



**T.C.
DÜMLUPINAR ÜNİVERSİTESİ**

Bilgisayar Mühendisliği Bölümü

**HASTANE OTOMASYON
PROJESİ**

Ferat AKTAŞ

**Danışman
Yrd. Doç. Dr. Muammer Akçay**

**Haziran, 2013
Kütahya**

Bu çalışma/....../2013 tarihinde ařađıdaki jüri tarafından Bilgisayar Mühendisliđi Bölümünde Lisans Bitirme Projesi olarak kabul edilmiştir.

Bitirme Projesi Jürisi

Danışman Adı		
Üniversite		
Fakülte		

Jüri Adı		
Üniversite		
Fakülte		

Jüri Adı		
Üniversite		
Fakülte		

ÖNSÖZ

“Hastane Otomasyonu” konulu bu çalışma, Dumlupınar Üniversitesi, Bilgisayar Mühendisliği Bölümü’nde “Bitirme Tezi” olarak hazırlanmıştır.

Projenin Araştırma-Tasarım-Geliştirme aşamalarında karşılaştığım zorluklarda benden yardımlarını esirgemeyen arkadaşlarıma ve proje danışmanım Yrd. Doç. Dr. Muammer Akçay’a teşekkürlerimi bir borç bilirim.

Ayrıca eğitimim süresince bana her konuda tam destek veren aileme ve bana hayatlarıyla örnek olan tüm hocalarıma saygı ve sevgilerimi sunarım.

Haziran, 2013

Ferat AKTAŞ

İÇİNDEKİLER

ÖZET	- VI-
SUMMARY	-VII-
1.GİRİŞ	-07-
2.GENEL BİLGİLER	-08-
1. Yazılım	-08-
1. Bir Yazılımda Olması Gereken Temel Özellikler	-08-
2. Yazılım Çeşitleri	-09-
2. Programlama Araçları	-09-
1. Programlama Dili	-09-
2. Programlama Dilleri Çeşitleri	-10-
3. Programlama Dillerinin Tarihi Gelişimi	-10-
3. Orta Seviyeli Diller	-13-
1. C Programlama Dili	-13-
1. C Dilinin Erken Tarihi	-13-
2. C Dilinin Geçmişi	-13-
3. ANSI C ve ISO C Dilleri	-14-
4. C99	-15-
2. C++ Programlama Dili	-16-
3. C# Programlama Dili	-17-
1. Tasarım Hedefleri	-17-
2. Eleştiri	-18-
3. Performans	-18-
4. Platform	-18-
4. Microsoft Visual Studio	-19-
1. Mimarisi	-19-
2. Özellikleri	-20-
1. Kod Editörü	-20-
2. Hata Ayıklayıcı (Debugger)	-20-
3. Tasarımcı (Designer)	-20-
3. Desteklenenler	-22-
4. Versiyonlar	-23-
1. Versiyon Geçmişi	-25-
5. Veritabanı Sistemleri	-26-
1. Veri Nedir?	-26-
2. Veri Tabanı Nedir ?	-26-
3. Neden Veritabanı Kullanılır?	-27-
4. Veri Tabanı Yönetim Sistemi Nedir?	-27-
5. Veri Tabanı Yönetim Sisteminin Sağladığı Yararlar	-28-

6. Microsoft SQL Server	-29-
1. Tarihi	-29-
2. Versiyonlar	-31-
1. Temel (Mainstream) Versiyonlar	-31-
2. Özelleştirilmiş (Specialized) Versiyonlar	-32-
3. PROJE	-33-
1. Yazılım Geliştirme Sürecinde Geliştirme Aşaması	-33-
2. Yazılım Geliştirme Süreci	-34-
1. Birinci Aşama: Tanımlama/Planlama	-34-
2. İkinci Aşama: Geliştirme	-34-
3. Üçüncü Aşama: Bakım/Destek	-34-
3. Projemizin Geliştirme Aşaması	-35-
1. Birinci Aşama: Tanımlama/Planlama	-35-
2. İkinci Aşama: Geliştirme	-36-
4. Projede Kullanılacak Teknolojiler	-36-
1. Üçüncü Aşama: Bakım/Destek	-37-
6. Örnek Modül Görünüm ve Açıklamaları	-38-

ÖZET

Bu bitirme tezinde ilk olarak programlama dilleri hakkında bilgiler verilmiştir. Daha sonra nesneye yönelik programlama dillerinden olan C# dili üzerinde arařtırmalar yapılmıřtır. C# programlama dili derleyicisi olarak kullanılan Visual Studio 2012 programı incelenmiř ve Visual Studio 2012 ilk defa kullanacaklara yönelik programın arayüzüne ařına olabilmeleri için arayüz bileřenleri verilmiřtir. Bunlarla birlikte bitirme tezimizin konusunu oluřturan otomasyon sistemi için kullanılmak üzere veri tabanları incelenmiřtir. Projemizde kullanılan MS SQL veritabanı hakkında bilgiler iřlenmiřtir.

Tezin önemli bir kısmını her hangi bir hastane servisinin gereksinim duyacađı, hastanenin ihtiyaçlarına cevap verebilecek bir otomasyon sistem yazılımı içermektedir. Ancak böylesi bir SAP proje çalıřmalarında gereksinimler hastanelerden hastanelere farklılık göstereceđi için bu projede genel ortak gereksinimlere göre bir yazılım ürünü ortaya konulmuřtur.

Yazılım geliřtirme süreç ařamaları iřlenmiř ve proje görselleriyle birlikte tanıtılmıřtır.

Anahtar Kelimeler: C#, MS SQL, Visual Studio 2012, SQL Servers, Microsoft SQL Server 2012, Database Management Systems.

SUMMARY

This Project is a kind of SAP projects. Even though the SAP Projects have their own solution ways which is different of hospital to hospitals. This project aims to find a general solutions about Hospital Management Systems' general necessities.

1. GİRİŞ

Hastaneler birden çok fonksiyonun bir arada götürüldüğü kompleks işletmelerdir. Öncelikle hastane bir işletmedir, daha sonra tıbbi hizmet veren bir kuruluştur, ayrıca otelcilik hizmeti verirler. Binlerce kalemi bulan ilaç ve tıbbi malzemenin yönetilmesi bile başlı başına ayrı bir hizmet olarak alınabilir. Bu karmaşık yapı içerisinde bilgisayar kullanımı birçok sorunu çözebileceği gibi beraberinde birçok sorunu da gündeme getirebilir.

