



**T.C.
HALIÇ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI**

**CMMI REHBERLİĞİNDE PRINCE2 İLE SERVİS ODAKLI
MİMARİ TABANLI UYGULAMA GELİŞTİRME**

YÜKSEK LİSANS TEZİ

**Hazırlayan
Umut ŞİMŞEK**

**Danışmanı
Prof. Dr. Halûk GÜMÜŞKAYA**

İstanbul – Ocak 2011

İÇİNDEKİLER

	Sayfa No.
KISALTMALAR LİSTESİ	IV
ŞEKİLLER LİSTESİ.....	V
TABLOLAR LİSTESİ.....	VI
1. GİRİŞ.....	1
1.1. Tezin Amacı ve Kullanılan Yöntemler	1
1.2. Tezin Yapısı	2
2. İLGİLİ ÇALIŞMALAR.....	3
2.1. Giriş	3
2.2. Akademik Çalışmalar.....	3
3. İLGİLİ KONULAR.....	7
3.1. Servis Odaklı Mimari: SOA.....	7
3.1.1. Servis Odaklı Mimari Öncesi	7
3.2. PRINCE2	10
3.2.1. PRINCE2 Yapısı.....	11
3.2.2. Prensipler.....	11
3.2.3. Temalar.....	12
3.2.4. Süreçler.....	13
3.3. CMMI	15
3.3.1. CMMI Modelindeki Sürekli ve Aşamalı Gösterimler	15
3.3.2. CMMI'nın Amaçları.....	19
4. PRINCE2'NUN SOA UYGULAMASINDA KULLANILMASI.....	21
4.1. SOA Uygulaması Programı.....	21
4.2. PRINCE2 Prensiplerinin SOA Prensipleri İle Birlikte Kullanılması	23
4.3. PRINCE2 Temalarının SOA İle Birlikte Kullanılması	24
4.4. PRINCE2 Süreçlerinin SOA Projelerinde Kullanılması	25
4.5. PRINCE2'nun SOA Projelerine Uyacak Şekilde Biçimlendirilmesi	26
5. CMMI REHBERLİĞİNDE PRINCE2 İLE SOA UYGULAMASI	27
5.1. Yönetilirlik (Managed) Olgunluk Seviyesi	27
5.2. Tanımlanmışlık (Defined) Olgunluk Seviyesi	28
5.3. Nicel Yönetilebilir (Quantitatively Managed) Olgunluk Seviyesi	30
5.4. Optimize Olgunluk Seviyesi	31
6. ÖNERİLEN YAKLAŞIMLARIN BİR PROJEDE GÖSTERİLMESİ.....	32

6.1. SOA Bařlangıç Projesi.....	32
6.1.1. SOA Rol ve Sorumluluklarının Belirlenmesi	33
6.1.2. SOA Eđitimi	34
6.1.3. SOA Prensiplerinin Oluřturulması	36
6.1.4. SOA Teknoloji Beklentilerinin Oluřturulması	37
6.1.5. Mevcut Sistemlerin Analizi	38
6.2. SOA Teknolojik Altyapı Projesi	38
6.2.1. NetBeans IDE 6.x	39
6.2.2. Bileřik Uygulamalar ve JBI.....	40
6.2.3. Servis Motorları.....	40
6.2.4. Bađlama Bileřenleri.....	40
6.2.5. BPEL Tasarımcısı.....	41
6.2.6. WSDL Editörü.....	41
6.2.7. XML řema Tasarımcısı.....	41
6.3. SOA Takip ve Geliřtirme Projesi.....	41
6.4. İř Çıktısı Odaklı SOA Projesi	42
6.4.1. Gerekler Analizi ve Servis Odaklı Modelleme.....	42
6.4.2. Adım Adım Servis Modelleme.....	46
6.4.3. Servis Odaklı Mimaride Deđiřiklik Yönetimi	48
6.5. Rezervasyon İř Sürecinin NetBeans Ortamında Oluřturulması	53
7. SONUÇLAR.....	59
8. KAYNAKLAR.....	60

KISALTMALAR LİSTESİ

BPEL: Business Process Execution Language
CCTA: Central Computer and Telecommunications Agency
CMMI : Capability Maturity Model Integration
CORBA: Common Object Request Broker Architecture
DCOM: Distributed Component Object Model
ESB: Enterprise Service Bus
HTTP: Hypertext Transfer Protocol
IDE: Tümüleşik Geliştirme Ortamı
IEC: International Electrotechnical Commission
IEEE: Institute of Electrical and Electronics Engineers
ISO: International Organization for Standardization
JB1: Java Business Integration
NDIA: The National Defense Industrial Association
OGC: Office of Government Commerce
PRINCE2 : PRojects IN Controlled Environments
RPC: Remote Procedure Call
SEI: Software Engineering Institute
SG: Specific Goal
SOA : Service Oriented Architecture
SOAP: Simple Object Access Protocol
SP1: Specific Practice
USB: Universal Serial Bus
XML: Extensible Markup Language

ŞEKİLLER LİSTESİ

	Sayfa No.
Şekil 3.1. SOA ’in temel öğeleri.	9
Şekil 3.2. PRINCE2 metodunun yapısı.	11
Şekil 3.3. PRINCE2 süreç modeli.	14
Şekil 3.4. Bir CMMI modelinin iki farklı gösterimi.	17
Şekil 3.5. CMMI kavramı.	20
Şekil 4.1. OGC’ye göre Proje ve Program arasındaki farklar.	21
Şekil 4.2. SOA uygulama programı katmanları.	22
Şekil 6.1. SOA başlangıç projesinde yer alabilecek faaliyetlerin bir Gantt şeması ile gösterilmesi.	32
Şekil 6.2. SOA analiz süreci.	44
Şekil 6.3. Servis modelleme süreci.	46
Şekil 6.4. Değişiklik yönetimi süreci.	49
Şekil 6.5. Netbeans 6.x entegre geliştirme ortamında EJB Projesi oluşturmak.	54
Şekil 6.6. Web servisi oluşturma.	54
Şekil 6.7. Web servisi içinde bir sınıfının oluşturulması.	55
Şekil 6.8. Netbeans 6.x entegre geliştirme ortamında BPEL modülü oluşturma.	55
Şekil 6.9. Netbeans 6.x entegre geliştirme ortamını.	56
Şekil 6.10. Assign ögesinin içinde atamaların düzenlenmesi.	56
Şekil 6.11. Netbeans ortamında bileşik uygulamanın oluşturulması	57
Şekil 6.12. Netbeans üzerinde JBI Modülü	57
Şekil 6.13. Rezervasyon servisinin BPEL tasarımcısında görünümü.	58

TABLÖLAR LİSTESİ

	Sayfa No.
Tablo 3.1. CMMI'ın gelişimindeki önemli kilometre taşları.	16
Tablo 3.2. CMMI Süreç Alanları, Kategoriler ve Olgunluk Seviyeleri.	18
Tablo 4.1. PRINCE2 metodolojisinin SOA uygulamasına katkıları.	26
Tablo 5.1. CMMI yönetilirlik süreç alanlarının PRINCE2'nun karşılama seviyesi	28
Tablo 5.3. CMMI-DEV'in SOA uygulamasına pratik katkıları.	31
Tablo 6.1. Bir SOA program ve proje organizasyonu örneği.	34
Tablo 6.2. SOA başlangıç projesinde yer alacak çalışma grupları.	34
Tablo 6.3. SOA Uygulama projelerinde değişiklik süreci rol ve sorumlulukları.	50
Tablo 6.4. Değişiklik süreci rol ve sorumluluklarının gerçek kişilere atanması	52

GENEL BİLGİLER

Adı ve Soyadı : Umut Şimşek
Anabilim Dalı : Mühendislik
Programı : Bilgisayar Mühendisliği
Tez Danışmanı : Prof. Dr. Halûk Gümüşkaya
Tez Türü ve Tarihi : Yüksek Lisans – Ocak 2011

CMMI REHBERLİĞİNDE PRINCE2 İLE SERVİS ODAKLI MİMARİ TABANLI UYGULAMA GELİŞTİRME

ÖZET

Bu tezde, kurumsal ortamlarda Servis Odaklı Mimari (Service Oriented Architecture - SOA) tabanlı bir uygulama geliştirilirken, PRINCE2 (PRojects IN Controlled Environments) proje yönetimi metodolojisi ve CMMI (Capability Maturity Model Integration) süreç geliştirme modeli analiz edilmiş ve SOA uygulamasını daha başarılı kılmak için bu yaklaşımların ne tür katkılar sağlayabileceği araştırılmıştır. Bu araştırma sonucunda elde edilen verilere göre bir yol haritası oluşturulmuş ve bu yol haritasında gerçekleştirilecek çalışmalar tarif edilmiştir. Tezde öne sürülen öneriler geliştirilen örnek bir SOA uygulaması ile desteklenmiştir.

Tezin ana çalışmalarından ilki olarak, PRINCE2 prensiplerinin SOA uygulamalarına ne tür katkılar sağlayabileceği üzerinde durulmuştur. Bu metodoloji kullanılarak bir uygulama planı çıkarılmıştır.

Tezde yapılan ikinci çalışma, CMMI modelinin üzerinde durduğu konular genel olarak incelenerek SOA tabanlı uygulama geliştirmeyi iyileştirmede bunlardan nasıl faydalanabileceği üzerinedir.

Tezdeki üçüncü ve son çalışma, analizler sonunda ortaya çıkan modelin gerçekleştirildiği bölümdür. Örnek SOA'nın stratejik bir karar olarak kavramsal, kuramsal ve ardından teknolojik olarak altyapısı hazırlandıktan sonra, bu ortamda bir iş sürecini gerçekleştiren SOA uygulaması geliştirilmiştir.

Bu tez çalışmasında daha önce yapılmış araştırma çalışmalardan farklı olarak, SOA tabanlı uygulama geliştirmede PRINCE2 ve CMMI birlikte kullanılmıştır.

Anahtar Kelimeler: Servis Odaklı Mimari, Yazılım Geliştirme, PRINCE2 Proje Yönetimi Metodolojisi, CMMI Süreç İyileştirme Modeli, SOA

GENERAL INFORMATION

Name and Surname : Umut Şimşek
Field : Engineering
Program : Computer Engineering
Supervisor : Prof. Dr. Halûk Gümüşkaya
Degree and Date : Master – January 2011

DEVELOPING SERVICE ORIENTED ARCHITECTURE BASED APPLICATIONS USING PRINCE2 UNDER CMMI GUIDANCE

ABSTRACT

In this thesis, PRINCE2 (Projects IN Controlled Environments) project management methodology and CMMI (Capability Maturity Model Integration) process improvement model are analyzed and evaluated with the search of any contributions for more successful implementations of SOA (Service Oriented Architecture) based applications in enterprise environments. A roadmap was created using the findings of our research and the tasks which are parts of the roadmap are described and supported with a sample application.

As a first study of our thesis, the focus point is possible contributions of PRINCE2 method to improve SOA based application development. An implementation plan is created using this study.

Then, the emphasis points of CMMI model are analyzed generally in order to improve SOA based application development as second part of our study.

Finally, findings of our research are implemented in the third and last part of the thesis. After the creation of a SOA environment considering strategic decisional, conceptual, principal and technologic infrastructure aspects of SOA, a business process is realized as an example.

The main contribution of the thesis is the combination and adaptation of PRINCE2 and CMMI methodologies in order to improve the success of SOA based application development projects.

Key Words: Service Oriented Architecture, Software Development, PRINCE2 Project Management Methodology, CMMI Process Improvement Model, SOA

1. GİRİŞ

Son yıllarda bilgi teknolojilerinde lider ve yazılım üreticisi birçok büyük firma, yazılım geliştirme felsefelerinde ve geliştirdikleri uygulamalarında Servis Odaklı Mimari (Service Oriented Architecture - SOA) [1, 2] tabanlı bir mimariyi kullanmakta ve bu mimari ile yeni ürünler geliştirmektedir. Bunun yanında bu mimari yaklaşımın daha iyi hale getirilmesi ya da alternatifleri üzerinde durulması konusunda da değişik görüşler söz konusudur.

Yazılım mühendisliğinin kritik başarı faktörlerinden sayılan yönetim bilimleri ve en iyi pratiklerin gün geçtikçe önemi daha da artmaktadır. Bu konuda iki önemli standart olan PRINCE2 (PRojects IN Controlled Environments) [3, 4, 5] ve CMMI (Capability Maturity Model Integration) [6, 7] dünyada çok yaygın kabul görmekte ve çok değişik sektörlerde uygulanmaya çalışılmaktadır. Bu standartların alternatiflerini ve farklı yaklaşımları tercih edenlerin sayısı da az değildir.

Bu tezde SOA yaklaşımı ile bir kurumsal yazılım sistemi tasarımı teknik ve metodolojik olarak incelenmiştir. Bu alanda kullanılan teknolojiler ve yöntemler üzerinde durulmuş ve PRINCE2 ve CMMI gibi proje yönetimi ve yazılım geliştirme standartlarının kullanılabilirliği araştırılmıştır. Bu tezde tüm bu konular ele alınmış ve birlikte kullanımlarının nasıl olabileceği konusunda bir araştırma yapılmış ve bunların pratiğe geçirilmesi konusu üzerinde çalışılmıştır.

1.1. Tezin Amacı ve Kullanılan Yöntemler

Bu tezin amacı SOA yaklaşımı ile bir kurumsal yazılım sistemi tasarlamak ve bu projeyi tasarlarken konu ile ilgili teknolojileri incelemek ve standart haline gelmiş CMMI ve PRINCE2 gibi kurum süreç modellerinin ve metodolojilerinin SOA yaklaşımı ile birlikte kullanımını incelemek ve gerekli adaptasyon ve yeni yöntemler üretmek ve bunu uygulamaktır.

Bir SOA uygulama ortamının oluşturulması ve yazılım proje geliştirme aşamalarında PRINCE2 ve CMMI metodolojilerinin içerdiği ya da önerdiği kavramlar, araçlar ve süreçler incelenmiş ve bunların SOA uygulamalarına fayda katacak şekilde

uyarlanarak bir SOA uygulama programı oluşturulmuştur. Tezde önerilen yaklaşımları göstermek için süreç ve geliştirme örnekleri verilmiştir.

1.2. Tezin Yapısı

Bu tez sekiz bölümden oluşmaktadır. İlk bölümde giriş yapıldıktan sonra, ikinci bölümde, ilgili çalışmalar incelenerek özetlenmiştir. Üçüncü bölümde servis odaklı mimari genel olarak incelenmiş, bu mimarinin temel öğeleri ve önemli özellikleri üzerinde durulmuştur.

Dördüncü bölümde, PRINCE2 metodolojisi incelenmiştir. Bu metodolojinin prensipleri, temaları ve süreçleri incelenerek, bunların servis odaklı mimaride nasıl kullanılacağı üzerine görüşler sunulmuştur. Bir SOA gerçekleştirmesinin kurumsal bir ortamda PRINCE2 yardımı ile nasıl yapılabileceği üzerine öneriler verilmiştir.

Beşinci bölümde, CMMI kurumsal süreç iyileştirme modeli incelenmiştir. Bu modelin çeşitli olgunluk seviyelerinde yer alan süreçleri incelenerek, bunların servis odaklı mimari projelerine nasıl rehberlik edebileceği üzerinde görüşler aktarılmıştır. Bir SOA gerçekleştirmesinin kurumsal bir ortamda CMMI desteği ile nasıl yapılabileceği açıklanmıştır.

Altıncı bölümde, dördüncü ve beşinci bölümde analiz edilen ve belirlenen süreç ve modellerin örnek süreçler rehberliğinde bir prototip uygulamayı hayata geçirmesindeki önemli adımlar sunulmaktadır. Yol haritasında yer alan başlangıç, altyapı, takip ve geliştirme projelerinde bir iş sürecini SOA mimarisinde uygulayan örnek uygulamanın önemli noktaları üzerinde durularak anlatılmaktadır.

Yedinci bölümde tez çalışmasında elde edilen sonuçlar özetlenmekte ve sekizinci bölümde tezin akademik ve diğer kaynakları verilmektedir.

2. İLGİLİ ÇALIŞMALAR

2.1. Giriş

SOA ile ilgili çalışmalar, proje yönetimi ve yazılım geliştirme metodolojileri olmak üzere iki ana grupta incelenecektir. SOA proje yönetimi, bu tarz bir projenin gerçekleştirilmesini başarılı kılacak katkıyı sağlarken, yazılım geliştirme ise özellikle yazılım geliştirmeye yönelik çalışmalarda bizi destekler.

2.2. Akademik Çalışmalar

Bir SOA projesinin başarılı bir şekilde gerçekleştirilmesi için bazı teknikleri ve temel prensipleri öneren çalışmalar mevcuttur. Bunlar Servis Odaklı Mimari Yönetimi kavramı altında toplanmaktadır [1, 8, 9]. Servis Odaklı Mimari Yönetimi, çeşitli seviyelerdeki SOA kararlarının doğru yönetilmesini sağlayan faaliyetlerdir. Bir organizasyonda SOA çalışmalarının doğru şekilde yapılmasına katkı sağlayacak faaliyetlerin bazıları aşağıda sıralandığı gibidir [2]:

- SOA çalışmasının hedef ve stratejilerinin belirlenmesi
- Organizasyon içinde gerekli alt yapının, süreçlerin ve yönetim mekanizmasının belirlenmesi,
- Çalışmaların gelişmesini gösterecek bir takım ölçüm setlerinin geliştirilmesi,

Yukarıda sıralananlar, yapılan çalışmanın kapsam ve amacına bağlı olarak çok daha geniş, detaylı veya zaman içinde gelişim gösteren bir yapıya sahip olabilir.

Bir SOA projesinin yönetimi sırasında kapsam, maliyet ve harcanacak çaba tahminleri ile ilgili çalışmalardan olan [8] SMAT-AUS metodolojik altyapısını kullanarak bir kısım yönetsel aktivitelerin SOA projelerinde kullanılmasını önermektedir. Teknik, sosyal, kültürel ve kurumsal açıların yanında organizasyonun tecrübesi ile birlikte olgunluk seviyesini de inceleyerek, birbirinden farklı SOA projelerinin tiplerini de göz önüne alarak kapsam, maliyet ve çaba kestirilmeye çalışılır.

Özellikle geniş çaplı SOA tabanlı bir projenin System Dynamics modelleme metodolojisi ile daha başarılı olduğunu savunan diğer bir çalışmada [9] SOA uygulamalarının bu metodolojiden faydalanması hedeflenmiştir. System Dynamics iş dünyası hedefleri ve sosyal sistemler gibi karmaşık geri beslemeli ortamları inceleyen ve

yöneten bir metodolojidir. Bu çalışmada System Dynamics modellemeye dayanan çeşitli SOA senaryo analizleri yapılmıştır. Bu farklı çalışmalar SOA'nın değişik faktörlerin toplam etkisinin nasıl bir etkiye sahip olduğunu ve bunların uzun vadede ne tür sonuçlar vereceğini ele almaktadır. SOA'nın iki önemli özelliği olan hızlı geliştirme ve birçok çözüm dokuları ve geliştirmelerde yüksek derecede yeniden-kullanılabilirlik olarak belirlenmiştir.

Bu konudaki diğer bir araştırmada [10] Amerika Birleşik Devletlerinde Carnegie Mellon Üniversitesi'ne bağlı Yazılım Mühendisliği Enstitüsü (Software Engineering Institute - SEI) tarafından geliştirilen, genel amaçlı ve geniş kapsamlı bir şekilde yazılım geliştirme ortamlarının metodolojik ve stratejik olgunluğunu belirleyen metodolojisi CMMI ile Office of Government Commerce (OGC) tarafından BT projelerini yönetmek üzere geliştirilen PRINCE2 arasında ilişkiler kurularak, her iki modelin planlama süreçleri açısından bir karşılaştırması yapılmıştır. Bu çalışmada PRINCE2 (Planlama Süreci) ile CMMI 1.1 Proje Planlama karşılaştırılmıştır. CMMI-sürekli modelinde, proje yönetimi süreç alanları, proje planlama, takip etme ve kontrol ile ilgili proje yönetimi faaliyetlerini kapsar. Proje planlama, CMMI'nin sekiz proje yönetimi süreç alanından biridir. Bu makalede yapılan analizin sonucu, PRINCE2 sertifikalı kurumlar her ne kadar CMMI süreç alanlarının tamamını gerektiği gibi karşılayamasa da CMMI-sürekli modelinin büyük bir çoğunluğunu karşılayacağını göstermiştir. PRINCE2 sertifikalı organizasyonların CMMI sertifikası almada genel olarak belirgin bir avantaj ve daha az zorluk yaşayacakları sonucuna ulaşılmıştır.

