sistem sayesinde bilgiye erişimde 10 dakikadan daha fazla ve büyük bir şirket için ise yılda 100 milyon dolardan daha fazla kar sağlanacağını vurgulamıştır.

Bu yaklaşım karmaşık verilere ve öncelikli olan değişiklik isteklerine erişimi modelleyerek mühendislerin zaman ve efor kaybetmesinin önler.

BÖLÜM 3

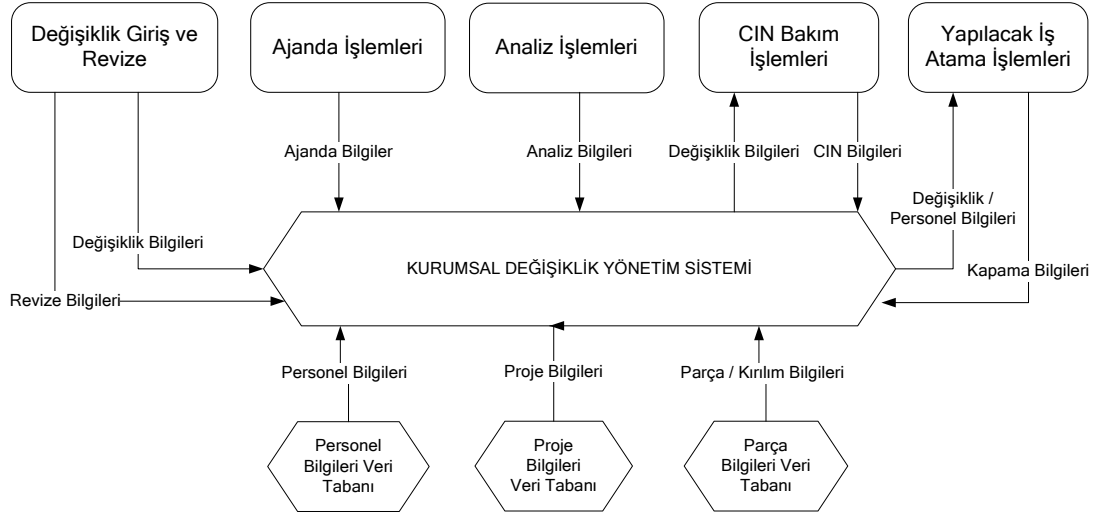
KURUMSAL DEĞİŞİKLİK YÖNETİM SİSTEMİ

Üretim, imalat ve tasarım süreçleri içerisindeki karmaşık işlemlerde ortaya çıkabilecek çok küçük değişiklikler süreçte büyük etkilere yol açar. Kurumsal Değişiklik Yönetim Sistemi (*KDYS*) bu değişikliklere bilimsel bir yaklaşımla, modern anlamda bir mühendislik yaklaşımı ile ele alarak yapılan işin kalitesini, güvencesini, hızını, organizasyonunu, yönetilmesini, mevcut yapıya uyumluluğunu, gelişmeye açık olması ve uygun değer maliyetle işlemin yapılmasını amaçlar.

3.1 KURUMSAL DEĞİŞİKLİK YÖNETİM SİSTEMİNE GENEL BİR BAKIŞ

Kurumsal Değişiklik Yönetim Sistemi ilk olarak değişikliğin sisteme girişi ile başlar, daha sonra sisteme girilen bu değişikliklerin analizinin yapılacağı bir ajanda oluşturulduktan sonra belirlenen saat, tarih ve yerde toplantı yapılarak değişiklik konuları konuşulur. Toplantı sonucunda yapılması planlanan değişiklikler, ilgili kişilere atanarak süreç devam eder. Bu süreç sırasında *KDYS* Yazılımı Personel Bilgileri Veritabanı, Proje Bilgileri Veritabanı ve Parça Bilgileri Veritabanları ile etkileşim içerisinde. Personel Bilgileri Veritabanından kullanıcıların yetki ve kimlik bilgileri alınır. Proje Bilgileri Veritabanından değişikliğin geçtiği proje bilgilerine ulaşılır, Parça Bilgileri Veritabanından ise değişiklik yapılacak parça bilgileri ile kırımlar bilgilerine erişilir, kırımlar bilgileri değişikliğin ürünün hangi bölgelerinde meydana geldiği hakkında kullanıcıya bilgi verir. Kapatılan değişikliklerden bir Yapılandırma Kimlik Numarası (*CIN*) oluşturulur, bu *CIN* numarası diğer sistemler tarafından referans olarak kullanılır. Süreç esnasında otomatik e-posta'lar ile kullanıcıyı bilgilendirici mesajlar ve bu mesajlar içerisinde hatırlatma ekleri gönderilir ve değişikliğin her aşamasında sistemden çekilebilen raporlar sayesinde kullanıcı değişikliğin tüm fazları hakkında detaylı bilgilendirilir.

KDYS'nin girdi ve çıktı bilgileri Şekil 4 de gösterilmiştir;



Şekil 4. KDYS Sistemine Genel Bir Bakış

3.2 OLUŞAN BİR DEĞİŞİKLİĞİN İZLEDİĞİ FAZLAR

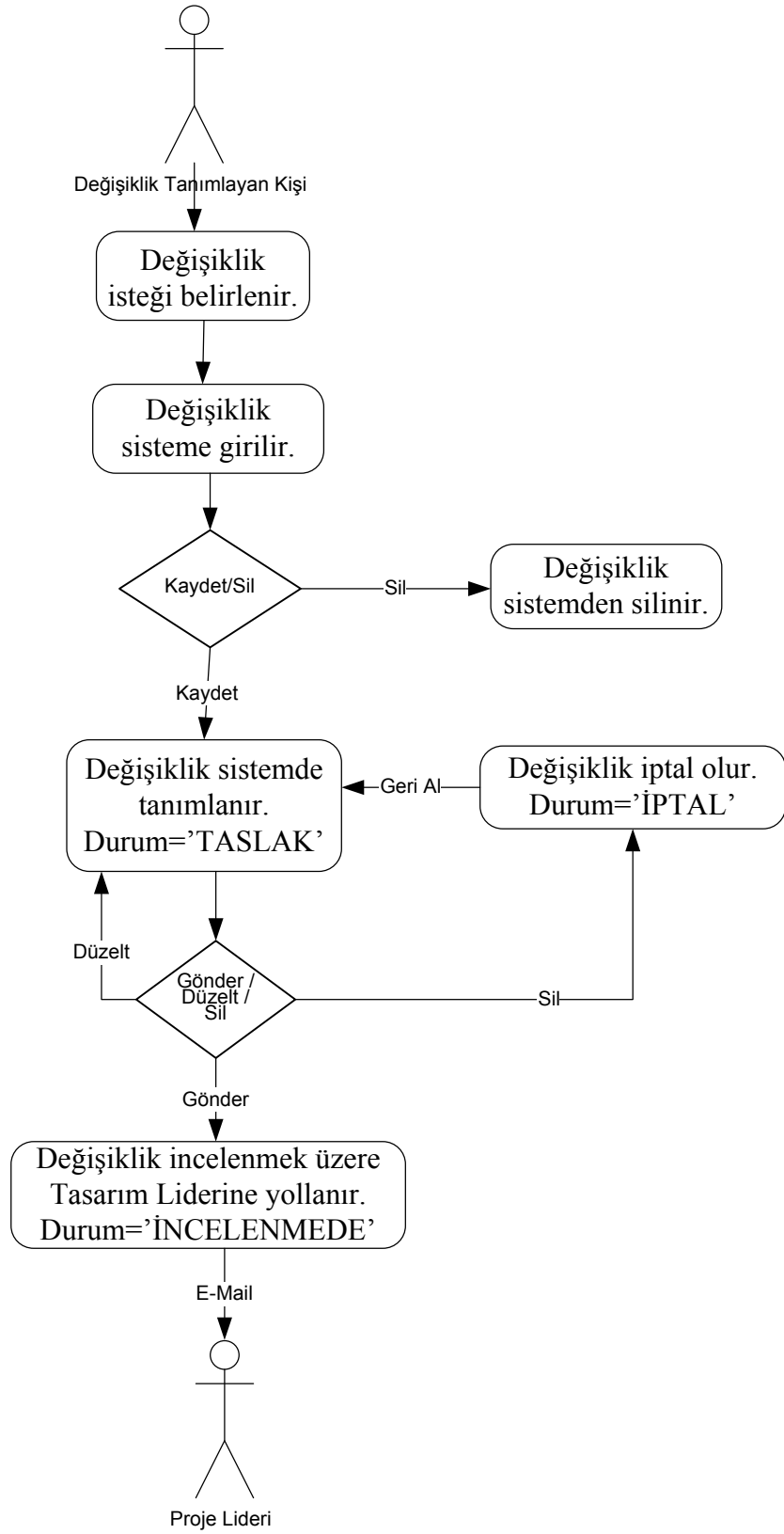
Bir değişiklik temel olarak dört fazdan oluşur. Bunlar sırasıyla başlatma fazı, değerlendirme fazı, inceleme fazı ve uygulama fazıdır.

3.2.1 BAŞLATMA FAZI

Bu faza projelerde görevli herkes dahil olabilir, tasarım ya da üretimin içinde herhangi bir noktada değişiklik olduğunu düşünen ya da gören kişiler sisteme yeni bir değişiklik isteği tanımlayabilirler. Bu istek tanımlanırken kullanıcı değişikliğin geçtiği projeyi, değişikliğin başlığını, tarihini, tanımını, çözüm önerisi varsa bu bilgiyi, değişikliğin etkilediği başka noktalar var ise bunları ve değişikliğin geçtiği proje liderini sistemden girerek değişikliği oluşturmuş olur. Tanımlanan değişiklik oluştuktan sonraki durum bilgisi “TASLAK” olarak ayarlanır.

Kullanıcı bu değişikliği değerlendirme fazına yollamadan önce her türlü düzeltmeyi sistem üzerinden yapabilir ve gerekirse değişikliği silebilir. Eğer değişiklik silinir ise değişikliğin durum bilgisi “İPTAL” olarak güncellenir. Silinen bir değişiklik tekrar “GERİAL” tuşu ile geri alınabilir ve durum bilgisi tekrar “TASLAK” konuma geçer.

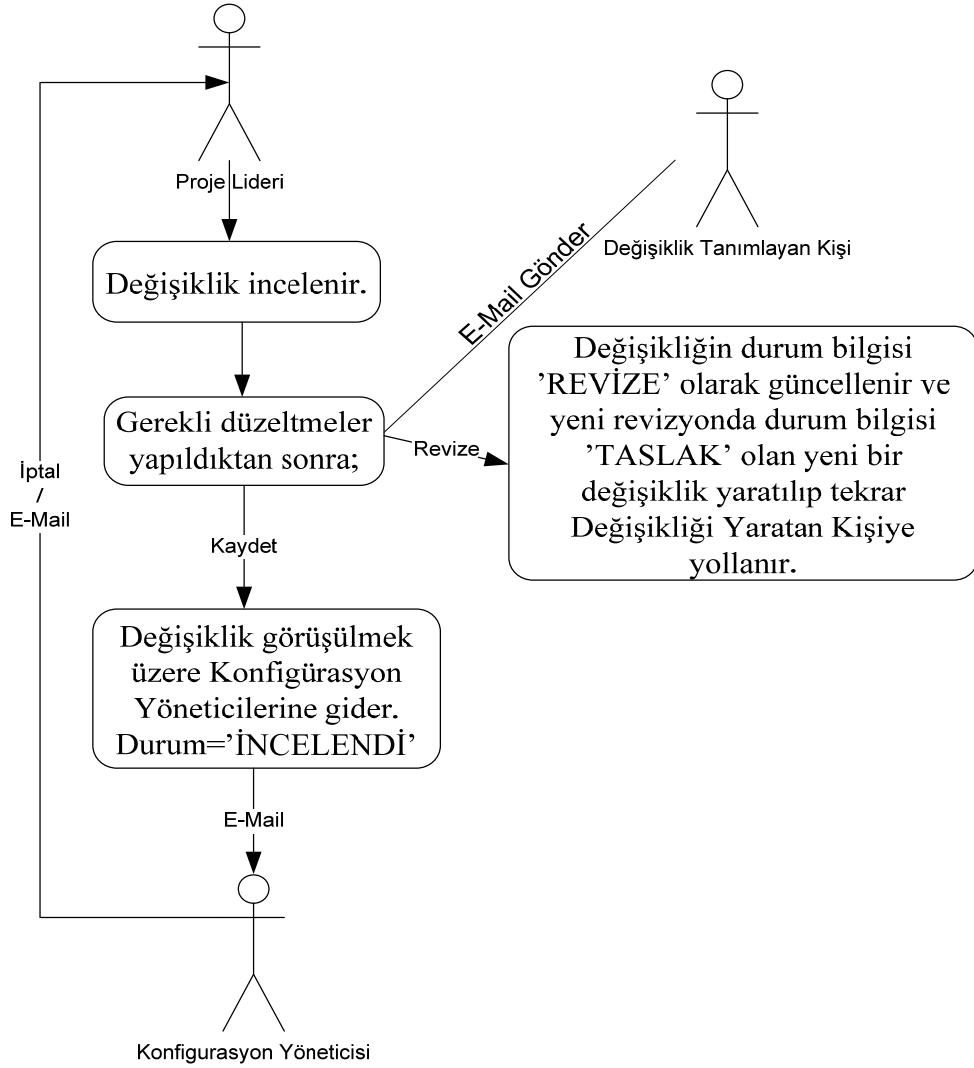
Sisteme girilen bu değişiklik bilgisi son halini aldı ise proje liderine e-posta yolu ile gönderilir. Bu işlem “İncelemeye Gönder” tuşuna basıldıktan sonra gerçekleşmiş olur ve değişikliğin durum bilgisi “İNCELENMEDE” olarak güncellenir. Bu aşamadan sonra artık bu değişiklik üzerinde değişikliği oluşturan kişi tarafından hiçbir işlem yapılamaz.



Şekil 5. Başlatma Fazı

3.2.2 DEĞERLENDİRME FAZI

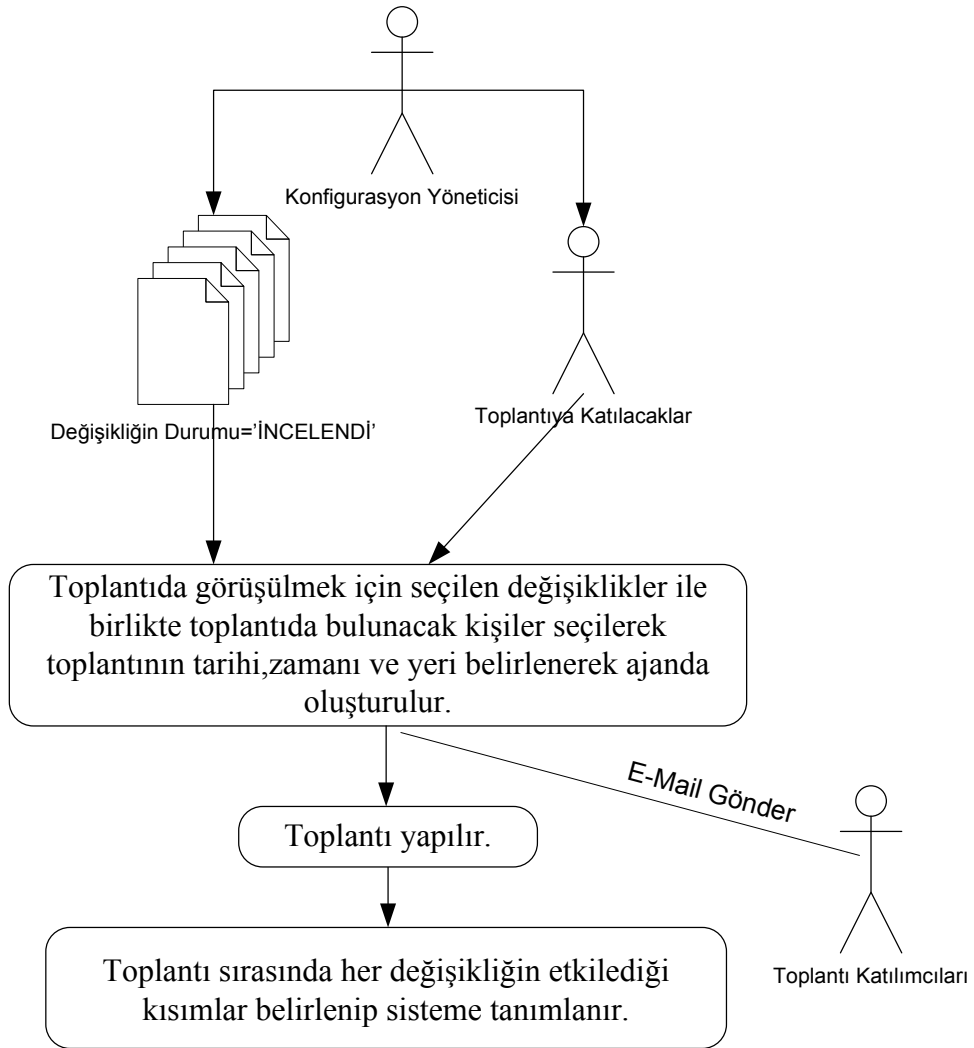
Proje liderine yollanan değişiklik bilgisi proje lideri tarafında incelenir, proje lideri bu değişiklik üzerinde güncellemeler yapıp revize edebilir, revize edilen değişiklik yeni revizyon numarası ile tekrar değişikliği tanımlayan kişiye gönderilir. Eğer proje lideri problemi onaylayıp kaydeder ise değişiklik konfigürasyon yöneticileri tarafından incelenmek üzere durum bilgisi="İNCELENDİ" olarak değişerek saklanmış olur. Durum bilgisi "İNCELENDİ" olan değişiklik üzerinde proje lideri bir daha değişiklik yapamaz. Eğer bu değişiklik proje lideri tarafından revize görür ya da iptal edilir ise bu değişikliği oluşturan kişiye sistem tarafından bilgilendirme amaçlı otomatik e-posta gönderilir ve yeni revizyon bilgisi ile durumu "TASLAK" olan yeni bir kayıt oluşturulup değişikliği tanımlayan kişiye yollanır.



