T.C.
DOKUZ EYLÜL ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
YÖNETİM BİLİŞİM SİSTEMLERİ ANABİLİM DALI
YÖNETİM BİLİŞİM SİSTEMLERİ PROGRAMI
YÜKSEK LİSANS TEZİ

AÇIK KAYNAK PLATFORM ÜZERİNDE İŞ SÜREÇLERİNİN TASARIMI VE YÖNETİMİ

Ahmet Volkan ERGÜL

Danışman
Prof. Dr. Vahap TECİM

İZMİR – 2023

TEZ ONAY SAYFASI

YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum “Açık Kaynak Platform Üzerinde İş Süreçlerinin Tasarımı ve Yönetimi” adlı çalışmanın, tarafımdan, akademik kurallara ve etik değerlere uygun olarak yazıldığını ve yararlandığım eserlerin kaynakçadan gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve bunu onurumla doğrularım.

17/08/2023

Ahmet Volkan ERGÜL

ÖZET

Yüksek Lisans Tezi

Açık Kaynak Platform Üzerinde İş Süreçlerinin

Tasarımı ve Yönetimi

Ahmet Volkan ERGÜL

Dokuz Eylül Üniversitesi

Sosyal Bilimler Enstitüsü

Yönetim Bilişim Sistemleri Anabilim Dalı

Yönetim Bilişim Sistemleri Programı

Bu tezin amacı işletmelerde iş akışlarını, veri alanlarını, formları, iş kurallarını, servis seviyelerini, rol ve yetkileri sürükle bırak mantığı ile tasarlayarak Bilgi Teknolojileri (BT) bağımlılığını azaltmak ve ilgili süreçlerin yol haritalarını canlı bir şekilde görmek ve takip etmektir. Ayrıca sürecin tamamlanmasının ardından elde edilen veriler analiz edilerek sistemi geliştirme, süreçlerinin iyileştirilmesi kapsamında yeni veri oluşmasına olanak sağlayacaktır. İşletmelerdeki tüm iş süreçleri farklı metotlarla takip ediliyor, sürdürülüyor gibi görünse de dağıtık yapıdaki bu tür çözümler zamanla bir dağ gibi bilgi kirliliğine, yönetiminin ve takibinin zorlaşmasına neden olmaktadır.

Projede 2 farklı süreç tanımlanmış ve süreçler bir değişiklik yönetim aracı üzerinde uygulanmıştır. Sürece özel durumlar ve durum geçişleri ile ilgili sorumlu roller tanımlanmıştır. Süreç içinde ve sonrasında analiz edilmesi planlanan verilerin de kaydının tutulması sağlanmıştır. Tanımlanan süreçler Düzeltici Önleyici Faaliyetler (DÖF) ve Giriş Kalite Kontrol (GKK) olarak akışları tasarlanmış ve ele alınmıştır. Süreçlerle ilgili olarak bir prototip çalışma yapılması, sonrasında işletmeden alınan geri bildirimler ve sürece yönelik çözüm tasarım raporu oluşturulmuştur. Sonrasında teknik kontroller yapılarak süreç yöneticileri ve sorumluları ile paylaşılarak tasarıma geçilmiştir.

İşletmedeki süreç sorumluları ile yapılan analiz toplantılarında sürece ilişkin veri yapıları aktarılmış ve ilgili dokümanlar paylaşılmıştır. Kullanıcı

entegrasyonları Active Directory ile bütünleştirilerek, işletmenin içinde oluşturulan organizasyonel şemaya bağlı kalınmıştır.

Anahtar Kelimeler: Bilgi Teknolojileri, Düzeltici Önleyici Faaliyetler, Giriş Kalite Kontrol, İşletme, Kişisel Verilerin Korunma Kanunu

ABSTRACT

Master's Thesis

Management and Design of Business Processes On

An Open Source Platform

Ahmet Volkan ERGUL

Dokuz Eylul University

Graduate School of Social Sciences

Department of Management Information Systems

Management Information Systems Program

The aim of this project is to reduce IT dependency by designing workflows, data fields, forms, business rules, service levels, roles and authorizations with drag-and-drop logic and to see and follow the roadmaps of related processes in a live way. In addition, the data obtained after the completion of the process are analyzed and data are obtained within the scope of system development and improvement of processes. Although it seems that all business processes in enterprises are followed and continued with different methods, such solutions in a distributed structure cause information pollution like a mountain over time and a formation that cannot be easily followed up.

In the project, 2 different processes were defined and the processes were implemented on a change management tool. Responsible roles for process specific situations and state transitions are defined. It was also ensured that the records of the data planned to be analyzed during and after the process were kept. The defined processes are designed and handled as Corrective Preventive Actions (CPA) and Input Quality Control (IQC). A prototype study was made regarding the processes, the feedback received from the business and a solution design report for the process were created. Afterwards, technical controls were made and the design was started by sharing with the process managers and responsible.

In the analysis meetings held with the process managers in the enterprise, the data structures related to the process were transferred and the relevant

documents were shared. User integrations were integrated with Active Directory, and the organizational chart created within the enterprise was adhered to.

Keywords: Information Technologies, Corrective and Preventive Actions, Input Quality Control, Business, Personal Data Protection Law

AÇIK KAYNAK PLATFORM ÜZERİNDE İŞ SÜREÇLERİNİN TASARIMI VE YÖNETİMİ

İÇİNDEKİLER

TEZ ONAY SAYFASI ii
YEMİN METNİ iii
ÖZET iv
ABSTRACT vi
İÇİNDEKİLER viii
KISALTMALAR xi
ŞEKİLLER LİSTESİ xiii
TABLOLAR LİSTESİ xiv

GİRİŞ 1

BİRİNCİ BÖLÜM
İŞ SÜRECİ

1.1. İŞ SÜRECİ İLE İLGİLİ TANIMLAMALAR 5
1.2. İŞ SÜRECİ YAPILANDIRMALARI 6
1.3. TÜRKİYE'DE UYGULAMALAR 7
1.4. DÜNYADA UYGULAMALAR 7
1.5. İŞ SÜREÇLERİ YAZILIMLARI 8
1.5.1 Açık Kaynak Yazılımlar 8
1.5.2 Ücretli Yazılımlar 9

İKİNCİ BÖLÜM
PROJE YÖNETİMİ

2.1. PROJE PLANLAMA 11
2.2. PROJE TASARIMI 12

2.3. PROJE YÖNETİMİ YAZILIMLARI 13

ÜÇÜNCÜ BÖLÜM
SİSTEM GELİŞTİRME YAŞAM DÖNGÜSÜ

3.1. PLANLAMA 15
3.2. ANALİZ 16
3.3. TASARIM 17
3.4. UYGULAMA 18
3.5. BAKIM VE DESTEK 19

DÖRDÜNCÜ BÖLÜM
GELİŞTİRİLEN AÇIK KAYNAK İŞ AKIŞLARI VE FORM TASARIMI YÖNETİM SİSTEMİ

4.1. VERİTABANI TASARIMI 22
4.2. FORM TASARIMI 23
4.2.1 İsimlendirme Yöntemi 23
4.2.2 Form Tasarımı 23
4.2.3. Form Kayıt Listesi 25
4.2.4. Sayısal Sahalar 26
4.2.5. Font Kullanımı 26
4.2.6. Renk Kullanımı 26
4.2.7. Stil Kullanımı 26
4.2.8. Client Side Kodlama 27
4.2.9. Server Side Kodlama 28
4.2.10. Özelliklerde Expression Kullanımı 28
4.2.11. Url Parametreleri 29
4.2.12. Dosya İlişkilendirme (Attachment) 29
4.2.13. Çoklu Seçimler 30
4.2.14. Süreç İle Bütünleşik Formlar 30
4.2.15. Kayıt Saklama Sonrası 30

4.2.16. Verı Doğrulama Ve Güvenlik 30
4.2.17. Özel Tema Kullanımı 31
4.2.18. Tasarım Doğrulama Sonuçları 31
4.2.19. Tasarım Doğrulama 31
4.2.20. Performans 31
4.2.21. Ek Form Kullanımı 32
4.2.21.1. Grid 32
4.2.21.2. Popup 34
4.3. TABLO VE INDEX TASARIMI 35
4.4. VIEW TASARIMI 36
4.5. RAPOR TASARIMI 37
4.5.1. RDL 37
4.5.2. XTRAREPORT (REPX) 38
4.6. WORKFLOW TASARIMI (WF) 39
4.7. PROJE SÜREÇLERİ 40
4.8. PROJE ZAMAN ÇİZELGESİ 41
4.9. KULLANICI VE ORGANİZASYON ENTEGRASYONU 42
4.10. ENTEGRASYON WEB SERVİS GEREKSİNİMLERİ 45
4.11. UYGULAMALAR 49
4.11.1 Düzeltici Önleyici Faaliyetler (DÖF) Uygulaması 49
4.11.2 Giriş Kalite Kontrol Uygulaması 55
4.12. RAPORLAMALAR 56

SONUÇ 57
KAYNAKÇA 60

KISALTMALAR

8D	8 Disiplin
B2B	Business to Business
B2C	Business to Consumer
BPM	Business Process Management
ÇKK	Çıkış Kalite Kontrol
DFS	Distributed File System
DÖF	Düzeltici Önleyici Faaliyet
FMEA	Failure Mode Effect Analysis
FORM	Form Tasarım Ortamı
GİB	Gelir İdaresi Başkanlığı
HTML	HyperText Markup Language (Türkçe: Hiper Metin İşaretleme Dili)
IP	Internet Protocol
JSON	JavaScript Object Notation
LAN	Local Area Network (Yerel Alan Ağı)
LDAP	Lightweight Directory Access Protocol
DFSADMIN	DocHuman middleware tasarım ortamı
POKA YOKE	Hata Engelleme
PDKS	Personel Devam Kontrol Sistemi
PPM	Parts Per Million
RDL	ReportDefinition Langauge
REPX	Xtra Reports
SDLC	Systems Development Life Cycle (Türkçe: Sistem Geliştirme Yaşam Döngüsü)
SGÇ	Sağlık Güvenlik Çevre
UI	User Interface
XML	Extensible Markup Language (Türkçe: Genişletilebilir İşaretleme Dili)
YÖK	Yüksek Öğretim Kurumu
WEB	İnternet Ağı

WF	Workflow Tasarım Ortamı
WWW	World Wide Web, Dünya Çapında Ağ (kısaca WWW veya Web)

ŞEKİLLER LİSTESİ

Şekil 1: Sistem Geliştirme Yaşam Döngüsü s.14
Şekil 2: Dochuman Middleware Mimarisi s.21
Şekil 3: DFSUser Tablo Hiyerarşisi s.22
Şekil 4: Dochuman Trace s.32
Şekil 5: Dochuman Kullanıcı Tanımları Ekranı s.44
Şekil 6: UserData XML Dönüş Deseni Örneği s.46
Şekil 7: UserData JSON Dönüş Deseni Örneği s.47
Şekil 8: Manuel DÖF Formu s.50
Şekil 9: Dochuman Uygunsuzluk Bildirim Formu s.51
Şekil 10: DÖF Formu Akış Şeması s.52
Şekil 11: GKK Modülü Alt Form Listesi s.56
Şekil 12: Şartlı Kabul Talep Formu s.56
Şekil 13: DÖF Raporları 1 s.57
Şekil 14: DÖF Raporları 2 s.57
Şekil 15: GKK Raporları 1 s.58
Şekil 16: GKK Raporları 2 s.58

TABLOLAR LİSTESİ

Tablo 1: Ücretli Yazılımlar s.9
Tablo 2: Proje Zaman Çizelgesi s.41
Tablo 3: Veri Sözlüğü Kullanıcı Bilgileri s.43
Tablo 4: Veri Sözlüğü Birim Bilgileri s.43
Tablo 5: Veri Sözlüğü Kullanıcı Grubu s.44
Tablo 6: Veri Sözlüğü Kullanıcı Grup Üyeleri s.44

GİRİŞ

Giderek artan rekabete bağlı olarak gelişen profesyonelleşme anlayışı, işletmeleri yeni atılımlar yapmaya itmiş, bunun sonucu olarak ürün ve hizmetler daha süratli ve kaliteli sunulmaya başlanmıştır.

Dünya çapında yapılan inceleme ve kıyaslama çalışmaları açıkça göstermiştir ki, değişime ayak uydurabilen rekabetçi bir işletme olabilmek için ürün ve hizmetleri geliştiren ve sunan süreçlerin düzgün yönetilmeleri ve sürekli iyileştirilmeleri gerekmektedir. Çünkü süreç yönetimi ve iyileştirme, farklılaşma yaratarak rekabetçi olmayı, dolayısıyla karlılığı sağlamaktadır (Selimoğlu, 2005:1).

Teknolojinin hızlı bir şekilde gelişmesi ile birlikte işletmelerde ürün ve hizmetleri ortaya çıkaran çalışmalar müşteriye katma değer oluşturan işler olarak görüldüğü için, farklı bir ifade ile gerçekleşecek işi oluşturduklarından kullanılan verilerin önemi her geçen gün daha da büyük önem arz etmektedir. Faaliyetleri oluştururken kullanılan bilgilerin etkili ve doğru bir şekilde yönetilmeleri, programlanmaları ve bir plana dahil edilmeleri gerekmektedir.

Bir işletme veya faaliyet yürüten bir kişi için, faaliyetler sırasında kullanılan bilgilerin yönetimi oldukça önemlidir. Bu, işletmenin verimliliğini artırabilir, karar alma süreçlerini iyileştirebilir ve kaynakları daha etkili bir şekilde kullanılmasını sağlayabilir.

İş süreçlerinin oluşturulması aynı zamanda bilgi yönetiminin de bir parçasıdır. Aynı şekilde bilginin toplanması, depolanması, analizi, paylaşımı ve kullanımı ile ilgilidir. Bu süreç, doğru kaynaklardan doğru bilgilerin toplanmasını, verilerin doğru şekilde saklanmasını, analiz edilmesini, yorumlanmasını ve karar alma süreçlerinde kullanılmasını içerir.

İşletmelerde süreç yönetimi, iş süreçlerinin etkin ve verimli bir şekilde planlanması, uygulanması, kontrol edilmesi ve sürekli olarak iyileştirilmesi anlamına gelir. Süreç yönetimi, işletmelerin rekabet avantajı elde etmesine ve operasyonel mükemmelliği sağlamasına yardımcı olur.

İş süreçlerinin dijitalize edilmesi, bir veritabanında tutulması, bilgi güvenliği ve gizliliğinin korunmasını da içerir. İşletmeler, faaliyetleri sırasında topladıkları hassas bilgileri dağıtık yapılardan ziyade kendi oluşturacakları ve yönetecekleri bilgi

yönetimi kaynaklarında korumak için uygun zemini de hazırlamalıdır.

Bu tezde planlanan çalışmalar, bir işletmenin iş süreçlerini dijital ortama taşımak, bu ortamda takip etmek, bunları hızlı bir şekilde iş akışına sokmak ve raporlamalar hazırlamaktır. Bağlanan iş akışları, işletmenin organizayonel yapısına uygun olarak bir sürece girecek ve olumlu/olumsuz olarak yeni bir iş süreci başlatılmıştır.

Birinci bölümde iş süreçleri ile ilgili tanımlar, yapılandırmalar ve uygulamaların Türkiye ve dünya çapındaki örneklerine değinilmiştir. Süreç yönetim yazılımları açık ve kapalı kaynak kodlu olarak iki ayrı başlık altında detaylı bir şekilde incelenmiş ve örneklendirilmiştir.

İkinci bölümde proje yönetimi esas alınmış ve teorik olarak genel hatlarıyla bilgilendirmeler yapılmıştır. Proje planlama süreçleri, tasarım ve yönetim yazılımlarına değinilerek bilgilendirme yapılmıştır.

Üçüncü bölümde etkin bir sistem geliştirme yaşan döngüsü ele alınarak bilgiler verilmiş ve müşteri beklentilerini karşılayacak, zaman ve maliyet değerlemelerinde, mevcut ve planlanacak olan Bilgi Teknolojileri yapısında etkin ve verimli çalışan yüksek kaliteli bir sistemle sonuçlanması gerektiği anlatılmıştır.

Dördüncü ve son bölümde tasarlanacak sistemin veritabanı ve arayüz tasarımları hakkında bilgiler verilip kullanıcıları ve ilgili hakları oluşturularak uygulamadaki rolleri belirtişmiştir. Arayüzler üzerinden çekilen raporlamalardan örnekler gösterilip süreç yönetimi hakkında uyglama örneklerine değinilmiştir.

Sonuç kısmında çalışmanın detaylı çıktıları, ulaşılan noktalar ve geliştirmelere yol açmak amacıyla yapılacak çalışmalara teşkil olacak detaylara yer verilmiştir.

BİRİNCİ BÖLÜM
İŞ SÜRECİ

İş akış yönetimi, iş süreçlerini standart bir hale getirerek yeniden yapılandırmayı sağlayan ve yönetimi merkezileştiren yapılardır. İş akış planı oluşturularak iş süreçlerini uçtan uca kolayca takip edip yönetebilen yazılımlar ile akış hızlandırılabilir, yönetim süreci çok daha kaliteli bir hale getirebilir. Her çalışanın görev tanımlarının net bir şekilde belirlendiği bu sistemde işlerin tamamlanma süreci de daha kolay hale geliyor. Verimliliğin de artması nedeniyle iş süreçlerinin zaman ve maliyetten de tasarruf etmeyi sağlığı görülmektedir. Firmaya ait verimlilik ve kârlılığı artırmak dijital ortamda iş akış yazılımları ile süreçlerini en etkin şekilde yönetebilmek mümkün olmaktadır (Logo Yazılım, 2023).

İş süreci, bir kurumda çalışan insanların belirli bir sonuç elde etmek için iş gücü, para, malzeme, teknoloji, iletişim gibi kaynakların kullanılarak, sonuçta fayda, ürün, hizmet, bilgi ve karar gibi değerleri müşteriye sunmak için gerçekleştirilen işlemler topluluğudur (Logo Yazılım, 2023).

Süreçlerin oluşturulması, ekip oluşturma, bireysel sorumluluk bilinci ve değişikliği kabul ettirme becerisini gerektirir. Bu sebeple değinilen kurallara değer veren bir kurum kültürünün öncelikle oluşturulması gerekmektedir. İşletme olarak iş süreçlerini yönetebilmeleri her bakımdan özellikle de başarı ve verimlilik açısından çok önemlidir (Yıldız, 2021).

Süreçler birden fazla birimin kesiştiği kümelerdir. Müşterilerine değer katmak isteyen kurumlar bu kesişen kümelerin birleşiminden oluşmaktadır. Bu kapsamda şirket çalışanları kişisel uzmanlıklarından ziyade, bu oluşumun üyesi olduklarını fark ederlerse, birimler arası iş birliği artar ve böylelikle süreçlerin verimliliği de arttırılmış olur.

Organizasyonların başarılarını ve üretkenliklerini iyileştirmek için yapmaları gereken en önemli şey, iş süreçleri analizinin yapılarak, iyileştirme çalışmalarının yapılmasının sağlanmasıdır. Süreçlerin iyileştirilmesi ile birlikte, maliyet, zaman, kalite ve üretkenlik gibi unsurlar da optimize edilmiş olacaktır.

İş Süreçleri Yönetimi (İSY-BPM-Business Process Management), işletmelerin iş süreçlerini tanımlama, tasarlama, uygulama, izleme, analiz etme ve sürekli olarak

iyileştirme sürecidir. Bu süreçler, işletmelerin müşteri ihtiyaçlarını karşılamak, faaliyetleri optimize etmek ve maliyetleri azaltmak için kritik öneme sahiptir (Adesole ve Baine, 2005). Özellikle kamuda iş süreçlerinin modellenmesi gerektiği kamuda verimlilik ve etkinlik çerçevesinde zorunluluk olarak ele alınmaktadır Aysolmaz vd, 2011). İş süreçleri ile ilgili projelerin yürütülmesinde üst yönetimin yeterince destek verememesi ve süreci yürütenlerin bilgi eksikliği gibi bir çok konuda başarısızlığın seepleri olarak sayılabilir (Antony ve Gupta, 2018).

İSY, işletmelerin süreçlerini daha verimli hale getirmek ve kaynakları daha etkili bir şekilde kullanmak için tasarlanmış bir yönetim yaklaşımıdır. İş süreçleri, bir işletmenin faaliyetlerinin temel bileşenleridir ve işletmenin başarısı için kritik öneme sahiptir. BPM, iş süreçlerini standartlaştırmak, otomatikleştirmek ve optimize etmek için kullanılan araçlar ve teknikler içerir.