SOA kurumsal uyarlaması hayat döngüsü şeklinde yapılan diğer bir çalışmada [11], bir proje kapsamında giriş, yol haritası geliştirme, planın uygulanması ve gözden geçirme ve yol haritasının güncellenmesi kilometre taşları ile belirlenmiştir. Birinci aşamada proje hedeflerinin belirlenmesi, takımın oluşturulması, zaman çizelgelerinin ve proje çıktılarının belirlenmesi önerilmektedir. Yol haritası aşamasında, iş tarafı, uygulama, teknoloji ve veri prensipleri, altyapı, iş tarafı referans mimarisi oluşturmaktadır. Uygulama aşamasında, portföy, proje yönetimi ve benzeri temel çalışmalar yer almaktadır. Bu bölümde gerçekleşen gerçek duruma göre gözden geçirme ve yol haritası güncellemeleri yapılmaktadır.

PRINCE2 proje yönetimi metodolojisinin Model Sürümlü Mimari (MDA - Model Driven Architecture) yazılım geliştirme yaklaşımında nasıl kullanılabileceği kaynak 12’te sunulmaktadır. Bu yolla proje yönetim süreçleri, yazılım geliştirme süreçleri ile bir araya getirilerek, projenin daha uygun bir yol izlemesine kılavuzluk edilmesi söz konusudur. Bu çalışmada, kontrollü bir ortamda BT projelerinde MDA için bir çözüm sunulmaktadır. Yazılım mimarisinin kısıtlamalarından ve yönetim gereksinimlerinden dolayı, yazılım geliştirmek amacıyla uzun süreli MDA esas alınmakta ise, PRINCE2 gibi bir iskeletin kullanılması tavsiye edilmektedir. Bu yolla, proje yönetim süreçlerinin, yazılım geliştirme süreçleri ile birlikte kullanımı projeye daha doğru kılavuzluk edecektir.

SOA’nın günümüz hızlı değişen iş gereksinimlerini karşılayan yönü açısından, çevik yazılım geliştirmenin (agile software development) en uygun bir yaklaşım olduğunu öne süren diğer bir çalışmada [13], çevik yazılım geliştirmenin temelleri incelenerek, SOA tabanlı sistemlere uygunluğu değerlendirilmiştir. Bu çalışma ışığında varılan sonuca göre iki tip servis vardır. Bunlar ince bölümlenmiş servisler ve kabaca bölümlenmiş servisler. Uygulama içi servisler için çevik geliştirme yaklaşımı çalışacaktır. Çünkü bunların arayüzlerini değiştirmek sadece uygulamanın kendisine etki edecektir. Uygulamalar arasında kullanılacak servisler için servisi gerçekleyen uygulamadan arayüz bir gereksinim olarak istenmektedir ki bu uygulama kendisi değişir ya da değişmeyebilir. Yine de çevik geliştirme projeleri arayüz değişikliklerine hazırlıklı olmalıdır. Bir SOA kavramı olan sürekli geliştirme, servis kompozisyonu ve servis arayüzlerinin modifikasyonu anlamını da çağrıştırmaktadır. SOA’da bu tip bir çeviklik, uygulama projelerinde de daha fazla çeviklik gerektirecektir. Buna göre bu tarz bir çalışmanın tam olarak birleştirilmesinin zor olduğu ifade edilmektedir.

Kaynak 14’te servis odaklı modelleme ve mimari kavramının önemli noktaları tartışılırken, ideal SOA ortamı oluşturulması için gerekli analiz ve tasarım temel faaliyetlerinin neler olacağı sunulmaktadır. Bu çalışmada SOA kavramsal meta-modeli UML diyagramları şeklinde oluşturulmuştur. Mimari sitil ve prensipler üzerinde durulmuştur. Mimari bir SOA şablonu oluşturulmaya çalışılmıştır. SOA kavramı katmanlar şeklinde modellenerek bunların tanımlanması yapılmıştır. Bu modelleme için

gerekli faaliyetler belirlenmiştir. Son olarak servis odaklı modelleme ve mimarisini meydana getiren bir süreç oluşturulmuştur.

Kaynak 15’de deneysel bir metodoloji, servis odaklı tasarım ve geliştirme için tanımlanmıştır. Bu metodoloji, tasarım ve geliştirme temellerini tanımlayan bir çabayı yansıtır ve web servisleri ve iş süreçleri aynı seviyede uygulanır. Metodoloji aşağıdan-yukarı, yukarıdan-aşağı ve melez gibi değişik geliştirme modellerini dikkate alır ve dayandığı referans modellere vurgu yaparak bir kaç servis gerçekleştirme senaryosunu göz önüne alır. Servis ve süreç tasarımı sırasında servis ve süreçlerin fonksiyonel gereksinimleri yanında, güvenlik, iletişimsel özellikleri ve politikaları gibi fonksiyonel olmayan konuları da kapsar.

3. İLGİLİ KONULAR

3.1. Servis Odaklı Mimari: SOA

İnternetin hızla tüm dünyada kullanımı, firmalar arasında etkileşimin artması, hızı ve entegrasyonu son derece önemli olmuştur. Kurumsal ortamlarda birçok yazılım birlikte kullanılmaktadır. Bu yazılımlar birbiri ile entegre çalışmada sorun yaşamaktadırlar. Bazı fonksiyonlar ise benzer olmasına karşın kullanım olanakları kısıtlı ve esnek olmayan bir yapıda olabilmektedirler. Son yıllarda, özellikle dağıtık yapıli kurumsal ortamlarda bu ve benzeri entegrasyon sorunlarını çözmek için servis odakli mimari (SOA) yaklaşımı geliştirilmiş ve uygulanmaya konmuştur.

SOA yaklaşımı, temel olarak tabii uygulamasının web servisleri olarak temsil edildiđi kurumsal bir stratejik anlayıştır. SOA, yalnızca bir ürün veya tek bir referans mimari olmaktan çok, en iyi pratikler, referans mimariler, süreçler, araçlar ve şablonlar yanında bir kurumun hızlı cevap vermesini ve kurumsal mimarinin çevikliğini arttırmaya yaracak her şeyi kapsar. Standartlar ve şablonlar SOA'nın kolay ve geniş çapta kabulünde büyük rol oynar [15].

SOA en genel tanımıyla bilgi sistemlerini servis şeklinde modülleştiren bir mimari tarzdir. Bu servisler daha sonra iş süreçlerini hayata geçirmek üzere düzenlenirler. Başarılı bir SOA uygulamasında, servisler hazır bir şekilde birleştirilerek yeni ve daha gelişmiş çeşitli iş süreçlerine dönüştürülür.

Aslında SOA, elli yıldan daha önce yapısal programlama ile başlamış yazılım modülleştirme tekniklerinin evrimleşmiş mantıksal devamıdır. SOA'nın özelliđi servis sağlayıcılar ve kullanıcılar ya da müşterilerin konumu ve uygulama teknolojileri seçiminde ileri düzeyde esneklik sağlamasıdır. Soyutlanmış servis arayüzleri sayesinde tedarikçi ve müşterilerin kendi içinde bağımsızca gelişmesine imkân sağlar [16].

3.1.1. Servis Odaklı Mimari Öncesi

Ana sunucu sistemler, 1960'lı ve 70'li yıllarda çok nadiren birbirleri ile haberleşirdi. Bir iş fonksiyonu ile ilgili tüm fonksiyonlarını sağlamak, bu tip sunucuların satın alınma sebebiydi. Bir haberleşme söz konusu olduğunda ise bunu bir grup teyp verinin transferi için kullanılmakta idi. Zaman geçtikçe, sistemler arası gerçek zamanlı erişim, özellikle

bir organizasyon içinde birden fazla sistem bulununca, gereksinim haline geldi. Şirketler arası gerçek zamanlı işlemlerin gerektiği, finansal piyasalarda bu gereksinim özellikle görülmekteydi.

Başlangıçta, gerçek zamanlı erişim düşük seviye soket haberleşmeleri ile yapılmaktaydı. Genelde Assembly ve C dilinde yazılan soketler karmaşıktı ve tabanında yatan ağ protokollerini ileri düzeyde anlamayı gerektiriyordu. Bazı şirketler, mesajlaşma ve haberleşmeyi kolaylaştırmak için orta-seviye yazılımlar geliştirmeye başladılar. Sonunda, uzak prosedür çağrıları (RPC) ile dağıtık yapıda uygulamalar geliştirmek uygulanabilir seviyeye geldi. Böylece uzak prosedür çağrıları ile farklı fonksiyonlar, uzak bilgisayarlar tarafından çağrılabilir ve kullanılabilir duruma geldi.

1980'li yıllarla birlikte, kişisel bilgisayarlar sahneye çıktı ve geliştiriciler masaüstü bilgisayarlarının gücünü daha etkin kullanım yollarını araştırmaya başladılar. Donanım fiyatlarının hızlı düşüşü kurumsal sunucuların sayısını çok arttırdı. Bu eğilimler RPC'nin olgunluğunun artmasıyla daha da güçlendi ve dağıtık bilgisayar mimarisine iki önemli gelişimin önderi oldu.

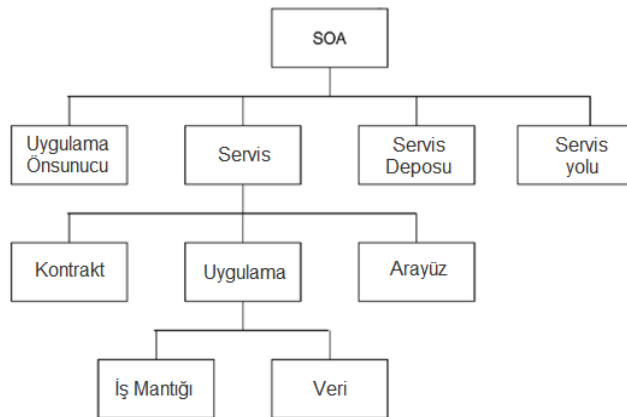
CORBA (Common Object Request Broker Architecture, Ortak Nesne İstem Aracısı Mimarisi) programlama fonksiyonlarının dağıtık kullanımını standartlaştıran bir kavram olarak 1991 yılında çıkmıştır. Üreticiler arasındaki uyumsuzluklar ve uyarlanmaları ticari olarak pahalı olduğundan kabulü yavaş olmuştur. Her ne kadar 1998'de 2.0 versiyonu ile ek birkaç dil desteği ve önceki standartlardaki sıkıntıları gidermesine rağmen, Java ile gelen RMI (Remote Method Invocation, Uzak Metot Çağırımı), dağıtık sistemleri, büyük oranda basitleştirmiştir ve sonunda XML ile CORBA'nın büyük oranda terk edilmesine neden olmuştur.

DCOM (Distributed Computing Object Model, Dağıtık Bileşen Nesne Modeli), CORBA'ya karşılık geliştirilmiş bir Microsoft teknolojisidir. İlk uygulaması 1993 yılında görülmüştür. Microsoft dünyasında başarılı uygulanması yanında, Microsoft'a ait bir teknoloji olması bariz bir kısıtlamadır. Kurumsal sınıfta büyük uygulamalar (ERP, Enterprise Resource Planning), o dönemde ortaya çıkmıştır ve genelde Microsoft dışı teknolojiler kullanılmıştır. Daha sonra Java'nın Kurumsal JavaBeans (Enterprise JavaBeans, EJB) platformu DCOM'un Java alternatifi olacaktır.

İnternet'in 1990'lı yılların sonunda yaygın kabulü ile firmalar, kendi platformlarını, ortakları ve müşterilerine sunacak şekilde genişletmenin faydalarını anlayacaktır. Bundan önce organizasyonlar arası bağlantılar kiralık hatlar (leased lines) üzerinden yapılmaktaydı. CORBA veya DCOM'un İnternet üzerinden kullanımında ise ciddi oranda network kısıtlamaları vardı. CORBA veya DCOM'un piyasa paylaşımı hedefleri ise başka bir kısıtlamaydı.

SOAP (Simple Object Access Protocol, Basit Nesne Erişim Protokolü) 2000 yılında ortaya çıktığında, farklı platformlarda iletişimi sağlayabilen XML'e dayandığından tüm sorunları çözecekmiş gibi bir izlenim uyanmıştı. SOAP, geniş anlamda CORBA ve DCOM'a bir RPC alternatifi olarak görülmektedir. Çünkü RPC'ler dağıtık sistemlerin baskın modeli olduğundan, SOAP aynı kapasitede kullanılmaya başlanmıştır. Aslında SOAP, RPC'nin daha çok platformu anlaştıran XML'e dayalı gelişmiş halidir. Günümüzde SOA Kavramı, internetin ve özellikle HTTP'nin yaygın kullanımı ile başladı. Gartner Group'tan Roy Schulte, 2003 yılında, SOA kavramından bahsetti ve çabucak yaygınlaştı. Tam olarak SOA'yı ifade etmek ilk anda zor oldu ancak zaman içinde birçok tanımlamalar ortaya çıktı [17].

Günümüzde SOA açık, çevik şekilde genişletilebilir, birleşmiş, mimari olarak birleştirilebilir, etkileşime uygun, keşfedilebilir ve potansiyel olarak yeniden kullanılabilir, web servisleri olarak gerçekleştirilebilir olarak sunulmaktadır [1]. SOA, kurumsal uygulamalarda bulunan farklı fonksiyonları, iş gereksinimlerini karşılamak üzere birleştirilebilen ve yeniden kullanılabilen standart tabanlı olan servisler olacak şekilde organize eden bir BT stratejisidir [18].



Şekil 3.1. SOA'nın temel öğeleri.

SOA, uygulama arayüzü (application frontend), servis, servis deposu (service repository) ve servis yolu (service bus) anahtar kavramlarını temel alan bir yazılım mimarisidir. SOA'nın tüm odaklanma alanı, Şekil 3.1'de de görüldüğü gibi bir iş altyapısının tanımına dayanır. Uygulama önyüzü iş sürecinin sahibi olmasına karşın, servisler, uygulama önyüzü ve diğer servislerin kullandıkları iş fonksiyonlarını yerine getirirler. Bir servis, iş mantığı ve verisi, fonksiyonellik, kullanım ve bir istemci için ve fiziksel olarak sunumu sağlayan servis arayüzü kısıtları belirleyen servis anlaşmasını sağlayan bir uygulamayı içerir. Servis deposu, SOA'nın her bir servisinin servis anlaşmalarını tutar ve servis yolu uygulama önyüzü ve servisleri birbirine bağlar [19].

3.2. PRINCE2

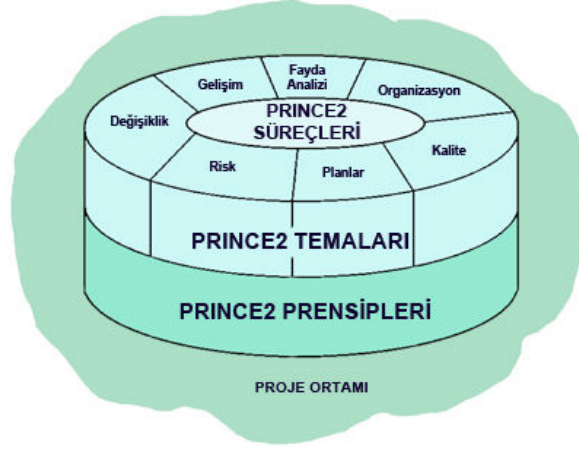
PRINCE2 yoğun olarak İngiliz Hükümetinin kullandığı mecburi standart proje yönetimi metodolojisi olmasının yanında dünyada en geniş kabul bulup kullanılan standartlardan biridir. Binlerce projeden elde edilmiş deneyimler üzerine kurulmuş ve sayısız proje destekçisi, proje yöneticisi, proje takımı akademisyenleri, eğitimcileri ve danışmanların katkılarıyla oluşturulmuş yapısal bir proje yönetimi metodudur [3, 4]. PRINCE2 belirli bir sahibi olmayan bir metottur ve proje yönetimi olarak dünya çapında en geniş anlamda kabul görmüş metotlardan biridir. Bunun en büyük sebebi PRINCE2'nun gerçek anlamda jenerik olmasındandır.

İlk olarak PRINCE ismiyle 1989 yılında CCTA (the Central Computer and Telecommunications Agency) tarafından oluşturulmuştur. Bu kuruluşun ismi daha sonra OGC (the Office of Government Commerce) olarak değiştirilmiştir. Avrupa organizasyonlarından oluşan bir şirketler birliği ile PRINCE2 sürümü 1996'da çıkarılmıştır. 2009 yılında ise PRINCE2 son sürümü duyurulmuştur.

PRINCE2 biçimlendirilebilir, takibi basit, projelerin organizasyon, yönetim ve kontrol konularını içeren jenerik bir proje yönetimi metodolojisidir. Öncelikli amacı, bir projede, doğru ürünlerin ya da çıktıların zamanında ve öngörülen bütçede başarılı olarak teslim edilmesini sağlamaktır. Risklerin yönetimi, kalite kontrol ve etkin değişiklik yönetimi PRINCE2 tarafından sağlanır.

3.2.1. PRINCE2 Yapısı

PRINCE2, proje yönetimini dört entegre öge ile adresler. Bu ögeler, Prensipler, Temalar, Süreçler ve Proje Ortamıdır ve bu yapı Şekil 3.2’de gibi bir diyagramda gösterilir. PRINCE2 süreç (proses) tabanlı bir yaklaşımla projeleri yönetir ve bu süreçlerin anahtar girdileri ve çıktıları belirli nesnelere ve faaliyetleri vardır.



Şekil 3.2. PRINCE2 metodunun yapısı.

3.2.2. Prensipler

PRINCE2 7 prensip üzerine kurulmuştur. Bu kılavuz zorunluluklar ve en iyi pratikler projenin gerçekten PRINCE2 kullanılarak yönetildiğini belirlerler. Bir projede bu prensiplerin tamamı uygulanmadığında bu bir PRINCE2 projesi olamaz. Bu prensipler:

İş tarafı uyumluluğu: Bir proje sürekli olarak iş uyumluluğuna sahip olmalıdır.

Deneyimden öğrenme: PRINCE2 proje ekibi daha önceki deneyimlerinden öğrenir. Proje yaşam döngüsü boyunca dersler, gözlemlenir, kaydedilir ve kendileri üzerinde çalışılır.

Roller ve sorumluluklar: Proje, iş tarafı, kullanıcı, katılımcı ve tedarikçilerden oluşan organizasyon içerisinde belirlenmiş ve anlaşılabilir rol ve sorumluluklara sahiptir.

Aşamaları yönetme: Proje aşama aşama planlanır, takip edilir ve kontrol edilir.

İstisnaları yönetme: Delege edilmiş otoritenin limitlerini oluşturmak için projede belirli toleranslar oluşturulur.

Ürün odaklılık: Bir PRINCE2 projesi ürünün (çıktının) tanımına ve teslimine ve özellikle de kalite gereksinimlerine odaklanır.

Bıçimlenirlik: PRINCE2, projenin ortamına, büyüklüğüne karmaşıklığına, önemine, yeteneklerine ve risklerine uyacak şekilde bıçimlendirilir.

3.2.3. Temalar

PRINCE2 prensipleri üretmede faydalanılan yaklaşımlar 7 temada vurgulanmıştır. Bu temalar proje boyunca sürekli ve paralel olarak adreslenmesi gereken yönleri tanımlayıcıdır. Bu 7 tema çeşitli proje yönetimi disiplinlerini ve neden onların gerekli olduklarını PRINCE2 tarafından gerekli ele alış bıçimini açıklar. Bunlar:

Fayda analizi: Projenin nedenini cevaplar. Proje, organizasyon için değer taşıyan bir fikir ile başlar. Bu tema, bu fikrin nasıl geçerli ve tutarlı bir yatırım teklifine geliştirileceğini ve proje boyunca nasıl organizasyonun hedeflerine odaklılığın takibinin yönetileceğini adresler.

Organizasyon: Projede kimlerin katılımcı olacağını belirler. Projeyi destekleyen organizasyon, projeden sorumlu olacak ve tamamlanmasına kadar sevk ve idare edecek bir yöneticiyi atamak durumundadır. Projeler çapraşık fonksiyonludur ve normal fonksiyon yapısı çizgisi uygun değildir. Bu tema, projeyi etkin olarak yönetecek geçici PRINCE2 proje yönetimi takımının rol ve sorumluluklarını tanımlar.

Kalite: Başlangıçtaki fikir ancak genişçe taslağının oluşturulmasıyla anlaşılabilir. Bu tema her katılımcının teslim edilmesi gereken ürünün kalite özelliklerinin neler olacağını anlayacağı tasarlamanın nasıl geliştirileceğini ve sonra bu gerekliliklerin nasıl sağlanacağını anlatır.

Planlar: PRINCE2 Projeleri bir seri onaylanmış planlar üzerinde yürür. Bu tema, planların geliştirilmesi için gereken adımları ve uygulanması gereken PRINCE2 tekniklerini tanımlayarak kalite temasını tamamlar. Planlar, PRINCE2’da organizasyonun çeşitli seviyelerdeki personelinin gereksinimlerini karşılar. Proje boyunca bunlar iletişim ve kontrol odaklıdır.