Şekil 6. Değerlendirme Fazi

3.2.3 İNCELEME FAZI

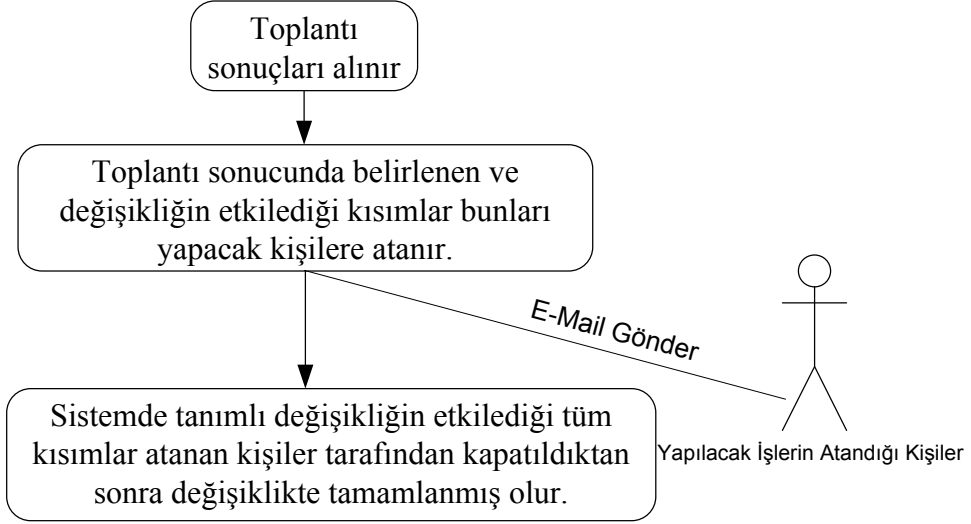
Durum bilgisi “İNCELENDİ” olan tüm değişiklikler konfigürasyon yöneticisinin önüne listelenir. Konfigürasyon yöneticisi listelenen bu değişiklikleri konuşmak üzere bir toplantı oluşturulur ve bu toplantıya katılacak kişileri belirler. Oluşturulan toplantının saat, tarih ve yer bilgileri katılımcılara otomatik mail atılarak bilgilendirilir. Toplantı sırasında her değişikliğin etkilediği kısımlar tartışılıp belirlendikten sonra sisteme tanımlanır.



Şekil 7. İnceleme Fazi

3.2.4 UYGULAMA FAZI

Toplantı sonucunda belirlenen ve değişikliğin etkilediği kısımlar yapılmak üzere ilgili kişilere atanır. İlgili kişilere atanan bu değişiklikler tamamlandıktan sonra sisteme kapandı olarak girilir. Değişiklik bilgisinin etkilediği tüm kısımlar, atanan kişiler tarafından kapatılır ise değişiklik yapılmış olur. Tüm bu süreçler sistem tarafından izlenir ve gerekli raporlar her aşamada alınarak kullanıcılara bilgi verilir.



Şekil 8. Uygulama Fazi

3.3 SİSTEM TARAFINDAN OLUŞTURULAN RAPORLAR

Sistem kullanıcılara süreçlerin her aşamasında değişiklikler ile ilgili detaylı raporlar verir. Bu raporlar;

- Değişiklik İsteği Raporu (*Change Request Report*) : Değişikliğin tüm detay bilgilerinin gösterildiği rapordur.(Ek 4)
- Değişiklik İsteği Tarihçe Raporu (*Change Request History Report*) : Bir değişiklik hangi aşamalardan, hangi zamanlarda geçmiş bilgisinin verildiği rapordur.(Ek 5)
- Yapılacak İşlerin Tarihçe Raporu (*Action Status History Report*) : Yapılacak bir işin (*action item*) kime atandığı, ne zaman açılıp ne zaman kapandığı bilgisinin verildiği rapordur.(Ek 6)
- CIN Dosya Raporu: Değişiklik bilgileri tarafından oluşturulan CIN numaralarının listelendiği rapordur.

BÖLÜM 4

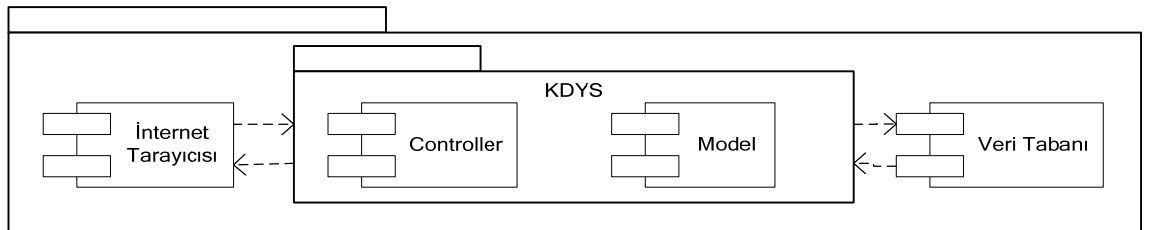
TASARLANAN SİSTEM HAKKINDA BİLGİ

Kurumsal Değişiklik Yönetim Sistemi en modern yazılım geliştirme araçları kullanılarak yazılmış ve alt yapısı bakımıyla ileriye yönelik adımlara ve teknolojilere açık bir şekilde geliştirilmiştir. Bu sistemde kullanılan donanım, yazılım ve haberleşme araçları aşağıda listelenmiştir. Kullanılan yazılım araçlarına ileriki kısımlarda daha detaylı bir şekilde anlatılacaktır.

- Veritabanı Yönetim Sistemi olarak IBM DB2.
- UML Diyagramlar için Microsoft Visio.
- Doküman versiyonlarını tutmak için CA Harvest.
- Yazılım Geliştirme Platformu olarak IBM Rational Application Development (*RAD*).
- Yazılım Geliştirme Teknolojisi olarak J2EE ve AJAX.
- Raporlar'ın hazırlanması için Crystal Reports.
- Haberleşme için kurumun kendi network altyapısı kullanılacaktır.

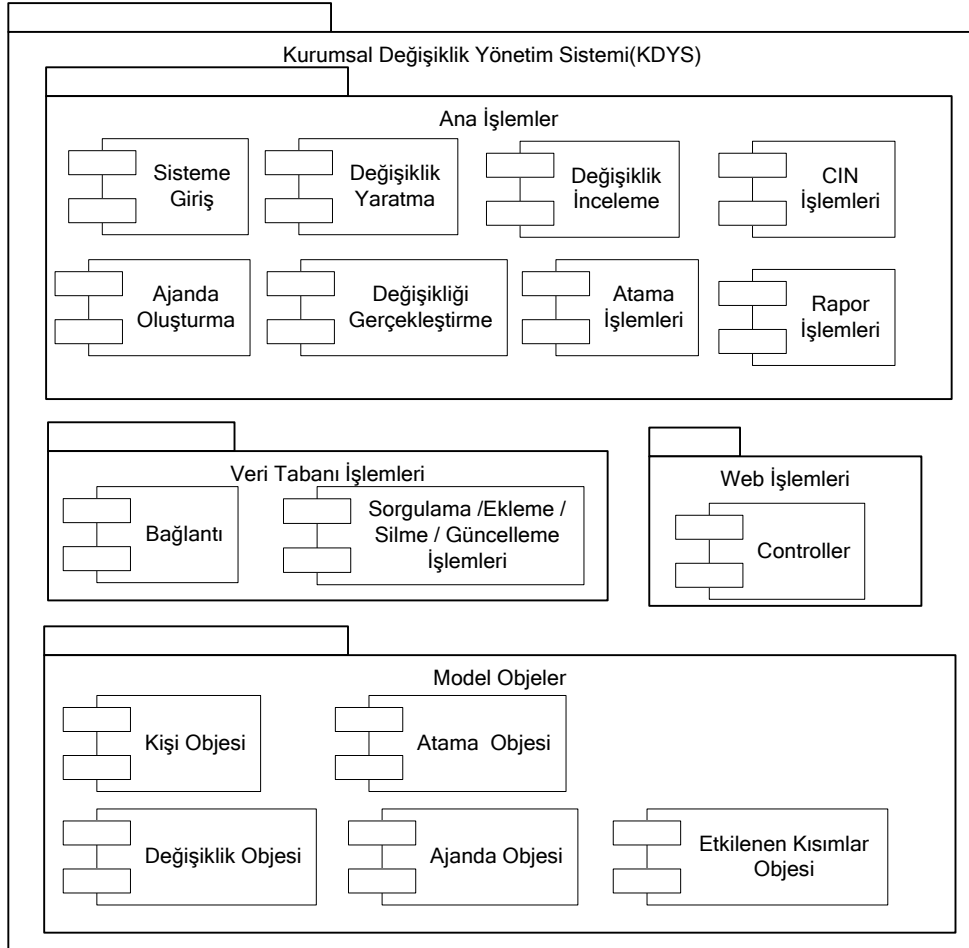
4.1 SİSTEMDE KULLANILAN BİLEŞENLER

Kurumsal Değişiklik Yönetim Sistemi üç ana bileşenden oluşmaktadır. İnternet tarayıcısı, sistemin kendisi ve veritabanı. İnternet tarayıcısı kullanıcıdan aldığı bilgileri ağ yardımıyla sisteme iletir. Sistem almış olduğu bu bilgileri veritabanı ile iletişime geçerek işlemeye başlar.



Şekil 9. Sistem Ana Bileşenleri

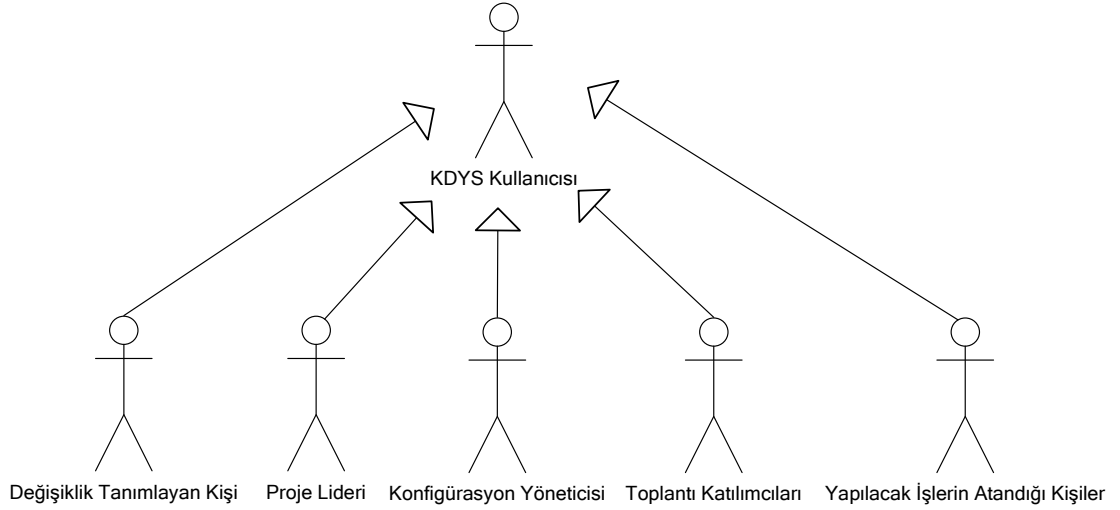
KDYS dört temel alt sistemden oluşmaktadır. Ana işlemler, veritabanı işlemleri, web işlemleri ve model nesnelere. Ana işlemler; KDYS'nin fonksiyonel yapısını sağlar. Veritabanı işlemleri; tüm bağlantı ve veritabanı ile olan iletişimi kurar. Web işlemleri; kullanıcının internet tarayıcısı tarafından girmiş olduğu tüm istekleri alır ve geri dönen cevapları da internet tarayıcısına iletir. Model nesnelere ise; verileri nesnelere atama ve atanmış verilerin geri alınma işini yapar.



Şekil 10. KDYS Alt Sistemleri

4.2 SİSTEMDE TANIMLI ROLLER

Sistemi kullanan toplam beş temel rol vardır. Değişiklik isteğini tanımlayan kişi, değişikliğin geçeceği projenin lideri, ajandayı oluşturan kişi ve etki analizi yapacak kişiler ile değişiklik işlerinin atandığı kişiler.

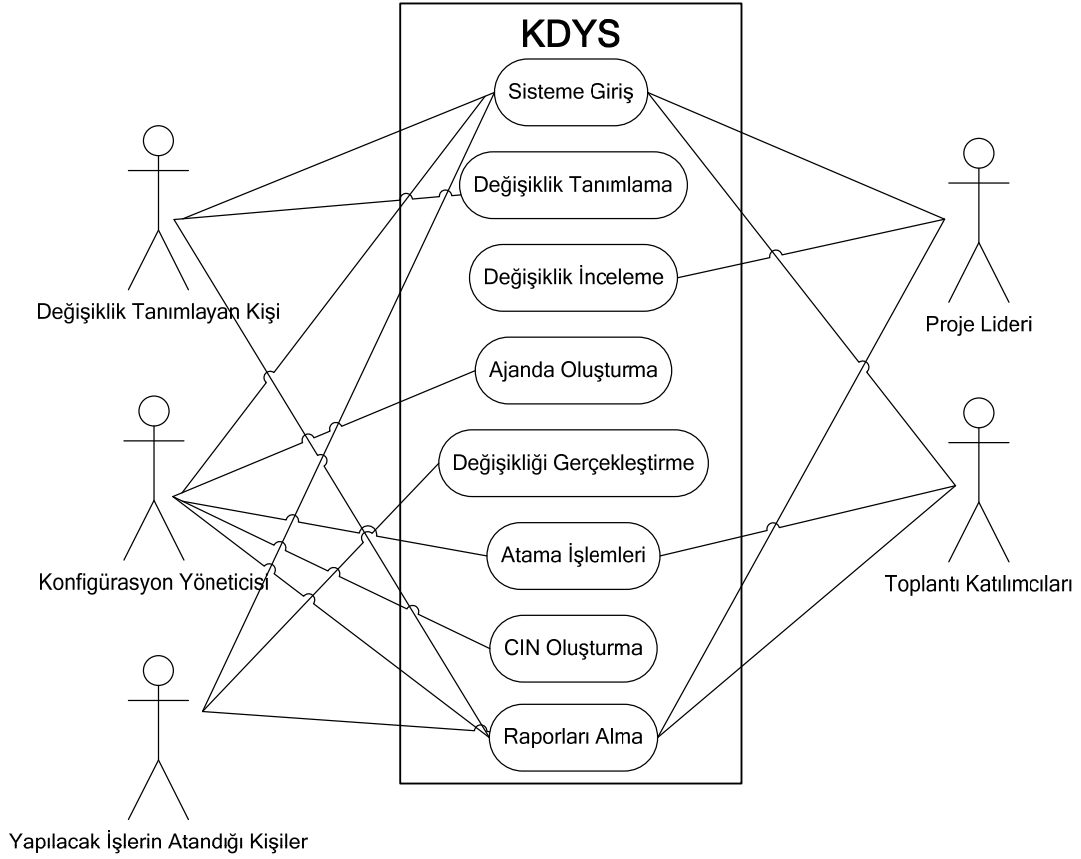


Şekil 11. Sistemde Tanımlı Roller

4.2.1 SİSTEMDE TANIMLI OLAN ROLLERİN GÖREVLERİ

- İlk olarak sisteme giriş yapmak isteyen kullanıcının hangi rol'e sahip olduğu bilgisine Personel Bilgileri Veritabanı yardımıyla bakılır.
- Değişikliği tanımlayan kişi, tasarım veya üretim süreçlerinin herhangi birinde görmüş olduğu değişikliği sisteme girer.
- Değişikliğin tanımlandığı proje lideri bu değişikliği gözden geçirmekle sorumludur.
- Konfigürasyon Yöneticisi toplantının yerini, tarihini ve zamanını belirleyerek toplantıda hangi değişikliklerin konuşulacağına ve toplantıya kimlerin katılacağına karar verir.
- Toplantı katılımcıları değişikliğin etkilediği başka kısımların olup olmadığına ve oluşan değişiklikleri kimin çözeceğine karar verir.
- Yapılacak işlerin atandığı kişiler bu değişikliğin hangi kısımlara ne gibi etkiler yapacağına karar verir ve değişikliği gerçekleştirirler.

Şekil 12 de rollerin hangi işlemleri yapmaya yetkisi olduğu gösterilmiştir. Kullanıcılar sadece yetkili olduğu işlemleri yapmakla sorumludurlar. Kullanıcılara farklı roller tanımlayıp, farklı yetkilendirmeler yapılmasındaki amaç oluşabilecek karışıklık ve yanlışlıkları önlemektir.



Şekil 12. Rollerin Sistemdeki Görevleri

4.3 KULLANILAN YAZILIM GELİŞTİRME TEKNOLOJİLERİ

Yazılım geliştirme ortamı olarak Yahoo'nun kullanmış olduğu YUI mimarisini ile beraber AJAX teknolojisi kullanılmıştır.

4.3.1 Yahoo User Interface

Yahoo User Interface (*YUI*), web geliştiricileri ve ara yüz tasarımcılarının işlerini kolaylaştırmak amacıyla Yahoo'nun uzman geliştiricileri tarafından esasen JavaScript ile yazılmış yardımcı yazılımlar ve kontrollerden oluşan bir kütüphanedir. Bu kütüphane Berkeley Software Distribution (*BSD*) lisansı ile lisanslanmış ve tamamen ücretsiz olarak her türlü kullanıma açılmıştır.

YUI'nin kullanıcılarına AJAX desteği, Javascript ve CSS bileşenleri ile kendi servisleri için Yahoo API'leri sağlamasının ötesinde en büyük getirisi ara yüz tasarımcıları ve web geliştiricileri için farklı tarayıcılar üzerinde çalışan adım izlenceler (*script*) üretebilme konusunda getirdiği kolaylıklardır [9].