Hastanelerde kaynakların yerinde kullanılması ve israfın önlenmesi, oluşturulmaya çalışılan kalite ve standartların tesis edilmesi ve korunması, hastanelerde sunulan sağlık hizmetinin en iyi şekilde verilebilmesi, gelir ve giderlerin izlenmesi, gelir kaçaklarının önlenmesi, hastane yönetimine verilecek önemli kararlarda bilgi desteğinin sağlanması, sağlanan bilgi desteğiyle ileriye yönelik doğru hedef belirlenmesi ve doğru kaynak yönetimi, eksiksiz tıbbi kayıt, muhasebe, depo, ambar, demirbaş kayıtları tutabilmek amacıyla kullanılan bilgi sistemleridir.

2. GENEL BİLGİLER

2.1. Yazılım

Yazılım, elektronik aygıtların belirli bir işi yapmasını sağlayan programların tümüne verilen isimdir.

Bir başka deyişle var olan bir problemi çözmek amacıyla bilgisayar dili kullanılarak oluşturulmuş anlamlı anlatımlar bütünüdür.

Kelime işlemci programları, bilgisayarınız açıldığında CD sürücünüzü, sabit sürücüleri, RAM'i tanıyan BIOS; işletim sistemi, web tarayıcınız, virüslerin kendileri, antivirüs programları hep birer yazılımdır.

2.1.1. Bir Yazılımda Olması Gereken Temel Özellikler

Doğruluk: Yazılımın belirtilmiş ihtiyaçlarını karşılamaıdır.

Güvenilirlik: Gerekli işlevi ne hassaslıkla yerine getireceđi beklentisidir.

Verimlilik: İşlevin gerçekleştirilmesi için kullanılması gereken bilgisayar kaynakları ve kod miktarıdır.

Güvenlik(Bütünlük): Yazılım ve bilgilerine, istenmeyen insanlarca ulaşımın ne derece engellenebildiđidir.

Kullanılabilirlik: Programın öğrenilmesi, çalıştırılması, girdi hazırlama ve çıktı yorumlama işlemlerinin kolaylık derecesidir.

Hata bulma kolaylıđı: Hatanın yerini bulma ve düzeltme kolaylıđıdır.

Esneklik: Yazılımda deđişiklik yapma kolaylıđıdır.

Taşınabilirlik: Programın farklı donanımlarda ve yazılım sistemi ortamlarında kullanılmasıdır.

Tekrar kullanılabilirlik: Yazılım tamamının ya da bir bölümünün farklı bir uygulamada kullanılabilmesidir.

Birlikte çalışabilirlik: Bir yazılım sisteminin diđerleri ile bağlantı sağlamaı kolaylıđıdır.

2.1.2. Yazılım Çeşitleri

Bilgisayar yazılımları genel olarak 2 ana grupta incelenebilir.

□ Sistem yazılımları (system software) : Bilgisayarı yöneten, denetleyen, kontrol eden yazılımlardır.

Örneğin: Linux, Pardus, Windows vb.

□ Uygulama yazılımları (application software) : Belli bir alana ve uygulamaya ilişkin olarak kullanıcılar için geliştirilmiş yazılımlardır.

Örneğin: Kelime işlemci programları, web hazırlama araçları, programlama araçları vb.

Bütün sistem programları içinde en temel yazılım işletim sistemidir ki, bilgisayarın bütün donanım ve yazılım kaynaklarını kontrol ettiği gibi kullanıcılara ait uygulama yazılımlarının da çalıştırılmalarını ve denetlenmelerini sağlar.

2.2. Programlama Araçları

2.2.1. Programlama Dili

İster genel ister özel amaçlı olsun tüm uygulama ve sistem yazılımları programlama dilleriyle yazılır. Bir programlama dili, insanların bilgisayara çeşitli işlemler yaptırmasına imkân veren her türlü sembol, karakter ve kurallar grubudur. Programlama dilleri insanlarla bilgisayarlar arasında tercümanlık görevi yapar. Programlama dilleri, bilgisayara neyi, ne zaman, nasıl yapacağını belirten deyim ve komutlar içerir.

Bir programlama dili şunlardan oluşur.

- **Genel komutlar:** Programlama dilinin anlayacağı komutlardır.
- **Gelişmiş komutlar:** Genel komutları kullanarak oluşturulmuş komutlardır.
- **API komutları:** İşletim sisteminin sunduğu özellikleri kullanan komutlardır.
- **Derleyici komutları:** Komut içinde çalışmayıp derleme esnasında alınan bilgilere göre derleme yapılmasını sağlar.

□ **Aktif nesnelere:** ‘Buton, Menü, Gösterge çubuğu ve Tabpanel’ gibi bileşenlerin genel adıdır.

2.2.2. Programlama Dilleri Çeşitleri

Bir programlama dili ya insan ya da makine anlayışına yakındır. İnsan anlayışına daha yakın programlara dillerine yüksek seviyeli programlama dilleri, makineye yakın olanlara ise düşük seviyeli programlama dilleri denir.

□ Yüksek seviye programlama ile yazılan projelerin kaynak kodları kısa, derlenmiş hâlleri ise uzun olur. Çalışma hızları ise yavaştır.

□ Alçak seviye programlama ile yazılan projelerin kaynak kodları uzun, derlenmiş hâlleri ise kısadır olur. Çalışma hızları ise en yüksek seviyededir. Programlama dillerini seviyelerine göre 5 ana gruba ayırabiliriz:

□ Çok yüksek seviyeli diller ya da görsel diller
Access, Foxpro, Paradox, Xbase, Visual Basic, Oracle Forms

□ Yüksek seviyeli diller (Bunlara algoritmik diller de denir.)
Fortran, Pascal, Basic, Cobol

□ Orta seviyeli diller
C, C++(C Plus) , C#(C Sharp) Orta seviyeli diller daha az kayıpla makine diline çevrilebildiğinden daha hızlı çalışır.

□ Alçak seviyeli programlama dilleri
Sembolik makine dili (Assembler).

□ Makine dili
En aşağı seviyeli programlama dilidir (Saf makine dili tamamen 1 ve 0 lardan oluşuyor.) .

2.2.3. Programlama Dillerinin Tarihi Gelişimi

Düşük seviyeli diller : Bilgisayarların ilk dönemlerindeki programlama dilleri, kullanımı çok zor olan makine dili ve assembly dilleriydi. Makine dili, geliştirilen ilk programlama dilidir ve ilk kuşağı temsil eder. Makine dilinde yazılan tüm komutlar 0 ve 1 lerden oluşur. Bütün talimatlar en detaylı bir şekilde tanımlanır ve iki tabanlı sayı sistemi kullanılarak kodlama yapılır.