Risk: Projeler kararlı faaliyetlerden daha fazla riskler içerir. Bu tema projenin planlarında ve daha geniş proje ortamında belirsizliklerinin nasıl yönetileceğini adresler.

Değişiklik: Bu tema proje yönetiminin projenin kabul edilmiş ana hatlarına etki edecek potansiyel durumların nasıl değerlendirilip onlar hakkında nasıl işlem yapılacağını tarif eder.

Gelişme: Bu tema planların devam eden geçerlilik ve yaşayabilirliklerini adresler. Tema, planların onaylanmasında karar verme sürecini, gerçekleşen performansın takibini ve plana göre gitmeyen olaylar varsa bunların tırmanış sürecini açıklar. Sonuçta gelişme teması projenin nasıl ilerlemesi gerektiğini belirler [5].

3.2.4. Süreçler

PRINCE2 prensiplerine ulaşmak amacıyla aşağıdaki süreçler belirlenmiştir. Proje başlangıcından kapanışına kadar, yaşamı boyunca adım adım gelişmeyi tarif eder. Her süreç kendine ait kontrol listeleri (checklist), tavsiye edilen faaliyetler, ürünler ve ilgili sorumlulukları sağlar. Bu süreçler:

Projeyi başlatma süreci: Bu PRINCE2'nun ilk sürecidir. Bu bir proje öncesi süreçtir ve projenin başlaması için gerekliliklerin yerli yerinde olmasından emin olunmasını sağlar.

Proje koordinasyon süreci: Projenin başlangıcından bitimine kadar devam eder. Süreç proje kuruluna yönlendirilmiştir. Proje kurulu raporlar ve bir kaç karar noktasını kontrol ederek yönetir ve takip eder.

Projeyi devreye alma süreci: Projenin başlangıç hedefleri: Projenin devam etmesi için gerekli sebeplerin olup olmadığında fikir birliğine varılması, devam edilebilecek kararlı yönetim temelinin kurulması, kabul edilebilir bir Fayda Analizi dokümanının oluşturulması ve onaylanması, işin başlamasından önce sağlam ve kabul edilmiş bir temelin varlığının sağlanması, projenin ilk sürecinde yer alacak kaynakların dahilîyetlerinde anlaşılması, proje kurulunun projeyi sahiplenmesinin olanak verilmesi ve cesaretlendirilmesi, projenin yaşam döngüsü için gerekli olan karar verme süreçlerine bir taban sağlanması, risklerin de hesaba katılarak proje için gerekli olan zaman ve çaba atmasının mantıklı ve gerçekçi yapıldığından emin olunması hedeflerinin içerir.

Bir aşamanın kontrolü: Bu süreç proje kuruluna projenin devam edip etmeyeceği konusunda anahtar karar noktaları sağlar.

OGC'ye göre Program, organizasyonun stratejik hedefleriyle ilgili ürünler, çıktılar veya yararlar sağlayacak bir grup birbiriyle ilişkili projenin koordinasyonu, yönetilmesi ve denetlenmesi amacıyla geçici ve esnek kurumsal yapıdır. Program genel itibariyle bir tek projeden daha uzun yaşama sahiptir. Program yapısı ve raporlama gereklilikleri ile kendisinin bir parçasını oluşturan projeleri etkiler.

3.3. CMMI

CMMI kurumların performansını arttıran etkin süreçlerin temel öğelerini destekleyen bir süreç geliştirme modelidir. CMMI bir proje sürecinde, bir kurumun bir alt bölümünde veya bütün bir kurum süreçlerinin geliştirilmesine yol gösteren bir kılavuz olarak kullanılabilir. Geleneksel olarak ayırık olan kurumsal fonksiyonların birleştirilmesine, süreç geliştirme hedefleri ve öncelikleri belirlenmesinde, mevcut süreçlerin değerlendirilmesinde referans noktası sağlama konularında yardımcı olur [6]. CMMI Modelinin gelişmesindeki önemli kilometre taşları Tablo 3.1'de verilmiştir.

CMMI, yetenek olgunluk modeli (CMM) ya da CMM Yazılım'ın devamıdır. CMM 1987 den 1997'ye kadar geliştirilmiştir. CMMI Sürüm 1.1 2002'de, Sürüm 1.2 Ağustos 2006'da ve Sürüm 1.3 Kasım 2010 tarihinde yayınlanmıştır.

CMMI günümüzde 3 ilgi alanına hitap eder:

1. Ürün ve Servis Geliştirme — CMMI for Development (CMMI-DEV),
2. Servis Kurulumu, Yönetimi ve Dağıtımı — CMMI for Services (CMMI-SVC) ve
3. Ürün ve Servis Edinme — CMMI for Acquisition (CMMI-ACQ).

3.3.1. CMMI Modelindeki Sürekli ve Aşamalı Gösterimler

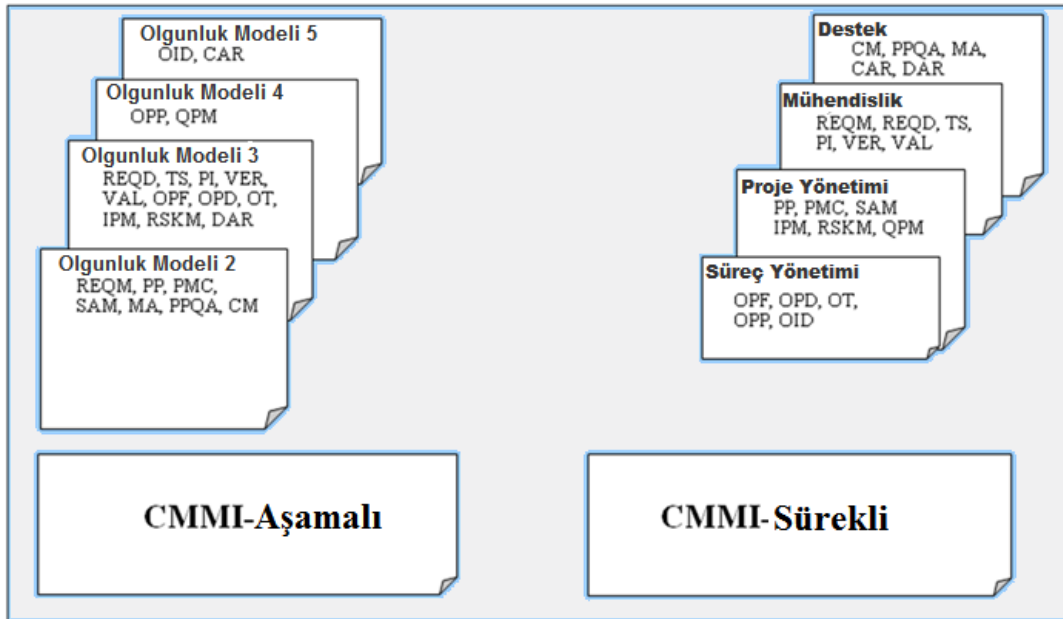
Bir kurum içinde kullanılan süreçler, çalışma alanı, kurum yapısı, büyüklüğü gibi çeşitli değişkenlere bağlıdır. CMMI modelleri birer süreç ya da süreç tanımı değildir. Süreçlerin gerçekleştirilmesinde birer yol göstericidir. CMMI modelleri süreç alanlarından oluşur. Bu süreç alanlarının her biri, kurum içinde var olan ya da kullanılan ve birbirleriyle organik bağı olan süreçlerin bir araya getirilmesinden oluşmaktadır. Süreç alanları birbirlerinden bağımsız olarak tanımlanmış olsa da kurumsal yapı içerisindeki etkileşimlerin herhangi bir süreç iyileştirme çalışmasındaki sonuçları mutlaka göz önünde tutulmalıdır.

Tablo 3.1. CMMI'nin gelişimindeki önemli kilometre taşları.

Yıl	Gelişme
1987	CMM (Capability Maturity Model) Carnegie Mellon Üniversitesindeki, Yazılım Mühendisliği Enstitüsü (SEI) tarafından, Amerikan Savunma Bakanlığı için geliştirilmeye başlandı.
1991	SEI SW-CMM'in (CMM for Software) ilk sürümü yayınladı.
1995	Enterprise Process Improvement Collaboration (EPIC), Systems Engineering Capability Maturity Model (SE-CMM) yayınladı.
1996	Systems Engineering Capability Assessment Model (SECAM), Council on Systems Engineering (INCOSE) tarafından geliştirilip yayınladı.
1997	SW-CMM Versiyon 2 taslağı ve Integrated Product Development CMM
1998	EIA 731 SECM oluşturuldu.
1998	İlk takım buluşması yapıldı.
1999	Operasyonların kavramı yayınladı. İlk deneme tamamlandı.
2000	Ek denemeler tamamlandı. CMMI-SE/SW sürüm 1.0 başlangıç amaçlı olarak yayınladı. CMMI-SE/SW/IPPD sürüm 1.0 başlangıç amaçlı olarak yayınladı. CMMI-SE/SW/IPPD/SS test amaçlı olarak yayınladı.
2002	CMMI-SE/SW sürüm 1.1 yayınladı. CMMI-SE/SW/IPPD sürüm 1.1 yayınladı. CMMI-SE/SW/IPPD/SS sürüm 1.1 yayınladı. CMMI-SW sürüm 1.1 yayınladı.
2006	CMMI-DEV sürüm 1.2 yayınladı. CMMI-DEV +IPPD sürüm 1.2 yayınladı.
2007	CMMI-ACQ sürüm 1.2 yayınladı.
2010	CMMI-ACQ, CMMI-DEV, CMMI-SVC modellerini kapsayan CMMI V1.3 son ve güncel sürüm olarak 1 Kasım 2010'da yayınladı.

CMMI modeli içinde 22 süreç alanı vardır. Bu süreç alanları Sistem Mühendisliği, Yazılım Mühendisliği, Bütünleşik Ürün ve Süreç Geliştirme ve Tedarikçi Kontrolü (SGTK) olmak üzere 4 ana disiplin altında toplanabilir. CMMI modeli değişik süreç iyileştirme modellerini bir araya getirme ve bu modellerden kolay bir dönüşüm sağlama hedefleri dolayısıyla birbirine denk sonuçlar veren iki farklı gösterime (Representation) sahiptir. Bu gösterimlerden ilki “Sürekli Gösterim” (Continuous Representation) olarak adlandırılır ve herhangi bir süreç alanında bir yetenek olgunluk seviyesi ile belirlenir. Bu gösterim sayesinde kurumunuzda istediğiniz süreç alanlarında istediğiniz yetenek olgunluk seviyesinde çalışmalar yapabilirsiniz.

Bir diğer gösterim de “Aşamalı Gösterim” (Staged Representation) olarak adlandırılır. “Sürekli Gösterim”den farklı olarak bu gösterimde belirli süreç alan gruplarında çeşitli aşamalar halinde iyileştirmeler sağlanır. Böylece süreç iyileştirme çalışmalarına yeni başlayan kurumlar önlerinde kesin olarak tanımlanmış bir yol görür. Bu iki görünümün aşağıdaki Şekil 3.4’te de açıklamaları verilmektedir.



Şekil 3.4. Bir CMMI modelinin iki farklı gösterimi.

Yetenek olgunluk modelinde Aşamalı Gösterimde süreçler 5 seviyelidir. Bu seviyeler:

1. Seviye: Süreçler önceden tahmin edilemez, kontrolleri zayıftır ve tepkiseldir. Anlık çözümler geliştirilir ve acil durumlarda süreçler tamamen bir kenara bırakılır.
2. Seviye (Yönetilebilir): Süreçler proje bazında tanımlanır ve yönetilir. Tablo 3.2’de görüldüğü gibi Proje Yönetimi ve Destek süreçleri bu olgunluk seviyesinde yer alır.
3. Seviye (Tanımlanmış): Süreçler kurum bazında tanımlanır ve yönetilir. Kurumun süreçleri bir havuzda toplanır ve tüm projelerde bu süreçler kullanılır. Tablo 3.2’de görüldüğü gibi Mühendislik, Süreç Yönetimi ve Proje Yönetimi ile ilgili süreçlerin 9 tanesi bu olgunluk seviyesinde bulunur.
4. Seviye (Nicel olarak yönetilebilir): Süreçler nicel olarak ölçülmekte ve kontrol edilmektedir.
5. Seviye (En iyilenir): Süreçler ve süreç iyileştirme odak noktasıdır. Bu olgunluk seviyesinde organizasyon, iş hedefleri ve performans ihtiyaçları nicel algılamasına dayalı olarak sürekli iş süreçlerini iyileştirir [20].

Tablo 3.2. CMMI Süreç Alanları, Kategoriler ve Olgunluk Seviyeleri.

CMMI Süreç Alanları, Kategoriler ve Olgunluk Seviyeleri				
Süreç Alanları	Süreç Alanları (İngilizce)	Kısaltmalar	Kategoriler	Olgunluk Seviyesi
Nedensel Analiz ve Çözüm	Causal Analysis and Resolution	CAR	Destek	5
Organizasyonel Performans Yönetimi	Organizational Performance Management	OPM	Süreç Yönetimi	
Organizasyonel Süreç Performansı	Organizational Process Performance	OPP	Süreç Yönetimi	4
Nicel Proje Yönetimi	Quantitative Project Management	QPM	Proje Yönetimi	
Karar Analiz ve Çözüm	Decision Analysis and Resolution	DAR	Destek	3
Entegre Proje Yönetimi	Integrated Project Management	IPM	Proje Yönetimi	
Organizasyonel Süreç Tanımı	Organizational Process Definition	OPD	Süreç Yönetimi	
Organizasyonel Süreç Odağı	Organizational Process Focus	OPF	Süreç Yönetimi	
Organizasyonel Eğitim	Organizational Training	OT	Süreç Yönetimi	
Ürün Entegrasyonu	Product Integration	PI	Mühendislik	
Gereksinimler Geliştirme	Requirements Development	RD	Mühendislik	
Risk Yönetimi	Risk Management	RSKM	Proje Yönetimi	
Teknik Çözüm	Technical Solution	TS	Mühendislik	
Konfigürasyon Yönetimi	Configuration Management	CM	Destek	
Ölçümleme ve Analiz	Measurement and Analysis	MA	Destek	2
Proje Takip ve Kontrolü	Project Monitoring and Control	PMC	Proje Yönetimi	
Proje Planlama	Project Planning	PP	Proje Yönetimi	
Süreç ve Ürün Kalite Güvence	Process and Product Quality Assurance	PPQA	Destek	
Gereksinimler Yönetimi	Requirements Management	REQM	Proje Yönetimi	
Tedarikçi Anlaşma Yönetimi	Supplier Agreement Management	SAM	Proje Yönetimi	

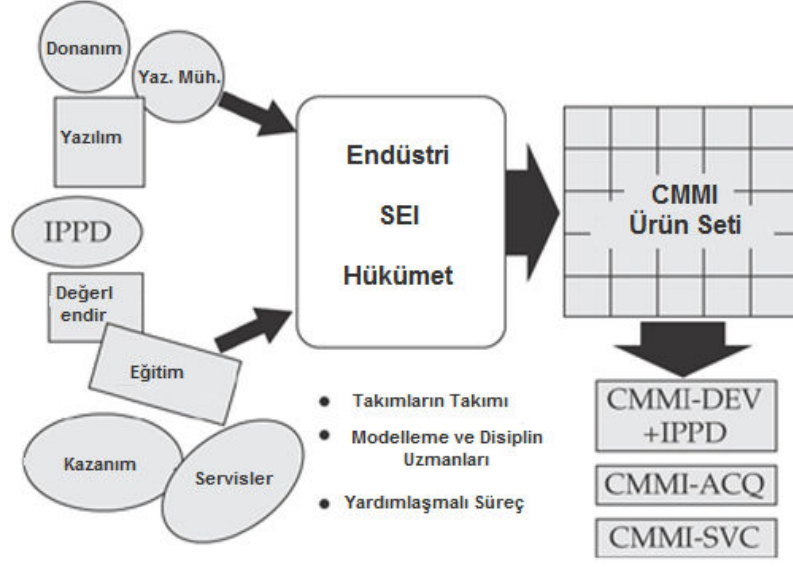
3.3.2. CMMI'in Amaçları

İşle ilişkili birçok faydasının olması yanında CMMI, proje destekleyicileri tarafından, daha verimli ve etkin süreç modellerinin geliştirilmesi gibi bir yönü vardır. Başlangıç ve uzun dönem amaçları vardır. Başlangıç amacı, yazılım, sistem mühendisliği ve entegre ürün ve süreç geliştirme olmak üzere 3 özel süreç geliştirme modellerinin birleştirilmesiydi. Bu entegrasyon, çok disiplinli model tabanlı süreç geliştirmenin maliyetlerini azaltmak için aşağıdakileri sağlaması hedeflenmiştir:

- Uyumsuzlukları ortadan kaldırmak
- Tekrarları azaltmak
- Açıklığı ve anlaşılabilirliği arttırmak
- Ortak bir terminoloji sağlamak
- Kalıcı bir tarz sağlamak
- Standart yapılaşmayı sağlamak
- Ortak bileşenleri sürdürmek
- ISO/IEC 15504 ile uyumluluğu temin etmek

CMMI 1.2'de CMMI takımının amaçlarından biri de mühendislik geliştirme faaliyetlerinin modelini geliştirmek ve basitleştirmektir. İkinci amaç ise servislerin belirlenmesi ve sunulmasının geliştirme dünyasına uygulanarak modelin kapsamının genişletilmesi idi. Şekil 3.4.'de bu amaçlar ve CMMI takımı tarafından geliştirilen ürün hattı yaklaşımı gösterilmektedir. Görülebileceği gibi gelecekte diğer disiplinlerin ve yeni fikirlerin CMMI ürün grubuna eklenebilecektir.

Hem mevcut hem de gelecek model entegrasyonunu kolaylaştırmak ve olanaklı kılmak için CMMI takımı, otomatik olarak genişletilebilir bir iskelet kurarak model bileşenleri, eğitim materyalleri bileşenleri ve değerlendirme materyallerini içine alabilecek duruma getirmiştir. Belirlenmiş kurallar, daha fazla disiplinlerin bu iskelete muhtemel eklentilerini yönetecek şekilde yönetir.

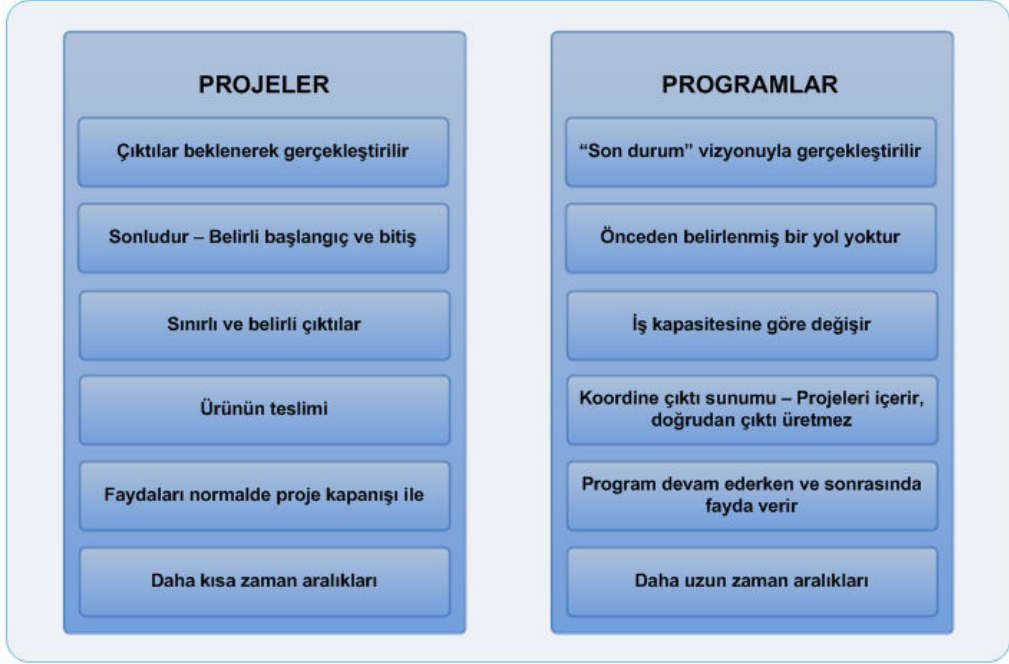


Şekil 3.5. CMMI kavramı.

4. PRINCE2’NUN BİR SOA UYGULAMASINDA KULLANILMASI

4.1. SOA Uygulaması Programı

Günümüzde dünya çapında en çok kullanılan proje yönetim metodolojilerinden biri olan PRINCE2’nun sahibi OGC Proje ile Program arasındaki farkları Şekil 4.1’de belirtmektedir. Bu kriterleri SOA uygulaması için dikkate aldığına “son hal vizyonuyla tetiklenir” ifadesindeki tetikleyici ideal SOA uygulamasıdır. SOA Projeleri ise, SOA prensipleri içinde doğrudan iş beklentilerini karşılamalıdır. SOA Programı çok belirgin bir başlangıç ve bitişe sahip değilken bir SOA Projesinin hedefi, başlangıç ve bitiş tarihlerinin olmasının yanı sıra en çevik bir şekilde iş beklentilerini karşılaması olmalıdır.