YUI 'nin Temel Özellikleri;

- Kolay ve esnektir, kütüphanesi az yer kaplar.
- Basit konular için zengin bir temel sunar.
- Modüler bir yaklaşım sunar; sayfalarda sadece gereken yerde yükleme yaparak gereksiz yük ve performans kullanımından kaçınır.
- YUI dosyaları Yahoo tarafından desteklenir.
- Ücretsizdir ve tamamen açık bir kütüphaneden oluşur.
- YUI kullanıcı ile ilgili olarak yardımcı dokümanlara, örnek çalışmalara rahatlıkla erişilebilir. Forumları kullanarak rahatlıkla destek alınabilir.
- Bütün web tarayıcılarında rahatlıkla çalışır.

YUI kütüphanenin çekirdeği aşağıdaki üç dosyayı içerir [10].

Yahoo Global Nesnesi

Yahoo Global Nesnesi tüm Yahoo kütüphanesinin kodlarının bulunduğu tek bir global ad sağlar ve kütüphane boyunca kullanılan birçok metodu içermektedir. YUI kütüphanesini kullanan her sayfa Yahoo Global Nesnesini içermelidir ve tüm diğer YUI öğelerinden önce görünmelidir, yani diğer tüm YUI öğeleri için temel gereksinimdir.

Yahoo Global Nesnesi tüm YUI öğeleri için ad ağacının en üstünde yer alır ve tiplene kontrolü, kullanıcı birimi algılama ve modül uzatma (*module extension*) için bazı kullanışlı metotlar içerir.

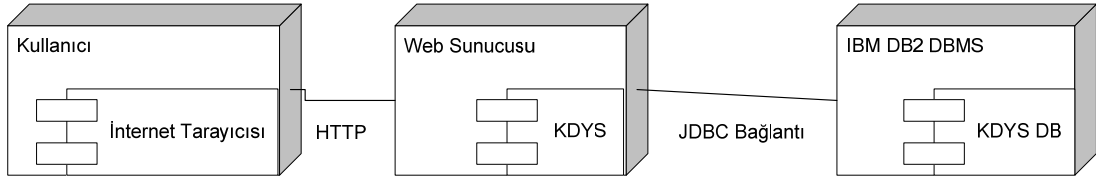
DOM Koleksiyonu

Dom, Dom-Scripting görevleri basitleştiren bir dizi için uygun metotlar (öğe konumlandırılması ve CSS stil yönetimi gibi) içerir. Aynı zamanda bu sırada çapraz tarayıcı (*cross-browser*) tutarsızlıklarını da normale çevirir.

Olay Yardımcı Programları

Olay Yardımcı Programları tarayıcının Olay nesnesinin özelliklerini incelemek ve DOM olaylarına kaydolmak için basitleştirilmiş bir ara yüz vererek, tarayıcıda olaya dayalı (*event-driven*) uygulamalarını kolaylaştırır. Aynı zamanda özel etkinlik (*Custom-Event*) nesnesi de içerir. Özel Etkinlik kodundaki ilginç anları ya da olayları

yayımlamayı sağlar ve sayfadaki diğer öğeler, bu olaylara kaydolur ve karşılık verirler.



Şekil 13. YUI Geliştirme Diyagramı

Kullanıcı: Kullanıcı internet tarayıcısı yardımı ile HTTP bağlantı sayesinde sistem ile iletişime geçer.

Web Server: KDYS uygulamasının kodlarını ve denetçilerin içinde bulunduğu bir Web Sunucusu tanımlıdır ve bu direk olarak kullanıcı isteklerinin metotlarla ilişkilendirir.

IBM DB2 DBMS: KDYS DB veritabanını kullanır. Web Sunucusu bağlantı yardımı ile veritabanı ile iletişime geçer. Bu iletişimde JDBC veritabanı bağlantısı kullanılır.

4.3.2 AJAX TEKNOLOJİSİ

Asynchronous JavaScript and XML sözcüklerinin kısaltması olan Ajax, etkileşimli web uygulamaları yaratmak için kullanılan bir web programlama tekniğidir. Temel amacı arka planda sunucuyla ufak miktarda veri değişimi sayesinde sayfayı daha hızlı güncellenebilen web sayfaları yapmak, dolayısıyla kullanıcının istediği her anda bütün web sayfasını güncellemek derdinden kurtulmaktır. Bu da web sayfasının etkileşimini, hızını ve kullanılabilirliğini artırmaktır [11].

Ajax Avantajları

Bant genişliğini ayarlamak; HTML'yi ağ tarayıcısı (*browser*) programında oluştururken, JavaScript işlemlerini ve işlenecek verileri kullanırken, sunucudan gelen sayfa bilgisi gerektiğinden daha az yer tuttuğundan dolayı Ajax web sayfalarının görece olarak daha hızlı yüklendiği gözlenebilir. İçeriğin “isteğe bağlı olarak yüklenmesine” ek olarak, bazı web uygulamaları ilk önce Olay Yöneticilerini (*event handler*), ardından ilgili fonksiyonları indirir. Bu teknik, karmaşık mekanizması ve fonksiyonu olan web uygulamalarının gereksinim duyduğu yüksek miktarda bant genişliği tüketimini önemli ölçüde hafifletir.

AJAX'ın kullanmış olduđu Teknolojiler;

- Bilgiyi biçimlendirmek ve görüntülerini deđiřtirmek için XHTML (ya da HTML) ve CSS.
- Görüntülenecek bilgiyi dinamik olarak göstermek ve onunla etkileřimli çalıřmak için JavaScript, JScript gibi ECMAScript olan client side (kullanıcı tarafı) tarafındaki bir adım izlenç dili yardımıyla erişilebilen DOM.
- Web sunucu ile bilgi deđiř tokuđu için kullanılan XMLHttpRequest nesnesi. Bazı Ajax frameworklerde ve bazı durumlarda web sunucu ile bilgi alışveriři yapmak için XMLHttpRequest yerine IFrame de kullanılabilir; Diđer uygulamalarda ise dinamik olarak eklenen JavaScript TAG'leri de kullanılabilir.
- Kullanıcıyla sunucu arasındaki bilgi alışverişlerinde çođunlukla XML formatı kullanılmaktadır. Önceden hazırlanan HTML, düz metin, JSON ve EBML dahi olmak üzere herhangi bir format da kullanılabilir. Bu dosyalar sunucu tarafında herhangi bir adım izlençesi ile bile üretilmiş olabilirler.

DHTML, SPA ve LAMP gibilerin aksine Ajax kendi kendine bir teknoloji deđildir, ancak bu terim birden çok teknolojinin bir arada kullanılmasını ifade etmektedir [11].

En başta Google pek çok servisinde AJAX tekniđini kullanmaktadır. Microsoft da bu tekniđi kullanan pek çok servis hazırlamıştır. Yahoo 'da diđerlerinden geri kalmamış, bu etkileyici tekniđi kendi servislerine entegre etmiştir. Ayrıca kendilerinden oldukça bahsettiren YouTube, Flickr ve Del.icio.us siteleri de AJAX tekniđini kullanan önemli web uygulamalarıdır [12].

4.3.3 E-POSTA SUNUCU

Tezde tasarlanan uygulamadaki tüm e-mail işlemleri için JavaMail API kullanılmıştır. JavaMail API platformdan ve protokolden tamamen bađımsız olarak çalıřarak uygulamalarda mesajlaşmayı sağlar. JavaMail Java EE üzerine inşa edilmiştir.

4.3.4 UML MODELLEME DİLİ

Birleştirilmiş Modelleme Dili (*UML*) bir programlama dili değildir. Bir diyagram çizme ve ilişkisel modelleme dilidir. İş sistemlerinin nasıl modellenebileceğini belirleyen ve açıklayan yöntemlerin bir araya toplanmış halidir.

UML daha çok şekiller ve onların oluşturduğu diyagramlardan oluşur. Böyle bir diyagram oluşturmak hem programı yazanlar için hem de programı kullanacaklar için büyük kolaylık ve bir birliktelik sağlamaktadır. Bazen öyle bir program olur ki bir çok parçası farklı insanlar tarafından yapılır. Onlar arasındaki uyum da UML diyagramları sayesinde sağlanır. Belli standart bir dili olduğu için UML diyagramı dünyanın neresinde olursa olsun herkes tarafından anlaşılabilir.

UML ile hazırlanmış bir yazılım hem daha az maliyetli hem daha etkili ve daha uzun ömürlü olur. UML ile dokümantasyonu yapılmış bir programın sonradan düzenlenmesi daha kolay olur.

UML 'nin faydaları maddeler halinde sıralanırsa;

- Kodlama işlemi daha kolaydır, çünkü UML ile programdan ne beklenildiği ve programlama ile neler yapılabilineceği profesyonel bir şekilde belirlenir.
- Programda gerçekleşen beklenmedik bir takım mantıksal hataları minimuma indirmiş olur.
- Tasarım aşaması düzgün yapıldıysa tekrar kullanılabilen kodların sayısı artacaktır. Bu da program geliştirme maliyetini büyük ölçüde düşürür.
- UML diyagramları programın tamamını kapsayacağı için bellek kullanımı daha etkili hale gelir.
- Programın kararlılığı artacaktır. UML ile dokümanlandırılmış kodları düzenlemek daha az zaman alacaktır.
- Ortak çalışılan projelerde programcıların iletişimi daha kolay hale gelir. Çünkü UML ile programı parçalara ayırıp bu parçalar arasında bir ilişki kurulmuştur.

UML temel olarak 9 diyagram türünden oluşur [22];

Sınıf Diyagramı (*Class Diagram*)

Sınıf Diyagramları UML 'in en sık kullanılan diyagram türü olup, nesne yönelimli analiz, tasarım ve programlama 'da esas teşkil eden sınıfları en iyi şekilde temsil etmeyi amaçlar. Gerçek hayattan alınan örneklere göre sınıflar, ona ait özellikler ve ifade edebileceği davranışlar belirlenir ve bunlar sınıf diyagramı olarak çizilir.

Nesne Diyagramı (*Object Diagram*)

Bir nesne sınıfın bir örneğidir. Bu tür diyagramlarda sınıfın yerine gerçek nesnelere kullanılır. Nesne diyagramında her bir sınıftan oluşturulmuş nesnelere yer alır.

Durum Diyagramı (*State Diagram*)

Gerçek nesnelere herhangi bir zaman içindeki durumunu gösteren diyagramlardır.

Sıralama Diyagramı (*Sequence Diagram*)

Sınıf ve Nesne diyagramları statik bilgiyi modeller. Halbuki gerçek zamanlı sistemlerde zaman içinde değişen interaktiviteler bu diyagramlarla gösterilemez. Bu tür zamanla değişen durumları belirtmek için sıralama diyagramları kullanılır.

Faaliyet Diyagramı (*Activity Diagram*)

Bir nesnenin durumu zamanla kullanıcı tarafından ya da nesnenin kendi içsel işlevleri tarafından değişebilir. Bu değişim sırasını faaliyet diyagramlarıyla gösterilir.

Kullanım Senaryosu Diyagramı (*Use-Case Diagram*)

Programın davranışının bir kullanıcı gözüyle incelenmesi Use Case diyagramlarıyla yapılır. Kullanım senaryosu diyagramı bir sistemden beklenen belirli bir davranışını gösterir ve sistemden beklenen gereksinimlerin tespit edilmesi amacıyla kullanılır. Tipik olarak kullanım senaryoları, bağımlılıkları, ilişkileri ile aktörler gösterilir.

İş Birliği Diyagramı (*Collaboration Diagram*)

Bir sistemin amacının yerine gelmesi için sistemin bütün parçalarının işlerinin yerine getirmesi gerekir. Bu işler genellikle birkaç parçanın beraber çalışmasıyla mümkün olabilir. Bu tür ilişkileri göstermek için İş Birliği Diyagramları kullanılır.

Bileşen Diyagramı (*Component Diagram*)

Özellikle birden çok geliştiricinin yürüttüğü projelerde sistemi bileşen denilen parçalara ayırmak, geliştirmeyi kolaylaştırır. Sistemin öyle modellenmesi gerekir ki her geliştirici ötekenden bağımsız olarak çalışabilsin. Bu tür modellemeler Bileşen Diyagramlarıyla yapılır.

Dağılım Diyagramı (*Deployment Diagram*)

Dağılım diyagramı bir yapısal diyagram 'dır ve modellenen sisteme belli bir açıdan bakışı gösterir. Bilgisayarlar arasındaki bağlantılar, programın kurulacağı makineler ve sistemlerdeki bütün aletler dağılım diyagramında gösterilir.

4.3.5 DB2 VERİTABANI YÖNETİM SİSTEMİ

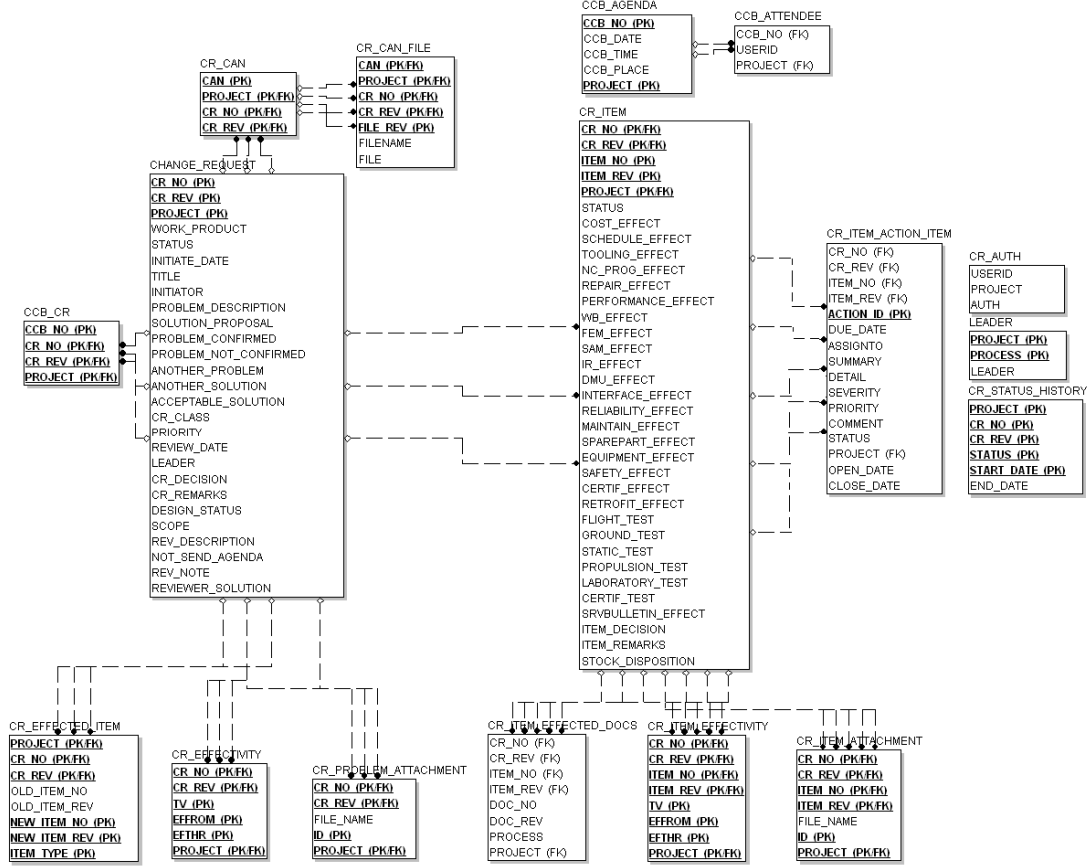
DB2 orta ölçekli ve büyük işletmelerin veri sunucusu gereksinimlerini karşılamak için tasarlanmıştır. Tek işlemciden yüzlerce işlemciye sahip olanlara kadar her boyuttan Linux, UNIX ya da Windows sunucusuna dağıtılabilir. DB2, boyutu birden çok terabayt olan büyük veri ambarları, 7 gün 24 saat yüksek performansla hizmet veren büyük hacimli işlem iş çözümleri ya da Web tabanlı çözümler gibi isteğe bağlı kurumsal ölçekli çözümlerin oluşturulması için ideal bir temel sağlar. DB2 diğer kurumsal veri kaynaklarıyla bağlantı, uyumluluk ve bütünleştirme olanakları sunar.

Temel özellikleri [23];

- Yenilikçi otomatik yönetim ve otomatik ayarlama yetenekleri sağlayarak veritabanı sunucularını yönetmeyle ilgili harcanan zamanı ve maliyetleri önemli ölçüde azaltır.
- Birleşik Web Hizmetleri ve XML için sektörün önde gelen desteğini sunar.
- Açık sektör standartlarına dayanır ve sektörün yaygın platformlarına taşınabilir.
- MDC (*Multi-Dimensional Clustering*) yeteneği aracılığıyla gelişmiş iş zekası desteği sunar.
- Java ve Microsoft ortamlarındaki geliştirici verimliliğini artıracak araçların yer aldığı Development Center olanağını içerir.
- İşbirliği için Lotus, yönetim için Tivoli ve dinamik e-iş için WebSphere gibi diğer IBM yazılımlarıyla bütünleştirmeyi destekler.

4.4 YAZILIM VERİTABANI ŞEMASI

Tez’de hazırlanmış olduğumuz KDYS’nin veritabanı şeması Şekil 14 deki veritabanı diyagramında detaylı olarak verilmiştir;



Şekil 14. Kurumsal Değişiklik Yönetimi Veritabanı Diyagramı

Şekil 14 deki veritabanı diyagramında sisteme girilen bir değişiklik isteğinin hangi tablolarda, hangi alanlarda tutulduğunu ve bu bilgilerin birbirleri ile olan ilişkileri gösterilmiştir. EK 7’de bu tabloların SQL dilindeki kodları verilmiştir.