Assembly dilleri : İkinci kuşak dilerdir ve geliştirilmesine 1950’li yılların başlarında Grace Hopper tarafından öncülük edilmiştir. Bu dillerde makine dili talimatları daha kolay bir şekilde anlaşılabilir ve hatırlanabilir olan sembollerle ifade edilir. Düşük seviyeli diller grubundan sayılmasına rağmen, makine diline göre

bir adım daha ileride olan assembly dilleri günümüzde bile programcılar tarafından kullanılmaktadır. Çünkü bu dillerle yazılan programlar genellikle çok hızlı çalışır ve daha az depolama yüzeyi gerektirirler. Bununla birlikte bu dillerle programlama yapmak çok yorucu, sıkıcı ve zaman alıcı olduğundan cazibesini kaybetmektedir.

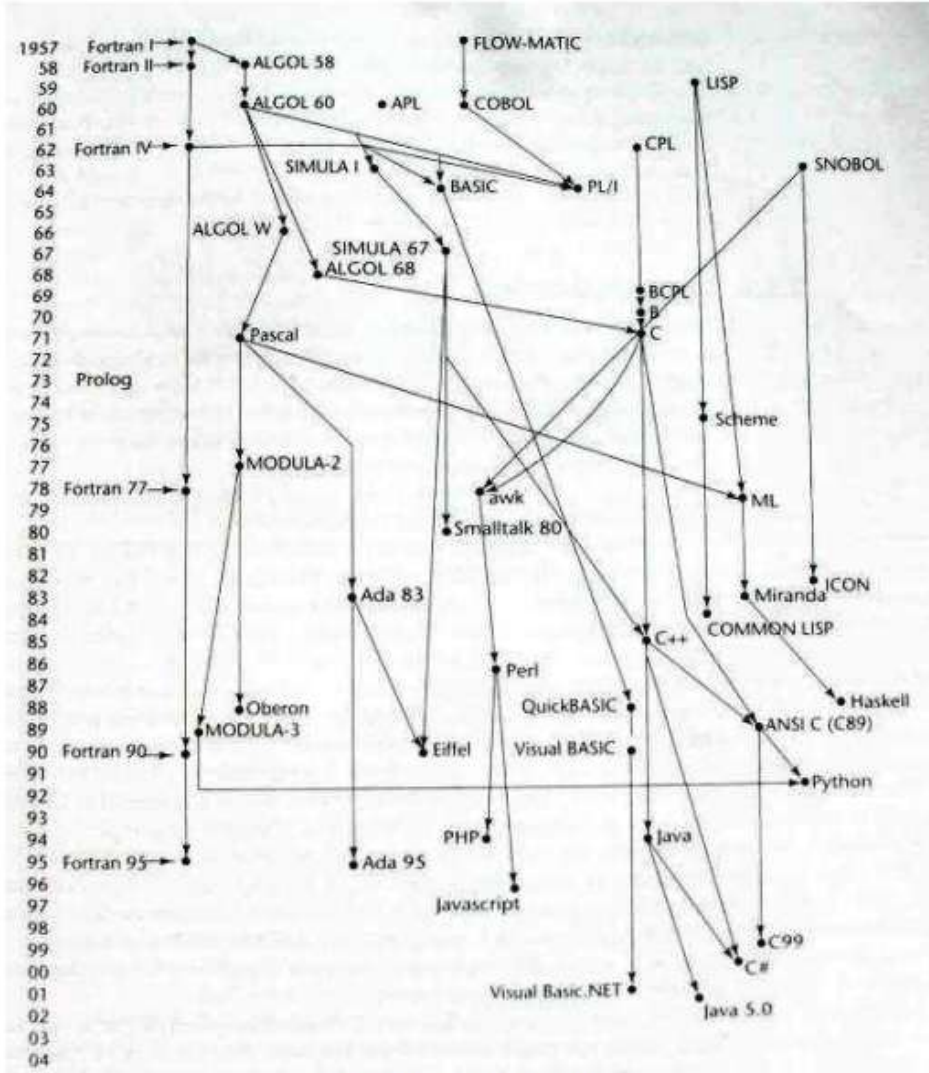
Yüksek seviyeli diller : Üçüncü kuşak da denilen yüksek seviyeli programlama dilleri öğrenilmesi daha kolay, program yazılması daha az zaman alan, daha iyi sonuçlar sağlayan programlama dilleridir. Yüksek seviyeli dillerden birinde yazılan bir kaynak program makine diline çevrilmek zorundadır. Dönüştürme işini gerçekleştiren programa derleyici (compiler) denir. Basic, Cobol, Fortran ve Pascal dilleri bu kategorideki programlama dillerine birkaç örnek oluşturur.

Çok yüksek seviyeli diller : Bu diller programlama işini çok daha kolaylaştırmıştır. Bu dillerdeki temel özellik kullanıcıların bilgisayara bir şeyin nasıl yapılacağını değil, ne yapılacağını ifade edebilmelerine imkan vermesidir. Örneğin; sayıların sıraya dizilmesi işlemini yapabilmek için bir yüksek seviyeli dil ile karmaşık bir mantık kullanarak 15-20 satırlık bir program yazmak gerekir. Bunun aksine bir çok yüksek seviyeli(dördüncü kuşak) dil ile bir kullanıcının yapacağı iş; sıraya dizilecek alanı, sıralama işleminin küçükten büyüğe mi yoksa büyükten küçüğe mi yapılacağını belirlemek ve ekrandaki bir ikona tıklayarak yada bir münüden “sırala” komutunu seçerek sıralama işlemini gerçekleştirmektir.

İlk geliştirilen dördüncü kuşak diller ile daha sonraları geliştirilen diller arasında bile takip edilen prosedürler ve kullanılan metotlar bakımından büyük farklılıklar vardır. Yeni geliştirilen diller çok daha karmaşık işleri daha kolay bir şekilde yapmaya imkan vermektedir.

Kuşak	Programlama Dili	Periyod
1	Makine dili	1940 – 1950 arası
2	Assembly dili	1950’li yıllardan itibaren
3	Yüksek seviyeli diller	1960’lı yıllardan itibaren
4	Çok yüksek seviyeli diller	1970’li yıllardan itibaren
5	Yapay zekaya yönelik diller	1980’li yıllardan itibaren

Tablo 1.1: Programlama dillerinin tarihi gelişimi



2.3. Orta Seviyeli Diller

2.3.1. C Programlama Dili

AT&T Bell laboratuvarlarında, Ken Thompson ve Dennis Ritchie tarafından UNIX İşletim Sistemi' ni geliştirebilmek amacıyla B dilinden türetilmiş yapısal bir programlama dilidir. Geliştirilme tarihi 1972 olmasına rağmen yayılıp yaygınlaşması Brian Kernighan ve Dennis M. Ritchie tarafından yayımlanan "C Programlama Dili" kitabından sonra hızlanmıştır. Günümüzde neredeyse tüm işletim sistemlerinin (Microsoft Windows, GNU/Linux, *BSD, Minix) yapımında %95' lere varan oranda kullanılmış, halen daha sistem, sürücü yazılımı, işletim sistemi modülleri ve hız gereken her yerde kullanılan oldukça yaygın ve sınırları belirsiz oldukça keskin bir dildir. Keskinliği, programcıya sonsuz özgürlüğün yanında çok büyük hatalar yapabilme olanağı sağlamasıdır.