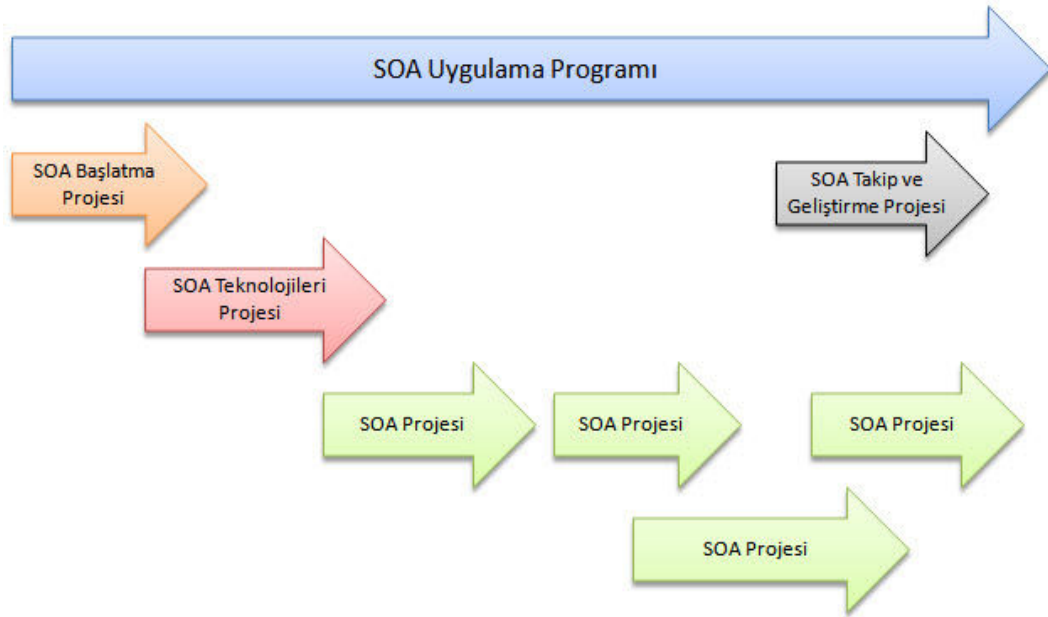


Şekil 4.1. OGC’ye göre Proje ve Program arasındaki farklar.

SOA Programında iki tipte projeler mevcuttur. Birinci tip projeler SOA ortamının oluşturulması, takip edilmesi, geliştirilmesi ve devam ettirilmesi ile ilgilidir. Diğer projeler ise sayısı, zamanı ve hedefi önceden belirli olmayan çok sayıda projeden oluşabilir. Gerçek dünyada tüm bu çalışmalar önemli bir projeden ya da SOA’nın uygulanmasına uygun bir projeden esinlenerek başlatılabilir. Bu durumda vizyonda çok sayıda SOA projesi söz konusu iken motivasyon olarak iş gereksinimine cevap veren

proje öne çıkabilir ve diğer tüm SOA ortamını gerçekleştiren projeler iş gereksinimi çıktıları veren projeye dayandırılabilir. Hatta tüm bu çalışmalar sadece tek bir proje gibi gösterilebilir. Prensip olarak bunun iyi bir yöntem olmaması nedeniyle BT içerisinde fiilen de olsa bunların kendi başına projeler olarak ele alınması gerekmektedir.

SOA Programının akışına büyük oranda iş tarafının beklentileri ve gerekleri yön vermelidir. Buna karşılık her iş odaklı SOA projesinin olabildiğince belirgin ve iyi tanımlanmış süresi ve çıktıları olmalıdır. SOA Programının çıktıları başarılı SOA projeleri, SOA Projelerinin çıktıları ise iş tarafı gereksinimlerine tam cevap verir şekilde olmalıdır. SOA'nın faydaları her projenin sonlandırılması ile ve program devam ettiği sürece ortaya çıkıyor olmalıdır. Bir SOA programı yıllarca sürebilir, buna karşılık bir SOA Projesinin çok kısa bir sürede gerçekleştirilmesi istenebilir. İş tarafı ile yüksek hassasiyette uyumlu olmak SOA'nın en can alıcı avantajlarından biri olduğundan SOA projeleri çok kısa zamanlı ve çevik olabilmelidir. Bütün bunların ışığında aşağıda Şekil 4.2'de görüldüğü gibi SOA Programı bir kaç tane projeden oluşabilir.



Şekil 4.2. SOA uygulama programı katmanları.

SOA Başlatma Projesi: Bu projede kurum ortamında SOA prensiplerinin analiz edilmesi ve oluşturulması gerçekleştirilmelidir. SOA prensipleri tüm ilgili SOA projeleri ile birlikte ve her SOA projesi gerçekleştirilmesi esnasında kullanılmalıdır. SOA

süreçleri ve kuralları bu adımda belirlenmelidir. Bir sonraki aşamada mevcut sistemlerin analiz edilir ve dökümü çıkarılır. Bu çalışmalar ışığında son olarak SOA uygulaması yol planı oluşturulur.

SOA Teknolojileri Projesi: İş gereksinimlerinden doğan bir SOA projesine başlamadan önce SOA altyapısının oluşturulması gerekmektedir.

SOA Takip ve Geliştirme Projesi: SOA projelerinin uygulamaya geçmesi ile birlikte mevcut, devam eden ve gelecekteki projelerin SOA'ya uygunluğunun teknolojik ve prensipler anlamında değerlendirildiği, raporlandığı ve iyileştirme, düzeltme ve güncellemelerin gerçekleştirildiği projeler SOA'nın getirisini artırıcı etkiler yapacaktır.

SOA Projeleri: SOA'nın Başlangıç ve Teknoloji projelerinin gerçekleştirilmesi ardından Tablo 4.1'de ele alındığı gibi doğrudan iş gereksinimlerini karşılayan, çıktıları iş tarafının beklentileri olan SOA proje ya da projeleri gerçekleştirilir. Bu tip projeler paralel olabileceği gibi iş tarafı ihtiyaçlarına göre birbiri ardı sıra ya da aralarında belirli zaman aralıklarının bulunduğu projeler olabilir. Bu yönüyle BT altyapısı iş tarafı durumuna uyumlu bir şekilde SOA altyapısında projeler üretme yeteneğine sahip olacaktır.

Kurumsal anlamda değişimi bir program altında ve çok sayıda projelerle gerçekleştirmeyi amaçlayan bu çalışma Tablo 4.1'de ilk sırada görülen SOA'nın Stratejik karar ve uygulama özelliğine karşılık gelmektedir.

4.2. PRINCE2 Prensiplerinin SOA Prensipleri İle Birlikte Kullanılması

SOA kavramının başarı prensipleri PRINCE2 ile birleştirildiğinde bu tarz uygulama projelerinin başarısı artmaktadır. PRINCE2 aşağıdaki prensipleri ile SOA'yı destekler:

Sürekli iş uyumluluğu ve ürün ve ürün kalitesi odaklılığı: Bu iki prensip her durumda bir proje için çok önemli olmasına karşılık SOA için ayrıca önemi vardır. Çünkü SOA, iş gereklerine hızlı cevap verme ve doğru çözüm üretme yönleriyle avantaj sunmaktadır. Dolayısıyla PRINCE2 ve SOA tam olarak bu noktalarda buluşurlar.

Deneyimden Öğrenmek: Bu prensip SOA'nın daha verimli hale gelmesine katkı sağlar. Örneğin SOA'nın temel öğeleri olan web servisleri ve bileşik uygulamalarda kullanılabilir. Elde edilen tecrübelerin biriktirilmesi ve bu bilginin benzer projelere aktarılıyor ve bu projelerde kullanılıyor olması daha iyi daha uyumlu daha iyi

dengelenmiş SOA bileşenlerinin oluşturulmasına büyük katkı sağlayacaktır. Bu şekilde SOA, yeniden kullanılabilirlik ve entegrasyon özelliklerini de gün geçtikçe daha etkin kullanabilecektir.

Özgün bir ürün ortamına biçimlendirilebilirlik: PRINCE2 jenerik olmasının yanında farklı tipte projelerde kullanılabilmesi amacıyla biçimlendirilebilirlik mekanizmasını kullanır. Bu özelliği ve mekanizması sayesinde özellikle yazılım projelerinde ve de SOA'nın esas alındığı projelerde büyük öneme sahiptir. Çünkü SOA projeleri yazılım yoğunluklu olması nedeniyle teknik detay, analitik kullanım ve olgunlukları açısından birçok farklılıklar gösterebilirken PRINCE2 ile bunları yönetmek mümkün olacaktır.

4.3. PRINCE2 Temalarının SOA İle Birlikte Kullanılması

PRINCE2'nun 7 teması her proje için uygulanmalıdır. Eğer bu temalar SOA projelerinde uygulanırsa SOA'nın vurgu yaptığı konuları güçlendirecektir ve başarı yüzdesini arttıracaktır.

Fayda Analizi: PRINCE2 ve SOA'nın en çok örtüşen konularından biri de iş ve ürün odaklı olmalarıdır. Bir PRINCE2 projesi, bu metodun ilgili sürecinden geçirilip, dokümanite edilmiş onaylanmış fayda analizi üzerine kuruludur ve projenin her aşamasında bu fayda analizini referans olarak alır. SOA ise değerini hızlı, çevik ve doğru iş gereksinimi çıktısını vermesi ile kazanmıştır. Bu nedenle SOA projesinin başlangıç noktası hızlı ve erken iş çıktısı üretmek olmalıdır.

Organizasyon: PRINCE2 organizasyon temasının amacı, projenin görev ve sorumluluklar altyapısının tanımlanması ve oluşturulmasıdır. PRINCE2 bir müşteri/tedarikçi ortamı üzerine konumlanmıştır ve projeden beklenen sonuçları tarif eden ve ortaya çıkacak bu sonuçlar için finansal kaynak olan bir müşteri ve istenen sonuçları ortaya çıkaracak kaynak ve yetkinliklerini sunan tedarikçinin var olduğu kabul edilmektedir.

PRINCE2 prensiplerinden biri olan belirlenmiş roller ve sorumluluklar gereğince bir PRINCE2 projesi üç ana paylaşımcı ilgilerine sahiptir. Bunlar iş, kullanıcı ve tedarikçidir. Bu üç grubun isteklerine cevap verildiği zaman bir proje başarılı kabul edilir. Tablo 4.1'de de görülebileceği gibi PRINCE2 iş ve kullanıcı odaklılığı, aynı öncelikli hedefleri paylaşan SOA'yı destekler.

Projelerin ortaya çıkardıkları ürünler proje yatırımını sağlayan iş gereksinimine uymalıdır ve finansal bir değer taşımalıdır. İş tarafı bakış açısı proje başlangıcında ve proje süresince temsil ediliyor olmalıdır. PRINCE2 iş hedefleri ile bu projenin çıktularından faydalananların ilgi ve gereklerini gözetir.

Kalite: Ürün odaklılık prensibi PRINCE2 metodunun ana kalite yaklaşımıdır. Bu yaklaşım ve SOA aynı hedefi paylaşır. PRINCE2 kalite teması iş beklentilerinin karşılanması ve arzulanmış faydaların elde edilmesi olarak tanımlanır.

Değişiklik: SOA çevik ve hızlı cevap verme ve servis odaklı yazılım geliştirme olarak kabul gördüğünde, çevik ve hızlı cevap verecek şekilde uygulanmalıdır. Bunu başarmak için SOA projeleri güçlü değişiklik mekanizmasına sahip olmalıdır. PRINCE2 bunu Değişiklik Teması ile belirleme, değerlendirme ve her tür potansiyel ve baz alınmış konularda onaylanmış değişiklikleri kontrol etme mekanizması ile karşılar.

Değişiklikler proje sırasında kaçınılmazdır ve her proje sistematik olarak değişikliğe neden olacak konularını belirleme, değerlendirme ve kontrol yaklaşımına ihtiyaç duyar. Proje takımı üyelerinden, paylaşımcı istek ve şikâyetlerinden olmak üzere birçok konuda değişiklik talebi gelebilirken PRINCE2 sorun ve değişiklik kontrolüne ortak bir yaklaşım sağlar ve proje performans hedeflerine etkisini uygun bir şekilde yönetir.

4.4. PRINCE2 Süreçlerinin SOA Projelerinde Kullanılması

Bir SOA projesinin SOA uygulama durumu PRINCE2 süreçleri yardımıyla kontrol edilebilir ve izlenebilir. SOA projeleri için oluşturulan “proje kurulu”, “bir projenin yönetilmesi” PRINCE2 sürecinde tanımlanır. Proje olağan raporları bu proje kurulunu bilgilendirmek için kullanılır. Proje kurulu, projenin başlangıç sürecinde gerekli onayı vermiş olmalıdır. Bir ürünün sunulması üzerine kurulan PRINCE2, bunu SOA tabanlı uygulamayı “ürün teslimi yönetimi” sürecini kullanarak yapmalıdır. Proje de kendi içinde zaman, iş, kilometre taşı açısından mantıklı aşamalara ayrılmalıdır ve her aşama proje kurulu tarafından SOA prensipleri de hesaba katılarak incelenmeli ve onaylanmalıdır. Her proje, kaydedilmiş çıkarılan dersler, teslim dokümanı ve uygulama sonrası gözden geçirmesi konularının da kontrolü yapılarak kapatma sürecinin işletilmesi ile sonlandırılmalıdır [21].

4.5. PRINCE2'nun SOA Projelerine Uyacak Şekilde Biçimlendirilmesi

PRINCE2'nun prensiplerinden biri de proje karmaşıklığı, çapı ve kültürü gibi konulara uyacak şekilde biçimlendirilebilir olmasıdır. Biçimlendirme, PRINCE2'nun projede doğru oranda planlama, kontrol, yönetim ve tema ve süreçlerin yönetiminin kullanılması olarak tanımlanır.

PRINCE2, “özel konular” olarak adlandırılan ve esnekliğiyle konunun ağırlık noktasını korumasını esas alan bir mekanizma sağlar. PRINCE2 jenerik olması nedeniyle endüstriye has faaliyetler ve metotlar içermez. Bu sebeple SOA tabanlı uygulamaları gerçekleştirmek için kullanılacak yazılım geliştirme yöntem ve tekniklerine kapı açar ve birlikte çalışacak ortamı sağlar. Tüm bu konular, PRINCE2 tarafından “özel konular” olarak belirlenir, değerlendirilir ve kapsama alınır.

Ele alınan konuda nesne tabanlı yazılım geliştirme metodolojileri ve SOA'ya has mimari ve modelleme konuları uyarlamaya dâhil edilmelidir. Örneğin Servis Odaklı Modelleme ve Mimari SOA tabanlı projenin olgunluğunu arttırmak amacıyla uygulanmalıdır [22]. Tablo 4.1 PRINCE2'nun bir proje geliştirme yöntemi olmasının yanında SOA uygulamalarına ek katkılarını göstermektedir.

Tablo 4.1. PRINCE2 metodolojisinin SOA uygulamasına katkıları.

Servis Odaklı Mimari Yaklaşım	PRINCE2 Metodolojisinin Katkı Noktaları
Stratejik karar ve uygulama	Program kavramı ile stratejinin gerçekleştirilmesi
SOA tabanlı uygulama	SOA programı ve ona bağlı projeler
SOA kavramının yerleştirilmesi	SOA programına bağlı SOA başlatma projesi
SOA teknolojik altyapısı	SOA programına bağlı SOA teknoloji altyapı projesi
İş gereksinimlerine uyumluluk	Fayda analizi, ürün ve sonuç odaklılık, kalite teması
Kullanıcı odaklılık	Organizasyon ile kullanıcı katılımı teması ile kullanıcı ve iş tarafı ile daha yakın çalışılır.
Tekrar kullanılabilirlik	Öğrenilen dersler prensibi ile tekrar kullanım oranı arttırılır.
Çevik ve hızlı cevap verme	Değişiklik yönetimi teması
Yazılım geliştirme metodolojileri	Biçimlendirilebilirlik prensibi ile adapte edilebilir.
SOA tabanlı yazılım geliştirme	Özel çalışmalar kavramı ile ayrıca ele alınmaktadır.

5. CMMI REHBERLİĞİNDE PRINCE2 İLE SOA UYGULAMASI

Bu bölümde SOA uygulaması geliştirmede, CMMI 1.3'ün CMMI-DEV aşamalı (staged) modeli rehberliğinde PRINCE2'nun SOA projelerine verebileceği katkılar incelenmiştir.

5.1. Yönetilirlik (Managed) Olgunluk Seviyesi

CMMI'da 2. seviye olan bu olgunluk seviyesi, bir yazılım geliştirme projesinde temel proje yönetim süreçlerinin uygulanıyor olmasını gerektirir. Aşağıda bu seviyedeki süreçlerin PRINCE2 ve SOA karşılaştırmaları verilecektir.

Gerekler Yönetimi (Requirements Management) kısmen PRINCE2 İş Analizi (Business Case) ve Sürekli İş Odaklılık (Continued Business Justification) ile gerçekleştirilmesine karşılık yazılım geliştirmeye has konular bu analize dâhil edilmelidir. Bir başka ifadeyle PRINCE2 özel konular ve çalışmalar dikkate alınarak Gerekler Yönetimi CMMI jenerik hedef ve prensipleri ile doldurulabilir.

Proje Planlama (Project Planning) süreci PRINCE2 kullanılarak gerçekleştirilmektedir. Özellikle Planlar (Plans) teması bu konuda daha olgun bir destek vermektedir.

Proje Takibi ve Kontrolü (Project Monitoring and Control), PRINCE2'nun iki sürecinde de ele alınmaktadır. Bunlar Aşama Kontrol Süreci ve Bir Aşamanın Yönetilmesi süreçleridir. Ayrıca Gelişme (Progress) teması ile bu konu desteklenmektedir.

Tedarikçi Anlaşması Yönetimi (Supplier Agreement Management), kısmen PRINCE2 organizasyon temasında geçmekle birlikte bu süreç CMMI'nin tavsiyeleri ile desteklenmek durumundadır.

Ölçüm ve Analiz (Measurement and Analysis), program seviyesinde özel bir projede alınarak SOA uygulamasının özellikle SOA kavramına uygunluğu ölçümlenir ve analiz edilir. Ayrıca her proje kapsamında PRINCE2 gelişme süreci gibi süreçler bu konuda genel anlamda bir vurguyu yapmaktadır.

Süreç ve Ürün Kalite Güvencesi (Process and Product Quality Assurance), PRINCE2 kalite teması ile karşılanmaktadır.

Konfigürasyon Yönetimi (Configuration Management), PRINCE2 Değişiklik Yönetimi (Change) içinde yer alırken, yazılım geliştirmede daha ayrıntılı olarak ele alınıyor olmalıdır. Bu süreç alanının daha ayrıntılı incelenmesi ve örneği 6. Bölümde yer alacaktır.

Bu aşamadaki süreçler SOA uygulamasına büyük katkı sağlayacağı açıktır. Çünkü SOA kurumsal olarak hem yazılım geliştirme odaklı kurumsal CMMI süreçleri ve BT projeleri için geliştirilmiş ve daha sonra jenerik hale getirilmiş ve kurumsal altyapıda uygulanan PRINCE2 ile desteklenmiş olmaktadır. Tablo 5.1’de Yönetilirlik olgunluk seviyesinin gereksinimlerini, PRINCE2 ile büyük oranda karşılanabileceği görülmektedir.

Tablo 5.1. CMMI yönetilirlik süreç alanlarının PRINCE2’nun karşılama seviyesi

CMMI Süreç Bölgesi	PRINCE2 Süreç, Tema ve Prensipleri	Karşılama Seviyesi
Gerekler Yönetimi	Fayda Analizi (Tema), Sürekli İş Odaklılık (Prensip)	●●
Proje Planlama	Planlar (Tema)	●●
Proje Takibi ve Kontrolü	Gelişme (Tema), Aşama kontrol (?), Bir aşamanın yönetilmesi (Süreç)	●●
Tedarikçi Anlaşması Yönetimi	Organizasyon (Tema)	●○
Ölçüm ve Analiz	Gelişme (Tema), Aşama kontrol (Süreç), Bir aşamanın yönetilmesi (Süreç)	●●
Süreç ve Ürün Kalite Güvencesi	Kalite (Tema)	●○
Konfigürasyon Yönetimi	Değişiklik (Tema)	●○

●● : Karşılıyor. ●○ : Kısmen karşılıyor. ○○ : Karşılmıyor

5.2. Tanımlanmışlık (Defined) Olgunluk Seviyesi

Bu CMMI 3. seviyesidir. Bu seviyede birçok yazılım geliştirme süreci standart olarak tanımlanmıştır. Bu süreçlerin bazıları bilişim teknoloji projelerine özel olup Tablo 5.2’de görüleceği gibi büyük oranda PRINCE2 ile karşılanamamaktadır.