CHANGE_REQUEST: Sisteme girilen değişikliğin temel bilgileri bu tabloda saklanmaktadır. Bu tabloda bulunan değişiklik numarası(CR_NO), değişiklik revizyonu(CR_REV) ve değişikliğin geçtiği proje(PROJECT) bilgileri birlikte bir birinci anahtar (*Primary Key*) oluşturmaktadır. Bu birincil anahtar sayesinde tablodaki değişiklik bilgisi sistemdeki diğer tablolar ile ilişkilendirilmiştir.

CR_EFFECTED_ITEM: Değişiklik sisteme tanımlanırken etkilediği parçalar bu tabloda saklanır. Eğer değişiklik isteği proje lideri tarafından onay alırsa bu tablodaki

bilgiler EK 8 deki tetik (*trigger*) ile otomatik olarak CR_ITEM tablosuna aktarılır ve daha sonra deęişiklikten etkilenen parça bilgilerine bu tablodan bakılır.

CR_EFFECTIVITY: Bu tabloda deęişiklikten etkilenen kısımlar saklanır. Daha sonra bu tablodaki bilgiler EK 8 deki bir tetik sayesinde otomatik olarak CR_ITEM_EFFECTIVITY tablosuna aktarılır.

CR_PROBLEM_ATTACHMENT: Deęişiklikle ilgili her türlü eklenti bu tablodaki blob bir alanda saklanır. Bu eklentiler çizim veya dokümanlar olabilir.

CCB_CR: Deęişiklięin yapılacağı toplantının numarası bu tabloda tutulur

CR_CAN: Deęişiklik tamamlandıktan sonraki CIN numarası bu tabloda tutulur

CR_CAN_FILE: Her CIN numarasına ait bir veya birden fazla ek dosyası bu tabloda saklanabilir.

CR_ITEM: Bir deęişiklięin etkiledięi birden fazla parça olabileceęi için bu parça bilgileri ayrı bir tablo olan CR_ITEM tablosunda tutulmuştur.

CR_ITEM_EFFECTED_DOCS: Deęişiklięin geçtięi parçaların etkilemiş olduęu dokümanlar bu tabloda saklanır.

CR_ITEM_EFFECTIVITY: Deęişiklikten etkilenen kısımlar burada saklanır.

CR_ITEM_ATTACHMENT: Deęişiklikle ilgili her türlü eklenti bu tablodaki blob bir alanda saklanır. Bu eklentiler çizim veya dokümanlar olabilir.

CR_ITEM_ACTION_ITEM: Deęişiklięin geçtięi parçadaki problemi çözmek için atanan kişiler ve detay bilgileri burada saklanır.

CCB_AGENDA: Deęişikliklerin tartışılacağı toplantı bilgileri burada saklanır.

CCB_ATTENDEE: Deęişikliklerin tartışılacağı toplantının katılımcı bilgileri burada saklanır.

CR_AUTH: Sistem kullanıcılarının hangi projeleri görme ve işlem yapma yetkisi olduęu bu tabloda tutulur.

LEADER: Sistemde tanımlı bir projenin lider bilgileri bu tabloda tutulur.

CR_STATUS_HISTORY: CR üzerinde yapılan tüm işlemler bu tabloya EK 8 de bulunan bir tetik yardımı ile otomatik olarak atanır.

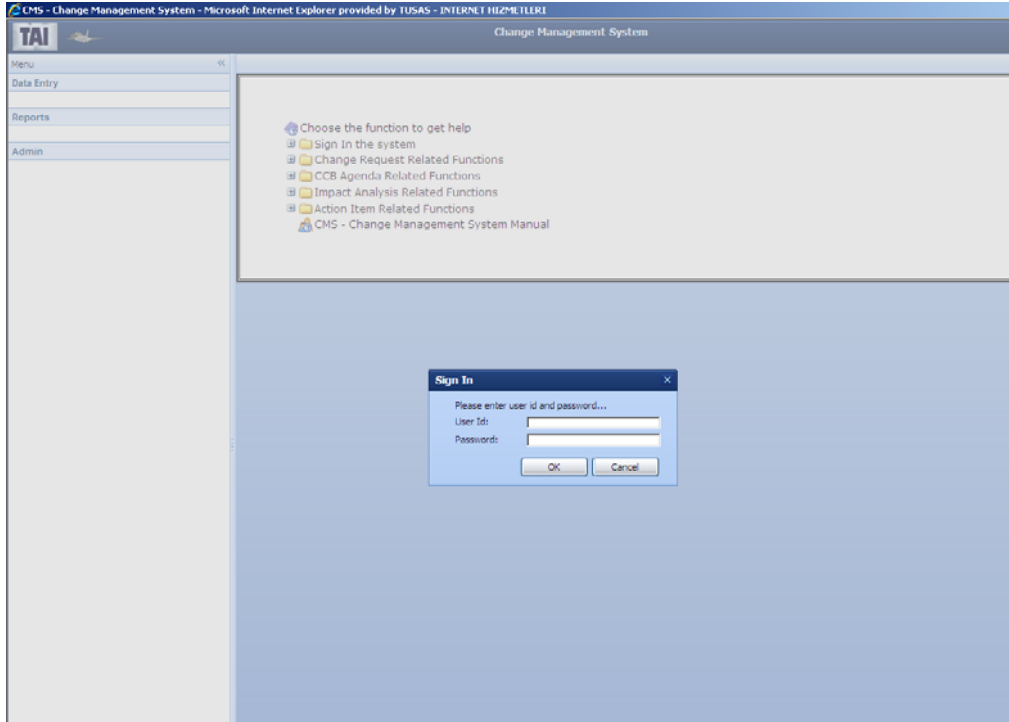
BÖLÜM 5

KURUMSAL DEĞİŞİKLİK YÖNETİM SİSTEMİNDE ÖRNEK BİR OLAYIN İNCELENMESİ

Bu kısımda, konunun daha iyi anlaşılabilmesi için örnek bir uygulama alanı üzerinde, ekran görüntüleri ile birlikte önemli bazı senaryolar detaylı bir biçimde anlatılacaktır. Kullanıcının sisteme giriş işlemlerinden, bir değişikliği nasıl tanımladığı ve bu değişikliğin kapanana kadarki geçirmiş olduğu tüm adımlar gösterilecektir.

5.1 SİSTEME GİRİŞ

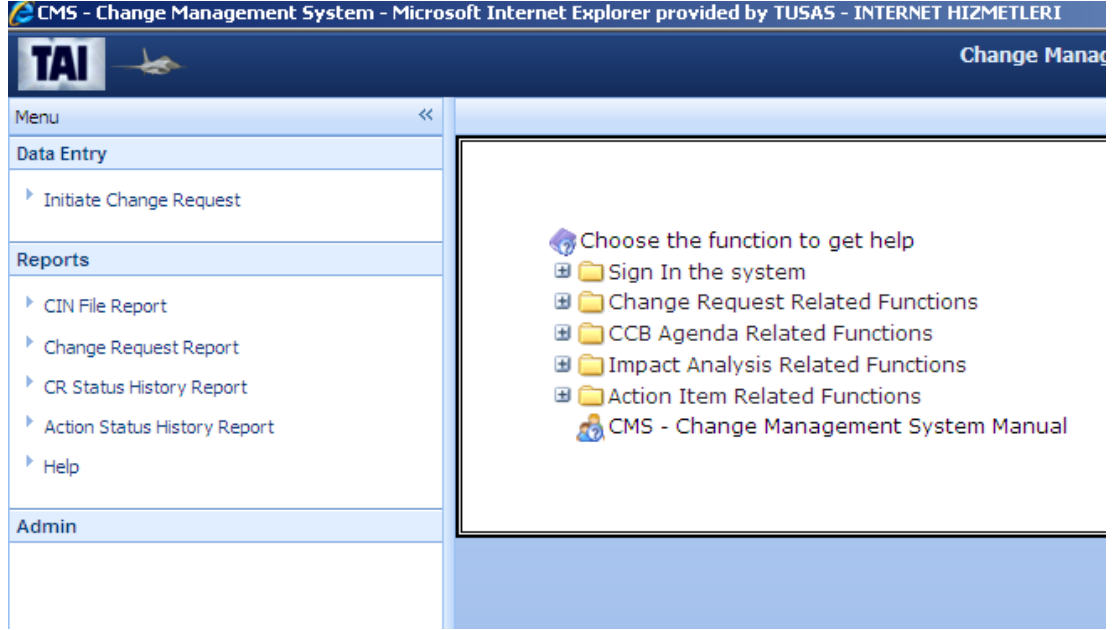
Sisteme yeni bir değişiklik tanımlamak isteyen kullanıcı ilk olarak kurumun ağına bağlı herhangi bir bilgisayarın internet tarayıcısına sistemin adresini yazarak sistemi çalıştırır. Program web tabanlı olduğundan son kullanıcının bilgisayarına kurulması gerekli bir program yoktur. İnternet tarayıcısı yeterlidir. Sisteme giriş ekranı kullanıcının karşısına Şekil 15 görüldüğü gibi gelir. Bu ekranda kullanıcıya kullanıcı adı ve şifre bilgileri sorulmaktadır.



Şekil 15. Sisteme Giriş Ekranı

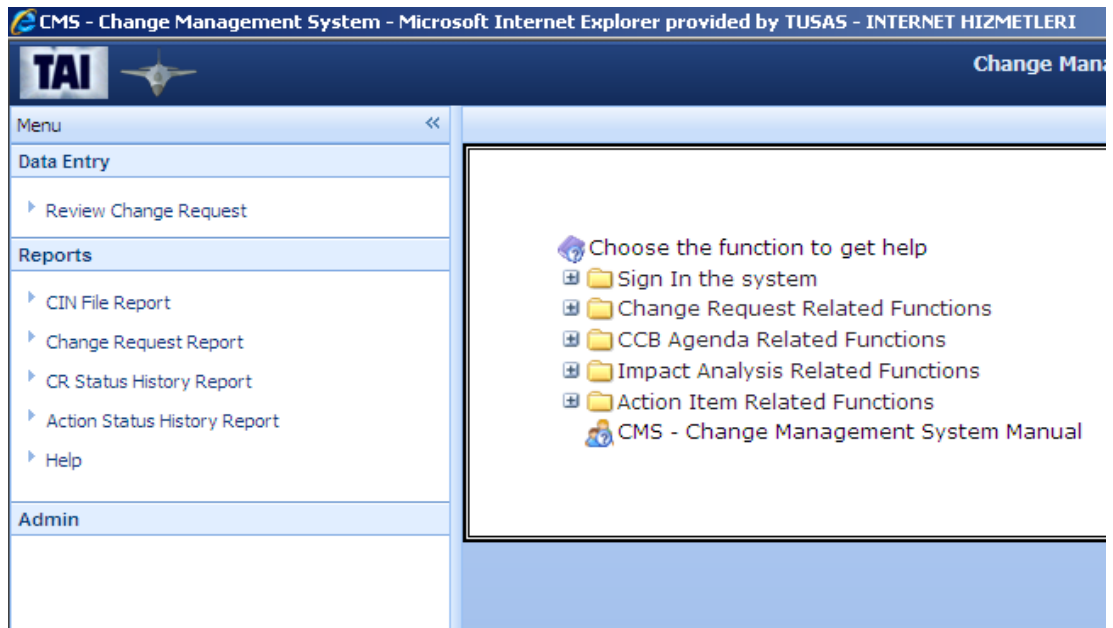
Kullanıcı bilgilerini ekrana girdikten sonra sistem tarafından personel bilgileri veritabanına bakılır ve kullanıcının sisteme yetkisinin olup olmadığı, yetkisi var ise sistemde tanımlı hangi ekranları kullanmaya yetkili olduğu bilgilerine bakılır.

Eğer kullanıcı sisteme sadece yeni bir değişiklik tanımlamaya yetkili ise bu kullanıcının karşısına Şekil 16'daki veri giriş menüsü çıkar;



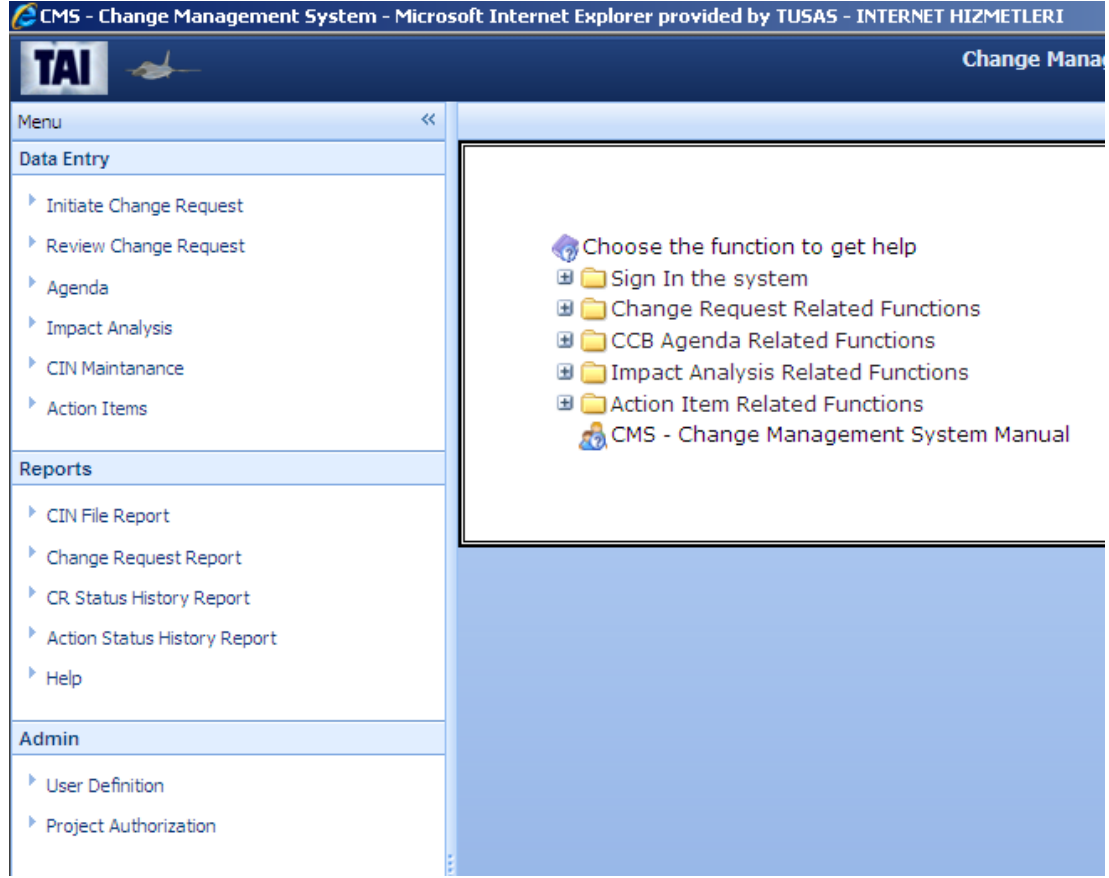
Şekil 16. Değişiklik Tanımlamaya Yetkili Kullanıcının Menüsü

Sisteme giren kullanıcı proje lideri ise bu kullanıcının sadece değişikliği inceleme yetkisi olduğundan ona Şekil 17'deki veri giriş menüsü açılır;



Şekil 17. Değişikliği İncelemeye Yetkili Kullanıcının Menüsü

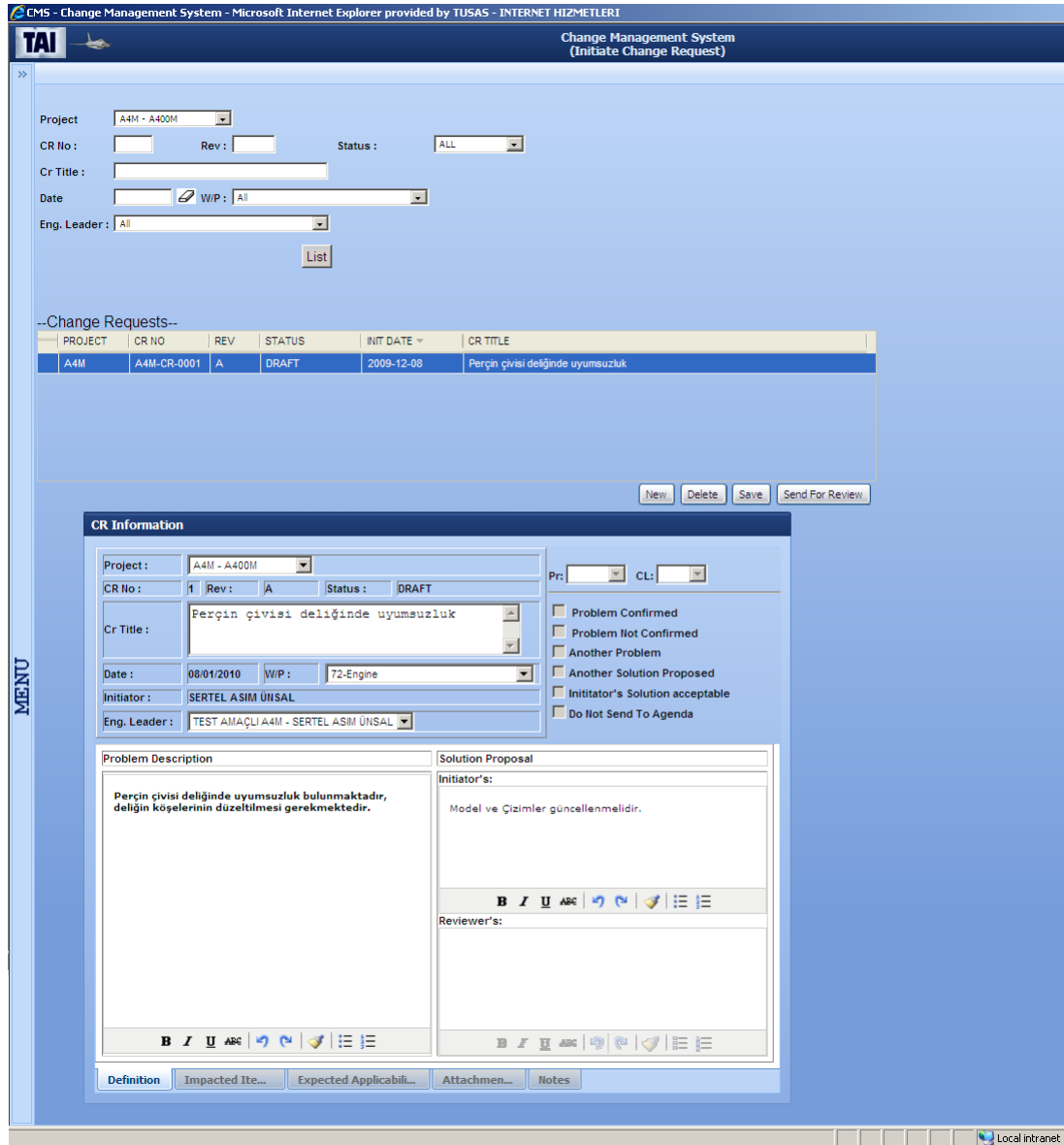
Kullanıcı yönetici (*administrator*) yetkisine sahip ise Şekil 18'deki ekran görüntüsünden de görüleceği gibi tüm işlemleri görme ve kullanma yetkisi bulunmaktadır.