Programlamanın gelişim süreciyle beraber programlamanın karmaşıklaşması, gereksinimlerin artması ile uygulama programlarında nesne yönelimliliğin ortaya çıkmasından sonra C programcıları büyük ölçüde nesne yönelimliliği destekleyen bir alt küme C++ diline geçmişlerdir.

2.3.1.1. C Dilinin Erken Tarihi

C'nin ilk gelişme safhaları 1969 ile 1974 arasında AT&T Bell Laboratuvarları'nda gerçekleşti. Ritchie'ye göre, en yaratıcı devre 1972 idi. Dilin pek çok özelliği "B" adlı bir dilden türediği için, yeni dile "C" adı verildi. "B" adının kökeni konusunda ise söylentiler değişik: Ken Thompson B'nin BCPL programlama dilinden türediğini söylemektedir, ancak Thompson eşi Bonnie'nin onuruna adını Bon koyduğu bir programlama dili de geliştirmiştir.

1973'e kadar C yeterince güçlü bir hale gelmiş ve ilk başta PDP-11/20 assembly dili ile yazılan UNIX'in çekirdeğinin büyük kısmı C ile yeniden yazılmıştı. Böylece UNIX, çekirdeği bir assembly dili ile yazılmayan ilk işletim sistemlerinden biri olmuştu.

2.3.1.2. C Dilinin Geçmişi

1978'de Ritchie ve Brian Kernighan The C Programming Language (C Programlama Dili) kitabının ilk baskısını yayımladılar.C programcıları tarafından "K&R" olarak bilinen bu kitap, yıllar boyunca C dilinin gayriresmi standardı olarak kullanıldı. C'nin bu sürümü bugün "K&R C" olarak adlandırılır. Bu kitabın ikinci baskısı ise aşağıda anlatılan ANSI C standardını içerir.

K&R dilde şu deęişiklikleri yaptı:

- struct veri tipleri eklendi
- long int veri tipi eklendi
- unsigned int veri tipi eklendi
- ==+ operatörü += olarak deęiştirildi

K&R C genellikle tüm C derleyicilerinin desteklemek zorunda olduęu dilin en temel kısmı olarak kabul edilir. Uzun yıllar boyunca, ANSI C'nin kabul edilifinden sonra bile, yüksek taşınabilirlik ((İngilizce) portability) istendiğinde, K&R C, C programcıları tarafından "ortak payda" olarak kabul edilmiştir çünkü bazı derleyiciler henüz ANSI C'yi desteklemek üzere güncellenmemişlerdi ve zaten iyi yazılmış bir K&R C programı aynı zamanda ANSI C'yi de destekliyordu.

K&R C'nin yayımlanmasını izleyen yıllar içine dile AT&T'nin derleyicilerinin ve bazı başka bilgisayar üreticileri tarafından desteklenen kimi "gayriresmi" özellikler eklendi. Bunların içinde aşağıdaki özellikler de vardı:

- void fonksiyonlar ve void * veri tipi
- struct ya da union veri tipi döndüren fonksiyonlar
- her bir struct'ın alan adları için ayrı bir ad alanı
- struct veri tipleri için atama
- bir nesneyi yazmaya karşı korumalı yapmak için const anahtar sözcüğü
- standart bir C kütüphanesi
- enumeration'lar
- single-precision float tipi

2.3.1.3. ANSI C ve ISO C Dilleri

1970'lerin sonunda C, en çok kullanılan mikrobilgisayar dili olarak BASIC'in önüne geçmeye başladı. 1980'lerde ise, IBM PC ile kullanılmak üzere benimsenmesiyle birlikte popülaritesi iyice artmaya başladı. Aynı zamanda, Bell Laboratuvarları'nda Bjarne Stroustrup ve iş arkadaşları C'ye nesneye yönelim eklemek üzere çalışmaya başlamışlardı. C bugün UNIX dünyasında en çok kullanılan dil olarak kalırken, Stroustrup'un geliştirip C++ adını verdiği dil Microsoft Windows işletim sisteminde en önemli dil oldu.

1983'te Amerikan Ulusal Standartlar Enstitüsü (ANSI) bir C standardı oluşturmak için bir kurul oluşturdu. Uzun ve yorucu bir çalışmadan sonra, bu kurul standardı 1989'da tamamladı ve standart ANSI X3.159-1989 "Programming Language C (C Programlama Dili)" olarak yayımlandı. Dilin bu versiyonu genellikle ANSI C olarak adlandırılır. 1990'da bu standart, küçük deęişikliklerle Uluslararası Standartlar Örgütü (ISO) tarafından da benimsenip ISO/IEC 9899:1990 olarak yayımlandı.

ANSI C'yi oluřturmanın amalarıdan biri K&R C'yi ieren ve dile sonradan katılan "gayiresmi" zellikleri de dile katan bir standart oluřturdu. Standart k fonksiyon prototiplerini ve daha yetenekli bir niřlemciyi de standarda ekledi.

Bugn artık ANSI C neredeyse tm derleyiciler tarafından desteklenmektedir. Gnmzde yazılmakta olan C programlarının ounluėu ANSI C standardına uygun olarak yazılmaktadır. Yalnızca standart C kullanılarak yazılmıř bir program, standarda uyumlu her derleyici ile doėru bir biimde derlenip alıřtırılabilir. Ancak, standart olmayan ktphaneler kullanılarak yazılmıř programlar belli bir platform ya da derleyici gerektirebilirler...

2.3.1.4. C99

ANSI standartlařtırma iřleminden sonra C dili uzun bir sre olduka sabit kaldı, ancak C++ geliřmeyi srdrd. Buna baėlı olarak, 1990'ların sonunda ISO standardı gncellendi ve 1999'da ISO 9899:1999 olarak yayımlandı. 2000 yılının Mart'ında ise, "C99" olarak bilinen bu standart ANSI tarafından da benimsendi.

C99'un yeni zellikleri řoye zetlenebilir:

- for ve bu gibi yineleme ifadelerinde parantez ii ilk deklarasyonlara izin verilmesi

rneėin; for(int i=0; i<10; i++) deyimi c99 da geerli iken c89 da geerli deėildir...