Gerekler Geliştirme (Requirements Development) süreci, İş Analizi ve Sürekli İş Odaklılık (Business Case ve Continued Business Justification) ile PRINCE2 ile kısmen karşılanmaktadır. Buna karşılık yazılım geliştirme odaklı projede buna eklenecek çalışmalarla desteklenmelidir.

Tablo 5.2. CMMI tanımlanmışlık süreç alanlarının PRINCE2'nun karşılama seviyesi

CMMI Süreç Bölgesi	PRINCE2 Süreç, Tema ve Prensipleri	Karşılama Seviyesi
Gerekler Geliştirme	Fayda Analizi (Tema) , Sürekli İş Odaklılık (Prensip)	●●
Teknik Çözüm	-	○○
Ürün Entegrasyonu	Ürün odaklı olmak (Prensip), Ürün tesliminin yönetilmesi(Süreç)	●●
Doğrulama (Verification)	Gelişme (Tema), Aşama sınırı kontrolü (Süreç)	●○
Onaylama (Validation)	Gelişme (Tema), Aşama sınırı kontrolü (Süreç)	●○
Kurumsal Süreç Odağı	-	○○
Kurumsal Süreç Tanımı	-	○○
Kurumsal Eğitim	-	○○*
Entegre Proje Yönetimi	-	○○
Risk Yönetimi	Risk (Tema)	●●
Karar Analizi ve Çözüm	Gelişme (Tema), Kontrol (Süreç), Raporlamalar (Süreç)	●○

●● : Karşılıyor. ●○ : Kısmen karşılıyor. ○○ : Karşılamıyor

*: Bu alanda program dahilinde başlangıç ve takip projelerinde yer alan aktivitelerle destek verilmektedir.

Teknik Çözüm (Technical Solution) süreci PRINCE2 dışındadır ve özel çalışmalar içinde ve biçimlendirme aşamasında projeye dışarıdan dahil edilebilir.

Ürün Entegrasyonu (Product Integration), Ürünler odaklı olma (Focus on Products) prensibi ve Ürün Tesliminin Yönetilmesi (Manage Product Delivery) ile ele alınan bir konudur. Bu süreçlerde CMMI'nin tavsiyeleri doğrultusunda ek geliştirmeler yapılabilir.

Doğrulama (Verification) ve Onaylama (Validation) alanları PRINCE2 Gelişim ve Aşama sınırı kontrolü ile kısmen yapılabilecektir.

Bu seviyede yer alan Kurumsal Süreç Odağı (Organizational Process Focus), Organizasyonel Süreç Tanımı (Organizational Process Definition) gibi bazı süreçler proje kavramının üzerinde olduğundan ve programda tüm süreçlerde yer almamaktadır. Yalnızca SOA'ya ait tasarım ve geliştirme sürecinde ve daha sonra benzer süreçlerin kurulum ve işletilmesinde rehberlik edecektir.

Kurumsal Eğitim (Organizational Training), SOA uygulama yol haritasında özellikle başlangıç projesinde detaylı olarak yerine getirilmektedir.

Entegre Proje Yönetimi (Integrated Project Management) farklı CMMI sürümleri ile ilgilidir. Kurumsal süreç tanımı ve süreç odağı konuları da çalışmanın amacını aşmaktadır.

Risk yönetimi de PRINCE2 Risk Teması ile sağlanmaktadır.

Karar Analizi ve Çözüm (Decision Analysis and Resolution), PRINCE2 proje kurulu tarafından raporların ve olayların takibi ve proje yöneticisi açısından kontrol sürecinde ele alınır.

5.3. Nicel Yönetilebilir (Quantitatively Managed) Olgunluk Seviyesi

Bu CMMI 4. seviyesidir. Bu seviyede Kurumsal Süreç Performansı (Organizational Process Performance) ve Nicel Proje Yönetimi (Quantitative Project Management) süreç alanları bulunmaktadır. Bu seviyede proje üstü bir bakış açısı sergilenerek kurumsal süreç performansı ve nicel proje yönetimi ele alınır. Bu seviyede PRINCE2 program kavramı kısıtlı anlamda bazı geliştirmelere imkân tanısa da bu süreçlerin yürütülmesi için CMMI esas olarak alınmak durumundadır. Bu seviye SOA uygulama programının da asıl amacının dışında yer almaktadır. Her ne kadar bu seviyenin, SOA uygulamasına faydaları olsa da mevcut bir altyapı varsa kullanılır, eğer yoksa bu konular bağımsız bir çalışma ile ele alınmalıdır.

5.4. Optimize Olgunluk Seviyesi

Sürekli Süreç İyileştirme odaklı bu seviyede PRINCE2 kapsam olarak cevap vermemektedir. Bu noktada CMMI Kurumsal Yenilik ve Yayma (Organizational Innovation and Deployment) ve Nedensel Analiz ve Çözüm (Causal Analysis and Resolution) süreci ile SOA tabanlı yazılım geliştirme ile birlikte tüm BT organizasyonu süreçlerini içine alacak bir hedef çalışmayı tarif etmektedir [23].

Bu seviyenin belirlediği hedefler SOA uygulamasında Takip ve Geliştirme projesinde SOA'ya ait projeler için yapılmaktadır. Örneğin SOA çalışmalarının analizi, öğrenilen dersler ve tecrübeler ışığında yeni düzenlemelerin sağlanması ve bunların uygulamaya geçirilmesi, Tablo 5.3'de görüldüğü gibi, CMMI Kurumsal Yenilik ve Yayma alanı altında yer alan İyileşmeleri Seç (Select Improvements) ve İyileşmeleri Yay (Deploy Improvements) gibi hedeflere karşılık gelmektedir.

CMMI'ı rehber olarak aldığı anda özellikle bu konuda belirli bir olgunluk seviyesinin oluşturulmadığı kurumlarda Tablo5.3'de de görülen SOA uygulama projesi ile hayata geçirilebilecek süreçler söz konusudur. Bu süreçler PRINCE2 ile birlikte kısmen ya da doğrudan hayata geçirilerek SOA uygulamasına katkısı sağlanabilir.

Tablo 5.3. CMMI-DEV'in SOA uygulamasına pratik katkıları.

Servis Odaklı Mimari Yaklaşım	CMMI Metodolojisinin Katkı Noktaları
İş Gereklerine Uyumluluk	Gerekler Yönetimi, Gerekler Geliştirme
Değişiklikleri Yönetme	Doğrulama, Onaylama ve Konfigürasyon Yönetimi
SOA Kavramının Yerleştirilmesi	Kurumsal Eğitim
Tasarım ve Geliştirme Süreci	Kurumsal Süreç Odağı, Kurumsal Süreç Tanımı
SOA Takip ve Geliştirme	Kurumsal Yenilikçilik ve Uygulama

6.1.1. SOA Rol ve Sorumluluklarının Belirlenmesi

Bir SOA uyarlaması projesine, Bölüm 3.2 ve Bölüm 4'te anlatıldığı gibi, PRINCE2 organizasyon temasının önerdiği süreçte rol ve sorumluluklar belirlenir. PRINCE2 yaklaşımına göre 3 seviyeli bir organizasyon söz konusudur:

- Koordine Etme – Proje Kurulu: Kurulda BT başkanı (CIO) en azından proje destekçisi olarak, BT Mimarı, iş tarafından başkan yardımcısı, BT/iş strateji müdürü gibi personel yer alır.
- Yönetme – Proje Yöneticisi: Projenin günlük olarak işleyişini takip edip yönetecek kişidir.
- Gerçekleştirme – Takım Lideri: Proje yöneticisi tarafından kendi takımına atanan çıktılarının gerçekleştirilmesini sağlar. Proje yöneticisinden iş alır ve ona rapor verir.

Bu katmanlara göre SOA uyarlamasında bir SOA Mimarı tüm SOA programının ve altında yer alan projelerin kurullarında yer alır. Benzer şekilde diğer kurul üyeleri SOA uyarlamasının aşamalarından haberdar olur ve PRINCE2 süreçlerinde kendilerine atanan sorumlulukları yerine getirir.

Tablo 6.1'de verilen bir proje organizasyonu örneğinde görüldüğü gibi, organizasyonun en üst katmanında yer alan proje kurulu, SOA uyarlama programının tümünde rol almaktadır. Proje yöneticisi ise bu örnekte programın ilk projesi olan “SOA Başlangıç Projesi” sorumlusudur. Bu yöneticinin yeterlilik durumuna göre diğer SOA projelerinde rol alması mümkündür.

Aşağıda Tablo 6.2'de yer alan rol ve sorumluluklar Tablo 6.1'de yer alan Takım Liderleri altında görev yapar. PRINCE2 prensiplerine göre bu grupların belirlenmesi öncelikli bir faaliyettir. Bu grupların ayrıca gelecek SOA projelerinde yer alması tercih sebebidir. Bunun nedeni, bu çalışanların, SOA'nın kavramsal kısmının ve eğitimlerinin yer aldığı ve SOA kültürünün ve teknolojisinin yerleştirildiği aşamalarda yer almalarıdır.

Tablo 6.1. Bir SOA program ve proje organizasyonu örneği.

Proje Organizasyon Katmanları	Başlık	İsimler
Proje Kurulu	BT Başkanı (CIO)	Talip Soydaner
	BT Mimarı	Yasemin Oygan
	BT/İş Strateji Geliştirme Müdürü	Hasan Birkan
	BT Uygulamaları Müdürü	Koray Metiner
	Kurumsal İş Geliştirme Müdürü	Saygın Rasim
Yönetme	Proje Yöneticisi	Aylin Emirhan
Takım Lideri	Yazılım Geliştirme Lideri 1	Necip Mulalı
	Yazılım Geliştirme Lideri 2	Rozin Betjinov
	SOA Teknolojileri Lideri	Kartal Aygun

Tablo 6.2. SOA başlangıç projesinde yer alacak çalışma grupları.

Ekip	Rol ve Sorumluluklar	Personel
Yazılım Geliştirme Ekibi 1	Yazılım Geliştirme Analiz ve Tasarım	Jale Ersen Kaan Seyri Leyla Usta
Yazılım Geliştirme Ekibi 2	Yazılım Geliştirme ve Entegrasyon	Ruslan Mihmandar Tarkan Soma Onur Orhan Turgut Sezginer Dursun Hasır
SOA Teknolojileri Ekibi	SOA Teknolojik Altyapı Kurulum ve Sürdürme	Osman Yılmaz Yalçın Erkenci

6.1.2. SOA Eğitimi

PRINCE2 programı ve proje yaklaşımına göre oluşturulan SOA Programı altında ilk olarak uygulamaya konacak SOA projesinde yer alan eğitimler aynı zamanda CMMI 3. Seviye süreç alanlarından biri olan Kurumsal Eğitime karşılık gelmektedir.

CMMI'nın bu süreç alanına göre hedefler ve pratikler aşağıdaki gibi sıralanır [24]:

SG1 (Specific Goal) Kurumsal eğitim kabiliyetinin oluşturulması

SP 1.1 (Specific Practice) Stratejik eğitim ihtiyaçlarının belirlenmesi

SP 1.2 Kurumun sorumluluğunda olması gereken eğitimlerin belirlenmesi

SP 1.3 Kurumsal eğitim planının oluşturulması

SP 1.4 Eğitim yeteneğinin oluşturulması

SG2 Gerekli eğitimin verilmesi

SP 2.1 Eğitimin verilmesi

SP 2.2 Eğitim kayıtlarının oluşturulması

SP 2.3 Eğitimin etkinliğinin ölçülmesi

Yukarıda CMMI belirli hedef ve pratiklerini dikkate alınması ile eğitimin içeriği hedef gruplarına göre belirlenmiştir:

1. SOA kavramının iş tarafına katkı noktaları: SOA'nın iş tarafına sağlayacağı ya da sağlaması beklendiği konuları anlatan eğitime proje kurulu başta olmak üzere Tablo 6.1'de adı geçen tüm elemanların katılması sağlanır.
2. SOA kavramının temel konuları ve nitelikleri: Bu eğitim SOA'nın temel konularını, prensiplerini ve özellikle BT tarafında getireceği yenilikleri ve faydaları anlatır. Tablo 6.1'deki katılımcıların büyük bölümü de bu eğitimin verildiği kesimdir. Kurulda yer alan CIO ve BT mimarının ve proje yönetici ve takım liderleri bu eğitime katılırlar. Bu eğitime ayrıca tüm Tablo 6.2 personeli katılır.
3. SOA geliştirme teknolojileri: Bu eğitim daha çok SOA'nın teknolojik yanını anlatır. Geliştirme ve uygulama tarafındaki geliştirme ortamları ve araçları takım liderlerine ve Tablo 6.2'de yer alan geliştirme ekiplerine verilir.
4. SOA ortamı için altyapı oluşturma ve sürdürme: SOA'ya gerekli ortamı sağlayan uygulamaların kurulması, bakım ve sürdürülmesinin anlatıldığı bu eğitim, tercihen takım liderleri ve Tablo 6.2'de yer alan teknoloji ekibine verilir.

Bu eğitimler, CMMI Kurumsal Eğitim SP 1.3, SP 2.2 tüm organizasyonun ve BT ekibinin genel eğitim plan ve arşivlerine kayıtlarının yapılması sağlanır. Eğitim sonrası katılımların ve katılım sertifikalarının ve eğitim kalite değerlendirmeleri de sağlanarak

kayıt altına alınır. Bu aşamada verilen eğitimlerin pratikte ne kadar kullanıldığı SOA uygulama programında yer alan takip ve geliştirme projesi içinde değerlendirilecektir notu düşülerek eğitim faaliyeti tamamlanır.

6.1.3. SOA Prensiplerinin Oluşturulması

Kurumdan kuruma öncelik ve çeşitliliği değişebileceği gibi genel anlamda SOA uygulaması başlangıcında vurgulanacak temel konular aşağıda sıralanacağı gibidir. Zaman içinde ve elde edilen tecrübeler ışığında bu konular gözden geçirilebilir, yeniden önceliklendirilebilir ya da bunlara yenileri eklenebilir.

Çeviklik: SOA'nın en değerli faydalarından biridir. Projelerinin tümünün ortak zorluklarından biri değişimdir. Proje inşaat ve medya gibi bir sektörde de olsa değişim söz konusu iken soyut, esnek ve çok hızlı değişen tetikleyici faktörleriyle yazılım sektöründe değişim sürecinin yetersiz alınması durumunda başarısızlık oranı yüksektir. Kurumsal yapıların günlük, saatlik hatta saniyelik değişebilen beklentilerini karşılamak için buna cevap verecek çeviklikte uygulamalar için hem metodolojik hem de teknolojik olarak hazır olunmalıdır. SOA uygulamaları da, yeni iş gereksinimlerine göre en hızlı şekilde bekleneni yapıyor olmalıdır.

Yeniden Kullanılabilirlik: SOA bileşenlerinin gelecekteki kullanılabilirliği ve eldeki mevcut bileşenlerin kullanılabilirliğinin gözden geçirilmesi bir prensip olarak kabul edilmelidir. Aslında servislerin ve benzeri bileşenlerin gelecekte kullanımını da öngörerek doğru tasarımın yapılması zor ama bir o kadar da değerlidir. Ayrıca var olan bileşenler de istenilen özellikte olmayabileceğinden ek çalışma gerektirebilir. Önemli olan servislerin ve benzeri SOA bileşenlerinin yeniden kullanılabilirlik olasılığını göz önünde tutularak tasarlanması esas olan yaklaşımdır.

Yatırımın Geri Dönüşü (Return Of Investment, ROI): ROI, BT'nin problemleri konularından biridir. Çünkü kurum içinde BT, donanım, yazılım ve insan kaynağı olmak üzere maliyeti yüksek bir birimdir. SOA uygulamasının bu konuda da iyileştirmeler getirmesi gerekmektedir. Bunun için normal tarzda yapılan projelerle SOA geliştirme projeleri finansal açıdan incelenmeli ve aradaki fark ortaya konulmalıdır. SOA kendi içinde daha az maliyetli olmak durumundadır. Bunun nedeni, azaltılmış benzer

uygulamalar, kısa geliştirme süreleri, daha az maliyetli entegrasyon, bakım ve destek gerektirmesidir.

Entegrasyon: SOA'nın vazgeçilmez bir gereğidir. Tekrar kullanılabilirlik, basitlik, esneklik gibi özelliklerin ortaya çıkması için entegrasyon şarttır. SOA'nın yapı taşlarından biri olan entegrasyon için gerekli altyapı ve tasarım şartlarının sağlanması gerekir.

Güvenlik: SOA mimari olarak katmanlı bir yapıya sahiptir ve bu nedenle her katman için güvenlik kendi içinde düşünülmelidir. Bu hem tasarım aşamasında hem de destekleyici teknolojilerin oluşturulması sırasında dikkatle ele alınması gereken bir konudur.

Hizmete Hazır Bulunma (Availability): SOA'nın getirdiği avantajlardan biri de hizmete hazır olma kabiliyetinin yüksek olmasıdır. Hazır olabilirlik servislerin önem ve kullanım gibi belirleyici faktörleri göz önünde tutularak teknoloji ve tasarım çalışmalarında göz önünde bulundurulmalıdır. Bu şekilde örneğin önemli bir hizmeti sağlayan bir servisin yedeği ya da yedekleri olmalı ve bunu kullanacak yedekleme mekanizması sağlanmış olmalıdır.

6.1.4. SOA Teknoloji Beklentilerinin Oluşturulması

Bu bölümde ise bir önceki bölüme benzer fakat daha çok teknolojik açıdan ele alınacak konular yer almaktadır. Teknik anlamda farkındalık oluşturulacak ve hem altyapı hem de geliştirme aşamalarında SOA projelerinde aranacak teknik özellikler aşağıda sıralanmıştır. Basitlik: SOA genel anlamda endüstri standartlarına dayanan bir yapıya sahiptir ve çözüm açısından bakıldığında diğer entegre sistemlere göre kompleksliği azaltacak şekilde tasarlanabilir. Gelecekte oluşturulacak uygulamalara da uyumluluğu yüksek olacaktır.

Esneklik: SOA yeni nesil bileşik (composite) çözümleri desteklemektedir. Bu performans odaklı çözümler, çeşitli sistemlerden faydalanarak birçok iş sürecini basit bir arayüzde birleştirebilir. Bu tarz bir uygulama örnek olarak verilecektir.

Bakım Kolaylığı: SOA'nın modüler ve çok katmanlı yapısı bakım görevinin de kolay yapılmasını sağlar çünkü bakım çalışmaları daha belirli konulara odaklanabilir. SOA basit yapısı sayesinde daha hızlı ve basit bakım yapılabilir.

Kaynak Kodu Taşınabilirliği: SOA katmanlar, servisler ve bileşenleri arasında gerçek bir saydamlık sağlar. Özellikle SOA teknoloji yatırımına yedeklilik aracı da eklenmişse bir servis değişik bir sebeple (çökme, bakım, geliştirme, emekliye ayırma...) devre dışı kalmış ise başka bir sistem ya da sistemlerde konumdan da bağımsız olarak alınan yedek servis ile iş süreçlerinin devamlılığı sağlanabilir. Böylece servislerin farklı sistemlere taşınması veya dağıtılması istemci tarafından görünmeksizin yapılabilir.

Geliştirme Kolaylığı: Bir SOA tabanlı uygulama çok katmanlıdır. Bu sayede her katman kendine has rol ve sorumlulukları olan ayrı bir geliştirici grubuna atanabilir. Bu ayrıca organizasyonlara daha belirli konularda ve daha az deneyimli geliştiriciler kullanarak maliyet ve kaynak avantajı sağlayacaktır.

Servisler: Aslında bir dönem SOA denince akla gelen servis konusu gerçekten SOA'nın tamamı olmasa da önemli bir temel ögesidir. Servisler bağımsız çalışabilirler ve bağımsız test edilebilirler. Geliştiriciler tarafından birim (unit) testlerinden kolaylıkla geçirilebilirler. Bu şekilde bir uygulama kalite güvencisi testinin birim testlerinden önce yapmaya gerek kalmaz.

6.1.5. Mevcut Sistemlerin Analizi

Kurumda mevcut olan süreçler, kurumun öncelikleri, hedefleri zayıf ve güçlü yanları analiz edilir. Bu daha çok teknik olmayan kısımdır. Teknik açıdan bakılacak olursa kurumun sahip olduğu yazılımlar, servisler, donanımlar incelenmeli ve bunların nasıl değerlendirilebileceği ve SOA tarafından kullanılıp kullanılmayacağı, kullanılırsa neler yapılması gerektiği belirlenmelidir. Bunun fizibilitesinin ne olacağı gibi konular ele alınarak bunların dokümanite edilip gerektiğinde başvurulması gibi konuları içerir [21].