İSY' nin temel adımları aşağıda açıklanmaktadır:

- **Süreçlerin tanımlanması:** İşletme süreçleri, işletmenin faaliyetleri için kritik öneme sahip olan işlevleri tanımlamak için analiz edilir.
- **Süreçlerin tasarımı:** Süreçlerin daha verimli hale getirilmesi için tasarım aşaması gerçekleştirilir.
- **Süreçlerin uygulanması:** Tasarlanan süreçler, işletme süreçleri için gerekli kaynakları sağlayacak şekilde uygulanır.
- **Süreçlerin izlenmesi:** Süreçlerin performansı, işletmenin belirlenmiş hedefleri ile karşılaştırılarak izlenir.
- **Süreçlerin analizi:** Süreçlerin analizi, süreçlerin performansını iyileştirmek için önemli bilgileri sağlar.
- **Süreçlerin iyileştirilmesi:** Analiz sonuçlarına dayanarak, süreçlerin iyileştirilmesi ve daha verimli hale getirilmesi sağlanır.

İSY, işletmelerin daha verimli ve etkili hale gelmesine yardımcı olurken, müşteri memnuniyetini artırır, maliyetleri azaltır ve işletmenin sürekli olarak gelişmesine olanak tanır.

1.1. İŞ SÜRECİ İLE İLGİLİ TANIMLAMALAR

İş Süreci: İşletmenin belirli bir hedefi veya sonucu elde etmek için bir dizi adımdan oluşan faaliyetlerin organizasyonu ve yönetimidir. İş süreci, girdileri çıktılara dönüştürmek için kullanılan süreçler, prosedürler ve kaynaklarla ilişkilidir.

Süreç: İş sürecinin temel yapı taşlarından biridir ve belirli bir sonuç elde etmek için bir dizi adım veya aktiviteyi ifade eder. Süreçler, belirli bir sıra veya akış içinde gerçekleştirilir ve kaynakları kullanarak bir çıktı oluştururlar.

Girdi: Bir iş sürecinin başlatılmasını sağlayan veya bir adımın tamamlanması için gereken bilgi, malzeme veya kaynaklardır. Girdiler, sürecin sonucuna etki eden veriler, belgeler, malzemeler, beceriler vb. olabilir.

Çıktı: Bir iş sürecinin sonucunda üretilen veya sağlanan sonuçlardır. Çıktılar, süreç tarafından işlendikten veya dönüştürüldükten sonra ortaya çıkar ve belirli bir değer veya fayda sağlar.

Prosedür: Bir iş sürecinin nasıl gerçekleştirileceğini tanımlayan talimatlar, adımlar ve yönergelerdir. Prosedürler, sürecin uygun bir şekilde yürütülmesini ve istenen sonuçların elde edilmesini sağlamak için kullanılır.

Roller ve Sorumluluklar: İş süreçlerinde yer alan kişilerin belirli roller ve sorumlulukları vardır. Bu roller, sürecin etkin bir şekilde yürütülmesini sağlar ve iş birliğini kolaylaştırır.

İyileştirme: İş süreçlerinin verimliliğini, etkinliğini ve kalitesini artırmak için yapılan değişikliklerdir. İyileştirme, süreçlerin analiz edilmesi, zayıf noktaların belirlenmesi ve daha iyi sonuçlar elde etmek için optimize edilmesini içerir. Bu tanımlamalar, iş sürecinin temel kavramlarını ve bileşenlerini anlamak için kullanılabilir. İşletmelerde süreç yönetimi ve iş süreçleriyle ilgili diğer detaylar, işletme büyüklüğüne, sektöre ve işletmenin özel ihtiyaçlarına bağlı olarak değişiklik gösterebilir.

1.2. İŞ SÜRECİ YAPILANDIRMALARI

İş süreci yapılandırmaları, işletmelerin iş süreçlerini optimize etmek ve daha verimli hale getirmek için kullanılan farklı yöntem ve tekniklerdir. İş süreci yapılandırmaları aşağıdaki gibi olabilir:

İş Akışı Yapılandırması: İş sürecinin adımlarını, görevleri ve ilişkileri belirleyen bir yapılandırma yöntemidir. İş akışı yapılandırması, sürecin her bir adımını ve bu adımların sıralamasını tanımlar. Böylelikle, iş sürecinin etkin bir şekilde yönetilmesi ve iş birliğinin artırılması sağlanır.

Rol ve Sorumluluk Yapılandırması: İş sürecinde yer alan kişilerin rollerini, sorumluluklarını ve yetkilerini belirleyen bir yapılandırma yöntemidir. Her bir rolün görevleri, karar yetkisi ve iş birliği gereklilikleri net bir şekilde tanımlanır. Bu yapılandırma, iş sürecinin daha etkin ve verimli bir şekilde yönetilmesine katkı sağlar.

Teknoloji Yapılandırması: İş süreçlerinde kullanılan teknolojik araçlar, sistemler ve otomasyonlarla ilgili yapılandırmalardır. Teknoloji yapılandırması, iş sürecinin gereksinimlerine uygun teknolojik çözümlerin seçilmesini, uygulanmasını ve entegrasyonunu içerir. Bu yapılandırma, sürecin verimliliğini artırır ve hataları azaltır.

Veri Yapılandırması: İş süreçlerinde kullanılan verilerin toplanması, depolanması, analiz edilmesi ve paylaşılmasını düzenleyen bir yapılandırma yöntemidir. Veri yapısı, veri akışının sağlanması, doğruluğunun korunması ve veri kullanımının optimize edilmesi amacıyla tanımlanır.

İyileştirme Yapılandırması: İş süreçlerinin sürekli olarak gözden geçirilmesi, analiz edilmesi ve iyileştirilmesi için kullanılan bir yapılandırma yöntemidir. İyileştirme yapısı, süreç performansını izlemeyi, verimlilik sorunlarını belirlemeyi ve sürekli iyileştirme fırsatlarını tanımlamayı içerir.

Bu yapılandırmalar, iş süreçlerini daha iyi yönetmek, verimliliği artırmak ve sonuçlarını optimize etmek için kullanılan farklı yaklaşımlardır. İşletmeler, özel ihtiyaçlarına, sektöre ve iş süreçlerinin karmaşıklığına bağlı olarak bu yapılandırmalardan bir veya birkaçını kullanabilir.

1.3. TÜRKİYE'DE UYGULAMALAR

İş Süreçleri Yönetimi (İSY) işletmelerdeki süreçlerin tıkanıklığını ve yavaşlığını ortadan kaldırarak, genel iş verimliliğinde önemli bir artış sağlamaktadır. İSY üzerinde gerçekleştirilen süreç odaklı çözümlerle yöneticiler ve çalışanların birbiriyle ve verilerle uyumlu ve hızlı bir şekilde çalışabilmesine olanak tanınmaktadır.

Böylelikle:

- Dijital olmayan ortamda elle yürütülen tüm işlemler ve süreçler ortadan kalkmaktadır.
- İşlerin takibi kolaylaşmakta ve izlenebilirliği artmaktadır
- Hatalar azalmakta ve işlerin terminleri daha kısa sürede olmaktadır.
- İş süreçlerindeki sıkıntılar büyüyemeden çözülebilmektedir.

Ülkemizde kullanılan iş süreçleri çözümleri birçok farklı platformlardan oluşmaktadır. Microsoft tarafında bulut tabanlı geliştirilen Power Apps uygulamaları, yine aynı şekilde Power Automate ile birleştirilerek kurumdaki süreçleri bir akışa bağlamaya olanak sağlamaktadır. Paket program tarafında yüzlerce uygulamanın sunulduğu bu sektörde öncelikli olarak başı Trello, Microsoft Planner, Paperwork, LOGO, Koçsistem BPM gibi markalar çekmektedir.

1.4. DÜNYADA UYGULAMALAR

Dünya genelinde kullanılan iş süreçleri yazılımları ve şirketler hakkında genel bir liste aşağıdaki gibidir:

- SAP - SAP, işletmeler için ERP (Kurumsal Kaynak Planlama), CRM (Müşteri İlişkileri Yönetimi), SCM (Tedarik Zinciri Yönetimi) ve diğer iş süreçleri yazılımları sunmaktadır.
- Oracle - Oracle, bir dizi iş süreçleri yazılımı sunmaktadır, bunlar arasında ERP, SCM, CRM ve insan kaynakları yönetimi (HRM) yazılımları yer almaktadır.

- Microsoft Dynamics - Microsoft Dynamics, işletmeler için ERP ve CRM yazılımları sunmaktadır. Ayrıca Dynamics 365 olarak adlandırılan bir bulut tabanlı işletme uygulama paketi sunmaktadır.
- IBM - IBM, işletmeler için iş süreçleri otomasyonu (BPA), iş akışı yönetimi (WFM) ve iş süreçleri yönetimi (BPM) yazılımları sunmaktadır.
- Salesforce - Salesforce, müşteri ilişkileri yönetimi (CRM) ve işletmeler için diğer yazılım çözümleri sunmaktadır.
- ServiceNow - ServiceNow, işletmeler için IT hizmet yönetimi (ITSM), iş süreçleri yönetimi (BPM) ve müşteri hizmetleri yazılımları sunmaktadır.
- Pega - Pega, işletmeler için iş süreçleri otomasyonu (BPA) ve iş süreçleri yönetimi (BPM) yazılımları sunmaktadır.
- Appian - Appian, işletmeler için iş süreçleri otomasyonu (BPA) ve iş süreçleri yönetimi (BPM) yazılımları sunmaktadır.
- Nintex - Nintex, işletmeler için iş süreçleri otomasyonu (BPA) ve iş süreçleri yönetimi (BPM) yazılımları sunmaktadır.
- Camunda - Camunda, işletmeler için açık kaynaklı iş süreçleri otomasyonu (BPA) ve iş süreçleri yönetimi (BPM) yazılımları sunmaktadır.

Bu liste yalnızca bazı örnekler içermektedir. İş süreçleri yazılımları piyasası oldukça geniştir ve her yıl yeni şirketler ve yazılımlar piyasaya sürülmektedir.

1.5. İŞ SÜREÇLERİ YAZILIMLARI

1.5.1 Açık Kaynak Yazılımlar

Türkiye'de sektörlerin bir kısmı iş süreçleri yönetimlerini Bilgi Teknolojileri Departmanlarının yazılım ekiplerini zenginleştirerek uygulamaya aldıkları süreç yazılımları ile tasarlarken diğer kuruluşlar ücretli yada açık kaynak yazılımlar ile süreçlerini yönetmeye çalışmaktadırlar.

Açık kaynak tarafında uygulama geliştirilirken alınabilecek önlem ve aksiyonlar baştan belirlenmeli ve uzman yazılım ekibinin yanında sistem mühendisi, ağ uzmanları gibi Bilgi Teknolojilerinin diğer kollarından da destek almak gerekmektedir. Çünkü açık kaynak yazılımlar, kurumlara yazılımı geliştirme ve şirkete

göre kişiselleştirilmiş bir yazılım olanağı sunabilmenin yanında, kapalı/kriptolanmış kodlara göre olası hatalar için de uygulayıcıya açık kapı bırakabilmektedir.

1.5.2 Ücretli Yazılımlar

Kurumalarda iş süreçlerini takip edip doğru yönetebilmek için yurt içinde ve yurt dışında yeterli sayıda yazılımlar mevcuttur. Yurt içinde ve yurt dışında bulunan ve ücretli olan iş akış uygulamalarından bazıları aşağıda listelenmiştir.

Tablo 1: Ücretli Yazılımlar

	Dağıtım	**Platform**	**Fiyat**
Kissflow	Bulutta barındırılan ve Açık API	Tüm Platformlar	**Küçük işletme:** 15$ / ay / kullanıcı **Büyüyen İşletme:** 20$ / ay / kullanıcı
Monday	Bulut tabanlı ve Açık API	Tüm Platformlar	**Standart:** 10€ / kullanıcı / ay **Profesyonel:** 16€ / kullanıcı / ay
BPMsense	Bulutta barındırılan	Tüm Platformlar	Yıllık 950€ (kullanıcı başına aylık 79€).
Process Street	Bulutta barındırılan	Tüm Platformlar	**Profesyonel** (5 Kullanıcı aylık 100$)
Wrike	Bulut Tarafından Barındırılan ve Açık API.	Tüm Platformlar	**Bedava:** 5 kullanıcı için **Profesyonel:** 20$ / kullanıcı / ay **İş:** 25$ / kullanıcı / ay
Kovan	Bulut tabanlı ve Şirket içi	Tüm Platformlar	**Standart Lisanslama: 90$ / kullanıcı / ay.**
Appian	Bulutta barındırılan	Tüm Platformlar	**Standart Lisanslama:** 90$ / kullanıcı / ay. **Uygulama Lisanslama:** Teklif alın
PNMSoft	Bulutta barındırılan ve Şirket içi	Tüm Platformlar	
Pipefy	Bulutta barındırılan ve Açık API.	Tüm Platformlar	**Küçük Takımlar:** Bedava **Profesyonel:** 9$ doları / kullanıcı / ay **İş:** 18$ / kullanıcı / ay **Kurumsal:** Bir teklif alın.
Bizagi	Bulutta barındırılan ve Şirket içi	Tüm Platformlar	**Bedava:** Kişisel kullanım **Profesyonel:** Tek bir kullanıcı için ayda 8$ **Çalışma grubu:** Takımlar için 16 $ / ay

Kaynak: Yazar tarafından derlenmiştir.

İKİNCİ BÖLÜM

PROJE YÖNETİMİ

Proje yönetimi, belirli bir amacı yerine getirmek için kaynakların (insanlar, para, ekipman vb.) planlanması, organize edilmesi, yönetilmesi ve kontrol edilmesi sürecidir. Bu süreç, belirli bir hedefe ulaşmak için zaman, bütçe, kalite, kapsam, risk ve kaynak yönetimi gibi çeşitli faktörleri yönetir.

Proje yönetimi, herhangi bir endüstride, herhangi bir projede kullanılabilir. İster yazılım geliştirme, inşaat, yeni ürün geliştirme veya pazarlama stratejisi olsun, proje yönetimi yöntemleri geniş bir yelpazede uygulanabilir.

Proje yönetimi, çeşitli adımları içerir:

Planlama: Projenin hedefleri ve hedefleri belirlenir. Bu aşamada, zaman çizelgeleri, bütçeler, kaynak planları ve risk yönetimi planları hazırlanır.

Kaynaklar: Projenin başarısı için gerekli olan kaynaklar (insanlar, malzemeler, bütçe vb.) tahsis edilir ve yönetilir.

Uygulama: Projenin gerçekleştirilmesi için çalışmalar başlatılır. Bu aşamada, takım üyeleri arasında işbirliği yapılır ve her adımın doğru şekilde uygulandığından emin olmak için düzenli denetimler yapılır.

İzleme: Proje süreci, ilerleme raporları, bütçe raporları ve risk yönetimi gibi farklı unsurların izlenmesi ile takip edilir. Bu aşamada, proje planının doğru şekilde uygulandığından emin olmak için sıkı takip yapılır.

Kontrol: İzleme sonuçlarına dayanarak, proje planında gerekli değişiklikler yapılır ve süreç yeniden başlatılır.

Kapanış: Proje başarıyla tamamlandıktan sonra, sonuçlar değerlendirilir, tüm belgeler arşivlenir ve tüm tarafların memnuniyeti sağlanır.

Proje yönetimi, belirli bir amaca ulaşmak için karmaşık süreçleri yönetmek için kullanılan bir araçtır. İyi bir proje yönetimi, projelerin zamanında, bütçede ve belirlenen hedeflerde tamamlanmasını sağlar.

2.1. PROJE PLANLAMA

Proje planlama, bir projenin başarılı bir şekilde tamamlanması için kullanılan süreçtir. Bu süreçte, projenin amacı ve hedefleri belirlenir, kaynaklar tahsis edilir ve proje faaliyetleri planlanır.

Proje planlama, birçok adımdan oluşur:

Projenin Amacı ve Hedefleri: Projeyi başlatmadan önce, projenin amacı ve hedefleri netleştirilmelidir. Bu adım, projenin neden yapılması gerektiğini ve hangi sonuçların elde edilmesinin hedeflendiğini açıklar.

Kaynakların Belirlenmesi: Projenin başarıyla tamamlanması için gerekli olan kaynaklar belirlenir ve tahsis edilir. Bu kaynaklar arasında insanlar, bütçe, ekipman ve diğer materyaller bulunur.

Faaliyetlerin Tanımlanması: Projenin gerçekleştirilmesi için gerekli olan tüm faaliyetler belirlenir ve sıralanır. Bu aşamada, her bir faaliyetin ne zaman başlayacağı, ne kadar süreceği ve hangi kaynaklarla gerçekleştirileceği belirlenir.

Zaman Çizelgesinin Hazırlanması: Projenin tamamlanması için gerekli olan tüm faaliyetlerin sırası belirlenir ve bir zaman çizelgesi oluşturulur. Bu çizelge, her bir faaliyetin ne zaman başlayacağı, ne kadar süreceği ve ne zaman tamamlanacağı gibi detayları içerir.

Risklerin Tanımlanması: Projeye ilişkin olası riskler belirlenir ve bir risk yönetim planı hazırlanır. Bu plan, olası riskleri önlemek, risklerle başa çıkmak için kullanılacak stratejileri ve proje faaliyetlerinin devamını sağlamak için alınacak tedbirleri içerir.

İletişim Planının Hazırlanması: Proje ekibi ve paydaşlar arasındaki iletişim düzenlenir ve bir iletişim planı hazırlanır. Bu plan, tüm paydaşların bilgilendirilmesini, proje ekibi arasındaki işbirliğini ve projenin ilerlemesi hakkında düzenli raporlamayı içerir.

Proje planlama, projenin başarılı bir şekilde tamamlanması için çok önemlidir. İyi bir proje planlaması, projenin zamanında, bütçede ve hedeflerde tamamlanmasını sağlar ve risklerin yönetimine yardımcı olur. Proje planlaması, projenin her aşamasında kullanılan bir araçtır ve proje yönetimi sürecinin bir parçasıdır.

2.2. PROJE TASARIMI

Proje tasarımı, bir projenin belirli hedeflere ulaşmak için nasıl tasarlandığını ve organize edildiğini ifade eder. Proje tasarımı, proje ekibinin, proje hedefleri, işlevleri ve projenin geliştirilmesi için gerekli olan kaynakların belirlenmesi için kapsamlı bir strateji geliştirdiği bir süreçtir.

Proje tasarımı aşamaları aşağıdaki gibidir:

Hedeflerin Belirlenmesi: Projenin hedefleri belirlenmeli ve her hedefin öncelikleri belirlenmelidir. Hedefler, projenin amacını belirleyen anahtar bileşenlerdir.

Analiz Etme: Projenin, mevcut kaynaklar ve işlevler gibi çeşitli faktörler dikkate alınarak analiz edilmesi gerekmektedir. Bu adım, projenin uygun bir şekilde tasarlanmasına yardımcı olur.

Kaynakların Belirlenmesi: Proje tasarımı, ihtiyaç duyulan tüm kaynakların belirlenmesi ile başlar. Bu kaynaklar arasında insan kaynakları, bütçe, tesisler ve teknoloji bulunur.

Planlama: Proje tasarımının bu aşamasında, projenin adımları belirlenir ve kaynaklar, zamanlamalar ve faaliyetler için ayrıntılı bir plan hazırlanır. Planlama aşamasında, projenin hedefleri, hedeflerin öncelikleri ve diğer işlevler dahil olmak üzere projenin tüm yönleri dikkate alınır.

Yönetim: Projenin yönetimi için bir ekip kurulmalıdır. Bu ekip, projenin zamanında tamamlanmasını sağlamak için işlevleri yönetir ve proje ekibinin diğer üyeleriyle birlikte çalışarak projenin başarılı bir şekilde tamamlanmasını sağlar.

Risk Yönetimi: Proje tasarımının bir parçası olarak, proje riskleri belirlenmeli ve yönetilmelidir. Risk yönetimi, olası risklerin önceden belirlenmesini, risklerin önlenmesini ve yönetilmesini içerir.

Proje İzleme ve Değerlendirme: Proje tasarımının tamamlanmasından sonra, proje ekibi projenin performansını izler ve değerlendirir. Bu, projenin hedeflerine ulaşıp ulaşmadığını ve hangi faaliyetlerin iyi gittiğini ve hangi faaliyetlerin iyileştirilmesi gerektiğini gösterir.

Proje tasarımı, proje yönetimi sürecinin önemli bir aşamasıdır. İyi bir proje tasarımı, projenin başarılı bir şekilde tamamlanmasını sağlayan hedefleri ve hedefleri doğru bir şekilde belirler.

2.3. PROJE YÖNETİMİ YAZILIMLARI

Proje yönetim yazılımları, proje yönetimi sürecindeki faaliyetlerin planlanmasını, takibini, yönetimini ve raporlamasını sağlayan araçlardır. Bu yazılımlar, projenin tüm aşamalarını kapsayacak şekilde tasarlanmış olup, proje yöneticilerinin, takım liderlerinin ve proje çalışanlarının iş birliğini kolaylaştırmayı amaçlamaktadır.