Şekil 18. Sistemde Yönetici Olan Kullanıcının Menüsü

5.2 DEĞİŞİKLİK TANIMLAMA

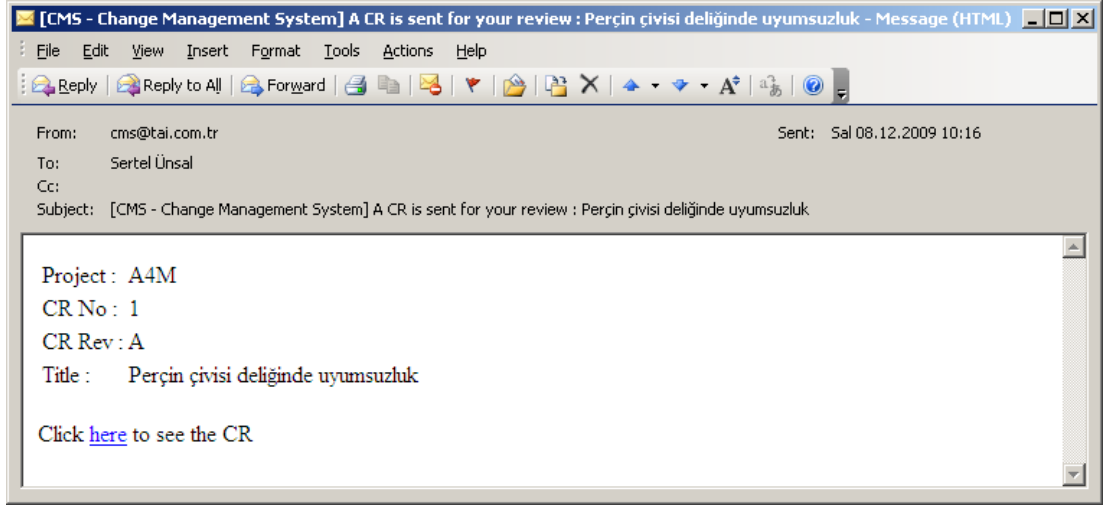
Bu ekranı sisteme yeni bir değişiklik tanımlamak isteyen kullanıcılar kullanır. Şekil 19'de de görülen ekranın üst kısmında daha önceden sisteme girilmiş olan değişiklikleri görüntülemek için arama kriterleri bulunur. Arama kriterlerine göre sistemde bulunan değişiklik bilgileri tabloda listelenir. Bu tablodan detaylarını görmek istediğimiz değişiklik seçilirse, bu değişiklik ile ilgili tüm detay bilgileri en alttaki "CR Information" panel'de gösterilir.



Şekil 19. Değişiklik Tanımlama Ekranı

Sisteme yeni bir değişiklik tanımlamak isteyen kullanıcı ilk olarak ekranda bulunan “New” tuşuna basar ve sistem tarafından otomatik olarak üretilen yeni bir değişiklik numarası ile tabloya kayıt eklenir. Tabloya eklenen bu kaydın detay bilgileri, altta bulunan panel vasıtasıyla sisteme girilir. Tüm veri giriş işlemleri tamamlandıktan sonra ekranda bulunan “Save” tuşuna basılarak değişiklik sisteme tanımlanmış olur. Sisteme tanımlanan değişiklik “Delete” tuşuna basılarak durum bilgisi “İPTAL” olarak güncellenir ve üzerinde başka bir işlem yapılmaz. Sisteme kayıt edilen değişiklik incelenmek üzere proje liderine “Send For Review” tuşuna basılarak gönderilir. Bu tuşa basılınca değişikliğin durum bilgisi “İNCELEMEDE” olarak

güncellenir ve proje liderini bilgilendirmek amaçlı otomatik bir e-posta Şekil 20'deki gibi gönderilir.



Şekil 20. Proje Liderine Gönderilen Bilgi E-Posta

Proje lideri gelen e-mail deki link'e tıklayarak incelenecek değişiklik bilgisini görüntüleyebilir.

5.3 DEĞİŞİKLİK İNCELEME

Proje liderine incelenmesi için gönderilen değişiklik tüm detayları ile birlikte Şekil 21'de gösterilir.

CMS - Change Management System - Microsoft Internet Explorer provided by TUSAS - INTERNET HIZMETLERI

TAI Change Management System (Review Change Request)

Project: A4M - A400M
CR No: [] Rev: [] Status: ALL
Cr Title: []
Date: [] W/P: All
Eng. Leader: All

List

--Change Requests--

PROJECT	CR NO	REV	STATUS	INIT DATE	CR TITLE
A4M	A4M-CR-0001	A	FNAL	2009-12-08	Perçin çivisi deliğinde uyumsuzluk

Revise Save

CR Information

Project: A4M - A400M
CR No: 1 Rev: A Status: REVIEWED
Pr: Low CL: Major
Cr Title: Perçin çivisi deliğinde uyumsuzluk
Date: 08/01/2010 W/P: 72-Engine
Initiator: SERTEL ASIM ÜNSAL
Eng. Leader: TEST AMAÇLI A4M - SERTEL ASIM ÜNSAL

Problem Confirmed
 Problem Not Confirmed
 Another Problem
 Another Solution Proposed
 Initiator's Solution acceptable
 Do Not Send To Agenda

Problem Description: Perçin çivisi deliğinde uyumsuzluk bulunmaktadır, deliğin köşelerinin düzeltilmesi gerekmektedir.

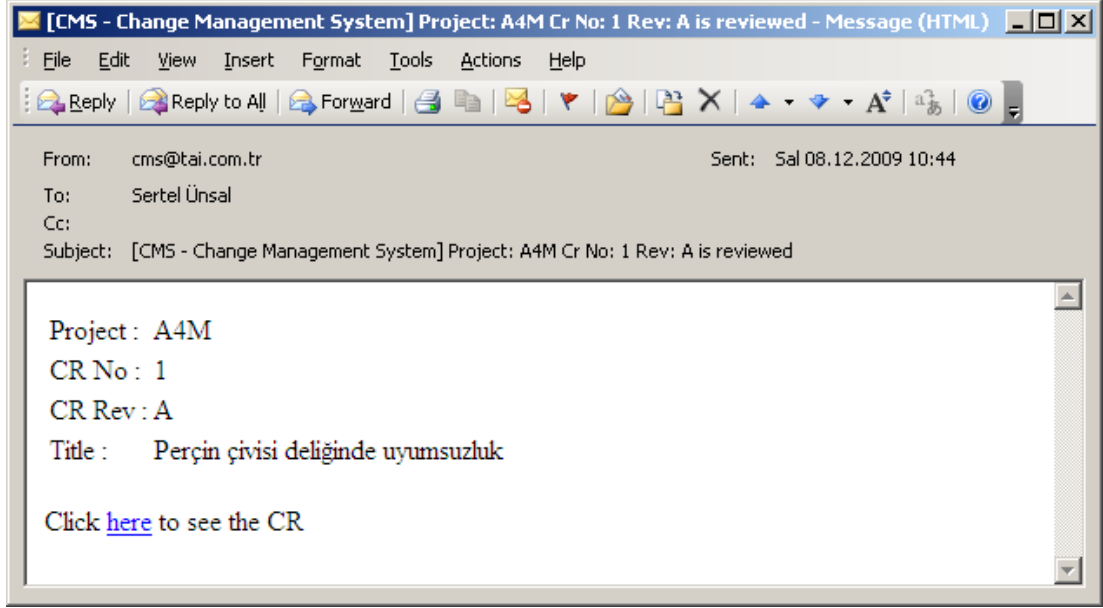
Solution Proposal: Initiator's: Model ve Çizimler güncellenmelidir.
Reviewer's: Uygundur.

Definition Impacted It... Expected Applicabili... Attachmen... Notes

Done Local intranet

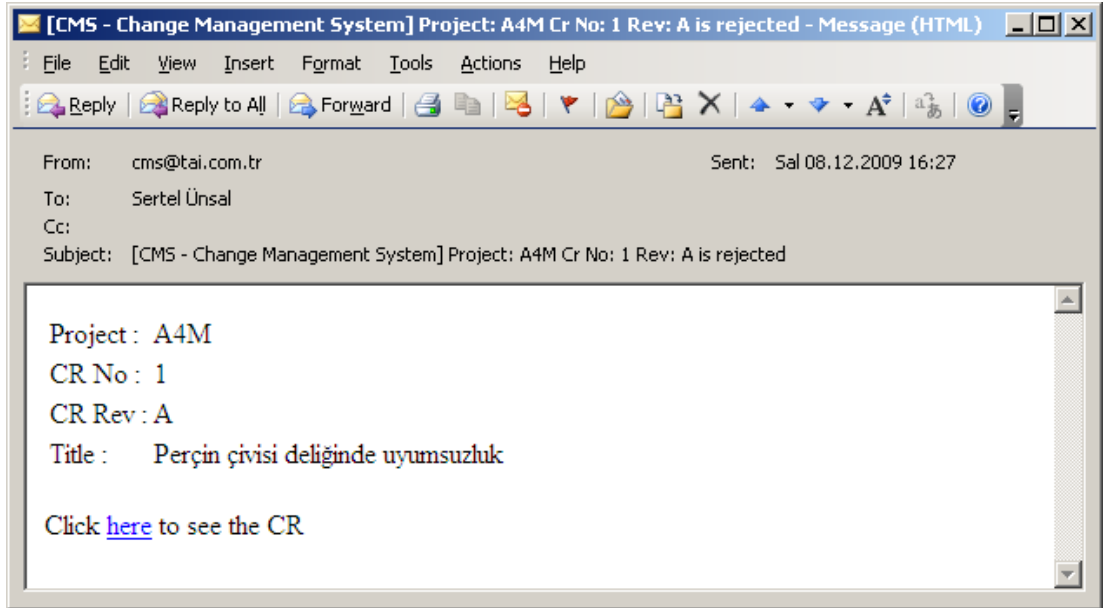
Şekil 21. Değişiklik İnceleme Ekranı

Değişikliği tanımlayan kullanıcının girmiş olduğu bilgilerin hepsini gören proje lideri ek olarak bu değişikliğe kendi görüşlerini, değişikliğin önceliğini ve doğruluğunu belirleyip “Save” tuşuna basarak değişikliği onaylar. Onaylanan değişikliğin durum bilgisi “İNCELENDİ” olarak güncellenir ve bu sırada değişikliği tanımlayan kullanıcıya bilgilendirici bir e-posta Şekil 22’deki gibi gönderilir.



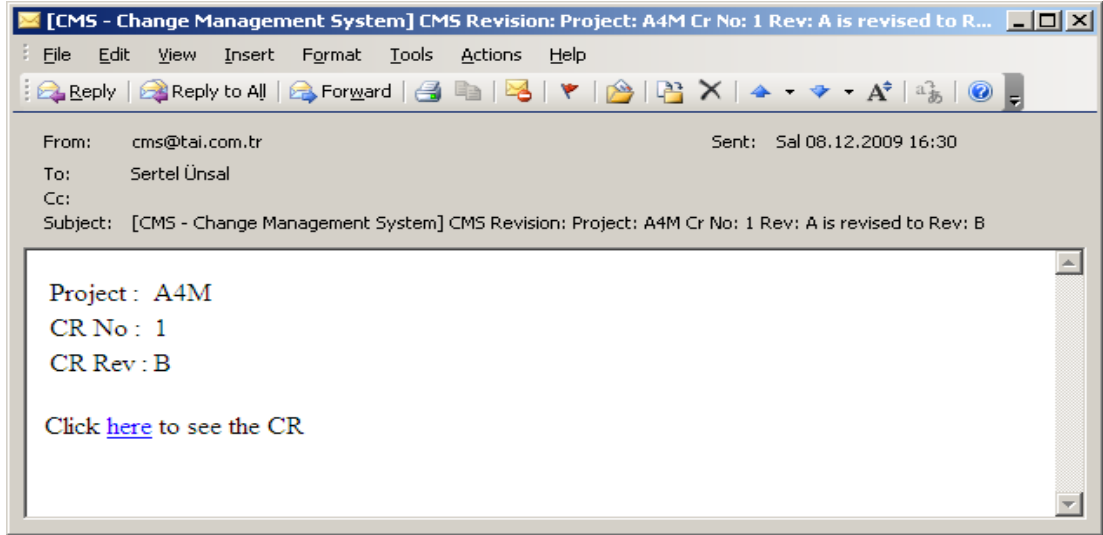
Şekil 22. Değişikliği Tanımlayan Kullanıcıya Gönderilen İncelendi Bilgisi E-Posta

Proje lideri tarafından onaylanmış olan değişiklik etki analizi yapılmak üzere artık konfigürasyon yöneticisi tarafından değerlendirilebilir. Proje lideri bu değişikliği gerekli görmez ve “Problem Not Confirmed” seçeneğini seçip “Save” tuşuna basarsa ise değişik sistemde durum bilgisi “İPTAL” olarak güncellenir ve bir daha bu değişiklik üzerinde hiçbir işlem yapılmaz. Değişikliğin iptal edilmesi durumunda değişikliği tanımlayan kullanıcıya Şekil 23’deki e-mail bilgilendirme amaçlı yolların.



Şekil 23. Değişikliği Tanımlayan Kullanıcıya Gönderilen İptal Bilgisi E-Posta

Eğer Proje lideri değişiklik bilgilerini eksik görür ve bazı yerlerinde düzeltme talep ederse “Revise” tuşuna basarak yeni bir revizyon numarası ile birlikte değişikliği sisteme tanımlayan kullanıcıya tekrar gönderilir. Bu sırada sistemi değişikliği tanımlayan kullanıcıya Şekil 24’deki bilgilendirici e-posta gönderilir.



Şekil 24. Değişikliği Tanımlayan Kullanıcıya Gönderilen Revise Bilgisi E-Posta

5.4 AJANDA OLUŞTURMA

Konfigürasyon Yöneticisi değişikliklerin konuşulması için bir toplantı tanımlar. Şekil 25’deki ekranın en alttaki tablosuna proje liderleri tarafından onaylanmış ve henüz hiçbir toplantıda görüşülmemiş tüm değişiklikler listelenir.

CMS - Change Management System - Microsoft Internet Explorer provided by TUSAS - INTERNET HIZMETLERI

TAI Change Management System (CCB Agenda)

Project : A4M - A400M Find

Outlook Lists: Tez Outlook List Fill Add All

-- CCB List --

Project	No	Date	Time	Place
A4M	A4M-CCB-0001	08/12/2009	14.00	TOPLANTI S...

Add CCB Remove CCB Save CCB

-- CR List of CCB --

Project	CR No	CR Rev	Title
A4M	A4M-CR-0001	A	Percin çivisi deliğinde uyumsuzluk
A4M	A4M-CR-0002	C	Uçağın ön kokpitindeki panel üzerindeki kablo değişimi.

Show CR

↑ ↓

-- CR List of Project --

Project	CR No	CR Rev	Title
A4M	A4M-CR-0001	A	Percin çivisi deliğinde uyumsuzluk
A4M	A4M-CR-0002	C	Uçağın ön kokpitindeki panel üzerindeki kablo değişimi.

Show CR Reject

MENU

Şekil 25. Ajanda Oluşturma Ekranı

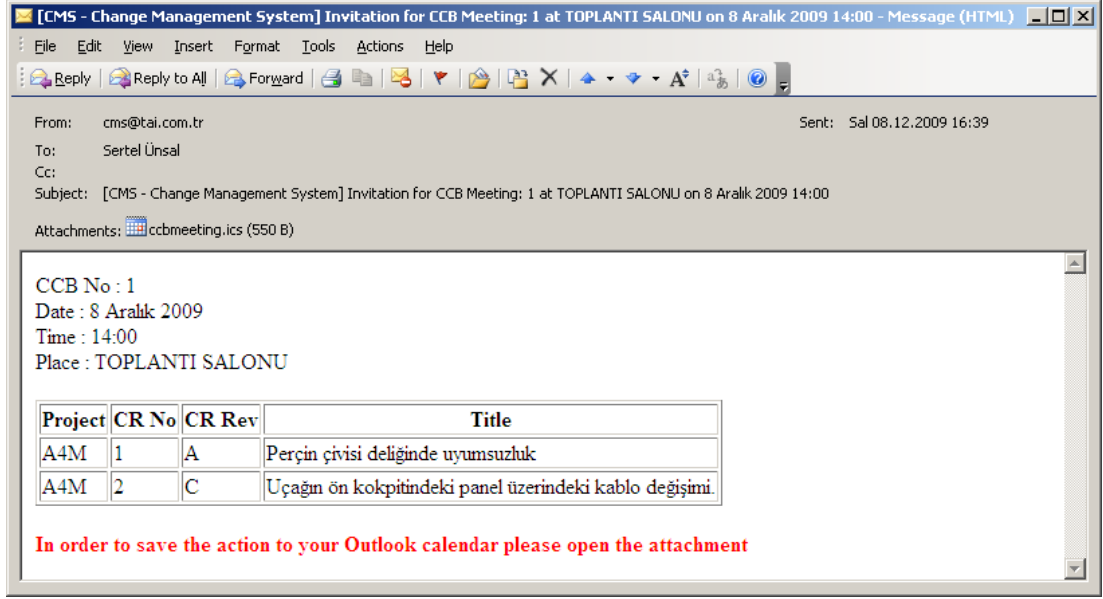
Bu listeden seçilen değişiklikler yukarı ok tuşuna basılarak toplantıda görüşülmek üzere listeye eklenir. Eklenen değişikliğin detaylı bilgilerinin görmek isteyen kullanıcı ekranda bulunan “Show CR” tuşu ile bu değişiklik bilgisine ait Şekil 26’da gösterilen raporu görüntülemiş olur.