- inline fonksiyonlar
- C++'da olduėu gibi artık deėiřkenler programın herhangi bir yerinde tanımlanabilirler
- long long int, boolean, complex gibi yeni veri tipleri
- deėiřken uzunluėa sahip diziler
- C++'dan alınan, // ile bařlayan tek satırlık program ii aıklamalar
- snprintf() gibi yeni ktphane fonksiyonları
- stdint.h gibi yeni bařlık dosyaları

C99'u bugn GCC ve bazı bařka derleyiciler desteklemektedirken, Microsoft ve Borland derleyicilerine C99 desteėi eklemekte isteksiz davranmaktadırlar.

2.3.2. C++ Programlama Dili

C++, Bell Laboratuvarlarından Bjarne Stroustrup tarafından 1979 yılından itibaren geliştirilmeye başlanmış, C'yi kapsayan ve çok paradigmalı, yaygın olarak kullanılan, genel amaçlı bir programlama dilidir. İlk olarak C With Classes (Sınıflarla C) olarak adlandırılmış, 1983 yılında ismi C++ olarak değiştirilmiştir.

Genel olarak her C programı aynı zamanda bir C++ programıdır, ancak her C++ programı bir C programı değildir. Bu durumun bazı istisnaları mevcuttur. C++'ı C'den ayıran özellikler C++'ın nesne paradigması kullanılarak programlamaya olanak tanıyan özelliklerdir. Sınıflar sayesinde yeni veri türleri yaratılabilir veya varolan türlerden yenileri türetilir. Ayrıca çokbüçümlilik sayesinde bir sınıf tanımıyla yazılmış kod, o sınıf türünden türetilmiş yeni sınıflarla da çalışabilir.

İşleçler: C++'ın C'den devraldığı ve onu geliştirdiği yapılardan biridir. Kısaca, özel olarak anlamlandırılmış fonksiyonlar kümesidir. Ancak bu fonksiyonlar, alışlageldik şekilde çağırılmaz, dilin semantik yapısı içerisinde özel olarak değerlendirilir. Operatörler çoğunlukla iki nesnenin arasına özel bir işaret konulmasıyla çağırılır. Örneğin, iki nesnenin toplanması için iki nesnenin arasına '+' işareti konulması yeterlidir. Bu işaret, derleyiciye '+' operatör fonksiyonunun çağırılacağını ifade eder. C'den farklı olarak, C++'ta operatörlerin çoğunu aşırı yükleme yoluyla özelleştirebilirsiniz.

Önişlemci : C++ Program kaynak kodlarının derlenmesi üç aşamadan oluşur: Önişlemci, kodun dönüştürülmesi ve bağlama. İlk aşama olan önişlemciyi basit bir yorumlayıcı olarak görülebilir. Bu aşamada, bir takım direktifler yardımıyla derlenecek kodlarda basit sözcüksel değişiklikler yapılabilir.

Önişlemci direktifleri # karakteriyle başlar, bu karakterden sonrası önişlemci tarafından yorumlanır ve bu özel karakterden önce beyaz boşluk(tab ve boşluk) dışında bir şey olmaması gerekir. Bu direktifler yardımıyla derlenecek kodlar üzerinde bir takım kurallar belirlenebilir. Ayrıca dosyaya bir başka dosyayı dahil etmek veya işlemciye özel direktifler vermek gibi işlemler içinde kullanılabilir.

Kalıtılma : Kalıtılma diğer veri türlerinin özelliklerinden bir veri türü elde etmeyi sağlar. Temel sınıftan kalıtım, "public", "protected" ve "private" olarak ilan edilebilir. Bu erişim belirteci ilgili olmayan ve türetilmiş sınıfların kalıtımını "public" ve "protected" üyelerine aktarabilir. Kalıtım ifadesi tek başına kullanılırsa "public" kalıtımını ifade eder. Diğer iki kalıtım türü "public" kalıtımdan daha az kullanılır. Geçiş belirteci ihmal edilirse "class" "private" olarak, "struct" "public" olarak kalıtılır. Temel sınıflar sanal olarak ilan edilebilir, bu sanal kalıtım olarak adlandırılır. Sanal kalıtım çoklu kalıtımın problemlerinden sıyrılarak temel yalnızca bir örneğinin kalıtım grafiğinde olmasını sağlar.

2.3.3. C# Programlama Dili

C# Programlama Dili (si şarp şeklinde telaffuz edilir), Microsoft'un geliştirmiş olduğu yeni nesil dildir. Yine Microsoft tarafından geliştirilmiş .NET Teknolojisi için geliştirilmiş dillerden biridir.

Microsoft tarafından geliştirilmiş olsa da ECMA ve ISO standartları altına alınmıştır.

C programlama dilinde bir tamsayı değişkeni 1 attırmak için ++ soneki kullanılır. C++ dili adını, C diliyle Nesneye Yönelimli Programlama yapabilmek için eklentiler (C With Classes) almıştır. Benzer şekilde C++ diline yeni eklentiler yapılarak ((C++)++) bir adım daha ileriye götürülmüş ve tamamen nesneye yönelik tasarlanmış C# dilinin isimlendirilmesinde, + karakterlerinin birbirlerine yakınlaşmış hali ve bir melodi anahtarı olan C# Major kullanılmıştır.

Bu dilin tasarlanmasına Pascal, Delphi derleyicileri ve J++ programlama dilinin tasarımlarıyla bilinen Anders Hejlsberg liderlik etmiştir.

Birçok alanda Java'yı kendisine örnek alır ve C# da java gibi C ve C++ kod sözdizimine benzer bir kod yapısındadır. .NET kütüphanelerini kullanmak amacıyla yazılan programların çalıştığı bilgisayarlarda uyumlu bir kütüphanenin ve yorumlayıcının bulunması gereklidir. Bu, Microsoft'un .Net Framewok'u olabileceği gibi ECMA standartlarına uygun herhangi bir kütüphane ve yorumlayıcı da olabilir. Yaygın diğer kütüphanelere örnek olarak Portable.Net ve Mono verilebilir.

Özellikle nesne yönelimli programlama kavramının gelişmesine katkıda bulunan en aktif programlama dillerinden biridir .NET platformunun anadili olduğu bazı kesimler tarafından kabul görse de bazıları bunun doğru olmadığını savunur.