6.2. SOA Teknolojik Altyapı Projesi

Bu projede SOA tabanlı uygulamanın gerçekleştirilmesi için gerekli olan altyapı araştırılacak ve bu araçlardan beklentilerin neler olduğu üzerinde durulacaktır.

Daha önce gelen SOA Başlangıç projelerindeki prensipler göz önünde tutularak, sektörde öne çıkan ürünlerin incelenmesi organizasyon için uygunluğu da göz önünde tutularak yatırım yapılması ve kurulması süreçleri de bu projede gerçekleştiriliyor olmalıdır.

Bu tezde olgunluğu ve pratikliğinin yanında ücretsiz oluşu da göz önüne alınarak SUN NetBeans IDE - SOA ortamı kullanılmıştır. Bu üründe kullanılan teknolojiler SOA dünyasının standartlarını karşılamaktadır. Ayrıca bu ürüne benzer araçlar kurumsal ortamlarda kullanımı her yönden de desteklenir şekilde sunulmaktadır.

6.2.1. NetBeans IDE 6.x

Bu tezde geliştirilen örnek SOA uygulamasında NetBeans'in 6.x sürümü kullanılmıştır. Bu tümleşik geliştirme ortamı (IDE), SOA süiti içermektedir. Bu uygulama ve eklentileri <http://www.netbeans.org> adresinden indirilebilir. Bu ortam GlassFish Uygulama Sunucusunu da içerir. Kurulum sırasında bu sunucunun seçilmesi gerekmektedir.

NetBeans ortamı açık kaynak kodlu birçok SOA aracına sahiptir. Bunlar Sun'ın Java Studio Enterprise ve Java CAPS gibi bileşik uygulama oluşturulmasını sağlayan, BPEL tabanlı Web Servisleri, güvenli Java EE Web Servisleri, XML şema ve WSDL araçlarını içerir. Bu araçlarla:

- Görsel Tasarım: XML şemalarında BPEL tasarım aracı kullanarak görsel tasarımlar yapılabilir.
- Test: Ortam içinde oluşturduğunuz uygulamanın testi yapılabilir.
- Güvenli Uygulamalar: Güvenli kimlik çözümlerini ortam sağlamaktadır.
- Entegre BPEL Motoru: Web servisleri için ayrıca uygulama sunucusu gerektirmemektedir.

GlassFish ve Sun Java System Application Server ortamı ile entegre çalışabilmektedir. BPEL motoru JBI (Java Business Integration) servis motoru ile iki sunucuda da kullanılabilir.

6.2.2. Bileşik Uygulamalar ve JBI

Bir bileşik uygulama (composite application), çeşitli altyapı sistemlerinden elde edilen servis, iş mantığı ve verilerinin kullanılarak belirli bir iş sürecini gerçekleştirmek amacıyla biraraya getiren bir çözümlene şeklidir. Tipik olarak bir bileşik uygulama, bir veya birkaç süreci, çeşitli süreç adımlarını bir araya getirebilir ve bunları iş gereksinimine göre tek bir arayüzden istemciye sunabilir.

SOA, bileşik uygulamayı anlatırken, istemci ve tedarikçi bileşenlerini bir araya getiren ve bir iş sürecini uygulayan ve konum şeffaflığı sağlayan uygulamalar olarak nitelendirir. Bu çalışmada JBI modülünü kullanan bir bileşik uygulama örnek olarak alınmıştır. JBI, mesaj değişim modeli aracılığıyla iletişim kuran çalışma zamanı mimarisini tanımlayan Java tabanlı bir standarttır. JBI, Sun Java System uygulama sunucusu ile önceden konfigüre edilmiş yaşam döngüsü modülü olarak entegre olarak çalışır.

6.2.3. Servis Motorları

Servis motorları, istemcilerine iş mantığı fonksiyonelliği sağlayan, istemci ya da sunucu olabilen WSDL üzerinden iletişime geçen entegrasyon araçlarıdır. Servis motorları Java iş entegrasyonu sunucusu üzerinde çalışır. Kurum servis yolu (ESB) üzerinden haberleşir.

6.2.4. Bağlama Bileşenleri

Bağlama bileşenleri (binding components), diğer JBI bileşenleri için geçiş ve protokol bağımsızlığı sağlarlar. JBI kabı, ayrık ve güçlü altyapı sağlayarak geliştirici ve entegratörlerin dayanıklı SOA uygulamalarını mümkün kılar.

FTP, dosya, SOAP, JDMS, JMS ve SMTP gibi birçok bağlama bileşeni vardır. Bunlar uygulama sunucusuna kurularak çalışır hale getirilir. Bağlama bileşenleri arayüzlerin standart setleri şeklinde geliştirilmiştir ve herhangi bir JBI uyumlu bağlama bileşeni uygulama sunucusuna kurulabilir.

6.2.5. BPEL Tasarımcısı

SOA, BPEL (Business Process Execution Language) gibi standartlara dayanan teknolojileri kullanarak sistemlerin geliştirme, yönetme ve dağıtmasını kolay hale getirir. BPEL endüstrisinde yaygın olarak kullanılan, zengin web servisi tabanlı ve organize edici, yönetici özellikleri olan bir teknolojidir. İş süreçleri olarak soyut ve işletilebilir şemaların üretilmesi ve uygun bir motor üzerinde çalıştırılmasını sağlar.

6.2.6. WSDL Editörü

WSDL (Web Services Definition Language), servisleri, servislerin konumlarının, ne tür operasyonlar sunduklarının tanımlanmasında kullanılır ve platform ve lisan bağımsız çalışırlar. Bu, kurumsal uygulama geliştiricisine zayıf bağlantılı (loosely coupled) servisler oluşturmaya ya da kullanmasına imkân tanır. Bu başarılı bir SOA uygulamasının anahtar ilkelerinden biridir.

NetBeans geliştirme ortamı WSDL dokümanı oluşturma ve düzenleme yeteneğine sahiptir. Ayrıca iki tip grafik arayüz ile WSDL dokümanını sunabilir.

6.2.7. XML Şema Tasarımcısı

Son yıllarda XML farklı sistemler arasında veri alışverişinin baskın teknolojisi haline gelmiştir. Düz dosyalar, genişleyebilir verileri ve kendini tanımlayabilir olması XML'in avantajlarıdır.

XML Şeması XML dokümanlarının sınıfları için XML dağırcığını tanımlama, tarif etme ve sınıflandırılmasında kullanılır. XML dosyalarını makine tabanlı değerlendirmesine izin veren tanımlamaları üretir. NetBeans XML şema tasarımcısına sahiptir [25].

6.3. SOA Takip ve Geliştirme Projesi

Bu proje belirli dönemlerde tekrarlanabilir ya da daha az adam gün harcayarak uzun süreli ve aşamalı hale de getirilebilir. Bu tezde oluşturulan yol haritasında bu tarzda bir kısa süreli proje görünmektedir. Bu projedeki SOA prensipleri revize edilir, eksiklikleri giderilir, gerekli uyarılar ve eğitimler verilir. Uygulanan SOA projeleri değişik araçlarla ölçülür, değerlendirilir ve gerekli SOA'yı iyileştirme adımları atılır.

Bu proje aynı zamanda SOA programı altında yer alan diğer projelerin kalitesinin değerlendirildiği ve kattığı değerlerin incelendiği ve PRINCE2 “öğrenilen derslerin” toplanıp incelendiği faaliyet öğelerini taşımaktadır. Yapılan bu çalışmalar ardından SOA iyileştirme adımları belirlenip mevcut proje içinde ve diğer projelerde uygulanacaklar diye iki ana kategoride ayrılır. Birince kategoridekiler hemen uygulanır. Diğerleri ise zaman içinde yürütülecek SOA iş çıktısı hedefli projelerde dikkate alınır. Böylece en başta yer alan ve SOA kavramsal ve teknik altyapısını oluşturmaya yönelik çalışmaların üzerine yeni bilgi ve tecrübeler eklenerek SOA olgunluğunun artması sağlanır.

6.4. İş Çıktısı Odaklı SOA Projesi

SOA altyapısı ve prensipleri oluşturulduktan sonra iş tarafının gereksinimlerini karşılayan projeler için ortam sağlanmış olur. Bu tür iş gereksinimini karşılayan örnek bir proje bu bölümde anlatılacaktır.

Bir otel, müşterilerine daha iyi hizmet vermek istemekte ve hizmet ağını genişletmeyi hedeflemekte ve firma müşterilerine kendi otelleri yanında anlaşmalı olarak birlikte çalıştığı otellerin odalarını seçme imkânı sunmak istemektedir. Bunun için diğer otellerin farklı konum ve teknolojik altyapılardaki mevcut sistemlerini de kullanacak bir yazılım kullanmaya karar vermiştir. Bu yazılım aynı zamanda gelecekte eklenebilecek yeni otellerin de rahatlıkla entegre olabileceği bir sistem olmalıdır. Bu sistem rezervasyon süreci ile başlayacaktır ve muhasebe, CRM, yardım masası gibi konularda da ortak çalışma ve yeni entegrasyonlara cevap verebilmelidir.

6.4.1. Gereksinim Analizi ve Servis Odaklı Modelleme

İş gereksinimlerinin belirlenmesi ve servis odaklı modellenmesi PRINCE2 metodu içinde doğrudan ele alınmaz. Bunun nedeni metodun jenerik olmasından kaynaklanmaktadır. Bu durumları karşılamak üzere geliştirilen “özel açılar veya konular” (special aspects) olarak nitelendirilen kategoride ürün ve çıktıya sektör ve konuya göre gereken ek çalışmaların yer almasıdır. CMMI ise yazılım geliştirme süreçlerinde 2. seviyede Gereksinim Yönetimi ve 3. seviyede Gereksinim Geliştirme süreç alanları ile bu konuda detaylı çalışmaların yapılmasını beklemektedir. Dolayısıyla

basit bir yaklaşımla PRINCE2'nun eksik olduğu alanı CMMI oldukça iyi doldurabilecek durumdadır.

Buna ek olarak SOA'nın kendine has mimari yapısının göz önüne alındığı gereklilik analizi çalışmalarından faydalanılması daha verimli bir proje ortaya konması açısından gereklidir. İş tarafı gereksinimlerinin belirlenme süreci servis odaklılık açısından bakıldığında servis odaklı analizin alanıdır.

Servis odaklı analizin amacı iki önemli soruyu cevaplamaktır:

- Hangi servisler oluşturulmalıdır?
- Hangi servis tarafından hangi iş mantığı kapsanmalıdır?

Genel anlamda servis odaklı analizi uygulamanın hedefleri aşağıdaki gibidir [6]:

- Başlangıç servis operasyon aday setlerini belirlemek.
- Servis operasyon adaylarını mantıksal olarak gruplamak. Bunlar servis adaylarını temsil ederler.
- Servis sınırlarını belirleyerek mevcut ve planlanan servislere taşmamalıdır.
- Yeniden kullanım potansiyeli ile kapsanan mantığı belirlemek.
- Kapsanan mantığın içeriğinin amacında kullanılması için uygunluğunu sağlamak.
- Bilinen başlangıç bileşen modelleri belirlemek.

Var olan bir BT ortamında yeni bir analiz sürecini gerçekleştirmeye çalışmanın kendine göre zorlukları vardır. Çünkü her organizasyon kendi problemlerine ve çözümlerine sahiptir ve yerleşmiş süreç ve dokümantasyonu yapılmış yatırımları söz konusudur. Dolayısıyla burada bahsedilen süreç, mevcut süreçleri kaldırmak değil onları destekleme amacına sahiptir.

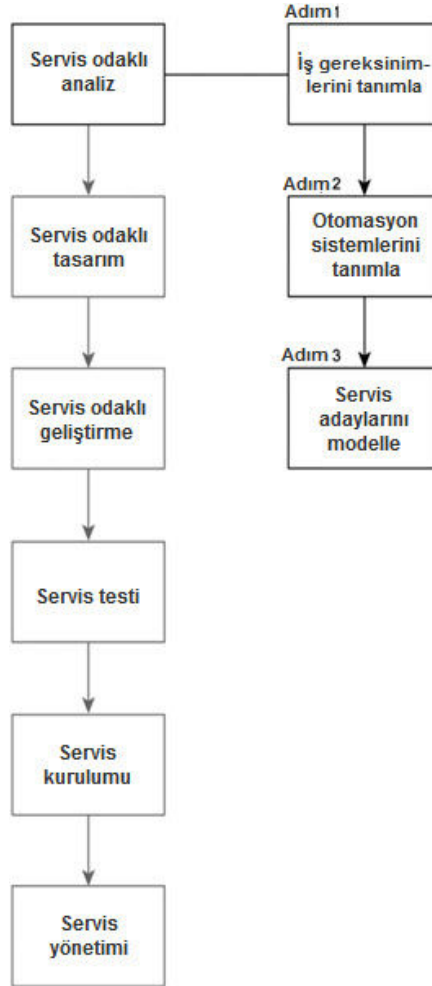
Servis odaklı analiz farklı seviyelerde uygulanabilir. Bu servisleri üretmek için hangi SOA yaklaşımının seçildiğiyle ilgilidir. Bu seçim, çözüm ortamının katmanlarını kapsayan soyutlama seviyesini belirler.

Analiz açısından bakıldığında her katman kendine özgü modelleme gereksinimine sahiptir. Örneğin uygulama servislerini tanımlamak için yapılan analizler, iş servisi

katmanındakilerden farklıdır. Bu nedenle süreci belirleyen daha önce sözü edilen SOA stratejisidir. Bunun yanında servis odaklı analiz aşağıdaki konuları içerir:

- Gerekli iş modelini ve ontolojisini kurabilmek için gerekli işler nelerdir?
- Analizi yapacak modelleme aracı hangisidir?
- Analiz SOA geçişinin (ya da gerçekleştirilmesinin) bir parçası olacak mıdır?

Dikkat edilecek olursa bazı sorular projenin kapsamıyla da ilgilidir. Büyük projelerde bunun için ayrıca bir proje yapılması da gerekebilir. Bu tezde ise bu ilk projede ele alınmıştır. Şekil 6.2’de servis odaklı analiz süreci genel bir yaklaşımla gösterilmektedir.



Şekil 6.2. SOA analiz süreci.

Adım 1 - İş otomasyon gerekliliklerinin belirlenmesi: İş gereksinimleri normal şekilde toplanır. Bunların dokümantasyonları analiz sürecinin başlaması için gereklidir. Servis odaklı çözümün desteklenmesi dikkate alınarak analizin kapsamı servis oluşturulması kapsamında tutulur ve sadece bu kapsam içindeki gereklilikler hesaba katılır.

İş gereksinimi üs seviye süreç otomasyonunun tanımlanabilmeyi mümkün kılacak olgunlukta olmalıdır. İş süreci dokümantasyonu 3. adımda tarif edilen servis modelleme sürecinin başlangıç noktasında kullanılacaktır.

Adım 2 – Var olan otomasyon sistemlerinin belirlenmesi: Mevcut iş otomasyon süreci 1. adımdaki gibi belirlenmelidir. Servis odaklı analiz, web servislerinin neyi kapsayacağı ya da eski sistemlerde neyi değiştireceğini belirlemediğinden potansiyel sistem etkilerinin sınırlarını kestirmeye yardımcı olmaz.

Mevcut sistemlerle web servislerinin nasıl ilişkili olacağı servis modelleme sürecinde yer alır. Bu bilgi 3. adımda, uygulama servisi adaylarını belirlemede yardımcı olur.

Adım 3 – Aday servislerin modellenmesi: Bir servis odaklı analiz hangi servis operasyon adaylarının belirleneceğini ve hangi mantıklı bağlamlar altında gruplanacağını servis modelleme kavramında ortaya koyar.

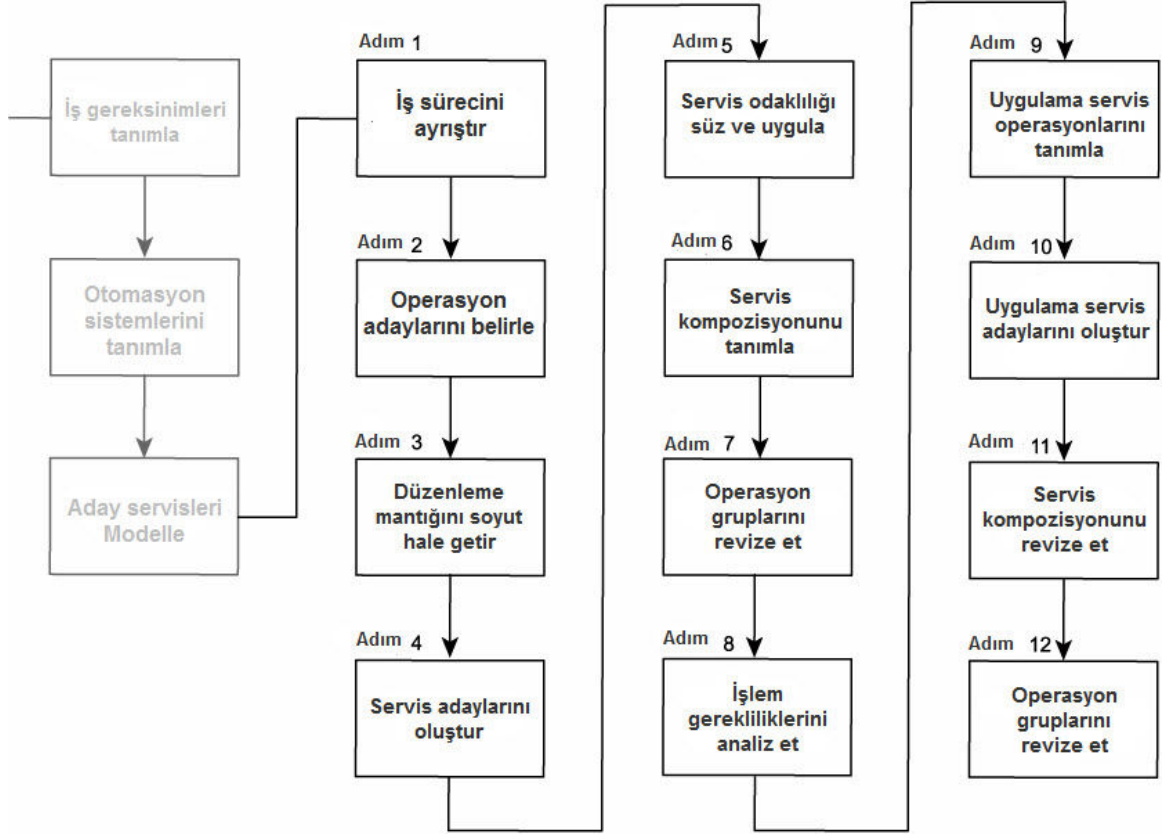
Otel yönetim sisteminde iş otomasyon gerekliliklerinin belirlendiğinde, müşteri web arayüzü, kullanıcı web arayüzleri, yönetici web arayüzü, çevrimiçi rezervasyon, çevrimiçi ödeme, çevrimiçi giriş-çıkış işlemleri gibi başlıca konular olarak ortaya çıkmaktadır.

Mevcut sistemlerde bu işler Yıldız Otel’de fiziksel kayıt defterinde ve muhasebe yazılımında, Güneş Otel’ de ise iki farklı eski otel doluluk ve muhasebe paket programı olduğu belirlenmiştir.

Servis adaylarını belirlemek için belirli süreçler incelenerek otel dışında, otelde iken ve muhasebe ile ilgili gerçekleşen işlemler servis adayı olarak belirlenmiştir.

6.4.2. Adım Adım Servis Modelleme

İlk aşamada servisler üretilmez; servis adayları üretilir. Aynı şekilde servis operasyonları tanınmaz, servis operasyon adayları öngörülür. Servis modelleme sürecinin sonucunda çıkan ürünler servis tasarım sürecinin hammaddesi olarak kabul edilir. Şekil 6.3'te servis modelleme adımları verilmektedir [1]:



Şekil 6.3. Servis modelleme süreci.

Adım 1 - İş sürecinin ayrıştırılması: İş süreci parçalara ayrılarak bir süreç akış mantığı oluşturulur.

Adım 2 – Servis operasyon adaylarının belirlenmesi: Bazı iş süreçlerinin bir servis adayı olmaya uygun olmadığı kolayca anlaşılır. Bu şekilde eleme yapılır.

Adım 3 – Düzenleme mantığının soyutlanması: Bir düzenleme katmanı söz konusu ise işleyen mantığın soyutlanması gerekir. Böylece iş kuralları, şarta bağlılık, istisna ve sıralı mantık durumları bu katman için uygun adaylardır.

Adım 4 - İş servis adaylarının oluşturulması: Mantıksal bağlamda gruplanabilen işlem adımları araştırılır. Her bağlantı bir iş servisi adayını temsil eder.