Proje yönetim yazılımlarının kullanımı, proje sürecini daha verimli ve etkili bir şekilde yönetmeyi sağlar. Bu yazılımlar, proje yöneticilerine ve takım liderlerine, proje ilerlemesi hakkında anlık bilgi sağlar ve takım üyelerinin görevlerini izlemelerine yardımcı olur. Ayrıca, proje yönetim yazılımları, projenin bütçesini ve zaman çizelgesini izlemeyi, değişiklikleri yönetmeyi ve iletişim kurmayı da kolaylaştırır.

Proje yönetim yazılımları, farklı özelliklere sahip olabilir ve çeşitli işletmelerin ihtiyaçlarına göre özelleştirilebilir. Bazı yaygın proje yönetim yazılımları şunlardır:

Microsoft Project: Proje yönetiminde yaygın olarak kullanılan bir yazılımdır. Bu yazılım, proje planlama, kaynak yönetimi, zaman çizelgesi oluşturma ve raporlama gibi birçok özellik sunar.

Asana: Takım iş birliği ve proje yönetimi için bir platformdur. Bu yazılım, görevlerin atanması, zaman çizelgelerinin oluşturulması, dosya paylaşımı ve takım üyeleri arasındaki iletişim için araçlar sunar.

Trello: Takım iş birliği ve proje yönetimi için kullanılan bir görsel organizasyon aracıdır. Bu yazılım, proje tahtaları, liste ve kartlar şeklinde organize edilmiş bir yapısı vardır.

Jira: Özellikle yazılım geliştirme projeleri için kullanılan bir proje yönetim yazılımıdır. Bu yazılım, hata izleme, değişiklik yönetimi ve zaman çizelgesi oluşturma gibi özellikler sunar.

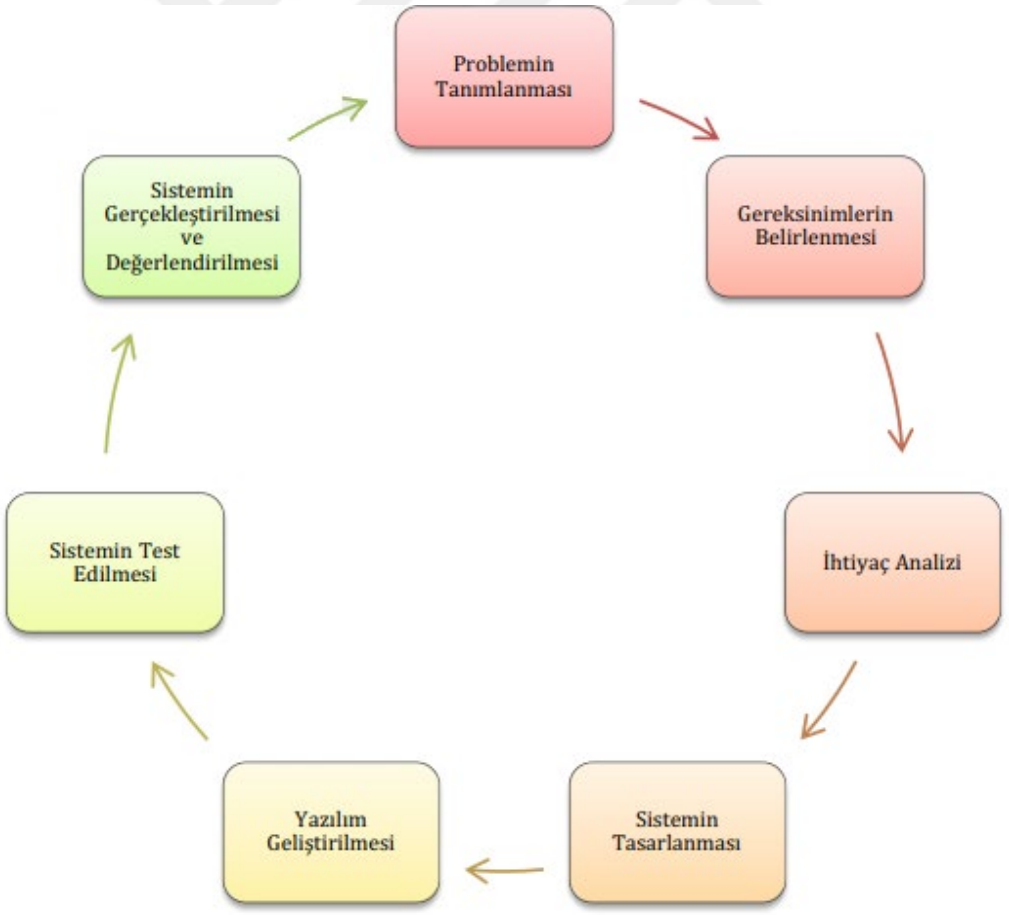
Basecamp: Ekip çalışmaları ve proje yönetimi için bir platformdur. Bu yazılım, görevlerin atanması, zaman çizelgesi oluşturma, dosya paylaşımı ve takım üyeleri arasındaki iletişim için araçlar sunar.

Bu yazılımlar, proje yöneticilerine ve takım liderlerine, proje sürecini daha verimli ve etkili bir şekilde yönetme imkânı sağlar.

ÜÇÜNCÜ BÖLÜM
SİSTEM GELİŞTİRME YAŞAM DÖNGÜSÜ

Sistem Geliştirme Yaşam Döngüsü (SGYD-SDLC: System Development Life Cycle), bir yazılım veya sistem projesinin başlangıcından sonuna kadar olan tüm aşamaların kapsamlı bir şekilde planlandığı ve uygulandığı bir yöntemdir. Bir sistemin geliştirilmesini, uygulanmasını ve yönetimini ele almaktadır. Genelde sistemler, yazılım, donanım, veri ve iş süreçlerini içermektedirler, bu nedenledir ki bir sistem geliştirme projesinin tüm aşamalarını içeren bir yöntemdir. "SGYD, sistem geliştirme sürecinin her bir aşamasını ayrıntılı olarak planlamak, uygulamak ve kontrol etmek için bir çerçeve sağlar" (Tecim, 2023). Bu dgü bütün bir sistemin veya iş süreçlerinin yönetilmesini amaçlamaktadır.

Şekil 1: Sistem Geliştirme Yaşam Döngüsü



Kaynak: Gökçen, 2005 ve Tecim, 2023'den uyarlanmıştır.

Bu aşamalar aşağıda açıklanmaktadır:

Planlama: Projenin gereksinimlerinin belirlenmesi için yapılan bir aşamadır. Bu aşamada, sistem kullanıcıları ve paydaşları ile görüşmeler yapılır ve kullanıcıların ihtiyaçları belirlenir.

Analiz: Bu aşamada, ihtiyaçların temelinde, sistemin mimarisi tasarlanır. Tasarım aşamasında, işlevsel ve teknik özellikler belirlenir, proje takvimi hazırlanır ve projenin gereksinimlerini karşılayacak bir sistem tasarlanır.

Tasarım: Bu aşamada, sistemin yazılım kodları yazılır ve test edilir. Programlama ve kodlama, tasarım belgeleri temel alınarak yapılır. Bu aşamada, projenin çeşitli testlerden geçmesi gerekmektedir.

Uygulama: Bu aşamada, tasarımı tamamlanmış sistemin hayata geçirilmesi çalışmaları yapılır. Kodlama, testler, genel olarak sistemin kurulumu ve tüm kullanıcıların uygulamadaki işlemleri yapmalarını sağlayan tüm diğer işlemleri içerir.

Bakım ve Destek: Sistem, kullanımda olduğu sürece bakım gerektirir. Bu aşamada, sistem üzerindeki sorunlar ve hatalar tespit edilir ve düzeltilir. Ayrıca, sistem özelliklerinin ve işlevlerinin güncellenmesi de gerekebilir.

SGYD, her aşamasının tamamlanması gereken bir döngüdür. Her aşama, sonraki aşamalar için bir temel sağlar ve projenin başarılı bir şekilde tamamlanmasını sağlayan önemli bir planlama, yönetim ve kontrol mekanizmasıdır.

3.1. PLANLAMA

Sistem Geliştirme Yaşam Döngüsü (SGYD) kapsamında, planlama adımı proje yönetimi sürecinin bir parçasıdır. Bu adım, projenin doğru bir şekilde planlanması, organizasyonu, yönetimi ve kontrol edilmesini sağlar. SGYD' nin planlama aşamasında, aşağıdaki aktiviteler gerçekleştirilir:

Proje gereksinimlerinin belirlenmesi: Projede gerçekleştirilecek işlevlerin ve gereksinimlerin belirlenmesi, projenin başarılı bir şekilde tamamlanabilmesi için önemlidir. Bu aşamada, projenin hedefleri, amaçları ve kapsamı tanımlanır. Gereksinimlerin belirlenmesi, proje planının oluşturulmasına ve projenin başlangıcına hazırlık aşamasıdır.

Proje planının hazırlanması: Proje yönetimi planı, projenin başarılı bir şekilde tamamlanması için gerekli olan tüm aşamaların ve süreçlerin belirlenmesiyle oluşturulur. Planlama aşamasında, proje takvimi, bütçe, proje yönetimi ekibi, kaynaklar ve risklerin yönetimi planı hazırlanır. Planlama aşamasında ayrıca, projenin sahibi, sponsorları, paydaşları ve diğer ilgili tarafların rolleri ve sorumlulukları da tanımlanır.

Proje yönetimi ekibinin oluşturulması: Proje yönetimi ekibi, proje planının uygulanması, izlenmesi ve kontrol edilmesinden sorumludur. Bu ekip, proje hedeflerini ve gereksinimlerini belirler, projeyi planlar, kaynakları tahsis eder, çalışma gruplarını yönetir ve proje takibini yapar.

Projenin yönetimi ve kontrolü: Projenin yönetimi ve kontrolü, proje yönetimi ekibi tarafından gerçekleştirilir. Bu aşamada, proje hedeflerinin, takviminin, bütçesinin ve performansının izlenmesi ve değerlendirilmesi sağlanır. Değişikliklerin yönetimi, risklerin yönetimi ve iletişim yönetimi de planlama aşamasında ele alınır.

Planlama aşaması, proje yönetimi sürecinde önemli bir adımdır. Bu adımın doğru bir şekilde tamamlanması, projenin başarılı bir şekilde tamamlanabilmesi için önemlidir. Planlama aşamasında, proje yönetimi ekibi, projenin tüm yönlerinin doğru bir şekilde planlanmasını ve projenin gereksinimlerine uygun olarak yönetilmesini sağlar.

3.2. ANALİZ

SGYD kapsamında, analiz adımı, sistem gereksinimlerinin belirlenmesi ve tasarımına yönelik bir adımdır. Analiz aşaması, önceden belirlenen hedeflere ulaşmak için gerekli olan fonksiyonel ve teknik gereksinimleri belirlemeyi amaçlar. Bu aşama genellikle aşağıdaki aktiviteleri içerir:

İhtiyaçların Belirlenmesi: Analiz aşaması, sistem gereksinimlerinin belirlenmesiyle başlar. Sistemde gerçekleştirilecek olan işlevler, kullanıcıların ihtiyaçları ve iş gereksinimleri analiz edilir. İhtiyaçların doğru bir şekilde belirlenmesi, proje başarısı için kritik bir öneme sahiptir.

Gereksinimlerin Doğrulanması: İhtiyaçların belirlenmesinin ardından, bu ihtiyaçların doğrulanması için bir dizi adım atılır. Bu adımlar, kullanıcılar, yöneticiler

ve diğer ilgili taraflar tarafından incelenen gereksinim belgelerinin hazırlanması, veri analizi, iş akışı analizi, prototip geliştirme, vb. gibi adımları içerir.

Tasarımın Hazırlanması: Analiz aşaması, gereksinimlerin doğru bir şekilde belirlenmesinin ardından, sistemin tasarımının hazırlanmasını içerir. Tasarım aşamasında, işlevsel gereksinimlerin nasıl karşılanacağına, teknik özelliklere ve sistem mimarisine karar verilir.

Prototip Geliştirme: Prototip geliştirme, analiz aşamasında bir dizi adımdan biridir. Prototip, sistemin işleyişini ve tasarımını görsel olarak gösteren bir modeldir. Prototip, müşterilerin ve diğer ilgili tarafların sistem hakkında daha iyi bir fikir sahibi olmasına yardımcı olur ve tasarımın doğruluğunu ve uygunluğunu doğrular.

Analiz Raporu Hazırlama: Analiz aşaması tamamlandıktan sonra, analiz raporu hazırlanır. Bu rapor, sistemin gereksinimleri, tasarımı ve analizi hakkında ayrıntılı bir açıklama içerir. Rapor, sistem geliştirme sürecinin bir sonraki aşamasına hazırlık yapmak için önemlidir.

Analiz aşaması, sistemin başarılı bir şekilde geliştirilmesi için kritik öneme sahip bir adımdır. Bu aşamada, müşterilerin ihtiyaçları doğru bir şekilde anlaşılır, gereksinimler doğrulanır, sistem tasarımı hazırlanır ve prototip geliştirilir.

3.3. TASARIM

SGYD kapsamında, tasarım adımı, sistem gereksinimlerine dayalı olarak detaylı bir tasarım belgesi hazırlamayı amaçlar. Bu belge, sistem mimarisini, işlevselliği, veri yapılarını, algoritmaları, ara yüzleri ve diğer ilgili özellikleri tanımlar. Tasarım adımı, genellikle aşağıdaki aktiviteleri içerir:

Mimari Tasarım: Bu adımda, sistem mimarisi belirlenir. Mimari tasarım, sistemin bileşenlerinin belirlenmesi ve bunların nasıl bir araya geleceği hakkında kararlar alınmasını içerir. Mimari tasarım, sistem performansı, güvenilirliği, ölçeklenebilirliği ve uygunluğu için kritik öneme sahiptir.

İşlevsel Tasarım: İşlevsel tasarım, sistemin işlevselliğinin detaylı bir şekilde tanımlanmasını içerir. Bu adımda, sistemin ne yapacağı, kullanıcı etkileşimi, işlem süreci, veri girişi, veri çıkışı ve diğer işlevler tanımlanır. İşlevsel tasarım, müşterilerin

ve geliştiricilerin sistemi anlamalarına ve gereksinimleri karşıladığından emin olmalarına yardımcı olur.

Veri Tasarımı: Veri tasarımı, sistemin veri tabanı yapısını belirleyen bir adımdır. Bu adımda, veri tabanı şeması, tabloların yapısı, alanlar, anahtarlar ve diğer ilgili özellikler belirlenir. Veri tasarımı, veri bütünlüğü, güvenilirliği ve uygunluğu için kritik öneme sahiptir.

Arayüz Tasarımı: Kullanıcıların sisteme erişmek ve işlevleri kullanmak için kullanacağı arayüzleri tanımlar. Bu adımda, kullanıcı arayüzleri, menüler, formlar, düğmeler, grafikler ve diğer ilgili öğeler tasarlanır. Arayüz tasarımı, kullanıcı deneyimini ve memnuniyetini artırmak için kritik öneme sahiptir.

Algoritma Tasarımı: Sistem işlevleri için gerekli olan algoritmaların belirlenmesini içerir. Bu adımda, hesaplama, sıralama, arama ve diğer işlevler için uygun algoritmalar belirlenir. Algoritma tasarımı, sistemin performansını ve doğruluğunu etkiler.

Tasarım adımı, SGYD' nin en kritik adımlarından biridir. Bu adımda, sistem mimarisi, işlevselliği ve son kullanıcı ara yüzü belirlenir.

3.4. UYGULAMA

SGYD kapsamında, uygulama adımı, tasarım belgesine dayalı olarak gerçek bir sistem oluşturma aşamasıdır. Bu adım, programlama, test etme ve sistemi gerçek ortama entegre etme süreçlerini içerir. Uygulama adımı, genellikle aşağıdaki aktiviteleri içerir:

Programlama: Bu adımda, tasarım belgesinde belirtilen gereksinimlere uygun olarak yazılım kodları yazılır. Programlama süreci, geliştiricilerin programlama dillerini kullanarak sistem gereksinimlerini karşılayan kodları oluşturmalarını içerir. Programlama aşamasında, kod kalitesi, okunabilirliği ve sürdürülebilirliği önemlidir.

Birim Testleri: Programlama aşamasından sonra, yazılımın doğru çalıştığından emin olmak için birim testleri yapılır. Birim testleri, her bir bileşenin doğru çalıştığından emin olmak için yazılımın her bir birimini test etmeyi içerir. Birim testleri, yazılımın kalitesini ve uygunluğunu artırmaya yardımcı olur.

Entegrasyon Testleri: Entegrasyon testleri, yazılım bileşenlerinin doğru bir şekilde birleştirilip bir arada çalıştığından emin olmak için yapılır. Entegrasyon testleri, yazılım bileşenlerinin uygun bir şekilde bir araya getirildiğinden ve gereksinimleri karşıladığından emin olmak için yapılır.

Sistem Testleri: Sistem testleri, yazılımın tamamının doğru çalıştığından emin olmak için yapılır. Bu adımda, yazılımın tüm işlevleri, performansı, kullanılabilirliği ve güvenilirliği test edilir. Sistem testleri, yazılımın müşteri gereksinimlerini karşıladığından emin olmak için önemlidir.

Kabul Testleri: Kabul testleri, yazılımın müşteri tarafından kabul edilebilir olduğundan emin olmak için yapılır. Kabul testleri, müşterinin yazılımın işlevselliğini, performansını ve diğer özelliklerini test etmesine ve onaylamasına olanak tanır.

Bu adımda, yazılımın gerçek bir sistem olarak hayata geçirilmesi sağlanır. Uygulama adımında yapılan testler, yazılımın kalitesini ve uygunluğunu artırır ve müşteri gereksinimlerinin karşılandığından emin olunur.

3.5. BAKIM VE DESTEK

SGYD' nin son adımı bakım ve sürdürülebilirliktir. Bu adım, yazılımın işlevselliğini korumak ve iyileştirmek için gerekli düzenlemeleri yapmayı ve yazılımı gelecekteki gereksinimlere uygun hale getirmeyi amaçlar. Bakım ve sürdürülebilirlik adımı, aşağıdaki aktiviteleri içerebilir:

Bakım: Bakım, yazılımın var olan işlevselliğini ve performansını korumayı ve iyileştirmeyi içerir. Yazılımın bakımı, hataların düzeltilmesini, performans sorunlarının çözülmesini, güvenlik güncelleştirmelerinin yapılmasını ve kullanıcı geri bildirimlerinin dikkate alınmasını içerir. Bakım işlemleri, yazılımın güncel kalmasını ve müşteri gereksinimlerine uygun kalmasını sağlar.

Sürdürülebilirlik: Sürdürülebilirlik, yazılımın gelecekteki gereksinimlere uygun hale getirilmesini sağlamayı amaçlar. Yazılımın sürdürülebilirliği, kodun okunabilirliğini, yeniden kullanılabilirliğini, değiştirilebilirliğini ve genişletilebilirliğini artırmayı içerir. Yazılımın sürdürülebilirliği, geliştiricilerin gelecekteki değişiklikleri kolayca yapmalarına olanak tanır ve yazılımın ömrünü uzatır.

Dokümantasyon: Dokümantasyon, yazılımın bakım ve sürdürülebilirliği için önemlidir. Dokümantasyon, yazılımın işlevselliği, tasarımı, kodu ve kurulum talimatları hakkında bilgi sağlar. Dokümantasyon, yazılımı anlamak ve gelecekteki değişiklikleri yapmak için gereken bilgiyi sağlar.

Bakım ve sürdürülebilirlik adımı, yazılımın hayat döngüsünün son adımıdır ve yazılımın ömrünü uzatmak ve uygunluğunu sağlamak için çok önemlidir. Bakım ve sürdürülebilirlik adımı, yazılımın güvenliğini, performansını ve işlevselliğini korumak için gereklidir ve yazılımın müşteri gereksinimlerine uygun olmasını sağlar.