C#, .NET orta seviyeli programlama dillerindedir. Yani hem makine diline hem de insan algısına eşit seviyededir. Buradaki orta ifadesi dilin gücünü değil makine dili ile günlük konuşma diline olan mesafesini göstermektedir. Örneğin; Visual Basic .NET (VB.NET) yüksek seviyeli bir dildir. Dersek bu dilin insanların günlük yaşantılarında konuşma biçimine yakın şekilde yazıldığını ifade etmektedir. Dolayısı ile buradan yola çıkarak VB.NET, C#.NET'ten daha güçlü bir dildir diyemeyiz. Programın çalışması gereken bilgisayarlarda framework kurulu olması gerekmektedir(Win7 ve Vista'da framework kuruludur)

2.3.3.1. Tasarım Hedefleri

ECMA tarafından C# dilinin tasarım hedefleri şöyle sıralanır:

- C# basit, modern, genel-amaçlı, nesneye yönelik programlama dili olarak tasarlanmıştır.

- Çünkü yazılımın sağlamlılığı, güvenilirliği ve programcılarının üretkenliği önemlidir. C# yazılım dili, güçlü tiplendirme kontrolü (strong type checking), dizin sınırlar kontrolü (array bounds checking), tanımlanmamış değişkenlerin kullanım tespiti, (source code portability), ve otomatik olarak veri toplama gibi özelliklerine sahiptir.

- Programcı portatifiği özellikle C ve C++ dilleri ile tecrübesi olanlar için çok önemlidir.

- Enternasyonal hale koymak için verilen destek çok önemlidir.

- C# programlama dili sunucu ve gömülü sistemler için tasarlanmıştır. Bununla birlikte C# programlama dili en basit işlevselli fonksiyondan işletim sistemini kullanan en teferruatlısına kadar kapsamaktadır.

- C# uygulamaları hafıza ve işlemci gereksinimleri ile tutumlu olmak üzere tasarlanmıştır. Buna rağmen C# programlama dili performans açısından C veya assembly dili ile rekabet etmek için tasarlanmamıştır.

2.3.3.2. Eleştiri

C# konusunda eleştiriler tabii ki var. En önemlisi using ile hangi referansları kullanacağımızı çok iyi bilmeliyiz. Ayrıca süslü parantez "{}" ve noktalı virgül ";" karakterlerini çok sevmemiz ve asla unutmamamız gerekmektedir.

2.3.3.3. Performans

Diğerleri gibi Sanal Makine'ye dayalı dillerden biridir, C# programlama dili direkt yerleşik kod'a derleyen dillerden daha yavaştır.

2.3.3.4. Platform

.NET Microsoft uygulama servisi Windows üzerinde geçerlidir. Fakat C# programlarını Windows, Linux veya MacOS üzerinde yürüten başka uygulamalar da yer almaktadır. Mono (software) ve DotGnu gibi.

2.4. Microsoft Visual Studio

Microsoft Visual Studio, Microsoft tarafından geliştirilen bir tümleşik geliştirme ortamıdır (IDE). Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework ve Microsoft Silverlight tarafından desteklenen tüm platformlar için yönetilen kod ile birlikte yerel kod ve Windows Forms uygulamaları, web siteleri, web uygulamaları ve web servisleri ile birlikte konsol ve grafiksel kullanıcı arayüzü uygulamaları geliştirmek için kullanılır.

Visual Studio IntelliSense'in yanı sıra "code refactoring" destekleyen bir kod editörü içerir. Entegre hata ayıklayıcı, hem kaynak-seviyesinde hem de makine-seviyesinde çalışır. Diğer yerleşik araçlar, GUI uygulamaları, web tasarımcısı, sınıf tasarımcısı ve veritabanı şema tasarımcısı yaratabilmek için bir form tasarımcısı içerir. Hemen hemen her işlevsellik düzeyinde dahil olmak üzere, kaynak kontrol sistemleri için destek (Subversion ve Visual SourceSafe gibi) sunan eklentileri kabul eder.

Visual Studio, değişik programlama dillerini destekler, bu da kod editörü ve hata ayıklayıcısının neredeyse tüm programlama dillerini desteklemesini sağlamaktadır. Dahili diller C/C++ (Görsel yoluyla C++), VB.NET (Visual Basic .NET üzerinden), C# (Visual C# ile), ve F# (Visual Studio 2010 itibarıyla) içermektedir.

2.4.1. Mimarisi

Visual Studio özünde herhangi bir programlama dili, çözüm veya aracı desteklemeyerek, bunun yerine, bir VSPackage olarak kodlanmış işlevsellik sağlar. Yüklendiğinde, işlevsellik, Hizmet olarak kullanılabilir. IDE üç hizmet vermektedir; SVsSolution, projeler ve çözümleri numaralandırma yeteneği sağlar; SVsUIShell, pencereleme ve UI işlevselliği (sekmeler, araç çubukları ve araç pencereleri dahil) sağlar; ve SVsShell, VSPackages kaydı ile ilgilenir. Dil Hizmetleri adı verilen belirli bir VSPackage kullanarak programlama dilleri için destek eklenmiştir. Bir dil hizmeti, çeşitli arayüzleri tanımlar, bu şekilde VSPackage uygulamaları, çeşitli fonksiyonlara destek eklenebilir. Bu şekilde eklenebilir işlevler, sözdizimi renklendirme, deyim tamamlama, araç eşleşen, parametre bilgileri, üye listeleri ve arka plan derleme hata işaretleri olarak bulunmaktadır.

Visual Studio, yerleşik herhangi bir kaynak denetimi desteği içermez ama IDE ile entegre etmek için kaynak kontrol sistemlerine iki alternatif yol tanımlar. Kaynak Kontrolü VSPackage kendi özelleştirilmiş kullanıcı arabirimini sağlayabilir. Buna karşılık, standart bir Visual Studio kullanıcı arayüzü ile MSSCCI (Microsoft Source Code Control Interface) kullanarak bir kaynak denetimi eklentisi, çeşitli kaynak denetimi işlevselliğini uygulamak için kullanılan işlevler kümesi sağlar.

2.4.2. Özellikleri

2.4.2.1. Kod Editörü

Visual Studio, herhangi bir diğer IDE gibi, sözdizimi vurgulama ve kod tamamlama destekleyen sadece değişkenler, fonksiyonlar ve yöntemler değil, aynı zamanda döngüler ve sorgular gibi dil yapıları için IntelliSense kullanan bir kod editörü içerir. IntelliSense, web site ve uygulamaları geliştirirken dahil edilen diller ile beraber XML, CSS ve JavaScript desteklemektedir. Otomatik tamamlama önerileri, modsuz bir liste kutusu içerisinde, kod editörü üzerinde açılır.

Visual Studio kod editörü aynı zamanda hızlı navigasyon için kod ayarı yer imleri destekler. Diğer seyir yardımcıları, çöken kod blokları ve artımlı aramaya ek olarak normal metin arama ve regex arama içerir. Kod editörü aynı zamanda bir çoklu öge panosu ve bir görev listesi içerir. Kod editörü ayrıca, tekrarlanan kod için kaydedilen ve üzerinde çalışılan proje içerisine yerleştirilebilen, kod parçacıklarını destekler.