A4M	A4M-CR-0001 (Rev. A)	STATUS: <i>REVIEWED</i>
A400M		INITIATE DATE: 25.07.2009
CHANGE REQUEST TITLE: Perçin çivisi deli?inde uyumsuzluk		LEADER: SERTEL ASIM ÜNSAL
		INITIATOR: ERDAL TEK?N
PROBLEM DESCRIPTION	Perçin çivisi deli?inde uyumsuzluk bulunmaktadır , deli?in kö?elerinin düzeltilmesi gerekir.	
SOLUTION PROPOSAL	<i>Initiator's:</i> Model ve çizimler güncellenmelidir. <i>Reviewer's:</i> Uygundur.	

Page 1 of 2

Şekil 26. Değişiklik İsteği Raporu

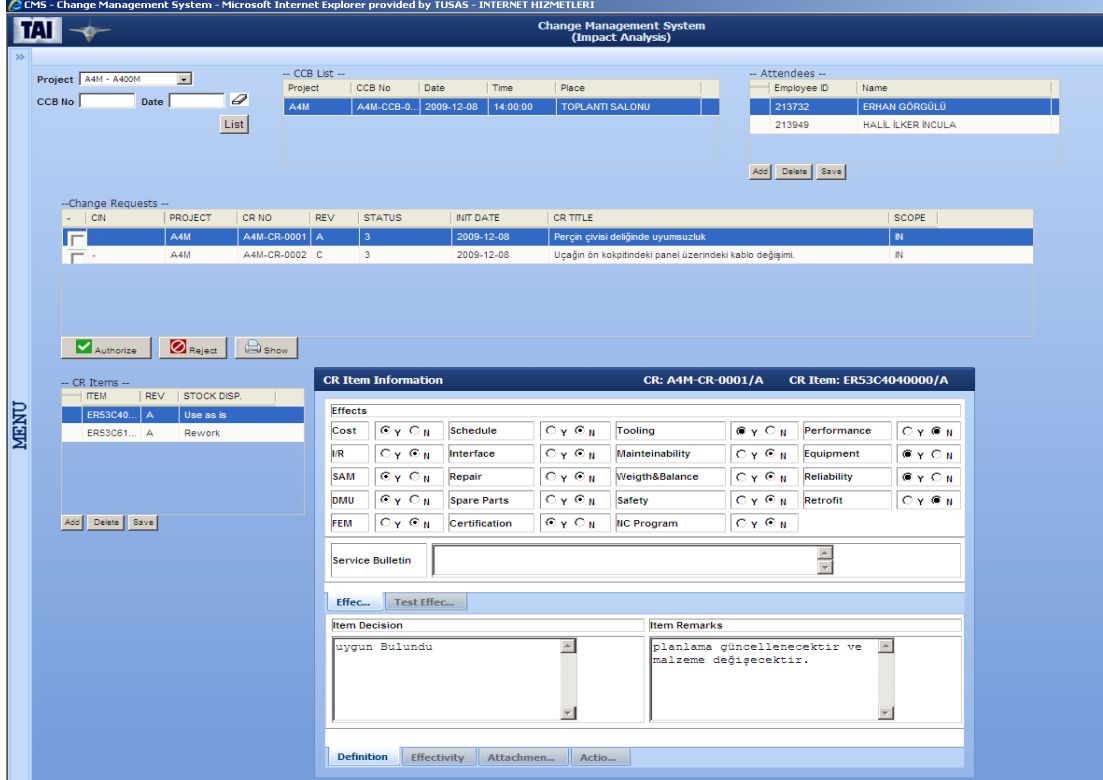
Şekil 25 de bulunan ekranın sağ üst kısmındaki tabloya da toplantıya katılacak kişiler girilir. Bu giriş işlemi ya direk “Add” tuşuna basılarak yapılır veya kullanıcının outlook’unda tanımlı olan listelerden otomatik olarak çekilir. Toplantı sisteme girildikten sonra toplantıya katılacak davetlilere “Invite” tuşuna basılarak Şekil 27’deki e-posta ile birer davet gönderilir. Katılımcı isterse davet ile beraber gönderilen ekteki hatırlatmayı outlook’undaki takvime ekleyebilir, bu sayede toplantı zamanını ve tarihini zamanı gelince hatırlamış olur.



Şekil 27. Toplantı Katılımcılarına Davet E-Posta

5.5 ETKİ ANALİZİ

Toplantı sırasında konuşulan değişikliğin etkilediği parçalar Şekil 28'deki ekrandan girilerek her parça için detaylı bir etki analizi alttaki panel vasıtasıyla sisteme girilir.

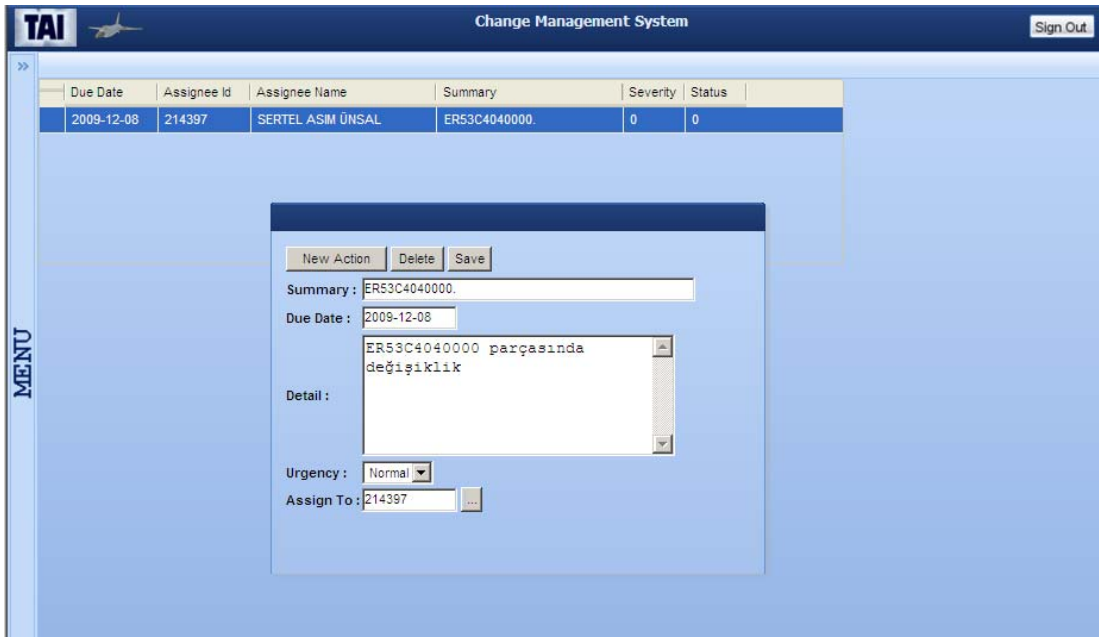


Şekil 28. Etki Analizi Ekranı

Panelde bulunan “Actions” sekmesi seçilerek her parçada oluşan değişiklikler ilgili kişilere atanır.

5.6 ATAMA İŞLEMLERİ

Etki analizi ekranında bulunan “Action” sekmesinde değişikliğin etkilediği parça için atama işlemi Şekil 29’deki ekran yardımıyla yapılır. Bu ekranda yapılması istenilen değişiklik ile ilgili bilgiler, ne yapılması gerektiği, önceliği, tarihi ve kimin bu değişikliği yapacağı bilgileri sisteme “Save” tuşuna basılarak girilir.



The screenshot shows the 'Change Management System' interface. At the top, there is a header with the TAI logo and a 'Sign Out' button. Below the header is a table with the following data:

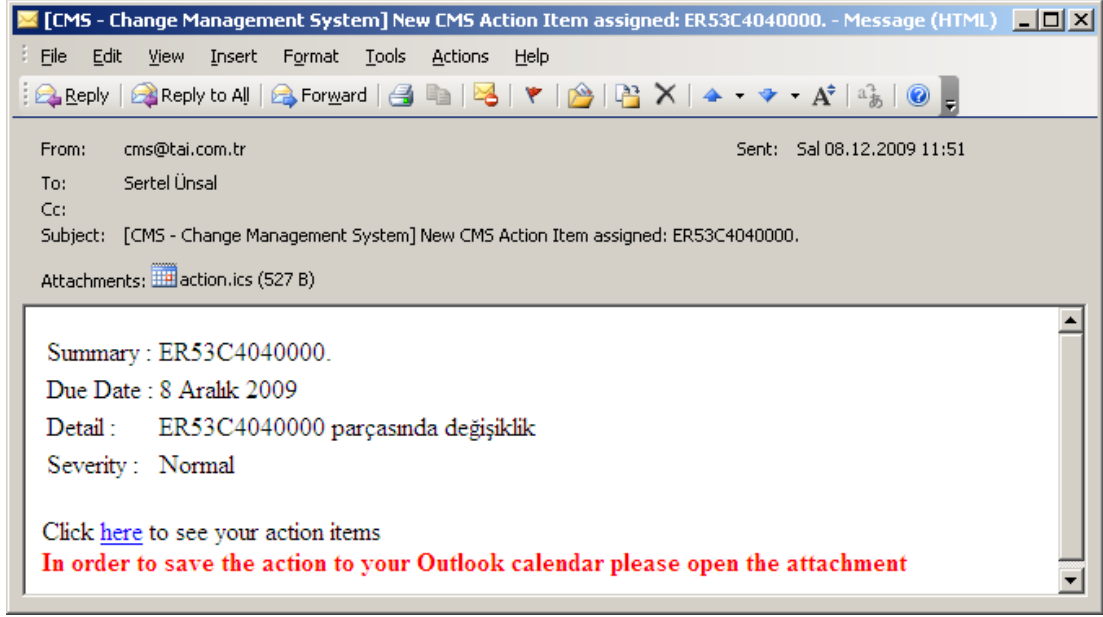
Due Date	Assignee Id	Assignee Name	Summary	Severity	Status
2009-12-08	214397	SERTEL ASIM ÜNSAL	ER53C4040000.	0	0

Below the table, there is a 'MENU' sidebar on the left. The main content area shows a modal window for adding a new action. The modal window has the following fields and controls:

- Buttons: New Action, Delete, Save
- Summary: ER53C4040000.
- Due Date: 2009-12-08
- Detail: ER53C4040000 parçasında değişiklik
- Urgency: Normal (dropdown menu)
- Assign To: 214397 (dropdown menu)

Şekil 29. Değişikliği Atama Ekranı

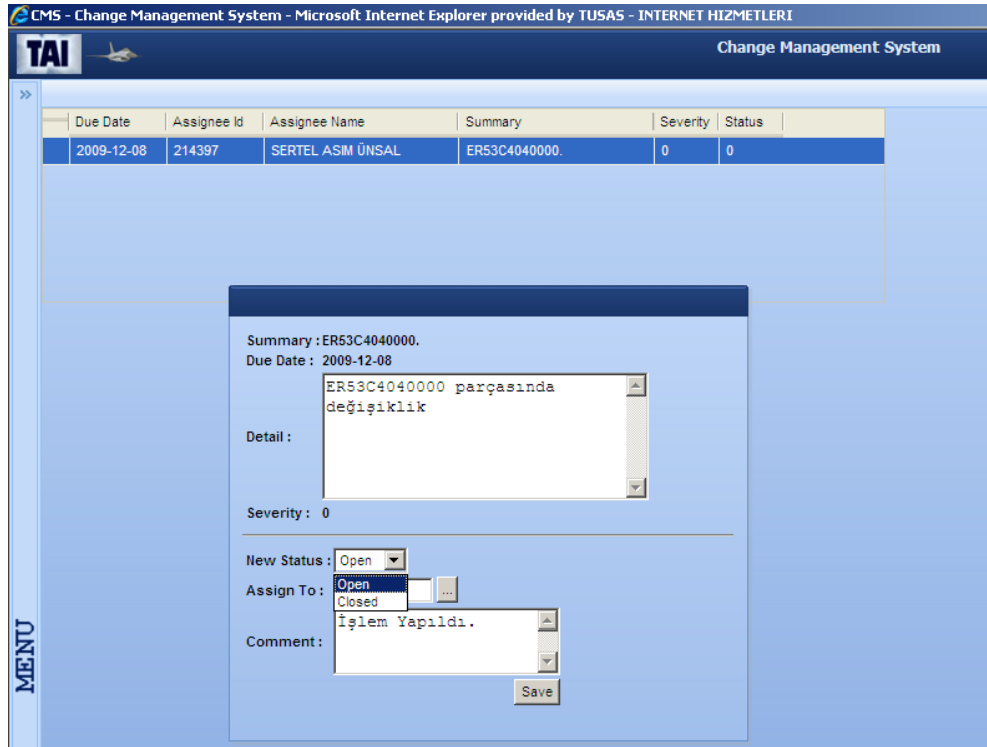
Değişikliğin atandığı kişiye bilgilendirme amaçlı Şekil 30’deki e-mail gönderilir.



Şekil 30. Değişikliğin Atandığı Kişiye Gönderilen E-Posta

Bu mailde ek olarak yapılması istenen değişiklik bilgisi hatırlatma olarak bulunur ve bu hatırlatmayı kullanıcı isterse açarak outlook'undaki takvime kaydedebilir.

Atanan kişi bu değişikliği tamamladıktan sonra Şekil 31'deki ekran yardımıyla sisteme değişikliğin kapandığı bilgisini girer. Tüm değişiklikten etkilenen parçalar atanan kişilerce kapandıktan sonra değişiklik de otomatik olarak kapanmış olur.



Şekil 31. Değişiklik Kapatma Ekranı

5.7 CIN OLUŞTURMA İŞLEMLERİ

Tüm değişiklikten etkilenen parçalar atan kişilerce kapandıktan sonra bu değişiklik bilgisi Şekil 32'deki CIN oluşturma ekranında listelenir. Bu ekranda bir veya birden fazla değişikliğe bir CIN numarası sistem tarafından otomatik olarak atanır. Eğer tanımlanan CIN'e bir eklendi girilmek istenirse bu işlemde aynı ekranda yapılabilir.

--- CIN List ---

Project	ID
A4M	A4M-CIN-0001

--- CR List of CIN ---

Project	CR No	CR Rev	Title	Class	Status
A4M	A4M-CR-0001	A	Perçin çivisi deği...	2	3

Project : A4M - A400M Find

--- CR List ---

Project	CR No	CR Rev	Title	Class	Status
A4M	A4M-CR-0002	C	Uçağın ön kolpilin...	2	3

Şekil 32. CIN Numarası Oluşturma Ekranı

BÖLÜM 6

SONUÇ

Üretim, imalat ve tasarım faaliyeti gösteren kurumlar süreç aşamalarında hiç istenmese de çeşitli nedenlerden dolayı değişiklik istekleri ile her an karşı karşıya kalmaktadırlar. Bu değişiklik isteklerinin çıkması üretilen ürünün potansiyel olarak sorunlu olması anlamına gelmez. Değişiklik ve değişiklik isteği ürünün yaşam dinamiğinin içinde varolan bir gereksinim olarak karşımıza çıkmaktadır.

Araştırmamızda, kurumlardaki süreçlerin hızını, çalışanların performansını ve karar mekanizmasının güvenilirliğini arttırıp, zaman ve kaynak israfını azaltarak kağıtsız bilgi sistemine geçmek hedeflendi. Bu hedefte başarılı olabilmek için ilk olarak “*Değişiklik Yönetimi*” kavramının tarihi incelenip, kurumlarda değişikliğin yönetilmesi ihtiyacının ne zaman ve nasıl doğduğu araştırıldı. Bu ihtiyacı karşılamak için yazılmış diğer sistemler incelenip, geçmişten günümüze kadar geçirmiş oldukları evrelere bakıldı. Bu sistemlerin izlemiş olduğu adımlar ile kullanmış olduğu ortak metotlar belirlenip tasarlanmak istenen sistemde referans olarak kullanıldı.

Dünya piyasasındaki lider firmaların değişiklik yönetimi ve bunu destekleyen ürün veri yönetimi, ürün yaşam döngüsü yönetimi gibi uygulamaları diğer firmalara göre daha çok önemseydiği, değişiklik yönetimi adımlarından ise etki analizi adımına 5 kat daha fazla değer verdiği görülmüştür. Bu sayede ise lider firmaların diğerlerine göre daha doğru ve daha hızlı karar verebildiği anlaşılmıştır [21].