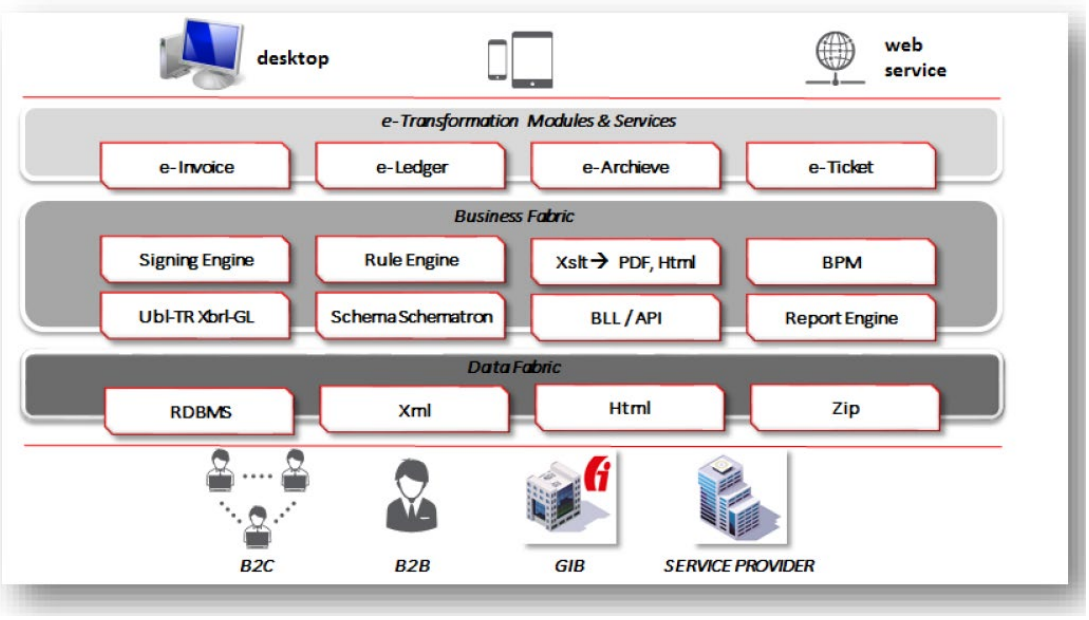
Bu çalışmada bir sanayi sektöründe süreç iyileştirme konusunda yazılım adaptasyonu üzerine uygulama geliştirilecektir. Buna benzer çalışmalar dondurma fabrikasında (Berber, 2017), piston üretiminde (Çevik ve Aran, 2014), gida işletmesinde (Kayışkan, 2016), sağlık kuruluşunda (Kılıç ve Aydınlı, 2015), mobilya sektöründe (Songül ve Oralhan, 2016) ve çelik üretiminde (Türkan ve Görener, 2017)

DÖRDÜNCÜ BÖLÜM
GELİŞTİRİLEN AÇIK KAYNAK İŞ AKIŞLARI VE FORM TASARIMI YÖNETİM SİSTEMİ

Tez kapsamında oluşturulan bu uygulama bir yardımcı sanayide faaliyet gösteren tüm iş süreçlerinin dijital ortamda yönetilmesini sağlayarak açık kaynak kod ile lisans maliyeti olmadan kuruma özel kişiselleştirilebilir bir platform edinilebilmesini sağlar. Uygulamanın mimarisi e-Dönüşüm modülleri ile birlikte Görsel 1'de yer alan sistem mimarisinde gösterilmiştir.

E-dönüşüm kapsamında kullanılan tüm modüller aynı orta-katman üzerinde farklı, ölçeklenebilir uygulama sunucuları üzerinde hizmet vermektedir. B2B, B2C ya da GİB iletişim katmanlarında özel entegratör platformundaki web arayüzleri ya da web servis işlevleri kullanılabilir.

Şekil 2: Dochuman Middleware Mimarisi



Kaynak: Netle Yazılım, 2023

4.1. VERİTABANI TASARIMI

Tez kapsamında özel bir üretim şirketinde işletilmekte olan Düzeltici Önleyici Faaliyetlerin (DÖF) Süreç Yönetimi, Giriş Kalite Kontrol (GKK) Süreç Yönetimi ve Akış Uygulamaları ile uyarlamaları yapılacaktır. Çalışmalara 28.01.2022 tarihinde başlanılmıştır.

Şekil 3: DFSUser Tablo Hiyerarşisi



Kaynak: Yazar tarafından derlenmiştir.

4.2. FORM TASARIMI

4.2.1. İsimlendirme Yöntemi

Form tasarımında oluşturulacak veritabanı tabloları belli bir kurala göre oluşturulmalıdır. Form, akış ya da rapor tasarımlarında "isimlendirme" yöntemi tekil olmalıdır. Önerilen isim yöntemi ise değişken ya da çalışma ortamındaki birimin sınıf türü kısaltılmalı ve amaca yönelik isimlendirme yapılmalıdır. İsimlendirme için İngilizce dili önerilmektedir.

Örnek: Form Tasarım Ortamı (Netle Yazılım, 2023);

Örnek: Form tasarım ortamı

Müşteri adı	Etiket	lbCustName, lbCustomerName, lbName
Müşteri adı	Metin Kutusu	tbcustName, tbCustomerName, tbName

4.2.2. Form Tasarımı

- Veri bağımlı form nesnelerinin data field alanı doğru şekilde tanımlanmalıdır. (DataField on DataFields)
- Veri girişlerinde düzenli ve pratik kullanım açısından veri giriş sırası denetlenmelidir. (Tab Order)
- Formu kaydeden kullanıcının, zorunlu olarak girmesi gereken alanlar belirlenmeli ve düzenlenmelidir. (Required on RequiredFields)
- Form nesneleri bazında güvenlik gerekli ise; alanların bazıları çeşitli durumlar bazında gizlenmesini (VisibleExpression) veya müdahale edilmemesini sağlamak (ReadonlyExpression) için gerekli düzenlemeler yapılmalıdır.
- Nesnelerin sağdan ve soldan hizalamalarına ayrıca uzunluk ve genişlik gibi şekillerinin düzenlenmesi görsel açıdan önem taşımaktadır. (Bileşen Düzeni)
- Form logo ve başlık gibi alanlar çeşitli görsel öğelerle desteklenmelidir. Bu görsel öğeler DocHuman web kurulum dizini altında yer alan images dizini altına saklanmalıdır. İlgili görsellerin boyutları form yüklenme hızı yavaşlığı, form kullanımı kolaylığı açışından belirli kriterlere göre ayarlanmalıdır.
- Form eğer akışa bağlanacaksa form bilgilerini güncelle menüsünden akış seçilmelidir.

- Form üzerinde veri girişi yapılacak alanlara gerekli durumlarda kullanıcılar tarafından kolay anlaşılabilecek açıklama mesajları yazılmalıdır. o Nesne türü Metin Kutusu türünde ise NullText sahası kullanılmalıdır. Nesne türü Metin Kutusu türünün dışında ise ToolTip özelliği kullanılmalıdır.
- Etiket nesneleri text-align değerinde left olacak şekilde düzenleme yapılmalıdır.
- Metin Kutusu nesnelerinde veri uzunluğuna göre maxlength değeri düzenlenmelidir
- Karmaşık ya da çok fazla sayıda UI tasarımlarında gruplandırma yapılmalıdır ve grup üst başlık (ve gerekiyorsa resim görselleri) kullanılmalıdır.

  Gruplandırma için aşağıdaki nesneler kullanılabilir.
  - Panel
  - Grup Panel
  - Tab Sayfası
- Formda detay tablolara veri girişi çok fazlaysa veri girişini hızlandırmak veya form veri girişi sırasında başka tablolardaki veriler önemliyse kullanıcı kullanımını en iyileştirmek için Popup penceresinden form bileşeni kullanılmalıdır.
- Kullanıcı tarafından serbest metin (text, memo, nvarchar(max), clob vb.) girişlerinde;
  - Sabit format kullanılacaksa RichTextBox (Metin Kutusu) nesnesi kullanılmalıdır.
  - Html format kullanılacaksa ve kullanıcı format vermek istiyorsa Html Editor nesnesi kullanılmalıdır.
- Formun uluslararası boyut kazanması için çoklu dil desteği ile form tasarımı yapılabilir.
- Formun web görünümünde bulunan “Kabul/Ret” düğmesinin davranışını ve sırasını DFSAdmin “Parametre” ekranından düzenlenebilir. Bu parametrelerle onay işlemlerinde not girilmesini sağlanabilir. Ek olarak “Akışa Gönder” düğmesinin davranışı da DFSAdmin “Parametre” ekranından yönetilebilir.
- Metin Kutusu nesnesinde bulunan MaskSettings başlığı altındaki özellikler ile belirli bir formatta veri girişi kuralı tanımlanabilir.
  - Metin kutusu kuralı ipucu için ShowHints özelliği kullanılmalıdır.

- Metin Kutusunun veri girişi öncesinde boş alanlar için yazacağı değerin girilmesi için PromptChar özelliği kullanılmalıdır.
- Maske ifadesinin düzenlenmesi için Mask özelliği kullanılmalıdır. Mask özelliğinde 9 opsiyonel alanı, 0 ise zorunlu alanı temsil eder.
- Veri girişinin kurallı yada metinsel olarak veri tabanına kayıt seçeneğini için IncludeLiterals özelliği kullanılmalıdır.
- Kural dışı veri girişinde verilecek hata metnin yazılması için ErrorText özelliği kullanılmalıdır.
    - **Örnek:** Metin kutusunun +(__) ___-__-__ formatında bir görünüm elde edebilmesi için;
      Mask özelliği: +(99) 000-00-00
      PromptChar özelliği: _

Metin Kutusu nesnesinde bulunan DisplayFormatString veri görünüm formatını düzenlemek için kullanılabilir (Netle Yazılım, 2018)

- **Örnek**: Metin kutusuna yazılan değerin "@domain.com.tr" uzantısı ile görünümünü sağlamak için; DisplayFormatString: {0}@domain.com.tr

4.2.3. Form Kayıt Listesi

- Kolon bazı güvenlik tanımı varsa, form listesi özellik ekranında ilgili kolon için visibleexpression değeri kullanılmalıdır.
- Anlaşılabilirliği arttırmak ve performans için listeye özel view yazılabilir.
- Satır bazı güvenlik tanımı varsa form özellikleri penceresinde yer alan "Kısıt" ifadesi ya da "Kullanıcı Tabanlı Güvenlik" etkin şekilde kullanılmalıdır.
- Performans ve güvenlik nedeniyle kayıt listesi görüntüsünde yer alan "Vt Bağlantı Kodu" için salt okunur özellikte açılmış bağlantı kodu kullanılmalıdır.
- Form kayıt listesinin varsayılan sıralaması "son girilen kayıt ilk sırada listelenir" şeklindedir. Bunun dışında bir gereksinim için liste özellik ekranında "Sıralama" alanına order by ifadesi girilmelidir.
- Listeye eklenen tüm kolonlar için length, fieldName, fieldDesc ve displayformatstring değerleri girilmelidir.

- Kolon bazında toplam, adet ya da ortalama gibi kümüle değerler gösterilecekse “Genel Özet Bilgi” ve “Grup Özet Bilgi” alanlarına eklenti yapılmalıdır.
- Form kayıt listesinin ekrana özel tasarımının (responsive tasarımının) sağlanması için listeye eklenen kolonlara lengthexpression değeri girilmelidir.

4.2.4. Sayısal Sahalar

Form üzerinde yer alacak sayısal sahalar için tasarım paletinde yer alan “Sayısal” sekmesindeki nesneler kullanılmalıdır.

Ölçeklendirme ya da uzunluk/genişlik gibi sınırlı değerler için seri çubuk kullanılmalıdır.

MaxValue ve MinValue değerleri düzenlenmelidir.

Grid üzerinden form, grid ve rapor tasarımlarında (RDL, RepX) sayısal sahaları için #,##.## gibi format tipleri kullanılabilir

4.2.5. Font Kullanımı

Projeye özel bir gereksinim yok ise, etiket nesneleri için “Segoe UI” font adı ve 11 point kullanılmalıdır.

Nesne özelliklerinden parent font özelliği true yapılarak, nesnenin arka nesnesinde yer alan font özelliklerinin kullanımı sağlanabilir.

4.2.6. Renk Kullanımı

Proje özel gereksinimleri yoksa standart olarak aşağıdaki durumlara özel renkler kullanılabilir.

- Başarılı ya da düzgün şekilde tamamlanmış bilgiler **Green**
- Başarısız ya da düzgün şekilde tamamlanmış bilgiler **Red**
- Dikkat edilmesi gereken bilgiler **Yellow**, **Orange**
- Bağlantı noktaları (Link, pencere ve detay görme) **Blue**

4.2.7. Stil Kullanımı

- Form tasarımında DFSAdmin tarafından desteklenmeyen tüm HTML5 CSS anahtar-değer özellikleri “Style” başlığı altında array olarak eklenmelidir.

- Bileşenlere DFS_CSS_Style özellikleri kullanılarak sayfaya uygun genişlik, gölgelendirme vb. ek stil özellikleri eklenebilir.
- Include dosyaları içerisinde ClientSide kodlama ve Style (CSS) birlikte kullanılabilir.
- Dochuman Web dizininde bulunan External dosyası içerisinde include dosyaları oluşturulabilir. Bu dosyalar içerisinde <script> etiketleri arasında JS, <style> etiketleri arasında da CSS özellikleri kullanılabilir. Dosyaların isimleri, içeriğinin kullanılacağı yere göre özelleştirilmelidir. Varsayılan olarak oluşturulmuş dosyalar: all.inc tüm form ve liste ekranlarında, allLists.inc tüm liste ekranlarında, allForms.inc tüm form ekranlarında çalışmaktadır. Bir form özelinde kullanılacak dosya ismi formFormId.inc şeklinde olmalıdır.

> ***Örnek:*** *Dochuman Web dizininde bulunan External dosyası içerisinde form özelinde kullanılacak include dosya adı: form8956.inc*

- Formların değişik çözünürlükteki ekranlara cevap verebilecek şekilde tasarlanmasına dikkat edilmelidir. "Elastik" sekmesi altında bulunan Taşıyıcı Panel, Satır Panel ve Kolon Panel nesneleri kullanılabilir. (Responsive Design)
- Genel olarak nesnelerin bireysel genişliği için de Width Expression özelliği kullanılabilir.
- Dochuman Web dizininde bulunan External dosyası içerisindeki include dosyalarına form içerisinde kullanılması için CSS sınıf yazılabilir ve bu sınıf CSS Class Expression özelliğinde kullanılabilir.

4.2.8. Client Side Kodlama

- Client side kodlama için DFSAdmin tarafında ilgili nesnelerin event alanları kullanılmalıdır ve JQUERY standartlarına göre geliştirme yapılmalıdır.
- Değişken ve yöntem tanımlamada javaScript isimlendirme ve önerilen yöntemleri kullanılmalıdır.
- Client side kodlamada property grid nesnesinde, event alanında varsa netleevent olayları kullanımda tercih edilmelidir. Bir nesnenin event isimlerinde hem

click hem de netleevent_click varsa ilgili işlem “netleevent” ile başlayan event içinde yapılmalıdır.

- Form üzerinde alan kapatma (liste link nesneleri, buton nesneleri, sol menu gibi) işlemlerinde "Netle.Client.UI” isim uzayı altındaki yöntemler kullanılmalıdır.

Örnek: *Form üzerinde butonları kapatma:*

Netle.Client.UI.HideAllButtons();

- Event kodlamaları içinde bir alan remark edilecekse "//alert(“ok”);” yerine “/*alert(“ok”);*/” yöntemi kullanılmalıdır.

4.2.9. Server Side Kodlama

- Sunucu katmanındaki kodlama gereksinimleri parametre değeri alan workflow nesneleri ile giderilmelidir.
- Amaca yönelik, özerk (otomatik) ve dar kapsamlı parametreler ile stored procedure mantığında, isim uzayı standartlarında geliştirilen WF nesneleri tasarlanmalıdır.
- Zorunlu parametre şeklinde gelen DFSUser değerine göre varsa güvenlik denetimi yapılmalıdır.
- Server Side kodlama ile bağlantılarda “netle client” java script isim uzayı altında yer alan yöntemler kullanılmalıdır.

4.2.10. Özelliklerde Expression Kullanımı

- Gereksinime göre nesne özellikleri declarative ya da imperative şekilde girilebilir.
- Visible Expression ile nesnenin özel durumlara göre görünürlüğü değiştirilebilir. Visible Expression özellliği imperative olarak kullanılabilir.
- DataValueExpression ile nesne value alanına, forma bağlı olduğu tablo haricinde, değişik kaynaklardan veri getirilebilir.

Örnek: *DataValueExpression özelliği için:*

DateTime.Now()

System.Convert.toString(DBUtil.SelectScalarValue(1, "select Column from TableOrViewOrFuncEtcName where Id = " &Model.Id.tostring()))

- Sahanın kullanıcı düzenlemesine kapalı olması gerekiyorsa ReadOnlyExpression kullanılıacak sadece okunur özelliğinin aktif edilmesi gerekir. EnableExpression veya Enabled özellikleri veri kaybına yol açabileceği için bu tür durumlar için kullanılmamalıdır. ReadOnlyExpression özellliği imperative olarak kullanılabilir.

***Örnek**: ReadOnyExpression özelliği için:*
System.Convert.toBoolean(IIF(UserId = 810, true, false))

4.2.11. URL Parametreleri

Formun farklı eylemlerindeki (action) URL bilgisindeki parametreleri ekrana yazdırılacaksa ya da JQUERY ile işlenecekse "webRequest" sınıfı kullanılarak alınmalıdır.

4.2.12. Dosya İlişkilendırme (Attachment)

- Form kayıt bilgisine özel bir ya da birden fazla dosya ilişkilendirme gereksinimleri için Dosya Yükleyicisi nesnesi forma eklenmelidir.
- Taranan dosyaların da forma dosya ilişiği olarak kullanılması isteniliyorsa; Netle Yazılım DocHuman Tarayıcı Uygulaması kurulmalı ve Dosya Yükleyicisi özelliklerinden UseScanner aktif edilmelidir.
- Formla ilişkilendirilen dosya/dosyalara iş akışından zorunlu parametre şeklinde gelen Attachment dizisi ile ulaşılabilir.
- Bir formda birden fazla Dosya Yükleyicisi olması durumunda iki bileşene yüklenen dosyaların kayıt görünümünde karışıklık yaratmaması için GroupName özelliği kullanılmalıdır.

4.2.13. Çoklu Seçimler

- Açılan Kutu ya da Radyo Düğmesi nesneleri çoklu seçimler için kullanılmalıdır.
- Eğer veri sayısı sınırlı ve en fazla 5 ise Radyo Düğmesi diğer durumda ise Açılan Kutu nesneleri kullanılabilir.
- Çoklu seçim verisi başka bir tablodan, view'den beslenecekse Rehber bileşeni kullanılmalıdır.
- Çoklu dil sorununu engellemek ve/veya veritabanında index üzerinden hızlı erişim sağlanması için metin değerlerine sayısal değerler işlenmelidir.

***Örnek**: ReadOnyExpression özelliği için:*
System.ConvcbCities.items özelliğine
0=İzmir
1=Ankara
2=İstanbul
değerleri eklenebilir.

4.2.14. Süreç İle Bütünleşik Formlar

Süreç içindeki gelişmeler, hata mesajları ya da son kullanıcı mesajları form üzerinde "Gelişmiş" altında yer alan süreç bağlantılı nesneler kullanılarak forma eklenmelidir.

4.2.15. Kayıt Saklama Sonrası

Yoğun yeni veri girişlerine göre tasarlanmış formlarda form özelliklerinde yer alan afterinsertmode değeri browsenewrecord olarak seçilmelidir.

4.2.16. Veri Doğrulama Ve Güvenlik

Form katmanında yapılan doğrulama ya da güvenlik tanımlamaları bir alt katmanda da tekrarlanmalıdır. Bir alt katman veritabanı katmanında ya da süreç yönetimi düzeyinde olabilir.

Örnek: *Ekranda bu saha required olarak işaretlenmelidir ve veritabanı tablosunda da not null özelliği belirtilmelidir.*

4.2.17. Özel Tema Kullanımı

Formlar özel bir tema altında çalıştırılıyorsa layoutname ve listlayoutname değerleri kontrol edilmelidir.

4.2.18. Tasarım Doğrulama Sonuçları

Her bir formun tasarımı sonrasında "Tasarım Doğrulama Donuçları" penceresinde yer alan hatalar ya da uyarılar dikkate alınarak tekrar düzenlenmelidir.

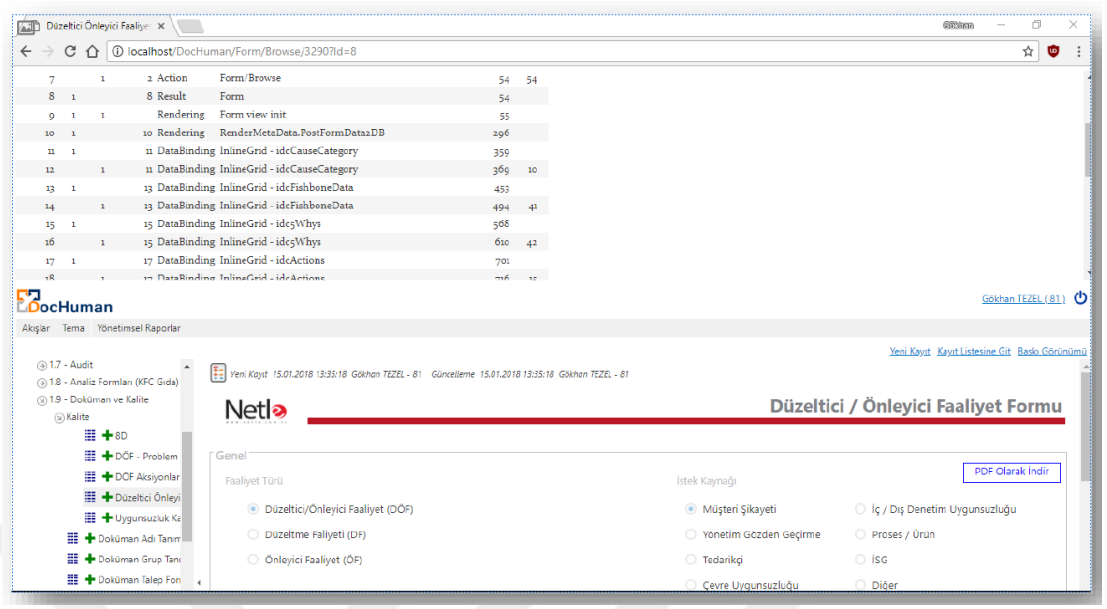
4.2.19. Tasarım Doğrulama

- Form açılışı ve güvenlik tanımlamaları test edilmelidir.
- Veri giriş kolaylığı ve ergonomi test edilmelidir.
- Tarayıcı javaconsole üzerinden script hatasının olmadığı kontrol edilmelidir.