Visual Studio arka plan derleme özelliği de içerir (artımlı derleme olarak da bilinir). Kod yazılırken, sözdizimi ve derleme hataları hakkında geri bildirim sağlayabilmek için (bu hataların altı kırmızı ile işaretlenir) Visual Studio arkaplanda derleme yapar.

2.4.2.2. Hata Ayıklayıcı (Debugger)

Visual Studio hem kaynak-seviyesi hem de makine-seviyesinde çalışan bir hata ayıklayıcı içerir. Bu, yönetilen kodun yanı sıra yerel kod olarak da çalışır ve Visual Studio tarafından desteklenen herhangi bir dilde yazılmış hata ayıklayıcı uygulamaları için de kullanılabilir. Buna ek olarak, çalışan süreçlere bağlanabilir ve bu süreçleri gözlemleyerek hata ayıklama yapabilir. Çalışan süreç için kaynak kodu erişilebilir ise, bu kodu çalışır olarak görüntüler. Kaynak kodu erişilebilir değilse, "sökme" gösterilir. Hata ayıklayıcı ayrıca, bellek dökümleri oluşturarak, daha sonra işlem yapmak için ayırabilir.

Hata ayıklayıcı, kırılma noktası ayarına izin verir (yürütme, belli bir pozisyonda geçici olarak durdurulabilir) ve izler (yürütme ilerledikçe, değişkenlerin değerlerini takip eder).

2.4.2.3. Tasarımcı (Designer)

Visual Studio uygulamaları geliştirme yardımcı olmak için bir dizi görsel tasarımcı içerir. Bu araçlar şunlardır:

Windows Form Tasarımcısı (Windows Form Designer)

Windows Forms tasarımcısı Windows Forms kullanarak GUI uygulamaları oluşturmak için kullanılır. Plan, diğer konteynerler içerisinde kontrollerin barındırılarak ya da formun yan tarafında kilitlenerek kontrol edilebilir. Veri görüntüleyen kontroller (metin kutusu, liste kutusu, ızgara görünümü, vb gibi), veri tabanları veya sorgular gibi veri kaynaklarına bağlı olabilir.

WPF Tasarımcısı (WPF Designer)

WPF Tasarımcısı, kod adı Cider, Visual Studio 2008 ile tanıtıldı. Windows Forms tasarımcısı gibi sürükle ve bırak metaforunu destekler. Windows Presentation Foundation'ı hedefleyen yazar kullanıcı arayüzleri için kullanılır. Veri bağlama ve otomatik düzen yönetimi de dahil olmak üzere tüm WPF işlevlerini destekler.

Web Tasarımcısı/Geliştirme (Web Designer/Development)

Visual Studio ayrıca, bir web-site editörü içerir, bu şekilde kişilerin sürükle ve bırak parçacığıyla web sitesi yazmasına olanak tanır. ASP.NET uygulamaları için kullanılır, ayrıca HTML, CSS ve JavaScript destekler. ASP.NET kodu ile bağlantı için bir "kod-arkası" modeli kullanır.

Sınıf Tasarımcısı (Class Designer)

Sınıf tasarımcısı, sınıfların, UML modelleme kullanılarak, yazılması ve düzenlenmesine olanak sağlar. C# ve VB.NET kodlarının ana hatlarını, sınıflar ve metodları için yaratabilir.

Veri Tasarımcısı (Data Designer)

Veri tasarımcısı yazdığınız tabloları grafiksel olarak, birincil ve yabancı anahtarlar ve kısıtlamalar dahil olmak üzere, veritabanı şemaları düzenlemek için kullanılabilir.

Eşleştirme Tasarımcısı (Mapping Designer)

Visual Studio 2008'den itibaren, eşleştirme tasarımcısı "LINQ to SQL" tarafından, verileri içeren veritabanı şemaları ve sınıflar arasında eşleştirmeyi tasarlamak için kullanılır.

Geniřletilebilirlik

Visual Studio, geliřtiricilere, programın kabiliyetlerini geniřletmek için Visual Studio uzantıları yazmalarına olanak tanır. Bu uzantılar, Visual Studio'ya "takılır" ("plug into") ve onun işlevselliğini geniřletmeyi hedefler. Uzantıları makrolar, eklentiler ve paketler řeklinde görebiliriz. Makrolar, geliřtiricilerin kayıt, yeniden oynatma ve dağıtma için programatik olarak kaydetmesi için tekrarlanabilir görev ve işlemleri temsil eder. Ancak, makrolar, yeni komutları uygulamak veya araç pencereleri oluşturmak için uygun deęildir. Eklentiler, Visual Studio nesne modeline erişim sağlar ve IDE araçları ile etkileşim için kullanılır. Paketler, Visual Studio SDK kullanılarak oluşturulur ve geniřletilebilirlięi en üst düzeye taşır. Tasarımcılar ve dięer araçları oluşturmanın yanı sıra dięer programlama dillerini entegre edebilir.

Visual Studio 2008, IDE'nin özelleřtirilmiř versiyonunun geliřtirilmesine olanak sağlayan Visual Studio Shell'i tanıtmıřtır. Visual Studio Shell, herhangi bir IDE'de gerekli işlevsellięi sağlamak için bir dizi VSPackages tanımlar. Bunun üzerinde, dięer paketler, yüklemeyi özelleřtirmek için eklenebilir.

2.4.3. Desteklenenler

Microsoft Visual C++

Microsoft Visual C++, Microsoft'un C ve C++ derleyicisi, dil hizmetleri ve Visual Studio IDE ile entegrasyon için özel araçlar içeren uygulamasıdır. C modu veya C++ modunda derlenebilir. C, kütüphane MS-özel eklemeleri ile ve C99 spec parçaları ile birlikte ISO C standardını takip eder. C++, ANSİ C++ spec ve birkaç C++0x özellięi ile birlikte çalışır. Aynı zamanda, C++/CLI spec'i, yönetilen kod ve karışık-mod kodu (yerli ve yönetilen kodun bir karışımı) yazmak için destekler.

Microsoft Visual C#

Microsoft Visual C#, .NET Framework ile birlikte Visual Studio IDE'nin C# projelerine destek vermesini sağlayan dil servislerini hedef alan, Microsoft'un bir C# dil uygulamasıdır. Dil servislerinin, Visual Studio'nun bir parçası olmasıyla birlikte, derleyici, .NET Framework'un ayrı bir parçası olarakta bulunmaktadır.