Endüstriyel alanda kullanılan değişiklik yönetim sistemlerinden bahsedilmiştir. Bu alanda yazılmış bir makaleden örnek alınan “A Virtual Environment for Engineering Change Management (*ADVICE*)” sistemi [24] üzerinde durulmuştur. Bu sistemde değişiklik yönetim sistemlerine ek olarak kazandırılan kabiliyetlere değinilmiş, süreçlerin grafiksel olarak yönetilmesi ve eski verilerden yola çıkarak önceliklendirme ile tahmin işlemlerinin yapılabilirliği anlatılmıştır. Bu sistemler sayesinde bilgiye erişimde 10 dakikadan daha fazla ve büyük bir şirket için ise yılda 100 milyon dolardan daha büyük karların sağlanabileceği vurgulanmıştır [30].

Süreçlerde oluşabilecek herhangi bir değişikliğin geçirdiği tüm fazlar ve bu değişikliğin tanımlanmasından tamamlanmasına kadar geçen zaman ve maliyetler çıkartıldı. Bu işlemlerin klasik yöntemlerle yapıldığı zaman kuruma ne gibi zararlar vereceğini ya da benzer hazır sistemleri dışarıdan temin edildiğinde kurumun yapısına uyarlanmasının zorluklarına değinildi. Kurumsal Değişiklik Yönetim Sistemini üretim, imalat ve tasarım yapan her türlü kuruma uygulanabilecek yapıda olduğu gösterip, çalışmanın sonunda bu beklentileri karşılamak adına örnek bir sistemi kurum üzerinde test edildi.

Bu tezde tasarlanan sistemin en modern yazılım geliştirme araçları kullanılarak kurumun kendi yazılım ekibi veya dış kaynaklar yardımı ile geliştirilebilir, sağlanabilir olduğu gösterildi. Sistemin alt yapısı(ara yüzler, veritabanı, programlama vb.) bakımıyla ileriye yönelik adımlara ve teknolojilere açık olduğunu anlatıldı.

Sonuç olarak bu tezde değişiklik yönetim sistemlerinin ne olduğu, gelişimi, önemi ve projelerdeki olumlu katkısı anlatılmıştır. Araştırmada tasarlanan sistem piyasadaki diğer sistemlere bir alternatif olarak geliştirilmiş ve bu sistemlerin temel özelliklerini kendi bünyesinde barındırıp bu sistemlere karşı bir takım avantajlar kullanıcıya sunmuştur. Oluşan değişikliklerin üretim, imalat ve tasarım süreçlerinin %8.5 lik bir kısmını almasından dolayı [20], Kurumsal Değişiklik Yönetim Sistemi bir kavram olmaktan çok kurumlar için bir çözüm olarak tavsiye edilmektedir.

Kurumsal Değişiklik Yönetim Sistemi her zaman yenilikleri destekleyici ve farklı kabiliyetler ile yeni yaklaşımlara açık şekilde tasarlanmıştır.

İleriki çalışmalarda Kurumsal Değişiklik Yönetimi sistemine ek olarak;

- Mühendislik alanlarında da kullanılmasında büyük yarar sağlayan 3D modelleme teknikleri ile süreçlerin daha kavramsal ve somut tasarım ile desteklenmesi sağlanabilir. Bu sayede kullanıcıların konuyu daha iyi anlamaları sağlanıp, daha çok hata yapmaları önenebilir.
- Daha önceden sisteme tanımlı verilerden yola çıkılarak, oluşan değişikliklerin etkileri önceden tahmin edilip bunların aralarında öncelendirme yapmak için veri madenciliği teknikleri kullanılabilir. Veri madenciliği teknikleri sayesinde sisteme daha önceden tanımlanmış benzer bir değişiklik var ise bu tesbit edilerek çift kayıt problemi önlenmiş ve geçmiş tecrübeler sayesinde de kurumlar için çok önemli olan zaman kazanılmış olunur.

- Oluşan deęişiklięin etkiledięi dięer ilişkisel parçaları bulmakta ve bunların aralarındaki baęımlılıklarını belirlemede yardımcı olan Change Propagation analizinin kullanılması önerilir. Bu sayede ürün bilgilerinin sentetik modeller oluşturulması sağlanır ve alınan kararlarda büyük kolaylıklar sağlanmış olur.

KAYNAKLAR

- [1] Güney Güler, Yönetim Bilgi Sistemi, İnönü Üniversitesi Eğitim Fakültesi, Yüksek Lisans, 2009, http://www.egitirim.gen.tr/gguler_bilgisis.htm
- [2] Boğaziçi Yazılım A.Ş., “Ürün Ömrü Yönetimi”, 2009
<http://www.cadcamdizayn.com/makale.asp?ID=108>
- [3] Mehmet Çağrı ELİBOL, Ankara TAI, 23 Ekim 2009 tarihli görüşme
- [4] TAI Configuration Control Full Change Process Document, 2006
- [5] Süleyman Müftüoğlu, “Ürün Yaşamçevrimi Yönetimi”, IBM Türk Ltd. Şti., Ocak 2003, <http://www.grupotomasyon.com.tr/makale-2.html>
- [6] Smarteam Company Web Page
<http://www.3ds.com/products/enovia/welcome/> Erişim Tarihi(01.06.2009)
- [7] SoftTech Company Web Page
<http://www.softech.com/company/about.php>, Erişim Tarihi(01.06.2009)
- [8] Wikipedia, Change management (Engineering)
[http://en.wikipedia.org/wiki/Change_management_\(engineering\)](http://en.wikipedia.org/wiki/Change_management_(engineering)), Erişim Tarihi(01.06.2009)
- [9] YUI Web Page
<http://developer.yahoo.com/yui>, Erişim Tarihi(15.04.2009)
- [10] Ali Osman SERHATOĞLU, Yahoo Kullanıcı Arayüz Kütüphanesi, Ankara, 2009
- [11] AJAX teknolojisi
[http://tr.wikipedia.org/wiki/AJAX_\(programlama\)](http://tr.wikipedia.org/wiki/AJAX_(programlama)), Erişim Tarihi(25.04.2009)
- [12] AJAX teknolojisini kullanan dünya firmaları
http://www.bilgisayardershanesi.com/bilgisayar_dersleri/ajaxi-anlamak.html
- [13] UML modelleme dili, <http://www.uml.org/> , Erişim Tarihi(21.05.2009)
- [14] National Semiconductor, Change Management Sistem Tanımı
<http://www.national.com/analog/quality/changemgmt>, Erişim Tarihi(26.05.2009)
- [15] IBM DB2 SQL Workshop, IBM Certified Course Material, July 2005
- [16] Sun Microsystems Inc., Java, 2009
<http://java.sun.com>, Erişim Tarihi(06.04.2009)
- [17] IEEE Software Engineering Standards Committee, 1998. Document: IEEE Recommended Practice for Software Requirements Specifications

- [18] Barış GEBEŞOĞLU, TAI Software Coding Standards Departmental Instruction., 2007
- [19] Barış GEBEŞOĞLU, TAI Software Development Departmental Instruction, 2007
- [20] J.D. Blackburn, New product development: the new time wars, in: Time-based Competition Business One, Irwin Publishers, Homewood, IL, 1991.
- [21] Aberdeen Group, Better Business Decisions from Intelligent Change Management, September 2007
- [22] Sefer ALGAN, UML, 2002
<http://www.seferalgan.com>,
<http://www.csharpnedir.com/articles/read/?id=6>
- [23] IBM Home Web Page
www-142.ibm.com/software/products/tr/tr/db2enterprise-9 Erişim Tarihi(01-12-2009)
- [24] Koçar, V, Akgündüz A, “ADVICE: A Virtual Environment for Engineering Change Management”, Computers in Industry, 2009
- [25] J. Tavċ ar, J. Duhovnik, Engineering Change Management in individual and Mass production, Robotics and Computer-Aided Manufacturing 21 (3) (2005) 205–215.
- [26] J.D. Blackburn, New product development: the new time wars, in: Time-based Competition Business One, Irwin Publishers, Homewood, IL, 1991.
- [27] A.Y. Yap, O. Ngwenyama, K. Osei-Bryson, Leveraging knowledge representation, usage, and interpretation to help reengineer the product development life cycle:visual computing and the tacit dimensions of product development, Computers in Industry 51 (May (1)) (2003) 89–110.
- [28] A. Rutka, Guenov, D. Marin, Y. Lemmens, T. Schmidt-Schȧ ffer, P. Coleman, A.Rivie` re, Methods for engineering change propagation analysis, in: Proceedings of the ICAS2006 25th International Congress of the Aeronautical Science, Hamburg, Germany, 2006.
- [29] H.Y. Kan, V.G. Duffy, C. Su, An Internet virtual reality collaborative environment for effective product design, Computers in Industry 45 (2) (2001) 197–213.
- [30] N. Stojanovic, L. Stojanovic, Usage-oriented evolution of ontology-based Knowledge management systems, CoopIS/DOA/ODBASE (2002) 1186–1204.
- [31] K. Rouibah, K. Caskey, Change management in concurrent engineering from a parameter perspective, Computers in Industry 50 (2003) 15–34.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ÜNSAL,Sertel Asım
Uyruğu : T.C.
Doğum tarihi ve yeri : 26.07.1981 Ankara
Medeni hali : Evli
Telefon : 0 (532) 543 06 96
e-mail : sunsal@tai.com.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Başkent Üniversitesi/Bilgisayar Müh.	2003

İş Deneyimi

Yıl	Yer	Görev
2004-2005	CyberSoft Yazılım Firması	Yazılım Mühendisi
2005-2006	Teknokur Bilgisayar Sistemleri	Şirket Ortağı/Yönetici
2006-2009	TAI	BilişimYazılım Mühendisi

Yabancı Dil

İngilizce

EKLER

EK 1 DEĞİŞİKLİĞİN TANIMLANDIĞI DOKÜMAN

Change Request

		STATUS:
		LAST UPDATE:

CHANGE REQUEST TITLE:	ATA:
	EPAC/TDU: N/A
CONTEXT TITLE:	ACCOUNTABLE:
	CHANGE OWNER:

REASON:
ORIGIN DOCUMENTS:
EXPECTED APPLICABILITY:

DESCRIPTION OF REQUEST:
ADDITIONAL CONTEXT DESCRIPTION:

REQUESTER:
REQUESTED TARGET DATE FOR MOD OPENING:
URGENCY REASON:

EK 2 DEĞİŞİKLİĞİN DEĞERLENDİRİLDİĞİ DOKÜMAN

Evaluation Sheet

		STATUS:
		LAST UPDATE:

EWAL. SHEET ISSUE*		PRIORITY*	
--------------------	--	-----------	--

IMPACTED ACMT's*							
	MUXX (Wing)	MFXX (F&E)	MYXX (SYS)	MPXX (PWIP)	MAXX (FAL)	MMXX (MIT)	MNXX (OAD)
Proposed							
Confirmed							

ENGINEERING IMPACT*:

INDUSTRIAL IMPACT*:

MP / MOD BREAKDOWN*:

<p>COST*:</p> <p><input type="checkbox"/> Low <input type="checkbox"/> Medium <input type="checkbox"/> High</p> <p>Low: < xyz€, xyz€ < Medium < abc€, High: > abc€</p>	COMMENT:
--	----------

LEADING ACMT AGREEMENT TO LAUNCH EVALUATION	ACCEPT	REASON FOR REJECTION:
	REJECT	
	SHORT LOOP	

PROGRAMME CCC AGREEMENT TO LAUNCH INVESTIGATION	ACCEPT	REASON FOR REJECTION:
	REJECT	

LEADING ACMT CM NAME/DEPT *	SIGNATURE:	DATE *
-----------------------------	------------	--------

EK 3 DEĞİŞİKLİĞİN MODİFİKASYON DOKÜMANI

Modification Document

INITIAL MODIFICATION DOCUMENT FOR:					MODIFICATION PROPOSAL No:		
SUBJECT / TITLE					ORIGINATOR		
					NAME :		
					REF :		
					DATE :		
					SIGNATURE		
ORIGIN:					ATA / SUB-ATA		
DESCRIPTION OF REQUEST OR REASON FOR PROPOSAL							
DESIGN REPLY / SOLUTION PROPOSED							
Leading ACMT:							
ACMT DESIGN OFFICE INVOLVED	CENTRAL	F&E	WING	SYSTEMS	MIT	FAL	POWER PLANT
PRODUCTION INVOLVED	CENTRAL	F&E	WING	SYSTEMS	MIT	FAL	POWER PLANT
NECESSARY TO OBTAIN/ MAINTAIN TYPE CERTIFICATE: FLIGHT TEST NECESSARY TO CERTIFY MOD: PRODUCTION SOLUTION MOD RETROFITABLE: CERTIFICATION CLASSIFICATION: SPARES INTERCHANGEABILITY AFFECTED: SERVICE BULLETIN PROPOSED:					AUTHORIZED SIGNATURE		DESIGN REPLY RESPONSIBLE
					DATE		DATE
OTHER PROGRAM(S) AFFECTED: NONE					APPLICABILITY: ST1		

EK 4 DEĞİŞİKLİK İSTEĞİ RAPORU

A4M	A4M-CR-0001 (Rev. A)	STATUS:	REVIEWED
A400M		INITIATE DATE:	25.07.2009

CHANGE REQUEST TITLE: Perçin çivisi deli?inde uyumsuzluk	LEADER: SERTEL ASIM ÜNSAL
	INITIATOR: ERDAL TEK?N

PROBLEM DESCRIPTION	Perçin çivisi deli?inde uyumsuzluk bulunmaktadır , deli?in kö?elerinin düzeltilmesi gerekir.
---------------------	--

SOLUTION PROPOSAL	<i>Initiator's:</i> Model ve çizimler güncellenmelidir. <i>Reviewer's:</i> Uygundur.
-------------------	---

EK 5 DEĞİŞİKLİK İSTEĞİ TARİHÇE RAPORU

CHANGE REQUEST STATUS HISTORY

09.05.2009

PROJECT	CR NO	REV.	STATUS	ENTRY	EXIT	DUR. (DAYS)
A4M	A4M-CR-0001	A	DRAFT	25.07.2009	25.07.2009	0,00
A4M	A4M-CR-0001	A	DRAFT	25.07.2009	25.07.2009	0,00
A4M	A4M-CR-0001	A	DRAFT	25.07.2009	25.07.2009	0,00
A4M	A4M-CR-0001	A	DRAFT	25.07.2009	25.07.2009	0,00
A4M	A4M-CR-0001	A	DRAFT	25.07.2009	25.07.2009	0,00
A4M	A4M-CR-0001	A	FINAL	25.07.2009	25.07.2009	0,00
A4M	A4M-CR-0001	A	FINAL	25.07.2009	25.07.2009	0,00
A4M	A4M-CR-0001	A	FINAL	25.07.2009	25.07.2009	0,00
A4M	A4M-CR-0001	A	FINAL	25.07.2009	25.07.2009	0,00
A4M	A4M-CR-0001	A	FINAL	25.07.2009	25.07.2009	0,00
A4M	A4M-CR-0001	A	REVIEWED	25.07.2009	26.07.2009	1,00
A4M	A4M-CR-0001	A	REVIEWED	25.07.2009	26.07.2009	1,00
A4M	A4M-CR-0001	A	REVIEWED	26.07.2009		41,00
A4M	A4M-CR-0001	A	RELEASED	26.07.2009	26.07.2009	0,00
A4M	A4M-CR-0002	A	DRAFT	25.07.2009		42,00
A4M	A4M-CR-0002	A	DRAFT	25.07.2009		42,00
A4M	A4M-CR-0003	A	DRAFT	25.07.2009		42,00
A4M	A4M-CR-0004	A	DRAFT	25.07.2009		42,00

EK 6 YAPILACAK İŞLERİN TARİHÇE RAPORU

ACTION STATUS HISTORY

05.09.2009

PROJECT	CR NO	REV.	ITEM	ITEM REV.	ASSIGNEE	OPEN DATE	DUE DATE	CLOSE DATE	DUR. (DAYS)
A4M	A4M-CR-0001	A	1	A	SERTEL ASIM ÜNSAL	07.26.2009	26.07.2009	26.07.2009	0

EK 7 VERİ TABANINA TABLO OLUŞTURMA SQL ÖRNEKLERİ

```
CREATE TABLE "DYS"."CCB_AGENDA" (  
    "CCB_NO"          INTEGER NOT NULL,  
    "CCB_DATE"       DATE,  
    "CCB_TIME"       TIME,  
    "CCB_PLACE"      VARCHAR(30),  
    "PROJECT"        CHAR(10) NOT NULL  
)  
GO  
ALTER TABLE DYS.CCB_AGENDA ADD PRIMARY KEY (PROJECT, CCB_NO)  
GO  
  
CREATE TABLE "DYS"."CCB_ATTENDEE" (  
    "CCB_NO"          INTEGER NOT NULL REFERENCES  
                        DYS.CCB_AGENDA(CCB_NO),  
    "USERID"         CHAR(6) NOT NULL,  
    "PROJECT"        CHAR(10) NOT NULL REFERENCES  
                        DYS.CCB_AGENDA(PROJECT)  
)  
GO  
  
CREATE TABLE "DYS"."CCB_CR" (  
    "CCB_NO"          INTEGER NOT NULL,  
    "CR_NO"          INTEGER NOT NULL REFERENCES  
                        DYS.CHANGE_REQUEST(CR_NO),  
    "CR_REV"         CHAR(2) NOT NULL REFERENCES  
                        DYS.CHANGE_REQUEST(CR_REV),  
    "PROJECT"        CHAR(10) NOT NULL REFERENCES  
                        DYS.CHANGE_REQUEST(PROJECT)  
)  
GO  
ALTER TABLE DYS.CCB_CR ADD PRIMARY KEY (CCB_NO, CR_NO, CR_REV,  
    PROJECT)  
GO  
  
CREATE TABLE "DYS"."CHANGE_REQUEST" (  
    "CR_NO"          INTEGER NOT NULL,  
    "CR_REV"         CHAR(2) NOT NULL,  
    "PROJECT"        CHAR(10) NOT NULL,
```

```

"WORK_PRODUCT"          CHAR(10) NOT NULL,
"STATUS"                SMALLINT NOT NULL,
"INITIATE_DATE"        DATE NOT NULL,
"TITLE"                 LONG VARCHAR NOT NULL,
"INITIATOR"             CHAR(6) NOT NULL,
"PROBLEM_DESCRIPTION"  LONG VARCHAR NOT NULL,
"SOLUTION_PROPOSAL"    LONG VARCHAR NOT NULL,
"PROBLEM_CONFIRMED"    SMALLINT NOT NULL,
"PROBLEM_NOT_CONFIRMED" SMALLINT NOT NULL,
"ANOTHER_PROBLEM"      SMALLINT NOT NULL,
"ANOTHER_SOLUTION"     SMALLINT NOT NULL,
"ACCEPTABLE_SOLUTION" SMALLINT NOT NULL,
"CR_CLASS"              SMALLINT NOT NULL,
"PRIORITY"              SMALLINT NOT NULL,
"REVIEW_DATE"          DATE NOT NULL,
"LEADER"                CHAR(6) NOT NULL,
"CR_DECISION"           VARCHAR(30) NOT NULL,
"CR_REMARKS"            VARCHAR(100) NOT NULL,
"DESIGN_STATUS"        SMALLINT NOT NULL,
"SCOPE"                 SMALLINT NOT NULL,
"REV_DESCRIPTION"       VARCHAR(255) NOT NULL,
"NOT_SEND_AGENDA"      SMALLINT NOT NULL,
"REV_NOTE"              LONG VARCHAR,
"REVIEWER_SOLUTION"    LONG VARCHAR NOT NULL
)
GO
ALTER TABLE DYS.CHANGE_REQUEST ADD PRIMARY KEY (CR_NO, CR_REV,
          PROJECT)
GO

CREATE TABLE "DYS"."CR_AUTH" (
    "USERID"          CHAR(6) NOT NULL,
    "PROJECT"         CHAR(10) NOT NULL,
    "AUTH"            CHAR(10) NOT NULL
)
GO

CREATE TABLE "DYS"."CR_CAN" (
    "CAN"             INTEGER NOT NULL,
    "PROJECT"         CHAR(10) NOT NULL REFERENCES
                    DYS.CHANGE_REQUEST(PROJECT),