4.2.20. Performans

Tasarım doğrulamaya ek olarak formun performansı da gözden geçirilmelidir. Kullanılıan expression, rehber, grid vb. nesnelerin harcadıkları zamanlar kontrol edilmeli, gerekiyorsa müdahale edilmelidir. Bu kontrol için Dochuman Trace kullanılmalıdır. Trace aktifleştirmek için ../DocHuman/Setup/EnableTrace bağlantısı kullanılır.

Şekil 4: Dochuman Trace



Kaynak: Netle Yazılım, 2018

4.2.21. Ek Form Kullanımı

Formlar oluşturulurken önceden belirlenen veritabanı tasarımına uygun olarak belirlenen tablolara bağlı olarak tasarım yapılır. Formlar genel olarak bir bilgisayar ekranının tamamını kaplayacak ve kullanıcıyı aşağı ve yukarı sürükleyerek yormayacak şekilde tasarlanır. Bu şekilde tasarlanan formlar bazı durumlarda yetersiz kalır ve ek form oluşturularak formlar birbirine bağlanır. Örneğin, seçeneklerden “A” olanı işaretlenirse ek bir form açılsın ”B” olarak işaretlenirse sayfa sonunda detaylı olarak belirlenen bilgiler verilsin denilebilir.

4.2.21.1. Grid

Form tasarımında "grid", bir sayfa veya ekranın düzenini ve bileşenlerin yerleşimini organize etmek için kullanılan bir tekniktir. Grid sistemi, sayfayı veya formu bölümlere ayırarak ve bileşenleri belirli hücrelere yerleştirerek düzenli bir yapı oluşturur. Grid, formun veya sayfanın bileşenlerini hizalamak, ölçeklendirmek ve dengelemek için bir kılavuz görevi görür. Grid sisteminin form tasarımında kullanılmasının bazı nedenleri şunlardır;

Düzenlilik ve tutarlılık: Grid, bileşenleri belirli hücrelere yerleştirerek düzenli bir yapı sağlar. Bu, kullanıcıların formu daha kolay anlamalarını ve kullanmalarını sağlar. Ayrıca, tutarlı bir görünüm ve his yaratır.

İletişim ve okunabilirlik: Grid sistemi, bileşenlerin ve içeriklerin yerleşimini optimize ederek formun okunabilirliğini artırır. Bileşenler arasındaki boşlukları ve hizalamaları düzenleyerek, kullanıcıların bilgileri daha kolay ve akıcı bir şekilde taramalarını sağlar.

Ölçeklenebilirlik: Grid, farklı ekran boyutlarına ve cihazlara uyum sağlamak için tasarımın ölçeklenebilir olmasını sağlar. Bileşenlerin hücrelere yerleştirilmesi, ekranın genişliğine veya yüksekliğine bağlı olarak otomatik olarak ayarlanabilir.

Hızlı ve verimli tasarım süreci: Grid sistemi, bileşenlerin ve düzenin hızlı ve verimli bir şekilde tasarlanmasını sağlar. Bileşenleri yerleştirmek için bir kılavuz sağlar ve tutarlı bir düzen oluşturmanıza yardımcı olur.

Sonuç olarak, grid sistemi form tasarımında düzenli ve tutarlı bir yapı oluşturmak, okunabilirlik ve kullanıcı deneyimini iyileştirmek ve tasarım sürecini hızlandırmak için kullanılır. Kullandığımız form tasarımında dikkat etmemiz gereken tanımlamalar ve kurallar şu şekildedir;

- Connection Code ve DataSource tanımlamalarına dikkat edilmelidir.
- Gereksiz kolonlar listeye eklenmemelidir.
- Boyuta göre yatay ve dikey scrollbar gösterilmelidir.
- Çok sayıda kayıt içeren detaylarda pagingSupport aktif edilmelidir.
- Sayısal alanlarda özet bileşenler kullanılmalıdır.
- İhtiyaca göre; yeni, sil ve güncelle linklerinin yerleri ve edit mode düzenlenmelidir.
- Grid (ListDetailControl) için KeyField alanı belirlenmelidir.
- Kullanım kolaylığı için; Title ve Table Wrap, Grouping ve Filter özellikleri kullanılmalıdır.
- Veri kısıtı ve güvenlikte; satır için WhereSQL ve WhereSQLExpression, kolon için kolona ait visibleExpression özellikleri kullanılmaldır.
- Veri girişini kolaylaştırmak için refEdit nesneleri kullanılmalıdır.
- Performans için DisplayDataSource kullanılmalıdır.

- Fazla kaydı olan ve sayısal olarak anlamlı veriler elde edilebilecek detaylar için PivotGrid kullanılmalıdır.

4.2.21.2. Pop-Up

"Pop-Up" (Açılır Pencere), kullanıcının ana sayfa veya içerikle etkileşim halindeyken belirli bir eylem gerçekleştirmesi gerektiğinde ortaya çıkan küçük bir penceredir. Pop-Up' lar, kullanıcılara ek bilgiler sunmak, tekliflerde bulunmak, kayıt olma veya abonelik gibi eylemleri gerçekleştirmek için kullanılır.

Pop-Up' ların kullanımı aşağıdaki amaçlarla gerçekleştirilebilir:

Promosyon ve Teşvik: Bir Pop-Up, kullanıcılara indirimler, özel teklifler veya kampanyalar hakkında bilgi verebilir. Bu şekilde, kullanıcılar satın alma veya kayıt gibi belirli bir eylemi gerçekleştirmeye teşvik edilir.

Bilgilendirme ve Rehberlik: Pop-Up' lar, kullanıcılara ek bilgiler sağlamak veya yol göstermek için kullanılabilir. Örneğin, bir formun doldurulması gereken alanlarını açıklayan bir pop-up kullanılabilir.

Abonelik veya Kayıt: Pop-Up' lar, kullanıcıların e-posta aboneliklerine veya üyelik kaydına yönlendirilmeleri için kullanılabilir. Bu şekilde, kullanıcılar daha fazla bilgi almak veya özel içeriğe erişim elde etmek için kayıt olabilirler.

Uyarılar ve Onaylamalar: Önemli bilgilerin veya onay mesajlarının iletilmesi için Pop-Up' lar kullanılabilir. Örneğin, bir formun başarıyla gönderildiğine dair bir onay mesajı Pop-Up olarak görüntülenebilir.

Pop-Up' ların doğru şekilde kullanılması önemlidir. Kullanıcı deneyimini olumsuz etkilememek ve kullanıcılara rahatsızlık vermemek için aşağıdaki faktörlere dikkat edilmelidir:

- Pop-Up' lar mantıklı bir zamanda ve uygun bir şekilde görüntülenmelidir.

-Kullanıcının ana içeriği veya görevi üzerinde büyük bir engel oluşturmamalıdır.

- Kullanıcılara kapatma veya gizleme seçenekleri sunulmalıdır.

- Pop-Up' lar mobil cihazlara uygun ve kullanıcı dostu olmalıdır.

Sonuç olarak, Pop-Up' lar form tasarımında kullanıcıların dikkatini çekmek, bilgilendirmek veya belirli eylemleri gerçekleştirmelerini sağlamak için kullanılan

etkili bir araçtır. Ancak, kullanıcı deneyimini düşünerek dikkatlice planlanmalı ve uygulanmalıdır. Uygulamada kullanılması gereken kurallar şu şekilde sıralanmıştır;

- Ek formda kullanılacak detayların fazla olması veya tasarımsal ihtiyaçlar olması halinde Pop-Up penceresinde form nesnesi kullanılmalıdır.
- Pop-Up form yeni kayıt veya düzenleme modunda açılabilir.
- Kolon sayısı fazla grid üzerinde listeleme ya da veri girişi yapılabilir ancak veri düzeltmede kullanılacak kolon sayısı çok fazla ise ve ergonomik değilse, grid üzerinden başka bir forma Pop-Up geçiş düzenlenebilir ve verilerin buradan düzenlenmesi sağlanabilir.
- Pop-Up olarak kullanılan formların kilitli olması yanlış veri girişlerini önler.
- Popup formuna ana formdan parametre geçirmek için beginCallback JS olayı kullanılabilir.

***Örnek**: Popup formuna ana formdan kayıt ıd değerini geçirmek:*

```
e.customArgs["RecordId"] = DFSRecordId;
```

- Pop-Up üzerindeki işlem bittikten sonra ana formda güncellenmesi gereken yerler varsa Pop-Up nesnesinin endCloseUp JS olayı kullanılmalıdır.

***Örnek**: Ana formda grid güncelleme:*

```
idcDetail.PerformCallback();
```

4.3. TABLO VE INDEX TASARIMI

- Tablo tasarımlarında proje kısaltmasına göre isimlendirme yapılmalıdır.

***Örnek**: Proje kısaltmasına göre isimlendirme;*

Proje Kısaltması: Ntts TabloAdı: NttsTblTalepler

- Tablo türüne göre suffix bilgisi eklenmelidir. Öneriler;
  - Sabit kayıtlar için MAS
  - Transaction kayıtları için TRA
  - Parametre kayıtları için PRM

***Örnek**: Şirket parametreleri NttsTblCompPrm*

Şirket geribildirimleri NttsTblCompFeedbackTra

- Tablo sabit bilgileri içerecekse Id kolunu int identity özellikte olmalıdır.

- Tablo transaction (hareket) bilgileri içerecekse ve kayıt sayısı fazla olacak Id bigint identity özellikte olmalıdır.
- Tablo ana bir tabloya bağlanacaksa ve grid kullanılacaksa MasterId değer ile türü seçilmelidir ve foreign key özelliği tanımlanmalıdır. Foreign key bağlantısında proje gereksinimine göre on delete casce özelliği tanımlanabilir.
- Tablo alanlarında constraint ifadeleri gereksinime göre düzenlenmelidir. (Kolon uzunluğu, null/notnull, default value vb.)
- Tablo tasarımında temel normalization (normalizasyon) kurallarına dikkat edilmelidir. Tablo içindeki yer alacak kolonların %80 oranında dolu olması beklenmelidir. Boş olan kolonlar gerekiyorsa farklı bir tabloda tasarlanmalıdır.

Tablo tasarımlarının yanında index tasarımları da yapımalıdır. Bu amaçla aşağıda maddeleri göz önünde bulundurmak gerekmektedir.

- Id değeri Primary key olduğu için ayrıca index tanımlaması yapılmamalıdır.
- MasterId gibi foreign key bağlantılı tüm kolonlar tek başına index olarak tanımlanmalıdır.
- Proje içinde, akışlarda, rehberlerde arama yapılacak sahalar bağlantı özelliğine göre segmented index olarak tanımlanmalıdır.
- Form listesinde kullanıcı tarafından yapılacak en yaygın aramalara ilişkin tek kolonlu ya da çok kolonlu index tanımlaması yapılmalıdır.

4.4. VIEW TASARIMI

- View isimleri proje kısaltmasına göre isimlendirilmelidir ve amaca yönelik son ek (suffix) eklenmelidir.

Örnek: *NttsVLTalepler: Ntts projesinde, View nesnesi, Liste için kullanılmaktadır.*
NttsVRTalepler: Ntts projesinde, View nesnesi, Rehber için kullanılmaktadır.

- Birden fazla ortak amaca yönelik view nesnesi varsa ek suffix değeri eklenebilir.

Örnek: *NttsVRTalepAra*
NttsVRTalepGirisi

- Rehber ya da grid üzerinde bağlantılı (masterId üzerinden ya da farklı kod üzerinden) sahalar için kullanıcıya açıklama değerini göstermek için inner/outer join özelliği tanımlanmalıdır ve form üzerinde listede ya da rehberde bu nesne kullanılmalıdır.

4.5. RAPOR TASARIMI

İş süreçlerinde rapor tasarımı, işletmelerin iş süreçleriyle ilgili verileri organize ederek, analiz edilebilir ve anlaşılabilir raporlara dönüştürmek için yapılan tasarım sürecidir. Raporlar, işletmelerin performansını izlemek, karar vermeye yardımcı olmak ve stratejik yönetim süreçlerini desteklemek için kullanılır.

4.5.1. RDL

Dochuman uygulamasında oluşturulacak olan her bir raporlama sayfası için ayrı ayrı RDL oluşturulmalıdır. RDL' ler raporlamalarda kullanılan şablonlardır.

- Sayfa header bilgisinde yer alacak bilgiler eklenmelidir.
    - Standart olarak rapor zamanı tarih biçiminde eklenebilir.
    - Şirket logosu ya da proje görseli üst bölüme eklenmelidir.
    - Rapor başlığı ortalanacak şekilde eklenmelidir.
- Sayfa footer bilgisinde yer alacak bilgiler eklenmelidir. Standart olarak aktif sayfa numarası ve toplam sayfa numarası değerleri yazdırılmalıdır.
- Tablo ya da liste halinde yapılacak raporlarda kolon genişlik değeri veriye göre düzenlenmelidir.
- Kolonlarda hizalama için aşağıdaki öneriler kullanılmalıdır:
    - Metin alanlarda SOL
    - Sayısal alanlarda SAĞ
    - Tarih alanlarında SOL

    - Enum ya da sabit değerlerde ORTA
- Sayısal alanlarda ya da tarih alanlarında işletim sisteminin (culture) kültürel etkiden bağımsız format değerleri kullanılmalıdır.
- Rapora kaynak olacak veri için gereksiz kolonlar select ifadesinden çıkarılmalıdır.
- Veri hacmi büyüklüğüne göre expression ya da calculated sahalar varsa veritabanında view katmanında çözülmelidir.
- Rapor body alanı, header ve footer yüksekliklerine göre aralarında boşluk kalmayacak şekilde düzenlenmeli. Arada boşluk kalması durumunda son sayfanın boş çıkma olasılığı olabilir.
- Rapor hazırlandıktan sonra run ile çalıştırılmalı ve print layout üzerinden sayfa genişliği ve yüksekliği taşma yapmaması için denetlenmelidir.

4.5.2. XTRAREPORT (REPX)

- Rapor sayfa boyutu ihtiyaca göre özelleştirilmelidir.
- Custom palette, calculated fields, function vb. kullanımları anlaşılır şekilde düzenlenmelidir.
- Kullanılan grafiklerin boyutlarına dikkat edilmelidir. Fazla küçük grafikler form çıktısında yeterli alan bulunamadı şeklinde hata verebilir.
- Kullanılan scriptler “Valide” ile kontrol edilmelidir.
- Veri seti güvenlik konularına dikkat edilerek oluşturulmalıdır.
- Rapor tasarımı, kullanılacak sistemde çalıştırılmalı ve kontrol edilmelidir. (Devexpress versiyon uyumsuzluğu olabilir.)
- Repx tasarım ortamında, iş akışı ile veri seti oluşturup, config ayarlarını bu veri setine bağlı düzenleyerek tasarım ortamında akış ile oluşturulan veri seti üzerinde tasarım yapılabilir. Bunun yanında config ayarları DFS ortamına bağlı düzenlenerek de çalışma yapılabilir. Böylece e-dönüşüm için özelleştirilmiş alanlar varsayılan olarak elde edilebilir.
- Script alanına eklenen scprit ile DFSAdmin veri tabanı bağlantılarında bulunan farklı veri tabanlarına erişerek veri elde edilebilir. Bu işlem için

DFS.Form.Bll isim uzayı altında bulunan DBUtil sınıfının SelectDataTable metodunu kullanılabilir.

Örnek: *SelectDataTable fonksiyonu kullanımı;*
DFS.Form.Bll.DBUtil.SelectDataTable(connectioncode, “”query”);

4.6. WORKFLOW TASARIMI (WF)

- Akış nesnesinin dışarıdan alacağı ya da dışarıya döneceği tüm parametreler Arguments bölümüne eklenmelidir.
- Akış içinde kullanılacak değişkenler Variable bölümüne eklenmelidir.
- Değişken tanımlamaları scope alanı sadece etkili olacağı alan için tanımlanmalıdır. İç içe akışlarda ya da gruplandırılmış iş mantıkların her değişken, sorumlu olduğu scope alanına göre özel olarak tannımlanmalıdır.
- Client Side tarafından formdan çağırılarak kullanılan akışlarda DFSUser değişkeni Arguments bölümüne input ve Int32 tipinde tanımlanmalıdır.
- Akışlar Hata Yakala (Try-Catch) nesnesi içinde tasarlanmalıdır. Catches bölümündekullanıcı dostu hata açıklamaları iletilmelidir.
- Akışlarda Tablo Oku nesneleri kullanıldığında sonrasında ilgili datatable ile işlem yapmadan önce datatable’ın dolu / boş olduğu kontrol edilmelidir.
- Akış tasarımının okunabilirliği açısında İç Akış nesnesi kullanılmalıdır.
- Kullanılan İç Akış nesnelerine veya herhangi bir akış bileşenine ek açıklamalar (annotations) ekleyerek akışın bölümlerine ilişkin notlar yazılabilir.
- Canlı ortam kullanımlarında DFSAdmin “Parametreler” ekranının “İş Akış” alanında bulunan “Akış başlangıç bitiş log kayıtları atılmasın” seçeneği işaretli olması önerilir. Bu parametre sayesinde veri tabanı boyundan ve akış süresinden avantaj sağlanır. İş akışının log kayıtlarının yazılması istenilen aralık için “Akış Logu Durdur”, “Akış Logu Başlat” nesneleri kullanılabilir. İş akışında yalnızca kişiselleştirilmiş log kayıtlarının yazılması için ise “İş Akış İçin Log“ nesnesi kullanılabilir.
- Kulllanılmayacak namespace’ler import edilmemelidir.

- Detaylı akış tasarımlarında Sıralı Bütün Akış, İç Akış gibi nesnelerle gruplandırma yapılmalıdır.
- Forma bağlı akışların yanı sıra Client tarafında anlık olarak işlemler için jquery attachment /sync akışları kullanılabilir. Burada JS kodlama sırasına dikkat edilmelidir.

Örnek:
```
var o = new Object();
Netle.Client.WFCore.startSyncUIWF (“yourWFName”, o,
function(data{if (data[“Result”]) {
alert(data[“Result”]);
return; }
alert(data[“myRes”]);
console.log(data[“myRes”]);
DFSDateTimePicker1.SetValue(new
Date(data[“myRes”]));
alert(“Islem Tamamlandi”);
});
```

- Akış motorunda serialize olmayacak nesne kullanımı olacaksa, örneğin NetOpenx nesnesi, FileInputStream, FileOutputStream, olası saklamalarda (onay bekleme, zaman bekleme vb.), akışın sağlıklı işlem yapabilmesi için ve hata vermemesi için serialize/deserialize olamayan nesneler NoPersistScope nesnesi içinde kullanılmalıdır.
- Akış içerisinde kullanılmayan hiçbir değişken variable bölümünde kalmamalıdır. Aynı değişken ortak scope içinde birden fazla tanımlanmamalıdır.

4.7. PROJE SÜREÇLERİ

- Süreç ile ilgili ön analiz toplantısı 28.01.2022 tarihinde başlanmıştır.
- Analiz toplantısında mevcut Düzeltici Önleyici Faaliyet Süreci ve sürece ilişkin veri yapıları aktarılmış ve ilgili dokümanlar paylaşılmıştır.

- Analiz toplantıları ile belirlenen akışa ilişkin çözüm tasarım raporu hazırlanmıştır.
- Çözüm tasarım raporu sunum ve onayı ardından akışa ilişkin prototip çalışması yapılmıştır.