Adım 5 – Servis odaklılık prensiplerini arıtma ve uygulama: Servis odaklılık prensipleri bu adımda uygulanır. Yeniden kullanılabilirlik, bağımsızlık, durumsuzluk, keşfedilebilirlik özellikleri bu aşamada uygulanır.

Adım 6 – Servis kompozisyonlarının belirlenmesi: Bir düzenleme katmanı söz konusu ise işleyen mantığın soyutlanması gerekir. Böylece iş kuralları, şarta bağlılık, istisna ve sıralı mantık durumları bu katman için uygun adaylardır.

Adım 7 - İş servis operasyon gruplarının gözden geçirilmesi: bir önceki adımın sonuçlarına göre servis operasyon gruplaması gözden geçirilir.

Adım 8 – Uygulama işleme gereklilikleri analizi: 6. Adımdan sonra iş odaklı bir servis katmanı oluşturulmasına başlanmıştır. Bu görünüm uygulama ve iş servisi adaylarını doğru bir şekilde sağlayabilir ama odak daha çok iş süreci mantığı üzerindedir.

Bundan sonraki adımlar seçimliktir ve tercihen çok geniş SOA ortamlarına uyar. Bu nedenle kısaca değinilecektir.

Adım 9 – Uygulama operasyon adaylarının belirlenmesi: Her bir uygulama mantığı işletim gereksinimlerinin adım serilerine ayrılmasıdır.

Adım 10 - Uygulama operasyon adaylarının oluşturulması: Bu işletim gereksinimlerinin önceden belirlenmiş bağlama göre gruplanmasıdır.

Adım 11 – Servis kompozisyon adaylarının gözden geçirilmesi: 5. Adımda belirlenmiş orijinal senaryonun faaliyetlerinin haritası yeniden incelenir ve be kez yeni uygulama servis adayları da incelemeye katılır.

Adım 12 – Servis operasyon gruplarının gözden geçirilmesi: Bir önceki adımdaki faaliyet senaryolarının üzerinden geçilince muhtemel yeni servis operasyon grupları ortaya çıkarılmasıdır.

Birinci seviyedeki adımlarla ortaya çıkan servis adayları üzerinden giderek iş süreci ayrıştırıldığında örnek olarak rezervasyon sürecinin web arayüzünden öncelikli otel seçeneklerinin sunulması, müşteri bilgilerinin alınması tercih ve konum tespitinin yapılması, boş yerlerin bu bilgilere göre sıralanması ve görüntülenmesi, temel

rezervasyon bilgilerinin alınması, bunların veri tabanına kaydedilmesi, müşteri ve resepsiyon sorumlusuna bildirilmesi şeklinde iş parçaları olduğu görülmektedir. Bunları operasyon olarak (nesne tabanlı mimaride fonksiyon ya da metot) ayrıştırıldığında örneğin verilerin algoritma boyunca mantıklı sırada okunması, bir operasyon olarak belirlenir.

Aralarında çalışacak mantıksal algoritma “yok ise sıradaki diğer adım” şeklinde işler örnekte BPEL tarafından organize edilecek şekilde modellemeye devam edilmiştir. Rezervasyon süreci geliştirilen yazılımda bir servis olarak ele alınmaya karar verilmiştir. Bu servis, servis odaklı mimarinin soyut, yeniden kullanılabilen gibi prensiplerine göre incelenerek servisin genel olarak her tür rezervasyon beklentilerini karşılayacak özellikler, örneğin kat tercihi, kampanya özelliği gibi sektörde ileride lazım olabilecek sorguları da içermesi istenmiştir. Ayrıca 3 adet yer ayırma da sorgulanabilecektir. Arayüzlerde geçen değişkenler yeniden gözden geçirilerek en iyi soyutlamanın araştırılması yapılmıştır. Örneğin, para ve tarih olarak belirlenen değişkenler karakter dizisi gibi okunup yazılarak ülke ve birim farklılıkları ayrıca alınarak karışıklıkları ortadan kaldırma hedeflenmiştir.

Bir sonraki adımda servislerin birlikte kullanılabilirliği rezervasyon ile ödeme ve otel giriş çıkış işlemlerinin birlikte nasıl çalışacağı belirlenmiştir. Bu belirlemelere göre rezervasyon servisinin neleri içermesi ve nasıl davranması gerektiği yeniden gözden geçirilmiştir. Bu durumda muhasebe kayıtlarında borçlu görünen bir müşterinin otel çalışma şekline göre ilgilendiği yüksek ücretli otellerde yer kalmadığının gösterilmesi için düzenleme istenmiştir.

Son olarak istenen tüm iş gereklerinin karşılandığı görülerek, rezervasyon servis olarak tanımlanmıştır. Bu servisin pratikte nasıl oluşturulduğu gerçek yazılım ortamını anlatan örnekte verilmektedir.

6.4.3. Servis Odaklı Mimaride Değişiklik Yönetimi

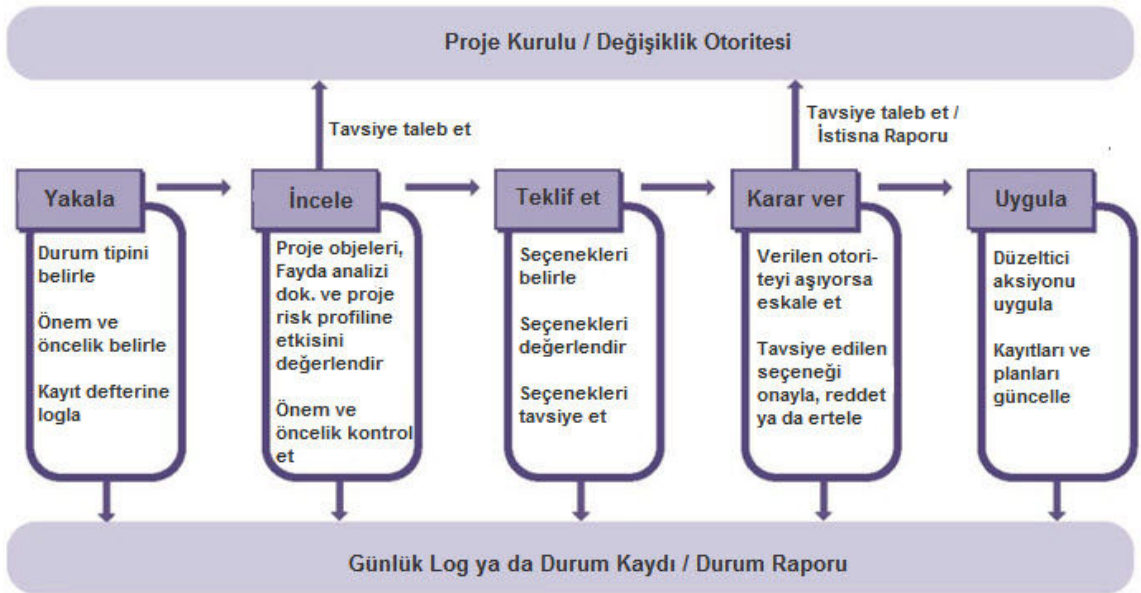
İş gereklerinin belirlenmesi ile planların güncellenmesi ve ardından da geliştirme çalışmaları başlatılmıştır. Bu proje bir iş çıktısı veren bir projedir ve analiz ne kadar iyi yapılırsa yapılsın, iş gereksinimlerinin değişmesi, teknik sıkıntılar, gibi birçok sebeple projenin devamı sırasında değişiklikler kaçınılmaz hale gelir. Bu tarz durumlar projenin

gecikmesi, yanlış tamamlanması, müşteri tarafından kabul edilmemesi veya tamamen başarısız olmasına neden olabilecek ciddiyette olabilir. Bu nedenle değişikliklerin yönetilmesi için hem genel proje metodolojilerinde hem de yazılım geliştirmeye has metodolojilerde değişik mekanizmalar geliştirilmiştir. SOA, hızlı değişen iş gereksinimlerine hızlı cevap verebilmesi avantajı ile öne çıkar ve bunu daha projenin devamı sırasında da en başarılı şekilde gerçekleştirebilmelidir. Bu durumda SOA'nın mimari yapısının katkısının yanında, iyi bir süreç ile değişiklikler yönetilmelidir.

CMMI, değişiklik yönetimini ilgili olduğu alanlara göre ayırmıştır. İş gereksinimleri ile ilgili bir değişiklik "Gereksinimler Yönetimi" sürecinde, sistemle ilgili değişiklikleri "Konfigürasyon Yönetimi" içinde ele almıştır. PRINCE2, daha önce ayrı bir süreçte ele aldığı Konfigürasyon Yönetimini 2009 yılında "Değişiklik Teması" altında ele almaktadır. Bu iki metodolojinin kılavuzluğunda oluşturulan değişiklik sürecine örnek olacak bir senaryo aşağıda gösterilecektir.

Değişiklik Yönetimi Sürecinin Adımları

Değişiklik Yönetimi, Şekil 6.4'de görüldüğü gibi pratik ve tüm durumları kapsayan yapısıyla PRINCE2 Teması ile gerçekleştirilebilir. Süreç 5 adımda gerçekleştirilir [5].



Şekil 6.4. Değişiklik yönetimi süreci.

Değişiklik Yakalama: Değişikliğe konu olan durumun tipinin belirlenmesi, ciddiyet ve önceliğinin belirlenmesi ve kayıt altına alınmasıdır.

Değişiklik İnceleme: Değişikliğin proje hedeflerine, fayda analizi dokümanına, proje risk profiline etkilerinin incelenmesi, ciddiyet ve önceliğinin teyit edilmesidir. Proje kurulunun tavsiyesi istenir.

Değişiklik Teklifi: Seçeneklerin belirlenmesi ve değerlendirilmesi ve doğru seçeneğin tavsiye edilmesidir.

Değişiklik Kararı: Eğer durum delege edilen otoritenin üzerindeyse proje kuruluna aktarılır ya da tavsiye edilen seçenek onaylanır ya da reddedilir.

Değişiklik Uygulama: Gerekli düzeltme aksiyonları alınır. Kayıtlar ve planlar güncellenir. Bütün bu yukarıdaki 5 aşamadaki bilgiler günlük kayıtlara ve raporlara eklenir.

Değişiklik temasıyla ilgili sorumlulukların gösterildiği örnek bir Tablo aşağıda Tablo 6.3'te yer almaktadır. Bu tablo ile süreç boyunca yer alacakların rol ve sorumlulukları açık olarak belirlenir [5].

Değişiklik Yönetiminin Bir Örnekle Gösterilmesi

Bir önceki bölümde geçen süreç, rol ve sorumluluklar, çalışmada SOA programı başında devreye alınarak tüm SOA alt projelerinde kullanılmak üzere kararlaştırılmıştır. Bu doğrultuda Tablo 6.4'te yer aldığı gibi rol ve sorumluluklar atanmıştır. Bu yapılanla birlikte SOA değişiklik yönetimi süreci işlemeye hazırdır.

SOA analiz ve modelleme aşamasında ortaya çıkan verilere göre Güneş Otel Rezervasyon Web Servisi, ilgili ekip tarafından kayıtlardadır. Proje planlarının onayından ve uygulamanın geliştirilmeye başlamasından sonra bu çalışmanın diğer firma tarafından yapılmadığı ve kısa bir zamanda da tamamlanamayacağı ortaya çıkmıştır.

Tablo 6.3. SOA Uygulama projelerinde deęişiklik süreci rol ve sorumlulukları.

Roller	Sorumluluklar
Program Yöneticisi	Deęişiklik ya da konfigürasyon yönetimi için program stratejisini yönetir.
Yönetici	Deęişiklik otoritesini ve deęişiklik bütçesini belirler. Durumlar için ciddiyet ve önceliklerini belirler. Proje Yöneticisinden gelen tavsiye beklentilerine cevap verir. Özellikle iş tarafı odaklı olmak üzere ortaya çıkan anlaşmazlık durumlarında kararlar verir.
Üs Düzey Kullanıcı	Proje Yöneticisinden gelen tavsiye beklentilerine cevap verir. Özellikle beklenen faydalar odak olmak üzere ortaya çıkan durumların kurula çıkarılmasına kararlar verir.
Üs Düzey Tedarikçi	Proje yöneticisinden gelen tavsiye beklentilerine cevap verir. Özellikle tüm çözümün bütünlüğü odak olmak üzere ortaya çıkan durumların kurula çıkarılmasına kararlar verir.
Proje Yöneticisi	Konfigürasyon yönetimi sürecini yönetir. Mümkün olduğu noktada proje destekten faydalanır. Durum ve deęişiklik kontrol yöntemini yönetir. Mümkün olduğu noktada proje destekten faydalanır. Durum Kayıt dokümanını oluşturur ve devamlılığını sağlar. Mümkün olduğu noktada proje destekten faydalanır. Düzeltilici aksiyonlar alır.
Takım Yöneticisi	Düzeltilici aksiyonlar alır.
Proje Güvence	Durumların deęerlendirilmesi ve çözümlenmesinde tavsiyelerde bulunur.
Proje Destek	Konfigürasyon, durum ve deęişiklik kontrol süreçlerini idare eder: Konfigürasyon Öğeleri kayıtlarını güncel tutar. Ürün Durum Hesapları üretir. Durum kayıtlarının takibinde proje yöneticisine yardım eder.

Tablo 6.4. Değişiklik süreci rol ve sorumluluklarının gerçek kişilere atanması

Roller	Personel
Program Yöneticisi	Yasemin Oygan
Yönetici	Hasan Ertunç Kasım
Üs Düzey Kullanıcı	Rikkat Hanlı
Üs Düzey Tedarikçi	Turgay Bezirgan
Proje Yöneticisi	Aylin Emirhan
Takım Yöneticisi	Necip Mulalı, Rozin Betjinov, Vahit Arslangil, Kartal Aygun
Proje Güvence	Cengizhan Sılagüz
Proje Destek	Cüneyt Düzgünel

Değişiklik süreci adımlarında görüldüğü gibi, Proje Yöneticisi Aylin Emirhan bu ortaya çıkan durumu net olarak tanımlar ve kaydeder. Tip olarak konfigürasyon değişikliği olduğu kanaatine varır. Öncelik ve ciddiyet açısından 4'ten 3 ve 4'ten 2 değerlerini verir. İkinci aşamada Proje Destek Cüneyt Düzgünel, Üst Düzey Kullanıcı Rikkat Hanlı ile birlikte inceler ve iş süreçlerini ne derece etkileyeceğini kestirmeye çalışır. Program Yöneticisi Yasemin Oygan'ın görüşüne başvurulur. Üçüncü adımda çözüm için ne tür seçenekler olabileceği tartışılır. Bu iş servisinin önemli bir ayağı olduğundan ertelenmesi, gecikmesi ya da iptali iş tarafına son derece olumsuz etki edeceği ortaya konur. Dördüncü aşamada iş tarafı faydasının yüksek yeni uygulamanın kısa bir sürede (2 adam/gün) gerçekleştirileceği öngörülerek ve verilen inisiyatif kullanılarak konfigürasyonun değiştirilerek SOA'yı uygulayan Yıldız Otel ekibi tarafından Güneş Otelin veritabanına bağlanmak suretiyle gerekli bilgileri edinen başka bir servis yazılmasına karar verirler. Bu aşamada projenin süresinin tolere edilebilir durumda olduğu için ek onay için proje kuruluna başvurulmamıştır. Son adımda gerekli kayıtlar alınır, planlar yeni duruma göre güncellenir ve uygulamasına geçilir. Bunun uygulama kısmı proje geliştirme adımlarının ikincisinde yer alacaktır.

6.5. Rezervasyon İş Sürecinin NetBeans Ortamında Oluşturulması

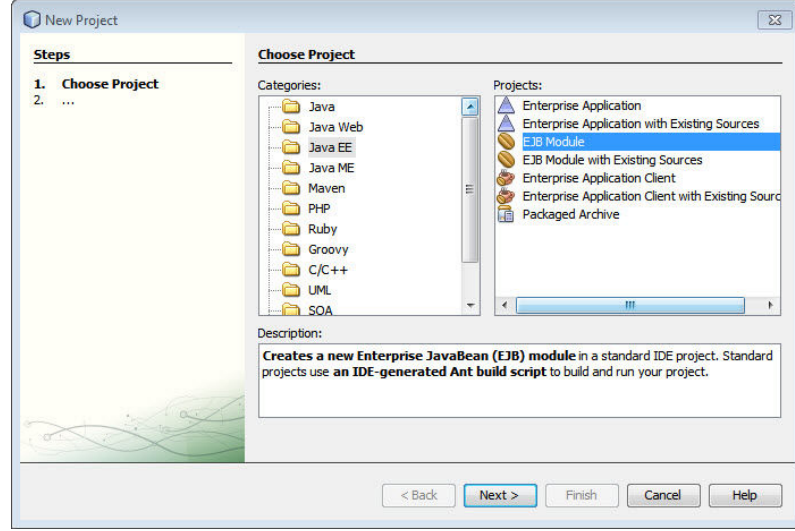
Daha önceki analiz ve modellemesi örneği yapılan SOA projesinde belirlenen rezervasyon servisi aslında birkaç web servisi ile beslenen ve mantıksal kısmının yine bir BPEL servisinin yerine getirdiği bir uygulamadır. Bu tarzda oluşturulan diğer iş süreçleri tüm bu işlemleri düzenleyen bir başka BPEL servisi aracılığıyla birleştirilip hizmete sunulabilecektir.

Sürecin alt adımları:

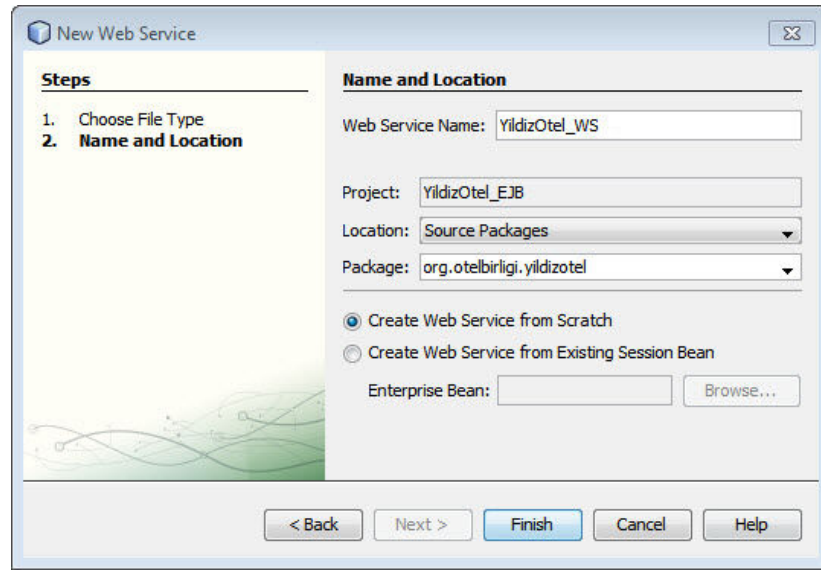
- Rezervasyon İş Sürecinin Oluşturulması ve Yıldız Otelin Web Servisi Aracılığıyla Entegrasyonu
- Güneş Otelin Veri Tabanı Üzerinden Entegrasyonu
- Ay Otelin FTP Aracılığıyla Entegrasyonu
- Venüs Otelin Web Servisi Aracılığıyla Entegrasyonu
- Şartlar ve Sıralamaların BPEL Üzerinde Uygulanması

Rezervasyon süreci ve buna Yıldız Otelin web servisinin eklenmesi için öncelikle YıldızOtelEJB oluşturulur. Menüden File, New Project sekmesi tıklandığında aşağıdaki ekrana ulaşılır. Şekil 6.5'deki gibi rezervasyon servisi için bir uygulama projesi oluşturulmuş olur.

SOA tabanlı uygulamanın en önemli bileşenlerinden biri olan web servisi NetBeans ortamında oldukça pratik bir şekilde oluşturulur. Bunun için örnek olarak Şekil 6.5'de TildizOtelWS adında bir web servisinin oluşturulması görülmektedir. EJB projesine tıklandığında görünen menüden yeni web servisi seçildiğinde aşağıdaki pencere açılır. Şekil 6.6 da görüldüğü gibi ikinci adımda bir web servisi ismi verilerek Finish tıklanır. Oluşan web servisi içine bir sınıf eklenir (Şekil 6.7). Bu sınıf rezervasyon sınıfıdır.