```

```

"CR_NO"      INTEGER NOT NULL REFERENCES
              DYS.CHANGE_REQUEST(CR_NO),
"CR_REV"     CHAR(2) NOT NULL REFERENCES
              DYS.CHANGE_REQUEST(CR_REV)
)
GO
ALTER TABLE DYS.CR_CAN ADD PRIMARY KEY (CAN, CR_NO, CR_REV, PROJECT)
GO

```

```

CREATE TABLE "DYS"."CR_CAN_FILE" (
  "CAN"      INTEGER NOT NULL REFERENCES
              DYS.CR_CAN(CAN),
  "PROJECT"  CHAR(10) NOT NULL REFERENCES
              DYS.CR_CAN(PROJECT),
  "CR_NO"    INTEGER NOT NULL REFERENCES
              DYS.CR_CAN(CR_NO),
  "CR_REV"   CHAR(2) NOT NULL REFERENCES
              DYS.CR_CAN(CR_REV),
  "FILE_REV" CHAR(2) NOT NULL,
  "FILENAME" VARCHAR(100) NOT NULL,
  "FILE"     BLOB(20971520) NOT NULL
)
GO
ALTER TABLE DYS.CR_CAN_FILE ADD PRIMARY KEY (CAN, CR_NO, CR_REV,
  PROJECT, FILE_REV)
GO

```

```

CREATE TABLE "DYS"."CR_EFFECTED_ITEM" (
  "PROJECT"  CHAR(10) NOT NULL REFERENCES
              DYS.CHANGE_REQUEST(PROJECT),
  "CR_NO"    INTEGER NOT NULL REFERENCES
              DYS.CHANGE_REQUEST(CR_NO),
  "CR_REV"   CHAR(2) NOT NULL REFERENCES
              DYS.CHANGE_REQUEST(CR_REV),
  "OLD_ITEM_NO"  VARCHAR(100),
  "OLD_ITEM_REV" CHAR(2),
  "NEW_ITEM_NO"  VARCHAR(100) NOT NULL,
  "NEW_ITEM_REV" CHAR(2) NOT NULL,
  "ITEM_TYPE"   SMALLINT NOT NULL
)
GO

```

```
ALTER TABLE DYS.CR_EFFECTED_ITEM ADD PRIMARY KEY (PROJECT, CR_NO,  
          CR_REV, NEW_ITEM_NO, NEW_ITEM_REV, ITEM_TYPE)
```

```
GO
```

```
CREATE TABLE "DYS"."CR_EFFECTIVITY" (  
          "CR_NO"      INTEGER NOT NULL REFERENCES  
                        DYS.CHANGE_REQUEST(CR_NO),  
          "CR_REV"    CHAR(2) NOT NULL REFERENCES  
                        DYS.CHANGE_REQUEST(CR_REV),  
          "TV"        CHAR(3) NOT NULL,  
          "EFFFROM"   CHAR(4) NOT NULL,  
          "EFTHR"     CHAR(4) NOT NULL,  
          "PROJECT"   CHAR(10) NOT NULL REFERENCES  
                        DYS.CHANGE_REQUEST(PROJECT)  
)
```

```
GO
```

```
ALTER TABLE DYS.CR_EFFECTIVITY ADD PRIMARY KEY (CR_NO, CR_REV,  
          EFFFROM, EFTHR, PROJECT, TV)
```

```
GO
```

```
CREATE TABLE "DYS"."CR_ITEM" (  
          "CR_NO"      INTEGER NOT NULL REFERENCES  
                        DYS.CHANGE_REQUEST(CR_NO),  
          "CR_REV"    CHAR(2) NOT NULL REFERENCES  
                        DYS.CHANGE_REQUEST(CR_REV),  
          "ITEM_NO"   VARCHAR(100) NOT NULL,  
          "ITEM_REV"  CHAR(2) NOT NULL,  
          "PROJECT"   CHAR(10) NOT NULL REFERENCES  
                        DYS.CHANGE_REQUEST(PROJECT),  
          "STATUS"    INTEGER NOT NULL,  
          "COST_EFFECT" SMALLINT,  
          "SCHEDULE_EFFECT" SMALLINT,  
          "TOOLING_EFFECT" SMALLINT,  
          "NC_PROG_EFFECT" SMALLINT,  
          "REPAIR_EFFECT" SMALLINT,  
          "PERFORMANCE_EFFECT" SMALLINT,  
          "WB_EFFECT" SMALLINT,  
          "FEM_EFFECT" SMALLINT,  
          "SAM_EFFECT" SMALLINT,  
          "IR_EFFECT" SMALLINT,
```

```

"DMU_EFFECT"          SMALLINT,
"INTERFACE_EFFECT"    SMALLINT,
"RELIABILITY_EFFECT" SMALLINT,
"MAINTAIN_EFFECT"     SMALLINT,
"SPAREPART_EFFECT"    SMALLINT,
"EQUIPMENT_EFFECT"    SMALLINT,
"SAFETY_EFFECT"       SMALLINT,
"CERTIF_EFFECT"       SMALLINT,
"RETROFIT_EFFECT"     SMALLINT,
"FLIGHT_TEST"         SMALLINT,
"GROUND_TEST"         SMALLINT,
"STATIC_TEST"         SMALLINT,
"PROPULSION_TEST"     SMALLINT,
"LABORATORY_TEST"     SMALLINT,
"CERTIF_TEST"         SMALLINT,
"SRVBULLETIN_EFFECT" VARCHAR(100),
"ITEM_DECISION"       VARCHAR(30),
"ITEM_REMARKS"        VARCHAR(100),
"STOCK_DISPOSITION"   VARCHAR(30)
)
GO
ALTER TABLE DYS.CR_ITEM ADD PRIMARY KEY (CR_NO, CR_REV, ITEM_NO,
      ITEM_REV, PROJECT)
GO
CREATE TABLE "DYS"."CR_ITEM_ACTION_ITEM" (
"CR_NO"          INTEGER NOT NULL,
"CR_REV"         CHAR(2) NOT NULL,
"ITEM_NO"        VARCHAR(100) NOT NULL,
"ITEM_REV"       CHAR(2) NOT NULL,
"ACTION_ID"      INTEGER NOT NULL PRIMARY KEY,
"DUE_DATE"       DATE,
"ASSIGNTO"       VARCHAR(45),
"SUMMARY"        VARCHAR(150),
"DETAIL"         VARCHAR(150),
"SEVERITY"       INTEGER,
"PRIORITY"       INTEGER,
"COMMENT"        VARCHAR(150),
"STATUS"         INTEGER NOT NULL,
"PROJECT"        CHAR(10) NOT NULL,
"OPEN_DATE"      TIMESTAMP NOT NULL,

```

```

        "CLOSE_DATE"    TIMESTAMP
    )
GO

CREATE TABLE "DYS"."CR_ITEM_ATTACHMENT" (
    "CR_NO"            INTEGER NOT NULL REFERENCES
                        DYS.CR_ITEM(CR_NO),
    "CR_REV"           CHAR(2) NOT NULL REFERENCES
                        DYS.CR_ITEM(CR_REV),
    "ITEM_NO"          VARCHAR(100) NOT NULL REFERENCES
                        DYS.CR_ITEM(ITEM_NO),
    "ITEM_REV"         CHAR(2) NOT NULL REFERENCES
                        DYS.CR_ITEM(ITEM_REV),
    "FILE_NAME"        VARCHAR(100) NOT NULL,
    "ID"               BIGINT NOT NULL,
    "PROJECT"          CHAR(10) NOT NULL REFERENCES
                        DYS.CR_ITEM(PROJECT)
)
GO

ALTER TABLE DYS.CR_ITEM_ATTACHMENT ADD PRIMARY KEY (CR_NO, CR_REV,
    ID, ITEM_NO, ITEM_REV, PROJECT)
GO

CREATE TABLE "DYS"."CR_ITEM_EFFECTED_DOCS" (
    "CR_NO"            INTEGER NOT NULL REFERENCES
                        DYS.CR_ITEM(CR_NO),
    "CR_REV"           CHAR(2) NOT NULL REFERENCES
                        DYS.CR_ITEM(CR_REV),
    "ITEM_NO"          VARCHAR(100) NOT NULL REFERENCES
                        DYS.CR_ITEM(ITEM_NO),
    "ITEM_REV"         CHAR(2) NOT NULL REFERENCES
                        DYS.CR_ITEM(ITEM_REV),
    "DOC_NO"           VARCHAR(100) NOT NULL,
    "DOC_REV"          CHAR(1),
    "PROCESS"          VARCHAR(100),
    "PROJECT"          CHAR(10) NOT NULL REFERENCES
                        DYS.CR_ITEM(PROJECT)
)
GO

CREATE TABLE "DYS"."CR_ITEM_EFFECTIVITY" (

```

```

"CR_NO"          INTEGER NOT NULL REFERENCES
                  DYS.CR_ITEM(CR_NO),
"CR_REV"         CHAR(2) NOT NULL REFERENCES
                  DYS.CR_ITEM(CR_REV),
"ITEM_NO"        VARCHAR(100) NOT NULL REFERENCES
                  DYS.CR_ITEM(ITEM_NO),
"ITEM_REV"       CHAR(2) NOT NULL REFERENCES
                  DYS.CR_ITEM(ITEM_REV),
"TV"             CHAR(3) NOT NULL,
"EFFFROM"        CHAR(4) NOT NULL,
"EFTHR"          CHAR(4) NOT NULL,
"PROJECT"        CHAR(10) NOT NULL REFERENCES
                  DYS.CR_ITEM(PROJECT)
)
GO
ALTER TABLE DYS.CR_ITEM_EFFECTIVITY ADD PRIMARY KEY (CR_NO, CR_REV,
              EFFFROM, EFTHR, ITEM_NO, ITEM_REV, PROJECT, TV)
GO

```

```

CREATE TABLE "DYS"."CR_PROBLEM_ATTACHMENT" (
  "CR_NO"          INTEGER NOT NULL REFERENCES
                    DYS.CHANGE_REQUEST(CR_NO),
  "CR_REV"         CHAR(2) NOT NULL REFERENCES
                    DYS.CHANGE_REQUEST(CR_REV),
  "FILE_NAME"     VARCHAR(100) NOT NULL,
  "ID"             BIGINT NOT NULL,
  "PROJECT"        CHAR(10) NOT NULL REFERENCES
                    DYS.CHANGE_REQUEST(PROJECT)
)
GO
ALTER TABLE DYS.CR_PROBLEM_ATTACHMENT ADD PRIMARY KEY (CR_NO,
              CR_REV, ID, PROJECT)
GO

```

```

CREATE TABLE "DYS"."CR_STATUS_HISTORY" (
  "PROJECT"        CHAR(10) NOT NULL,
  "CR_NO"          INTEGER NOT NULL,
  "CR_REV"         CHAR(2) NOT NULL,
  "STATUS"         SMALLINT NOT NULL,
  "START_DATE"    TIMESTAMP NOT NULL,

```



```

        "END_DATE"      TIMESTAMP
    )
GO
ALTER TABLE DYS.CR_STATUS_HISTORY ADD PRIMARY KEY (PROJECT, CR_NO,
        CR_REV, STATUS, START_DATE)
GO

CREATE TABLE "DYS"."LEADER" (
        "PROJECT"      CHAR(10) NOT NULL,
        "PROCESS"      VARCHAR(45) NOT NULL,
        "LEADER"      CHAR(6) NOT NULL
    )
GO
ALTER TABLE DYS.LEADER ADD PRIMARY KEY (PROCESS, PROJECT)
GO

```

EK 8 VERİ TABANINDA KULLANILAN TRIGGER ÖRNEKLERİ

```
CREATE TRIGGER CMS.TR_CRITEM_1
AFTER INSERT ON CMS.CR_ITEM
REFERENCING NEW AS N
FOR EACH ROW MODE DB2SQL
INSERT INTO CMS.CR_ITEM_EFFECTIVITY (PROJECT, CR_NO, CR_REV,
ITEM_NO, ITEM_REV, TV, EFFROM, EFTHR)
(SELECT N.PROJECT, N.CR_NO, N.CR_REV, N.ITEM_NO, N.ITEM_REV, TV,
EFFROM, EFTHR FROM CMS.CR_EFFECTIVITY WHERE
PROJECT=N.PROJECT AND CR_NO=N.CR_NO AND CR_REV=N.CR_REV)
```

```
(CREATE TRIGGER CMS.TR_CRITEM_1
AFTER INSERT ON CMS.CR_ITEM
REFERENCING NEW AS N
FOR EACH ROW MODE DB2SQL
INSERT INTO CMS.CR_ITEM_EFFECTIVITY (PROJECT, CR_NO,
CR_REV, ITEM_NO, ITEM_REV, TV, EFFROM, EFTHR)
(SELECT N.PROJECT, N.CR_NO, N.CR_REV, N.ITEM_NO,
N.ITEM_REV, TV, EFFROM, EFTHR FROM CMS.CR_EFFECTIVITY
WHERE PROJECT=N.PROJECT AND CR_NO=N.CR_NO AND
CR_REV=N.CR_REV))
```

```
CREATE TRIGGER CMS.TR_CHANGEREQUEST_1 AFTER
UPDATE ON CMS.CHANGE_REQUEST REFERENCING NEW AS N OLD AS O FOR EACH
ROW MODE DB2SQL BEGIN ATOMIC IF O.STATUS<>N.STATUS THEN
UPDATE CMS.CR_STATUS_HISTORY SET END_DATE=CURRENT_TIMESTAMP
WHERE PROJECT=O.PROJECT AND CR_NO=O.CR_NO AND CR_REV=O.CR_REV AND
STATUS=O.STATUS;
INSERT INTO CMS.CR_STATUS_HISTORY
(PROJECT ,CR_NO ,CR_REV ,STATUS ,START_DATE) VALUES
(N.PROJECT,N.CR_NO,N.CR_REV,N.STATUS,CURRENT_TIMESTAMP);
IF (N.STATUS=1 AND O.STATUS=3) OR (N.STATUS=0 AND O.STATUS=3) THEN
DELETE FROM CMS.CR_ITEM WHERE PROJECT=N.PROJECT AND
CR_NO=N.CR_NO AND CR_REV=N.CR_REV;
```

```

DELETE from CMS.CR_ITEM_EFFECTIVITY WHERE CR_NO=N.CR_NO AND
      CR_REV=N.CR_REV;
DELETE from CMS.CR_ITEM_ACTION_ITEM WHERE CR_NO=N.CR_NO AND
      CR_REV=N.CR_REV;
DELETE from CMS.CR_ITEM_EFFECTED_DOCS WHERE CR_NO=N.CR_NO AND
      CR_REV=N.CR_REV;
DELETE from CMS.CR_ITEM_ATTACHMENT WHERE CR_NO=N.CR_NO AND
      CR_REV=N.CR_REV;

END IF;
IF N.STATUS=3 THEN
INSERT INTO CMS.CR_ITEM
(
  PROJECT, CR_NO, CR_REV, ITEM_NO, ITEM_REV, STATUS
)
(
  SELECT
  PROJECT, CR_NO, CR_REV, NEW_ITEM_NO, NEW_ITEM_REV, 0
  FROM CMS.CR_EFFECTED_ITEM
  WHERE PROJECT=N.PROJECT AND CR_NO=N.CR_NO
        AND CR_REV=N.CR_REV
)
;
END IF
;
END IF
;
END

CREATE TRIGGER CMS.TR_CHANGEREQUEST_2 AFTER
INSERT ON CMS.CHANGE_REQUEST
REFERENCING NEW AS N FOR EACH ROW MODE DB2SQL
INSERT INTO CMS.CR_STATUS_HISTORY (PROJECT, CR_NO, CR_REV, STATUS,
      START_DATE) VALUES(N.PROJECT, N.CR_NO, N.CR_REV,
      N.STATUS, CURRENT_TIMESTAMP)

```