4.8. PROJE ZAMAN ÇİZELGESİ

Proje zaman çizelgesi, araştırmanın planlanan süresini ve hangi aşamaların ne zaman tamamlanacağını gösteren bir planlama aracıdır. Bu projeyle ilgili bir işletmede uygulamaya alınacak projenin bir çizelgesine Tablo 2'de yer verilmektedir;

Tablo 2: Proje Zaman Çizelgesi

Yapılacak Çalışma	**Başlama**	**Tamamlanma**	**Durum**
Proje Başlama Sunumu	18.01.2022	18.01.2022	Tamamlandı
Altyapı Çalışmaları	02.02.2022		Tamamlandı
Ortam Kurulumunun Gerçekleştirilmesi	02.02.2022	04.02.2022	Tamamlandı
Kullanıcı Tanımlarının Yapılması	05.03.2022	08.03.2022	Tamamlandı
Düzeltici Faaliyet Süreci Çalışmaları	10.03.2022	15.06.2022	Tamamlandı
Analiz Çalışmaları / Toplantıları	16.06.2022	01.07.2022	Tamamlandı
Çözüm Tasarım Raporu Çalışmaları	15.07.2022	02.08.2022	Tamamlandı
Prototip Çalışması ve Onayı	05.08.2022	27.08.2022	Tamamlandı
Entegrasyon Çalışmaları	01.09.2022	05.09.2022	Tamamlandı
Eğitim/Kullanım Materyallerinin Hazırlanması	10.09.2022	21.09.2022	Tamamlandı
Son Kullanıcı Eğitimleri	25.09.2022	15.10.2022	Tamamlandı
Canlıya Alma	16.10.2022	16.10.2022	Tamamlandı

Kaynak: Yazar tarafından derlenmiştir.

4.9. KULLANICI VE ORGANİZASYON ENTEGRASYONU

Uygulamada kullanılacak erişimler ve buna bağlı haklar işletmenin organizasyonel yapısına bağlı olarak oluşturulur. Her kullanıcı kendisine ait olan kullanıcı adı ile sisteme giriş yapar ve bu kullanıcıya verilen haklar doğrultusunda işlemlerini bir yetki çerçevesinde yürütür. Bu anlamda oluşturulan "Entegrasyon Kapsamı Veri Sözlüğü" prosedürü altındaki erişim bilgileri aşağıdaki gibidir;

- **Gereksinim**

Çalışmanın amacı sağlanacak web servis/servisleri üzerinden organizasyon kırılımı ve organizasyon ağaç bilgisi, kullanıcı bilgileri, kullanıcı/çalışma grubu bilgilerinin BPM Uygulama Yazılımı üzerine aktarılmasıdır.

Web servis/servislerinden gelen bilgiler, en az gün aşırı olarak alınabilmelidir. Bilgiler tekrarlı, çekilebilir olmalıdır.

Entegrasyon üzerinden gelen bilgiler ile yapılan takvimleme/takvimlemeler ile organizasyon şemasının güncel tutulması sağlanmalıdır.

- **Entegrasyon Kapsamı**

Entegrasyon aşağıdaki temel 3 tanımlama için çalışmaktadır.

- Kullanıcı Bilgileri
- Birim Bilgileri (Departman)
- Kullanıcı Grubu Bilgileri

- **Veri Sözlüğü**

Sistemde açılacak olan kullanıcı bilgilerinin veritabanında oluşturulacak tablo alanları ve veri tipleri Tablo 3'de belirtilmiştir.

Tablo 3: Veri Sözlüğü Kullanıcı Bilgileri

Alan	AlanKodu	Veri Tipi	Şartlar
Kullanıcı Adı	UserName	String	Zorunlu, Benzersiz
Kullanıcı Eposta Adresi	Eposta	String	Zorunlu, Benzersiz
Kullanıcının Adı	Name	String	Zorunlu
Kullanıcının Soyadı	SurName	String	Zorunlu
Birim (Departman) Adı	UnitName	String	Zorunlu
Birim (Departman) Kodu	UnitCode	String	Zorunlu
Yönetici User Name	ManagerUserName	String	Opsiyonel
Ünvan Adı	TitleName	String	Zorunlu
Ünvan Kodu	TitleCode	String	Opsiyonel

Kaynak: Yazar tarafından derlenmiştir.

Birim Bilgileri

Kullanıcılara ait bağlı bulundukları birimlerin bilgileri ve bu bilgilerin veritabanındaki tablo bilgileri Tablo 4'de verilmiştir. Farklı departmanlardaki kullanıcılar farklı yetkilerle işlem göreceklerdir.

Tablo 4: Veri Sözlüğü Birim Bilgileri

Alan	AlanKod	Veri	Zorunluluk
Birim	UnitName	String	Zorunlu
Birim	UnitCode	String	Zorunlu
Üst Birim	ParentUnitName	String	Opsiyonel
Üst Birim	ParentUnitCode	String	Opsiyonel

Kaynak: Yazar tarafından derlenmiştir.

Kullanıcı Grubu

Farklı departmanlarda bulunan kullanıcılar aynı grubun üyesi olabilirler. Örneğin planlama departmanındaki veri giriş personeli ile kalite departmanındaki veri giriş personelleri aynı grubun üyesi olarak kullanıcı grubunda yer alırlar. Bu tür durumlar için de ilgili veritabanında Tablo 5'deki gibi bir tablo üzerine Kullanıcı Grubu alanı yaratılabilir.

Tablo 5: Veri Sözlüğü Kullanıcı Grubu

Alan	AlanKod	Veri	Zorunluluk
Kullanıcı Grubu	UserGroupName	String	Zorunlu, Benzersiz
Kullanıcı Grubu Kodu		String	Zorunlu, Benzersiz

Kaynak: Yazar tarafından derlenmiştir.

Kullanıcı Grup Üyeleri

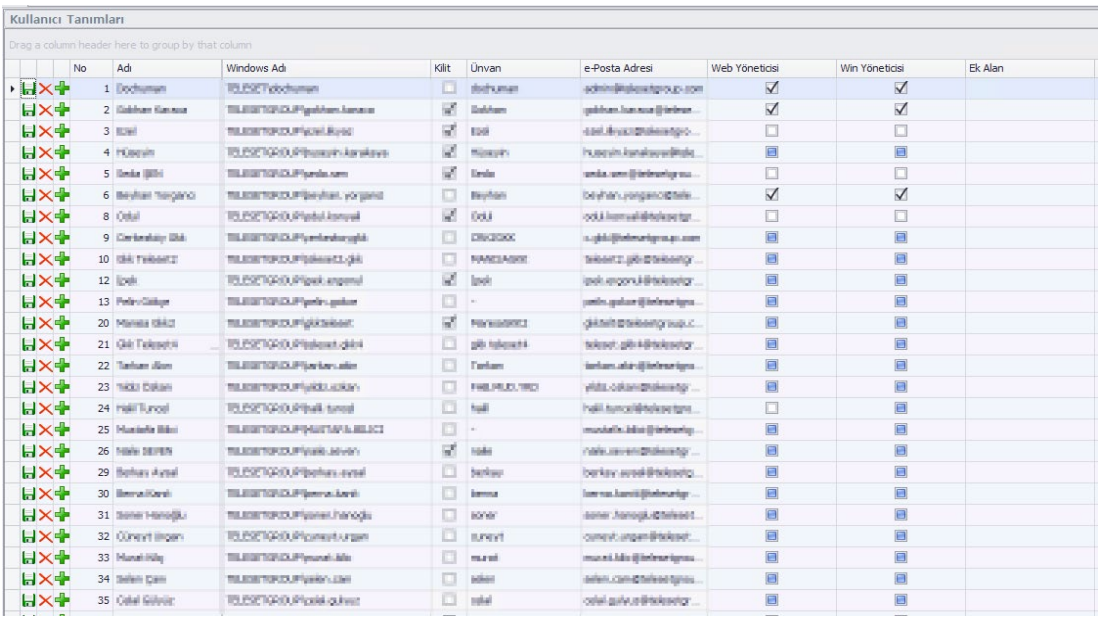
Kullanıcılar ve kullanıcı grupları ayrı olarak bir tabloda ilişkilendirilerek tutulmuştur. Grup Üyeleri kodu altında yaratılan kullanıcılar Giriş Kalite Kontrol birimi altında Depo sorumlusu kullanıcıları için kullanılmıştır. Tablo 6 kullanıcı grup üyelerini gösterirken Şekil 5 bunun yazılım üzerinden kulanımını göstermektedir.

Tablo 6: Veri Sözlüğü Kullanıcı Grup Üyeleri

Alan	AlanKod	Veri	Zorunluluk
Kullanıcı	UserName	String	Zorunlu
Kullanıcı Grubu Kodu	UserGroupName	String	Zorunlu

Kaynak: Yazar tarafından derlenmiştir.

Şekil 5: Dochuman Kullanıcı Tanımları Ekranı



Kaynak: Yazar tarafından derlenmiştir.

4.10. ENTEGRASYON WEB SERVİS GEREKSİNİMLERİ

- **Temel Gereksinimler**

Web servislerinin https protokolü üzerinden yayın yapılıyor olması gerekmektedir. WEb servisleri **soap** veya **restfull** olabilir. Restfull servislerin istek ve dönüş yapıları XML veya json olabilir.

Web servislerinin dönüş değeri olarak minimumda veri sözlüğü ile belirtilen dataları liste şeklinde dönmesi beklenmektedir.

Benzersiz alanlar haricinde tüm değerler büyük harf Türkçe karakter dönüş yapılması beklenmektedir.

- **Kimlik Doğrulama**

Kimlik doğrulama için aşağıdaki yöntemler desteklenmektedir.

- Basic Authentication
- Bearer Token
- Custom Header Key/Value
- Custom Header

- **Temel Entegrasyon Gereksinimi:**

Bu entegrasyon tipi çalışmak için, Kullanıcı Bilgileri Web Servisi gerektirir. Bu serviste minimum Kullanıcı Bilgileri Veri Sözlüğünde yer alan zorunlu alanların olması gerekir.

Bu entegrasyon tipi ile kullanıcı bilgileri, birimler ve birim ağacı entegrasyonu sağlanmaktadır.

Birimler kullanıcı bilgilerinden derlenir ve oluşturulur. Birim ağacı hiyerarşisi, birimlerde yer alan kullanıcıların yönetici tanımlamalarından yola çıkılarak yapılmaktadır. Birimdeki kullanıcı yöneticileri arasında sayısal olarak en çok adı geçen yöneticinin bulunduğu birim üst birim kabul edilir.

Oluşturulan her birime karşılık bir kullanıcı grubu oluşturulur. Birim altındaki kullanıcılar bu gruplara atanır.

“Manuel Yönetilecek” işaretli kullanıcılar ve sistem kullanıcıları bu

entegrasyona dahil edilemez.

Şekil 6'de oluşturulan veritabanını olan UserData dan XML'e dönüştürme örneği görülmektedir.

Şekil 6: UserData XML Dönüş Deseni Örneği

```
<UserData>
        <UserName>ali.username</UserName>
        <Email>ali@test.com</Email>
        <Name>ALİ</Name>
        <SurName>VELİ</SurName>
        <UnitName>MERKEZ</UnitName>
        <UnitCode>MRKZ01</UnitCode>
        <ManagerUserName></ManagerUserName>
        <TitleName>MÜDÜR</TitleName>
        <TitleCode>MDR</TitleCode>
        <PersonalID>0001</PersonalID>
        <PhoneNumber>02164442233</PhoneNumber>
        <GSMNo>05554442233</GSMNo>
</UserData>
<UserData>
        <UserName>veli.username</UserName>
        <Email>veli@test.com</Email>
        <Name>VELİ</Name>
        <SurName>ALİ</SurName>
        <UnitName>BİRİM</UnitName>
        <UnitCode>BRM01</UnitCode>
        <ManagerUserName>ali.username</ManagerUserName>
        <TitleName>PERSONEL</TitleName>
        <TitleCode>PRS01</TitleCode>
        <PersonalID>0002</PersonalID>
        <PhoneNumber>02164443322</PhoneNumber>
        <GSMNo>05552223344</GSMNo>
</UserData>
Result
```

Kaynak: Yazar tarafından derlenmiştir.

Şekil 7'de ise oluşturulan veritabanını olan UserData dan JSON formatında verinin aktarıldığı örneği görülmektedir.

Şekil 7: UserData JSON Dönüş Deseni Örneği

```
<UserData>
        <UserName>ali.username</UserName>
        <Email>ali@test.com</Email>
        <Name>ALİ</Name>
        <SurName>VELİ</SurName>
        <UnitName>MERKEZ</UnitName>
        <UnitCode>MRKZ01</UnitCode>
        <ManagerUserName></ManagerUserName>
        <TitleName>MÜDÜR</TitleName>
        <TitleCode>MDR</TitleCode>
        <PersonalID>0001</PersonalID>
        <PhoneNumber>02164442233</PhoneNumber>
        <GSMNo>05554442233</GSMNo>
</UserData>
<UserData>
        <UserName>veli.username</UserName>
        <Email>veli@test.com</Email>
        <Name>VELİ</Name>
        <SurName>ALİ</SurName>
        <UnitName>BİRİM</UnitName>
        <UnitCode>BRM01</UnitCode>
        <ManagerUserName>ali.username</ManagerUserName>
        <TitleName>PERSONEL</TitleName>
        <TitleCode>PRS01</TitleCode>
        <PersonalID>0002</PersonalID>
        <PhoneNumber>02164443322</PhoneNumber>
        <GSMNo>05552223344</GSMNo>
</UserData>
Result
```

Kaynak: Yazar tarafından derlenmiştir.

Birim Entegrasyonu Gereksinimi:

Organizasyon (Birim) bilgilerinin farklı bir web servis üzerinden oluşması gerektiği durumda uygulanmalıdır. Bu ikinci servis organizasyon hiyerarşisi ve birimlerin oluşturulması için kullanılır.

Bu entegrasyon tipi temel entegrasyon gereksinimlerine ek olarak, Veri Sözlüğü altındaki Birim bilgilerini içeren ikinci bir web servise ihtiyaç duyar.

Bu tip entegrasyonda birim bilgileri ve birim ağacı hiyerarşisi, birim web servis üzerinden gelen bilgiler ile tanımlanmaktadır. Birim web servisinden dönen değerler ile kullanıcı listesinin yer aldığı web servisteki birim ve birim kodu tanımlamalarının örtüşmesi gerekmektedir.

Oluşturulan her birime karşılık bir kullanıcı grubu oluşturulur. Birim altındaki kullanıcılar bu gruplara yerleştirilerek birim çalışma grupları oluşturulur.

Bu entegrasyon tipinde birim tanımlaması birim listesi ile uyuşmayan kullanıcılar ve manuel yönetilecek işaretli kullanıcılar, api kullanıcıları ve sistem kullanıcıları bu entegrasyona dahil edilemez. Entegrasyonda zorunlu bilgileri eksik olan kullanıcılar entegrasyona dahil edilmez.

Kullanıcı (Çalışma) Grubu Entegrasyonu Gereksinimi:

Ek kullanıcı veya çalışma grubu entegrasyonu yapılması gerektiğinde uygulanabilecek entegrasyon tipidir. Bu entegrasyon, birim entegrasyonu ile beraber çalıştırılabilir.

Bu entegrasyon ile birimler üzerinden oluşan kullanıcı gruplarının oluşmaması sağlanabilir. Tercihen birimler üzerinden oluşacak kullanıcı grupları oluşmadan sadece kullanıcı grubu entegrasyonu ile kullanıcı gruplarını oluşması sağlanabilir.

Bu entegrasyon tipi, grupların oluşturulması için 2 ek web servise ihtiyaç duyar.

- Kullanıcı grubu web servisi: Veri Sözlüğü-Kullanıcı Grupları altındaki model ile kullanıcı grubu bilgilerini dönen bir web servisi olmalıdır.
- Kullanıcı Gruplarındaki Kullanıcı web servisi; Kullanıcı gruplarındaki kullanıcıların listesini ileten bir servis olmalıdır. Data modeli Veri Sözlüğü-Kullanıcı Grupları başlığı altındaki alanlar olmalıdır.

Bu iki web servisinde yer alan kullanıcı grubu adları ve kodlarının uyumlu olması gerekmektedir. Kullanıcı grubu listesinde bulunamayan kodlar için entegrasyon

işlemi yapılmayacaktır.

Manuel yönetilecek işaretli kullanıcılar, api kullanıcıları ve sistem kullanıcıları bu entegrasyona dahil edilemez. Entegrasyonda zorunlu bilgileri eksik olan kullanıcılar entegrasyona dahil edilmez.

4.11. UYGULAMALAR

Bir işletme örnek alınarak 2 temel uygulama hayata geçirilmeye çalışılmıştır. Bunlardan ilki Düzeltici Önleyici Faaliyetler (DÖF) diğeri ise Giriş Kalite Kontrol Uygulamasıdır.

4.11.1. Düzeltici Önleyici Faaliyetler (DÖF) Uygulaması

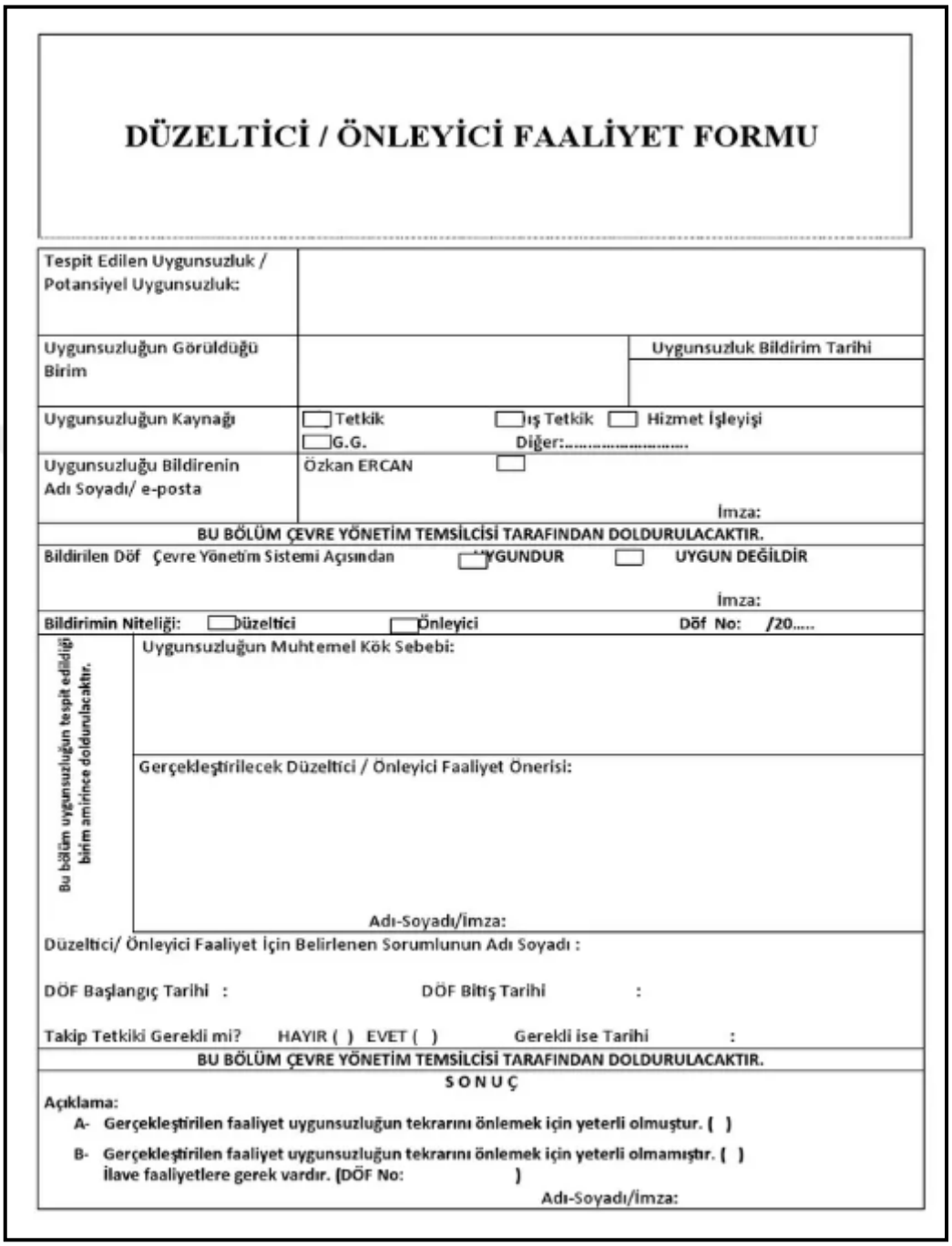
Düzeltici önleyici faaliyet, bir sorunun ya da hatanın meydana gelmesini engellemek veya meydana gelmiş bir sorunu düzeltmek için yapılan faaliyetlerin tümünü ifade eder. Bu faaliyetler genellikle bir organizasyonun kalite yönetim sistemi veya iş süreçleri yönetimi gibi sistemleri içinde yer alır.

Düzeltici önleyici faaliyetler, iki farklı ama birbirine bağlı amaç için uygulanabilir. Düzeltici faaliyetler, mevcut bir sorunu gidermek için yapılan faaliyetlerdir. Sorunlarla ilgili araştırma yaparak, kaynakları belirleyerek ve sorunu gidermek için eylem planları geliştirerek yapılırlar. Örneğin, bir üretim hatasının nedenini belirleyip, hatayı düzeltmek için gerekli işlemleri yapmak, bir düzeltici faaliyet örneği olabilir.