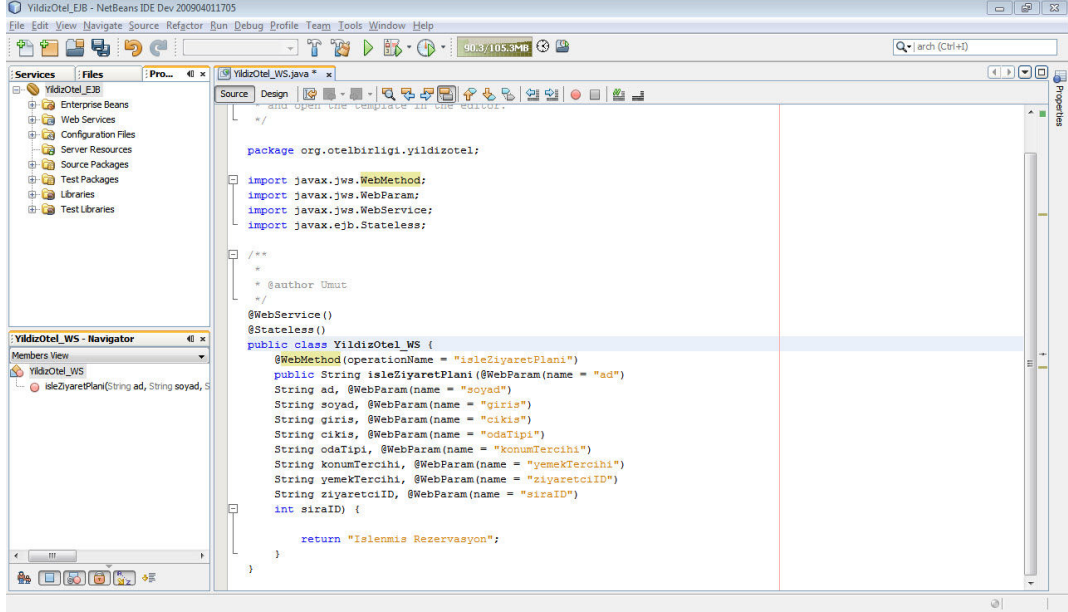


Şekil 6.5. Netbeans 6.x entegre geliştirme ortamında EJB Projesi oluşturmak.



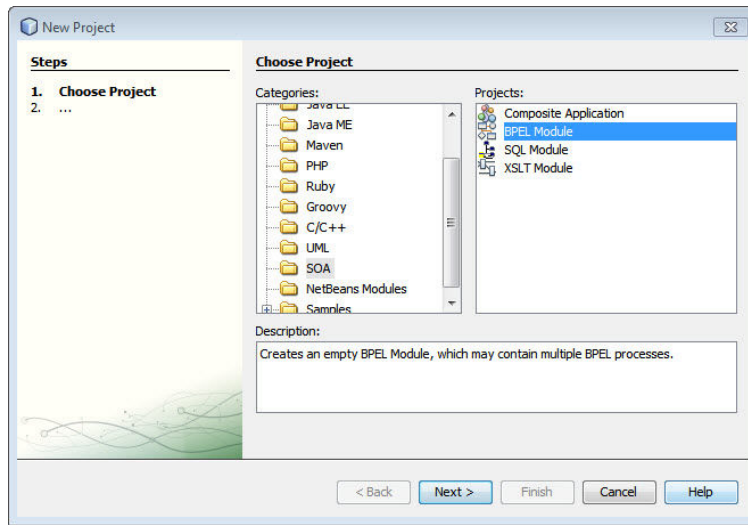
Şekil 6.6. Web servisi oluşturma.

Bu şekilde Yıldız Otelin bizim için sunacağı web servis kısmı hazır hale gelmiştir. Bundan sonraki aşamada bu web servisini ve diğer anlaşmalı otellerin bağlantısının ve yönetiminin sağlanacağı ve aynı zamanda süreci yönetecek olan iş süreci modülü oluşturulacaktır. Şekil 6.8’de görüleceği gibi yeni proje olarak SOA kategorisinde BPEL Modüle seçerek Next tuşuna tıklanır. Daha sonra BPEL modülü, Şekil 6.9 da görüldüğü gibi Rezervasyon_BPEL olarak isimlendirilir. İkinci aşamada oluşturmak istenen iş sürecine bir isim vererek işlem tamamlanır.



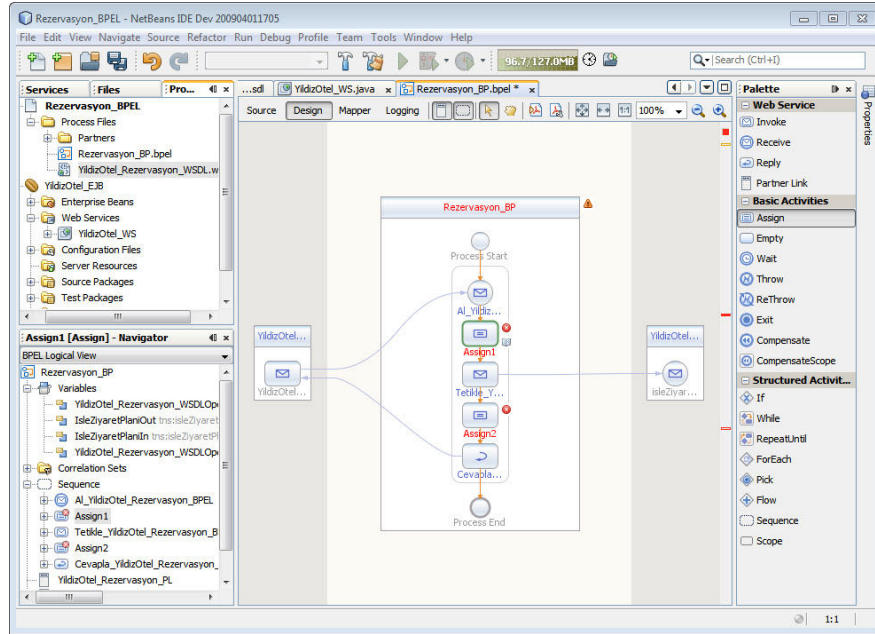
Şekil 6.7. Web servisi içinde bir sınıfın oluşturulması.

BPEL Modülü oluşturulması sırasında İş süreci ismi olarak Rezervasyon_BP adı verilir. İşlemden sonra oluşan bu iş süreci BPEL düzenleyici servis olarak çalışacaktır. Amacına uygun olarak farklı konum ve tipteki sistemlerle entegrasyon sağlayacak ve doğru iş sürecinin akışını da oluşturarak tam bir iş sürecinin sunulmasını sağlayacaktır. Bir BPEL modülünde oluşturulan iş süreci her ne kadar birçok servisi ve veri tabanı etkileşimini sağlıyorsa da kendisi de bir web servisi gibi çalışıyor olacak şekilde oluşturulmaktadır.

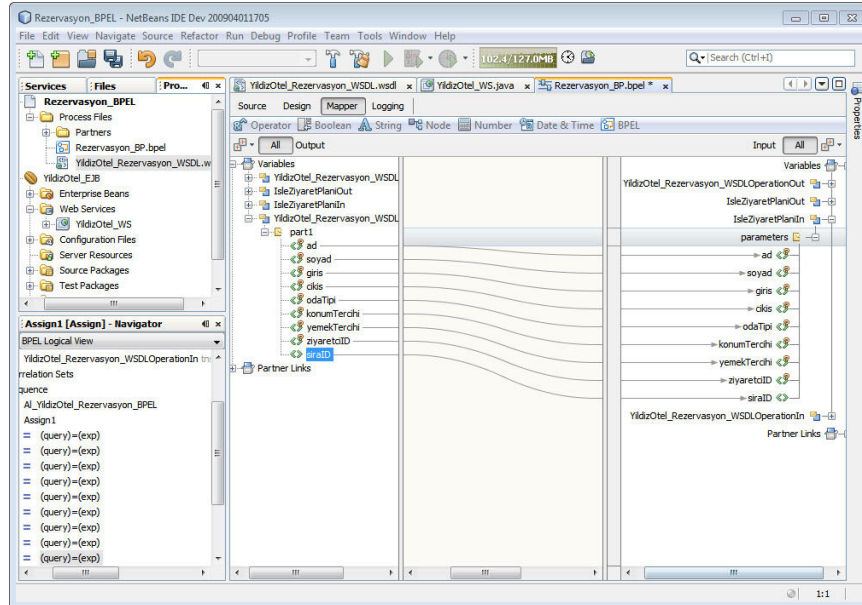


Şekil 6.8. Netbeans 6.x entegre geliştirme ortamında BPEL modülü oluşturma.

İş sürecinde ilk bağlantısını yapmak için bir Partner Link oluşturulması gerekmektedir. Rezervasyon_BP üzerine sağda Şekil 6.9’da görülen araç kutusundan sürükleyip bırak işlemiyle Reply, Assign, gibi adımlarla iş mantığı oluşturulur. Bu öğelerden biri olan Assign üzerine tıklanıp Mapper görünümüne geçilip tüm alanların eşleşmesi Şekil 6.10’daki gibi yapılabilir.

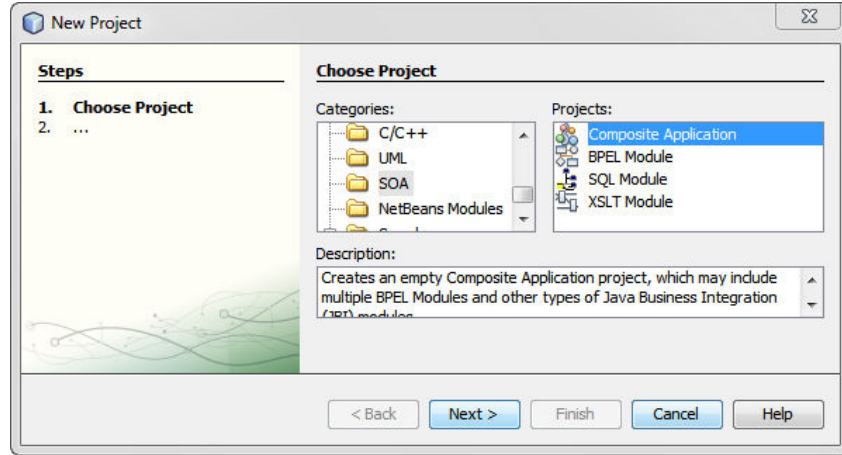


Şekil 6.9. Netbeans 6.x entegre geliştirme ortamını.



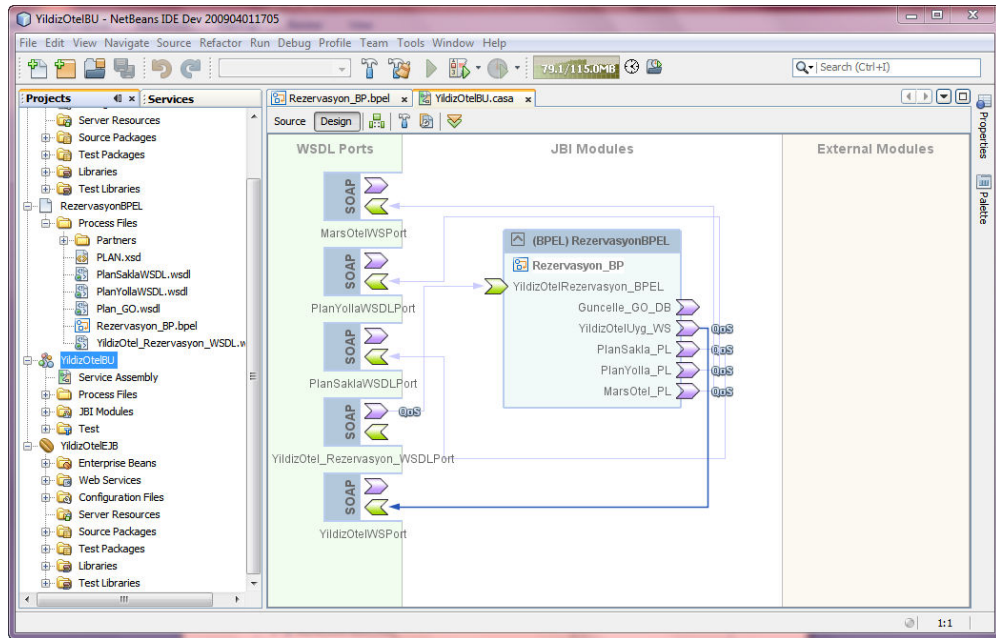
Şekil 6.10. Assign öğesinin içinde atamaların düzenlenmesi.

Rezervasyon iş sürecini yürüten bileşik uygulamanın oluşturulması için Şekil 6.11’de görüldüğü gibi yeni bir proje seçeneğinden SOA ve Composite Application tıklanarak yapılır.



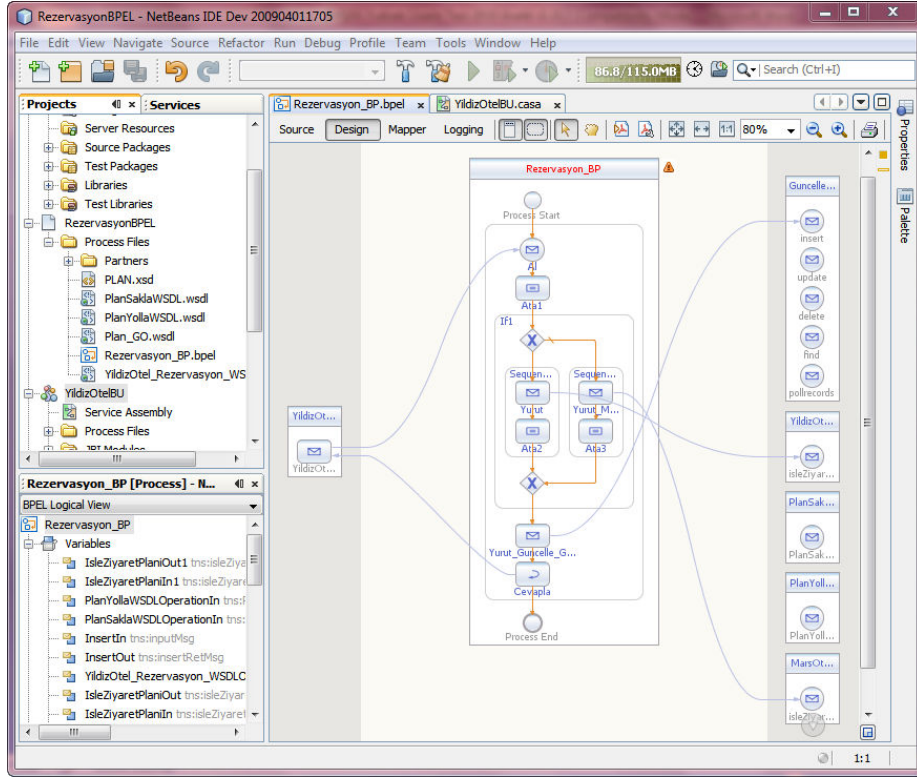
Şekil 6.11. Netbeans ortamında bileşik uygulamanın oluşturulması

Rezervasyon iş sürecinin diğer adımları da yukarıda anlatıldığı gibi tanımlanır ve birleştirilir. Bu sürecin çalışmasını sağlayan JBI modülünün son hali 6.12’de görüldüğü gibidir.



Şekil 6.12. Netbeans üzerinde JBI Modülü

Rezervasyon iş sürecini yürüten uygulama Şekil 6.13'de BPEL tasarımcısında görülen Rezervasyon_BP servisinin son halini yansıtmaktadır.



Şekil 6.13. Rezervasyon servisinin BPEL tasarımcısında görünümü.

7. SONUÇLAR

Bu tezde SOA tabanlı uygulamalarının en yüksek verimlilik ve fayda sağlayacak şekilde başarılması için uluslararası kabul görmüş ve standartlarla desteklenmiş farklı metodolojiler araştırılmış, bunlardan PRINCE2 ve CMMI'nin desteği ile birlikte göre SOA projeleri için uygun bir metodoloji olacağı sonucuna varılmıştır [21].

Tezin dördüncü ve beşinci bölümlerinde PRINCE2 ve CMMI metodolojilerinin SOA tabanlı proje ya da programların uygulanmasında bize sağlayacağı katkılar ele alınmıştır. Bu doğrultuda SOA uygulaması geliştirmede bir yol haritası oluşturulmuştur. Kavramsal anlamda SOA'nın desteklenmesi için SOA başlatma ve takip projeleri, teknolojik olarak SOA'nın desteklenmesi amacıyla SOA altyapı projesi oluşturulmuştur. Bu projelerin bağlı bulunduğu SOA uygulama programı altında iş tarafının gereksinimlerini çıktı olarak verecek ardışık, zaman aralıklı ve paralel SOA iş odaklı projeler yer almaktadır.

Tezin son bölümünde SOA uygulaması için tasarlanan yol haritasında yer alan anahtar projelerin bir örnek üzerinde hayata geçirilmesi yer almaktadır. Bu yol haritasına göre SOA Başlangıç projesinde SOA prensiplerinin ve kavramsal özelliklerinin belirlenmesi ve organizasyonda konumlandırılması amaç olarak belirlenmiştir. Teknolojik anlamda SOA kavramının işler hale getirilmesi için SOA altyapı projesi devreye alınmıştır. SOA uygulamasının SOA uyumluluğunun kontrol ve iyileştirilmesi için ilk aşamada bir SOA kontrol ve geliştirme projesi öngörülmektedir. Gerçek bir iş süreci olarak bir otel rezervasyonu sürecini gerçekleştiren SOA iş odaklı projesi ile çalışma sona ermektedir.

SOA yaklaşımına göre gerçekleştirilen uygulamaların başarısının artırılmasında PRINCE2 proje yönetimi ve CMMI süreç geliştirme metodolojilerinin çok önemli faydalar ve vizyonsal açılımlar getireceği sonucuna ulaşılmıştır.

8. KAYNAKLAR

1. E. Thomas, *Service-Oriented Architecture: Concepts, Technology and Design*, Prentice Hall, 2005.
2. N. M. Josuttis, *SOA in Practice*, O'Reilly Media, 2007.
3. The Office of Government Commerce (OGC), http://www.ogc.gov.uk/methods_prince_2.asp, 2010.
4. ILX Group, <http://www.prince2.com/what-is-prince2.asp>, 2010.
5. A. Murray, *Managing Successful Projects with PRINCE2*, OGC, Norwich, UK.
6. CMMI ana sayfası: <http://www.sei.cmu.edu/cmmi>, 2010.
7. D. M. Ahern, A. Clouse, R. Turner, *CMMI Distilled: A Practical Introduction to Integrated Process Improvement*, Third Edition, Addison Wesley Professional, 2008.
8. L. O'Brien, "A Framework for Scope, Cost and Effort Estimation for Service Oriented Architecture (SOA) Projects", *Australian Software Engineering Conference*, 2009.
9. JJ. Jeng, L. An, "System Dynamics Modeling for SOA Project Management", *IEEE International Conference on Service-Oriented Computing and Applications (SOCA'07)*, 2007.
10. A. Luqman, F. Hussain, Dr. S. Tauseef-ur-Rehman, "Mapping OGC PRINCE 2 to SEI CMMI 1.1", *IEEE Information and Communication Technologies Conference*, 2005.
11. S. Kumar, "The Enterprise SOA Implementation Lifecycle Explained", HCL Technologies Ltd. Noida, India, 2007.
12. S. R. M. Tayebi, S. S. Ostadzadeh, S. Mazaheri, "Employing MDA in PRINCE2 Framework", *Second International Conference on Future Information Technology and Management Engineering*, 2009.
13. P. Krogdahl, G. Luef, C. Steindl, "Service-Oriented Agility: An Initial Analysis for the Use of Agile Methods for SOA Development", *IEEE International Conference on Services Computing (SCC'05)*, 2005.
14. A. Arsanjani, "Service-Oriented Modeling and Architecture", IBM SOA/Web Services Center of Excellence, Software Group, Nov. 2004.
15. P. C. Brown, *Implementing SOA: Total Architecture in Practice*, Addison Wesley Professional, April 09, 2008.
16. B. A. Christudas, M. Barai, V. Caselli, *Service Oriented Architecture with Java*, Packt Publishing Ltd. Haziran 2008.
17. J. Davis, *Open Source SOA*, Manning Publications Co. 2009.

18. BEA white paper, “Domain Model for SOA: Realizing the Business Benefit of Service Oriented Architecture”, <http://eudownload.bea.com/uk/events/soa/soa.pdf>, 2005.
19. D. Krafzig, K. Banke, D. Slama, “Enterprise SOA: Service-Oriented Architecture Best Practices”, Prentice Hall PTR, Kasım 2004
20. CMMI Product Team, “CMMI for Development, Version 1.3”, Software Engineering Institute, November 2010.
21. U. Şimşek, H. Gümüşkaya, “Using PRINCE2 Project Management Methodology to Develop SOA Based Applications”, *2010 International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering*, Aralık 3 - 12, 2010.
22. M. Stevens, “The Benefits of a Service-Oriented Architecture”, www.developer.com, 2002.
23. Tutorials Point: <http://www.tutorialspoint.com/cmmi/cmmi-process-areas.htm>.
24. M. K. Kulpa, K. A. Johnson, *Interpreting the CMMI: A Process Improvement Approach*, Taylor & Francis Group, LLC, 2008.
25. D. Salter, F. Jennings, *Building SOA-Based Composite Applications Using NetBeans IDE 6*, Packt Publishing, 2008.