Önleyici faaliyetler ise, gelecekte ortaya çıkması muhtemel sorunların önlenmesi amacıyla yapılan faaliyetlerdir. Gelecekte olası sorunların belirlenmesi, sorunların nedenleri ve kaynakları ile ilgili araştırma yapılması ve bu sorunların ortaya çıkmasını engelleyecek önlemlerin alınması gibi işlemlerle gerçekleştirilirler. Örneğin, bir üretim hattındaki arıza sıklığının azaltılması için, makine bakım programının yeniden gözden geçirilmesi ve yeniden yapılandırılması bir önleyici faaliyet örneği olabilir.

Yapılan uygulama öncesinde Şekil 8'de gösterilen Düzeltici Önleyici Faaliyet Formu şirket içinde manuel form olarak geziyor, herhangi bir onay mekanizmasından geçmiyor, kayıt altına alınamıyor ve bir veritabanı oluşturulamıyordu.

Şekil 8: Manuel DÖF Formu



DÜZELTİCİ / ÖNLEYİCİ FAALİYET FORMU

Tespit Edilen Uygunsuzluk / Potansiyel Uygunsuzluk:		
Uygunsuzluğun Görüldüğü Birim		Uygunsuzluk Bildirim Tarihi
Uygunsuzluğun Kaynağı	☐ Tetkik ☐ Iç Tetkik ☐ Hizmet İşleyişi ☐ G.G. Diğer:.....	
Uygunsuzluğu Bildirenin Adı Soyadı/ e-posta	Özkan ERCAN ☐ İmza:	

BU BÖLÜM ÇEVRE YÖNETİM TEMSİLCİSİ TARAFINDAN DOLDURULACAKTIR.

Bildirilen Döf Çevre Yönetim Sistemi Açısından ☐ UYGUNDUR ☐ UYGUN DEĞİLDİR

İmza:

Bildirimin Niteliği: ☐ Düzeltici ☐ Önleyici Döf No: /20.....

Bu bölüm uygunsuzluğun tespit edildiği birim amirince doldurulacaktır.

Uygunsuzluğun Muhtemel Kök Sebebi:

Gerçekleştirilecek Düzeltici / Önleyici Faaliyet Önerisi:

Adı-Soyadı/İmza:

Düzeltici/ Önleyici Faaliyet İçin Belirlenen Sorumlunun Adı Soyadı :

DÖF Başlangıç Tarihi : DÖF Bitiş Tarihi :

Takip Tetkiki Gerekli mi? HAYIR ( ) EVET ( ) Gerekli ise Tarihi :

BU BÖLÜM ÇEVRE YÖNETİM TEMSİLCİSİ TARAFINDAN DOLDURULACAKTIR.

SONUÇ

Açıklama:

A- Gerçekleştirilen faaliyet uygunsuzluğun tekrarını önlemek için yeterli olmuştur. ( )

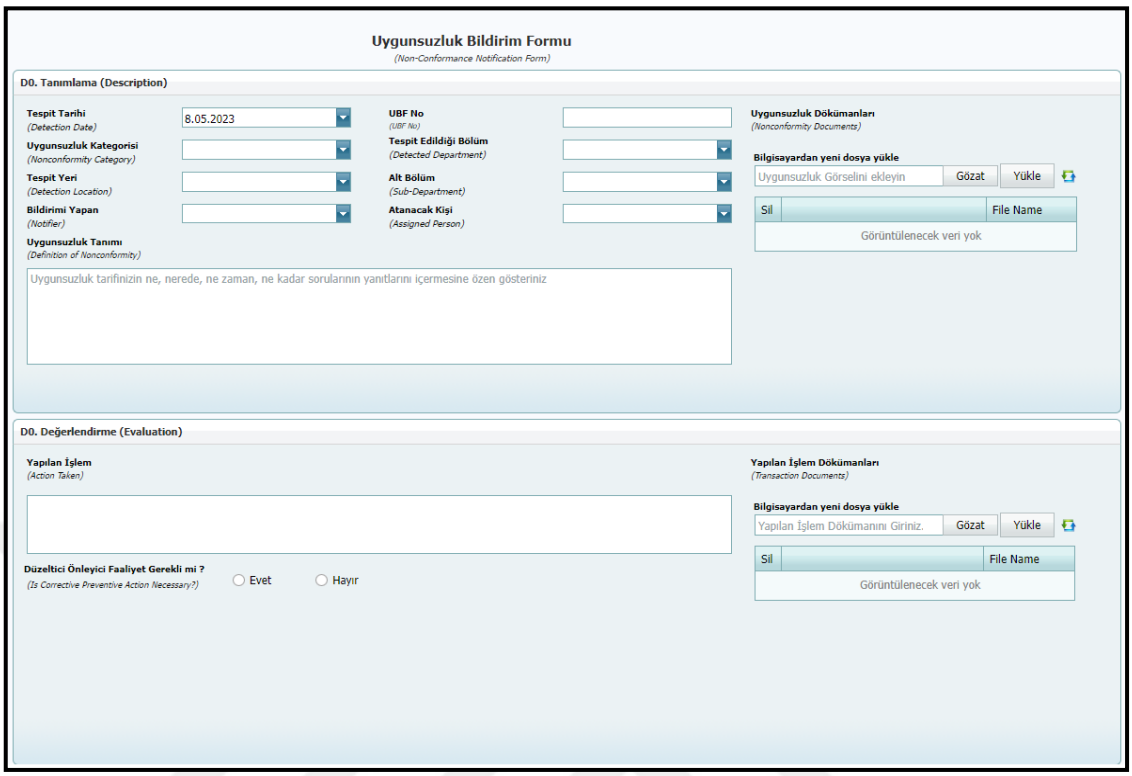
B- Gerçekleştirilen faaliyet uygunsuzluğun tekrarını önlemek için yeterli olmamıştır. ( )
İlave faaliyetlere gerek vardır. (DÖF No:)

Adı-Soyadı/İmza:

Kaynak: Yazar tarafından derlenmiştir.

İşletmede iş süreçlerini denetlemek açısından yapılan faaliyetlerde herhangi bir uygunsuzluk söz konusu ise Şekil 9'daki gibi bir formu doldurulması durumunda uygunsuzluk bildirimi anında yapılabilmektedir.

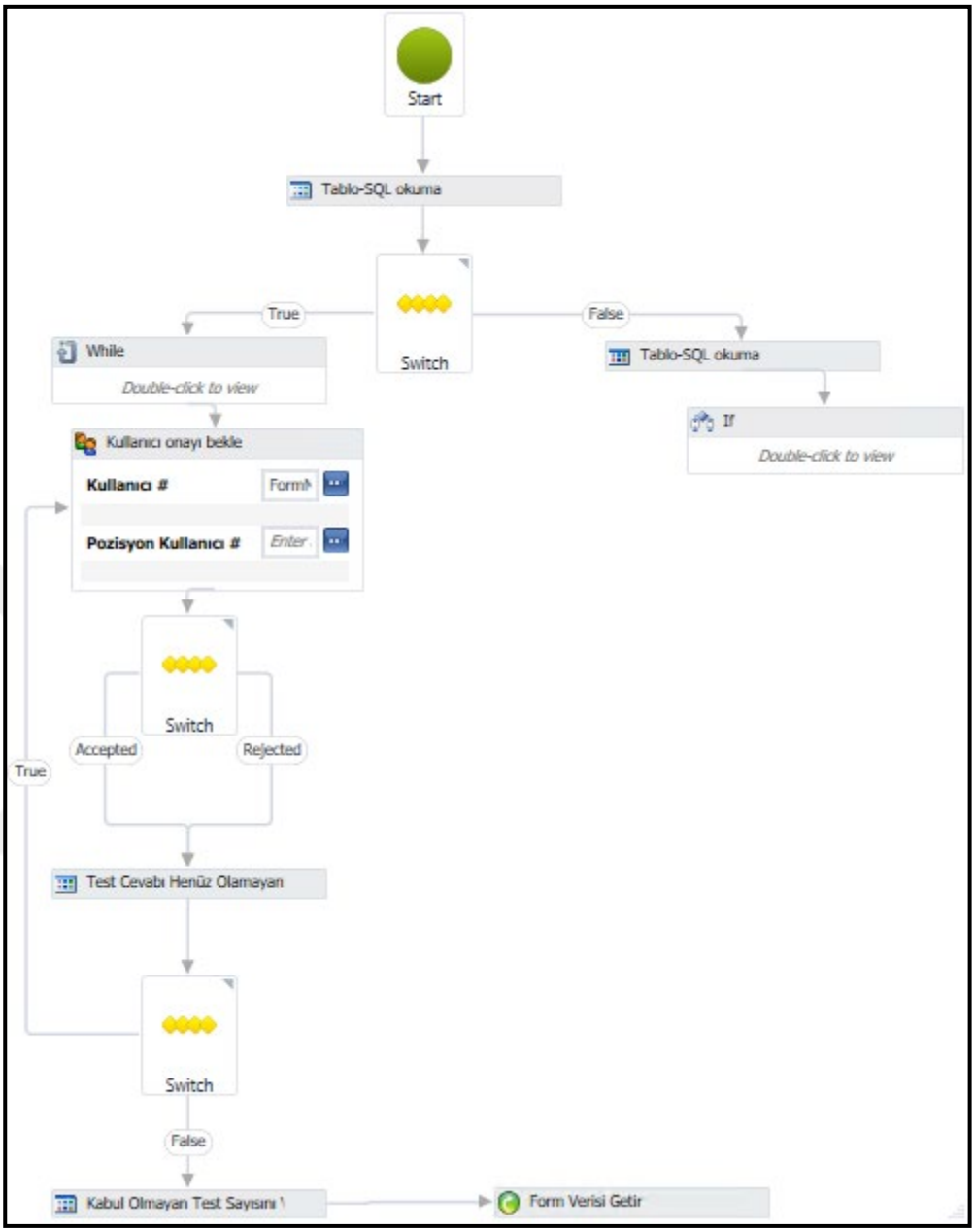
Şekil 9: Dochuman Uygunsuzluk Bildirim Formu



Kaynak: Yazar tarafından derlenmiştir.

Düzeltici önleyici faaliyetlerin akış şeması Şekil 10'da verilmektedir.

Şekil 10: DÖF Formu Akış Şeması



Kaynak: Yazar tarafından derlenmiştir.

Amaç:

Sistem uygunsuzlukları için yapılacak düzeltici faaliyetler için yöntem, yetki ve sorumlulukların tanımlanması.

Kapsam:

İşletmenin tüm bölümleri.

Genel Düzeltici Faaliyet:

Herhangi bir proses adımında karşılaşılan hata, tetkik sonucu bulunan uygunsuzluk ya da müşteriden gelen hata bildirimine istinaden hatanın giderilmesine yönelik yapılan faaliyetlerdir. DÖF Formu ve 8D Formu uygulamaları ile hatanın nedeni, alınacak aksiyonlar, termin ve sorumluları belirlenir.

Proses ve çıkış kalitede karşılaşılan hatalar proses kalite kontrol ve çıkış kalite kontrol formlarına işlenmektedir. Tespit edilen hata, 'Hata Kodları Listesi'ne' göre majör bir hata ise anında DÖF formu başlatılır. Minör hatalar için ise hazırlanan pareto analizi grafiklerinin ilk beş hatasının içinde ise DÖF formu başlatılır.

SGÇ faaliyetlerinde karşılaşılan uygunsuzlukların giderilmesi DÖF formu ile takip edilir. Uygunsuzluklar düzeltici faaliyet takip formunda sadece aksiyon olarak veya DÖF açılarak takip edilebilir. Çözümü kolay olan uygunsuzluklar sadece aksiyon olarak takip edilir, problem çözümü gerektiren uygunsuzluklar ise DÖF açılarak listeden takibi yapılır.

Müşteri Şikâyetleri:

Müşteri tarafında karşılaşılan hatalar için ise müşteri DÖF veya 8D formunu tarafımıza gönderdiği takdir de uygulama talep doğrultusunda başlatılır. Eğer müşteriden sadece bilgi geldiyse Kalite kontrol sorumlusu ile Fabrika Müdürü DÖF formunun kullanılıp kullanılmamasına karar verir.

Müşteri şikâyetleri için gerekli işlemler Müşteri Şikâyetleri Değerlendirme Talimatı'na göre yapılır.

Giriş, Proses ve Çıkış Kalite Kontrol:

Ürün kalitesini etkileyebilecek ya da müşteri şikayetine sebep verecek bir durum ile karşılaşıldığında, üretim esnasında veya çıkış kalite kontrol işlemi sırasında karşılaşılan ya da tekrar eden hatalar için hatayı tespit eden kişitarafından ilgili forma işlenir.

Üretimde tespit edilen hata üretim kaynaklı ise 'Tekrar İşleme Formu' doldurulur. Sonrasında Kalite Sorumlusu tarafından Hata Takip Listesine işlenir.

Tekrar işlem formlarının bir kopyası ÇKK raporuna eklenir ve bir nüshası Kalite Kontrol Sorumlusu tarafından Tekrar İşlen dosyasında saklanır. Tedarikçi kaynaklı ise 'Hatalı Malzeme Raporu' doldurularak giriş kalite kontrol birimine teslim edilir.

Üretimden hatalı malzeme raporu ile gelen malzemeler ve GKK sırasında tespit edilen hatalı malzemeler giriş kalitekontrol tarafından RET alanına alınır. DF Formu ya da 8D Raporlar açılarak tedarikçiye bildirim yapılır.

Tedarikçiden gelen DÖF formları, 8D raporları incelenir. Kalıcı aksiyonlar tespit edilen raporlar kabul edilir. Kök nedene yönelik aksiyon olmayan raporlar için tedarikçiye dönüş yapılarak kök neden aksiyonu adımları / detaylı açıklaması talep edilir.

Tedarikçi Kalite Denetimi sonrasında karşılaşılan uygunsuzluklar için tedarikçiden düzeltici faaliyet gerçekleştirmeleri istenir. Bu uygunsuzlukların giderilmesinden ve gerekli düzeltici faaliyetlerin uygulanmasının takibinden Kalite Güvence Sorumlusu sorumludur.

Faaliyetlerin Etkinliği ve Değerlendirmesi;

Başlatılmış olan her türlü faaliyetin etkinliği takip edilmek zorundadır. Başlatılan bir düzeltici faaliyetin etkinliği aşağıdaki durumlar gözetilerek takip edilir. Etkinlik takibi Faaliyet tamamlandıktan en az 1 ay sonrasında yapılır.

- POKA YOKE Uygulaması,
- JIDOKA Uygulaması,
- 8D Problem Çözme Tekniğinin tüm adımlarının eksiksiz uygulanması,
- PPM seviyesinde düşüş (Bu gözlem için bir dönemden fazla beklenebilir, düşüş oranı yönetici tarafından takdir edilir.)
- Gerektiği durumlarda Cpk ≥ 1,33 ve Cmk ≥ 1,66 (Müşteri isteği ile değişkenlik gösterebilir.)
- Prosedür, talimat, FMEA, kalite planı vb. revizeler gerçekleştirilmiş olması.

Üretimde veya müşteride minimum 1 ay süresince aynı hata ile karşılaşılmaması.

Proses ve Çıkış Kalite Kontrol işlemleri yanı sıra müşteri şikâyetlerine bağlı açılan uygunsuzlukların takibinden Kalite Kontrol Sorumlusu ile Fabrika Müdürü

birlikte sorumludur. Sistem uygunsuzluklarının takibinden Kalite Güvence Sorumlusu sorumludur. Tedarikçi kaynaklı uygunsuzluklarının takibi Kalite Güvence Yöneticisi ve Giriş Kalite Kontrol Elemanları tarafından yapılır. Tedarikçi Uygunsuzlukları takip listesi sürekli olarak güncellenir ve gözden geçirilir.

Tüm düzeltici faaliyetlerin güncel durumu ve etkinliği aylık değerlendirme ve yıllık Yönetimin Gözden Geçirme Toplantılarında görüşülür.

Düzeltici Faaliyet Formu'nda faaliyet etkin ise 'Etkinlik Tarihi' ile 'Faaliyetin Kesin Kapanış Tarihi' aynı kabul edilir.

Etkinliği ispatlanamamış ya da başlatılan faaliyetlerin yetersiz olduğu durumlardaki Düzeltici Faaliyetler sonlandırılmaz.

4.11.2. Giriş Kalite Kontrol Uygulaması

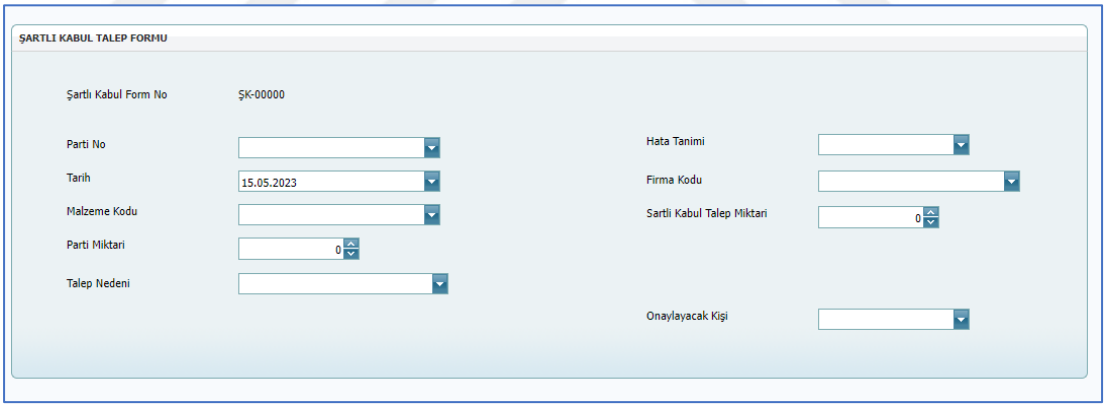
Giriş Kalite Kontrol (GKK) süreçlerinin dijital ortamda takibi ile tek bir forma bağlı kalmadan GKK süreçlerinin tamamı yazılım ile kontrol altına alınmıştır. Şekil 11 - GKK Modülü Alt Form Listesi'nde görüleceği üzere Kalite Kontrol adımları her bir iş süreci hesaplanarak alt formlar halinde patlatılmış ve sürecin her bir adımı akışa bağlanmıştır. Ana sanayiye gönderilen şahit numune malzemelerinde oluşabilecek minor hatalar "GKK Şartlı Kabul Formu" ile bütünleştirilerek RET olabilecek malzemelerin de önüne geçilmiştir.

Şekil 11: GKK Modülü Alt Form Listesi



Kaynak: Yazar tarafından derlenmiştir.

Şekil 12: Şartlı Kabul Talep Formu



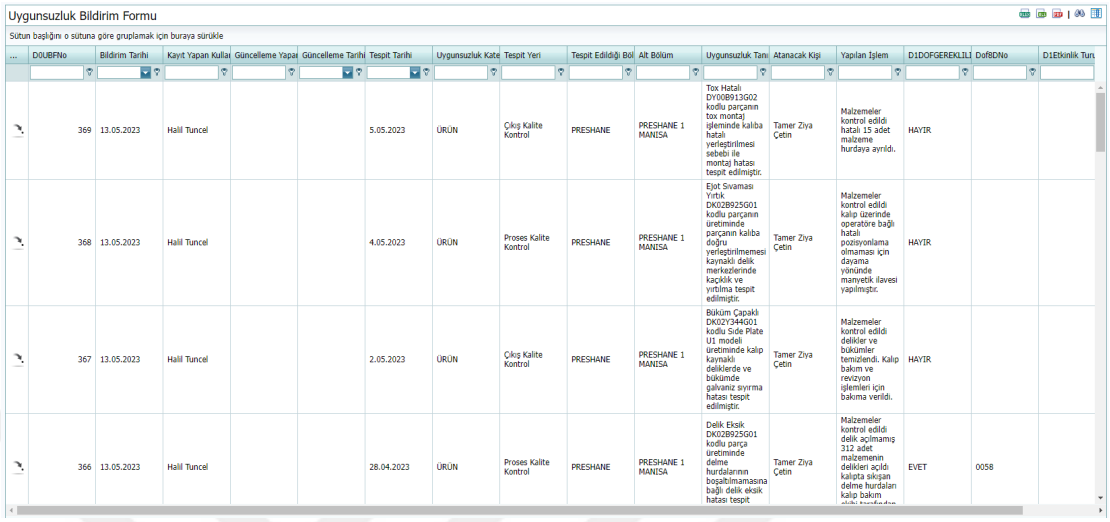
Kaynak: Yazar tarafından derlenmiştir.

4.12. RAPORLAMALAR

DÖF Uygulamasının verileri girildikçe, forma ait veriler ilgili veritabanı tablolarına aşağıdaki gibi yazılır. Kullanıcı bu bilgileri veritabanına bağlanma gerksinimi duymadan uygulamanın tablo görünümüne geçerek tüm verileri görebilir, süzebilir, filtreleyebilir ve dışarı aktarabilir. Bu veriler aşağıdaki rapor örneklerinde verilmiştir.

Şekil 13’de belli bir kullanıcının uygunsuzluk raporları listelemektedir.

Şekil 13: DÖF Raporları 1


Uygunsuzluk Bildirim Formu

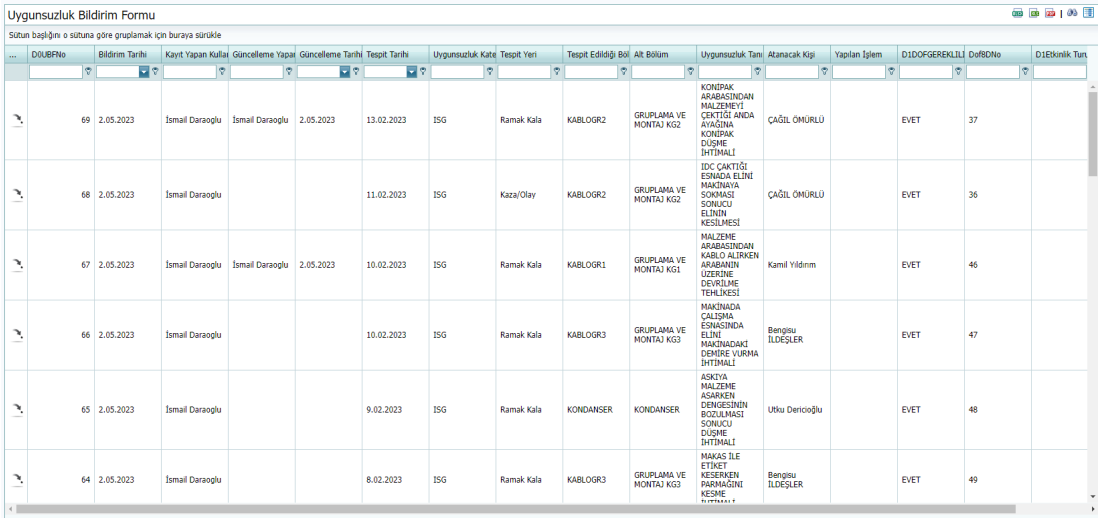
Sütun başlığını o sütuna göre gruplamak için buraya sürükle

...	D0UBFNo	Bildirim Tarihi	Kayıt Yapan Kullar	Güncelleme Yapar	Güncelleme Tarihi	Tespit Tarihi	Uygunsuzluk Kate	Tespit Yeri	Tespit Edildiği Böl	Alt Bölüm	Uygunsuzluk Tanı	Atanacak Kişi	Yapılan İşlem	D1DOFGEREKLİLİ	DofBDNo	D1Etkinlik Turu
	369	13.05.2023	Halil Tuncel			5.05.2023	ÜRÜN	Çıkış Kalite Kontrol	PRESHANE	PRESHANE 1 MANISA	Tox Hatalı DY0DB913G02 kodlu parçanın tox montaj işleminde kaliba hatalı yerleştirilmesi sebebi ile montaj hatası tespit edilmiştir.	Tamer Ziya Çetin	Malzemeler kontrol edildi hatalı 15 adet malzeme hurdaya ayrıldı.	HAYIR		
	368	13.05.2023	Halil Tuncel			4.05.2023	ÜRÜN	Proses Kalite Kontrol	PRESHANE	PRESHANE 1 MANISA	Ejot Sıvaması Yırtık DK02B925G01 kodlu parçanın üretiminde parçanın kaliba doğru yerleştirilmemesi kaynaklı delik merkezlerinde kaçıklık ve yırtılma tespit edilmiştir.	Tamer Ziya Çetin	Malzemeler kontrol edildi kalıp üzerinde operatöre bağlı hatalı pozisyonlama olmaması için dayama yönünde manyetik ilavesi yapılmıştır.	HAYIR		
	367	13.05.2023	Halil Tuncel			2.05.2023	ÜRÜN	Çıkış Kalite Kontrol	PRESHANE	PRESHANE 1 MANISA	Büküm Çapaklı DK02Y344G01 kodlu Side Plate U1 modeli üretiminde kalıp kaynaklı deliklerde ve bükümde galvaniz sıyırma hatası tespit edilmiştir.	Tamer Ziya Çetin	Malzemeler kontrol edildi delikler ve bükümler temizlendi. Kalıp bakım ve revizyon işlemleri için bakıma verildi.	HAYIR		
	366	13.05.2023	Halil Tuncel			28.04.2023	ÜRÜN	Proses Kalite Kontrol	PRESHANE	PRESHANE 1 MANISA	Delik Eksik DK02B925G01 kodlu parça üretiminde delme hurdalarının boşaltılmamasına bağlı delik eksik hatası tespit	Tamer Ziya Çetin	Malzemeler kontrol edildi delik açılmamış 312 adet malzemenin delikleri açıldı kalıpta sıkışan delme hurdaları kalıp bakım	EVET	0058	

Kaynak: Yazar tarafından derlenmiştir.

Şekil 14’de belli bir bildirim tarihinde uygunsuzluk kaydı yapılmış rapor görülmektedir.

Şekil 14: DÖF Raporları 2


Uygunsuzluk Bildirim Formu

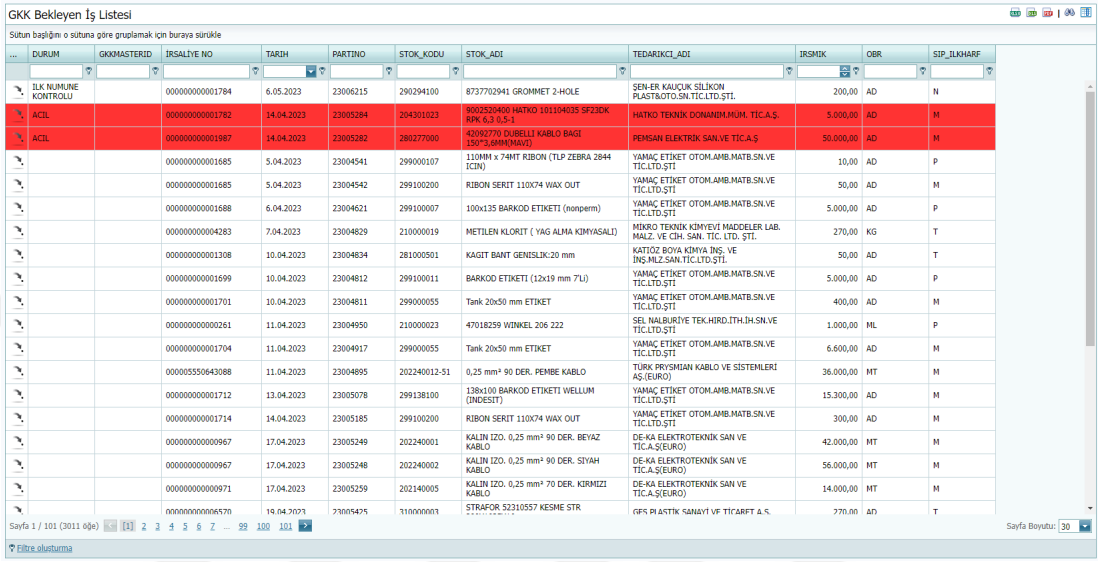
Sütun başlığını o sütuna göre gruplamak için buraya sürükle

...	D0UBFNo	Bildirim Tarihi	Kayıt Yapan Kullar	Güncelleme Yapar	Güncelleme Tarihi	Tespit Tarihi	Uygunsuzluk Kate	Tespit Yeri	Tespit Edildiği Böl	Alt Bölüm	Uygunsuzluk Tanı	Atanacak Kişi	Yapılan İşlem	D1DOFGEREKLİLİ	DofBDNo	D1Etkinlik Turu
	69	2.05.2023	İsmail Daraoglu	İsmail Daraoglu	2.05.2023	13.02.2023	ISG	Ramak Kala	KABLOGR2	GRUPLAMA VE MONTAJ KG2	KONİPAK ARABASINDAN MALZEMEYİ ÇEKTİĞİ ANDA AYAĞINA KONİPAK DÜŞME İHTİMALİ	ÇAĞIL ÖMÜRLÜ		EVET	37	
	68	2.05.2023	İsmail Daraoglu			11.02.2023	ISG	Kaza/Olay	KABLOGR2	GRUPLAMA VE MONTAJ KG2	IDC ÇAKTIĞI ESNADA ELİNİ MAKİNAYA SOKMASI SONUCU ELİNİN KESİLMESİ	ÇAĞIL ÖMÜRLÜ		EVET	36	
	67	2.05.2023	İsmail Daraoglu	İsmail Daraoglu	2.05.2023	10.02.2023	ISG	Ramak Kala	KABLOGR1	GRUPLAMA VE MONTAJ KG1	MALZEME ARABASINDAN KABLO ALIRKEN ARABANIN ÜZERİNE DEVRİLME TEHLİKESİ	Kamil Yıldırım		EVET	46	
	66	2.05.2023	İsmail Daraoglu			10.02.2023	ISG	Ramak Kala	KABLOGR3	GRUPLAMA VE MONTAJ KG3	MAKİNADA ÇALIŞMA ESNASINDA ELİNİ MAKİNADAKİ DEMİRE VURMA İHTİMALİ	Bengisu İLDEŞLER		EVET	47	
	65	2.05.2023	İsmail Daraoglu			9.02.2023	ISG	Ramak Kala	KONDANSER	KONDANSER	ASKIYA MALZEME ASARKEN DENGESİNİN BOZULMASI SONUCU DÜŞME İHTİMALİ	Utku Dericioğlu		EVET	48	
	64	2.05.2023	İsmail Daraoglu			8.02.2023	ISG	Ramak Kala	KABLOGR3	GRUPLAMA VE MONTAJ KG3	MAKAS İLE ETİKET KESERKEN PARMAĞINI KESME İHTİMALİ	Bengisu İLDEŞLER		EVET	49	

Kaynak: Yazar tarafından derlenmiştir.

Şekil 15'de Giriş Kullanıcı Kalite Kontrol bilgilerini listeleyen rapor görülmektedir. Burada özellikle durumu ACİL olan konular kırmızı ile vurguanmaktadır.

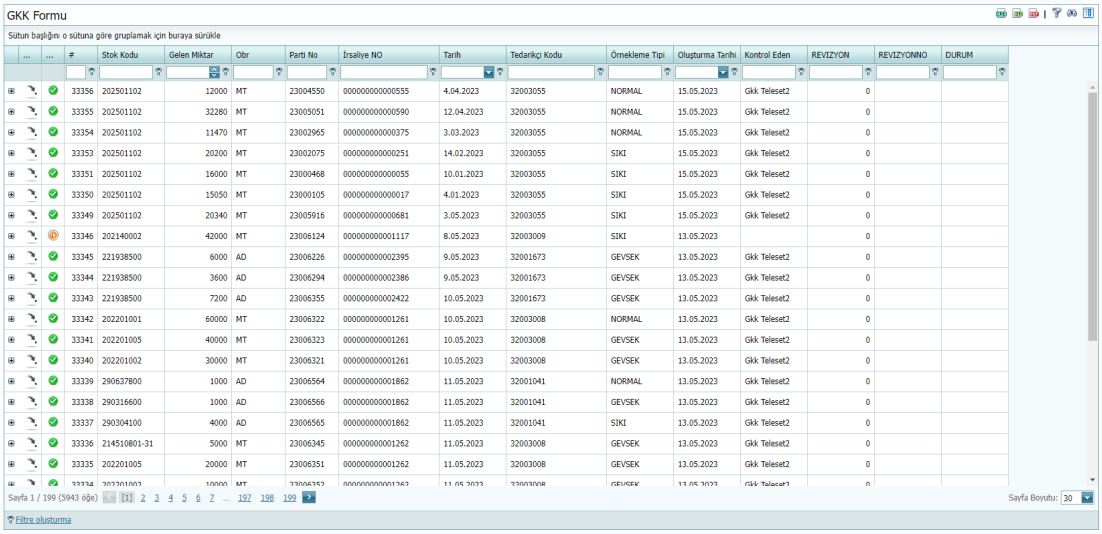
Şekil 15: GKK Raporları 1



Kaynak: Yazar tarafından derlenmiştir.

Şekil 16'da belli bir kontrol biriminin yaptığı bildirimler listelenmektedir.

Şekil 16: GKK Raporları 2



Kaynak: Yazar tarafından derlenmiştir.

SONUÇ

İşletmelerde iş süreçlerinin dijital ortama geçirilmesi, işletmelerin daha verimli, esnek ve rekabetçi olmalarını sağlayan birçok gelişmiş süreci beraberinde getirir. Bunlardan bazıları şunlardır:

- **Otomatikleştirilmiş iş akışı:** İş süreçleri dijital ortama taşındığında, işletmelerin manuel olarak yapılan işlerini otomatikleştirmeleri daha kolay olur. Bu sayede, tekrarlayan görevler otomatikleştirilir ve insan hatalarının riski azaltılır. Örneğin, bir fatura iş akışı otomatik olarak onaylanabilir ve ödeme işlemi otomatik olarak gerçekleştirilebilir.
- **Veri analizi ve raporlama:** Dijital iş süreçleri, işletmelere iş süreçlerine ilişkin verileri daha kolay bir şekilde analiz etme ve raporlama imkanı verir. Bu veriler işletmelere, süreçlerindeki zayıf noktaları belirleme, maliyetleri azaltma ve performanslarını iyileştirme konusunda yardımcı olur.
- **Mobilite ve esneklik:** Dijital iş süreçleri, işletmelerin çalışanlarına mobil cihazlar üzerinden erişim imkanı vererek, iş süreçlerini her yerde ve her zaman yönetmelerine olanak tanır. Bu da işletmelerin daha esnek olmalarını ve daha hızlı kararlar almalarını sağlar.
- **Müşteri memnuniyeti:** Dijital iş süreçleri, işletmelerin müşteri deneyimini geliştirmelerine yardımcı olur. Müşteriler, ürün veya hizmetleri daha hızlı ve kolay bir şekilde satın alabilirler. Ayrıca, müşteri soruları ve şikayetleri daha hızlı bir şekilde ele alınabilir.
- **Güvenlik:** Dijital iş süreçleri, işletmelerin verilerini daha güvenli bir şekilde yönetmelerine olanak tanır. Özellikle, işletmelerin müşteri verileri gibi hassas verileri daha iyi korunabilir. Dijital iş süreçleri, verilerin yedeklenmesi ve korunması için de daha iyi bir çerçeve sunar.

Sonuç olarak örnek işletmede, iş süreçlerinin 2 önemli ayağı olan Düzeltici Önleyici Faaliyetler (DÖF) ve Giriş Kalite Kontrol (GKK) faaliyetleri ele alınmıştır. Yapılan dijitalleştirme çalışmaları öncesinde her iki süreç manuel olarak formlar ile değerlendiriliyor ve tüm süreç elemanları elden ele gezen formlarla süreçleri tamamlamaları bekleniyordu. Fazlasıyla zaman kaybına neden olan bu durum iş süreçlerinin aksamasından kağıt israfına, müşterilerin süreçleri daha yavaş ve zor bir

şekilde takip etmelerine neden oluyordu. Ayrıca müşterilerin soruları ve şikayetleri de gecikmiş bir şekilde ele alınıyordu.

Düzeltici Önleyici Faaliyet (DÖF) ve Giriş Kalite Kontrol (GKK) süreçlerinin dijital ortamda yürütülmesi ile işletmenin daha verimli, hızlı ve etkili bir şekilde faaliyet göstermesi sağlanmıştır. Dijital ortamda DÖF ve GKK iş süreçlerinin takibi ile sağlanan faydalar şu şekilde sıralanmıştır:

1. **Daha Hızlı ve Etkili Faaliyetler:** İşletmenin daha hızlı ve etkili bir şekilde faaliyet göstermelerine yardımcı olmuştur. Bu faaliyetler, otomatikleştirilmiş süreçlerle gerçekleştirildiğinde, manuel hataları önlemiş ve daha hızlı bir şekilde tamamlanmıştır.
2. **Süreçlerin İyileştirilmesi:** İşletmenin süreçlerindeki hataları tespit etmelerine ve iyileştirmeler yapmalarına yardımcı olmuştur. Dijital süreçlerin takip edilmesi, hataların nerede oluştuğunu tespit etmeye yardımcı olmuş ve bu da süreçlerin iyileştirilmesine olanak tanımıştır
3. **Veri Toplama ve Analizi:** İşletmenin verileri toplamalarına ve analiz etmelerine olanak tanınmıştır. Bu da işletmenin süreçlerini daha iyi anlamalarına ve iyileştirmeler yapmalarına yardımcı olmuştur.
4. **İzlenebilirlik:** İşletmenin faaliyetlerini takip etmelerini ve izlemelerini daha kolay hale getirmiştir. İşletme, faaliyetlerin her aşamasını takip edebilir ve faaliyetlerin nerede takıldığını veya sorun yaşandığını tespit edebilir hale gelmiştir.
5. **Maliyet Tasarrufu:** Hem kağıt tasarrufu, hem adam/saat ile işçilik maliyetlerinin azalması hem de yazılım lisans maliyeti açısından bakıldığında dijital ortamda düzeltici ve önleyici faaliyetler, işletmenin ciddi anlamda maliyetlerinin azalmasına yardımcı olmuştur. Böylelikle manuel işlemleri azaltmalarına ve insan hatalarının önlenmesine olanak tanınmıştır.
6. **Kalite Kontrolünün İyileştirilmesi:** Dijital ortamda giriş kalite kontrolü, işletmenin malzemelerin kalitesini kontrol etmek için daha iyi bir yol sunmuştur. Bu da işletmenin kalite kontrolünü daha iyi anlamalarına ve iyileştirmeler yapmalarına yardımcı olmuştur.

KAYNAKÇA

Adesola, S. ve Baine, T. (2005). Developing and evaluating a methodology for business process improvement. Business Process Management Journal, 11(1), 37-46.

Antony, J. ve Gupta, S. (2018). Top ten reasons for process improvement project failures. International Journal of Lean Six Sigma.

Aysolmaz, B., Coşkunçay, A., Demirörs, O. ve Yıldız, A. (2011). Kamuda iş süreçleri modelleme: Gereği ve yararları. 5. Ulusal Yazılım Mühendisliği Sempozyumu UYMS'11.

Berber, G. (2017). Bir gıda işletmesinde süreç iyileştirme uygulaması: Dondurma fabrikası örneği. (Yayınlanmamış yüksek lisans tezi). İnönü Üniversitesi Sosyal Bilimler Enstitüsü.

Çevik, O. ve Aran, G. (2009). Kalite iyileştirme sürecinde hata türü etkileri analizi (FMEA) ve piston üretiminde bir uygulama. Selçuk Üniversitesi Sosyal ve Ekonomik Araştırmalar Dergisi, 8(16), 241-265.

Gökçen, H. (2005). *Yönetim Bilgi Sistemleri: Analiz ve Tasarım Perspektifi.* Ankara: EPİ Yayıncılık

Kayışkan, D. (2016). Bir gıda işletmesinde süreç yönetimi ve süreç iyileştirmesi üzerine uygulama. (Yayınlanmamış yüksek lisans tezi). Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü.

Kılıç, Ş. ve Aydınlı, C. (2015). Sağlık kurumlarında süreç yönetimi uygulamaları. İşletme Araştırmaları Dergisi, 7(3), 143-172.

Logo Yazılım (2023) Logo Flow - İş Akış Yönetimi. https://www.logo.com.tr/urun/logo-flow?waw_keyword=İş süreçleri. Erişim: 15 Mayıs 2023

Netle Yazılım (2023) *Netle Yazılım - DocHuman Tasarım Manifestosu* https://www.netle.com.tr/netle-yazilim-dochuman-tasarim-manifestosu-ver-2-6. Erişim: 15 Mayıs 2023,

Sarıgül, S. S. ve Oralhan, B. (2016). EFQM mükemmellik modeli ve mobilya sektörüne yönelik bir uygulama. Sosyal Ekonomik Araştırmalar Dergisi, 16(31).

Selimoğlu, N. (2005). *Süreç Yönetimi Ve Süreç İyileştirmede Bilgi Yönetiminin Rolü Ve Uygulamaya İlişkin Bir Araştırma.* İstanbul Üniversitesi Sosyal Bilimler Enstitüsü Yayınlanmamş Yüksek Lisans Tezi.

Tecim, V. (2023). Sistem Geliştirme Yaşam Döngüsü. https://vahaptecim.com.tr/sistem-gelistirme-yasam-dongusu/ (21.03.2023)

Türkan, T. ve Görener A. (2017). Süreç iyileştirme: Vasıflı çelik üretim sektöründe bir uygulama. Optimum Ekonomi ve Yönetim Bilimleri Dergisi, 4(2).

Yıldız, D. (2021) İş Süreçlerinin Modellenmesi, İyileştirilmesi, Performansının Ölçülmesi Ve Yönetilmesinin İşletme Verimliliğine Katkısı: Bir Uygulama, *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 59, Mayıs-Ağusto 2021, 91-118.