Microsoft Visual Basic

Microsoft Visual Basic, Microsoft'un VB.NET dili, birleşmiş araçlar ve dil servisleri uygulamasıdır. Visual Studio .NET (2002) ile tanıtıldı. Microsoft, Visual Basic'i Hızlı Uygulama Geliřtirme (Rapid Application Development) için konumlandırdı. Visual Basic, hem konsol uygulamalarında hem de GUI uygulamalarının yazımında kullanılabilir.

Microsoft Visual Web Geliştirici (Microsoft Visual Web Developer)

Microsoft Visual Web Developer, ASP.NET kullanarak web siteleri, web uygulamaları ve web servisleri oluşturmak için kullanılır. C# ya da VB.NET dilleri kullanılabilir. Visual Web Geliştirici kullanılarak grafiksel olarak web sayfa düzeni oluşturulabilir.

Team Foundation Server

Sadece Visual Studio Team System ile birlikte gelen Team Foundation Server, işbirlikçi uygulama geliştirme için planlanmış olup, sunucu tarafında arka uç kaynak kontrolü, veri toplama, raporlama ve proje izleme işlevselliği sağlamak gibi işlemler için tasarlanmıştır.

2.4.4. Versiyonlar

Visual Studio Express

Visual Studio Express Versiyonu, hafif, ücretsiz, bireysel IDE setlerinden oluşan, "per-platform" ve "per-language" bazında Visual Studio IDE'nin hafifletilmiş versiyonudur. Bireysel Visual Studio Shell App'ids üzerine desteklenen platformlar için geliştirme araçları (web, Windows, telefon) veya desteklenen geliştirme dillerini (VB, C#) yükler.

- Visual Basic Express
- Visual C++ Express
- Visual C# Express
- Visual Web Developer Express
- Express for Windows Phone

Visual Studio LightSwitch

Microsoft Visual Studio LightSwitch, mevcut .NET teknolojileri ve Microsoft platformları üzerinde inşa edilmiş "line-of-business" uygulamaları oluşturmak için özel olarak hazırlanmış bir IDE'dir. Üretilen uygulamalar mimari olarak 3 katmanlıdır: kullanıcı arayüzü Microsoft Silverlight üzerinde çalışmaktadır; mantık ve veri-erişimi katmanı, ASP.NET üzerinde çalışan, WCF RIA Servisleri ve Birim Çerçevesi (Entity Framework) üzerine kurulmuştur; ve ana veri deposu, Microsoft SQL Server Express, Microsoft SQL Server ve Microsoft SQL Azure'yi desteklemektedir.

Visual Studio Professional

Visual Studio Professional Edition, tüm desteklenen geliştirme dilleri için bir IDE sağlar. MSDN desteği, MSDN Essentials veya lisansa bağlı olarak tam MSDN kütüphanesi olarak kullanılabilir. XML ve XSLT düzenlemesi destekler ve sadece ClickOnce ve MSI kullanan kurulum paketleri oluşturabilir. Server Explorer ve Microsoft SQL Server ile entegrasyon gibi araçlar da içerir.

Visual Studio Premium

Visual Studio Premium Edition, Visual Studio Professional araçlarını içerir ve kod metrikleri, profil, statik kod analizi ve veritabanı birim testi gibi ek işlevler ekler.

Office için Visual Studio Araçları (Visual Studio Tools for Office)

Office için Visual Studio Araçları, Microsoft Office paketi için geliştirme araçları içeren, bir SDK ve eklentidir. Daha önce (Visual Studio .NET 2003 ve Visual Studio 2005 için) sadece Visual C# ve Visual Basic gibi dillerde desteklenen veya Team Suite'e dahil edilmiş ayrı bir SKU'ydu. Visual Studio 2008 ile, artık ayrı bir SKU ancak Profesyonel ve daha yüksek sürümleri ile birlikte gelmektedir.

Visual Studio Ultimate

Visual Studio Ultimate, Visual Studio Premium tarafından sağlanan özelliklere ek olarak, yazılım ve veritabanı geliştirme, işbirliği, ölçümler, mimari, test ve raporlama araçları kümesi sağlar.

Visual Studio Takım Sistemi (Visual Studio Team System)

Visual Studio 2010'dan önce, Visual Studio Team System için dört "rol-spesifik" sürümler;

- Takım Tarayıcısı (Team Explorer) -Temel TFS istemcisi-
- Mimari Versiyon (Architecture Edition)
- Veritabanı Versiyonu (Database Edition)
- Geliştirme Versiyonu (Development Edition)
- Test Versiyonu (Test Edition)

Test Profesyonel (Test Professional)

Visual Studio Test Professional sürümü Visual Studio 2010 ile tanıtıldı. Odağı, özel test rolü (dedicated tester role) ve test ortam yönetimi desteği üzerine yoğunlaştırmıştır.

Sürüm özellik tablosu																
Ürün	Uzantılar	Korulum Projeleri	MSDN Entegrasyonu	Hata Ayıklama	Profilleme	Statik Analiz	IntelliTrace	Birim Testi	Kod Kapsamı	Kodlu UI Testi	Test Eki Analizi	Test Yükleme	Lab Yönetimi	Mimari ve Modelleme	Mobil Geliştirme	
Ekspres	Hayır	Limitli	Essentials	Evet	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır	Sadece Windows Phone 7	
Professional	Evet	Evet	Essentials ya da Full	Evet	Hayır	Hayır	Hayır	Evet	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır	Sadece Windows Phone 7	
Premium	Evet	Evet	Full	Evet	Evet	Evet	Hayır	Evet	Evet	Evet	Evet	Hayır	Hayır	Sat-olunur	Sadece Windows Phone 7	
Ultimate	Evet	Evet	Full	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Sadece Windows Phone 7	
Test Professional	Hayır	Hayır	Full	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır	Evet	Hayır	Evet	Hayır	Hayır	

2.4.4.1. Versiyon Geçmişi

Visual Studio 4.0 versiyonundan önce, ayrı ürün olarak, Visual Basic 3, Visual C++, Visual FoxPro and Source Safe bulunmaktaydı.

Product name	Codename	Internal version	.NET Framework version	Release date
Visual Studio	Yok	4.0	Yok	Spring 1995
Visual Studio 97	<i>Boston</i>	5.0	Yok	1997
Visual Studio 6.0	<i>Aspen</i>	6.0	Yok	1998-06
Visual Studio .NET (2002)	<i>Rainier</i>	7.0	1.0	2002-02-13
Visual Studio .NET 2003	<i>Everett</i>	7.1	1.1	2003-04-24
Visual Studio 2005	<i>Whidbey</i>	8.0	2.0	2005-11-07
Visual Studio 2008	<i>Orcas</i>	9.0	3.5	2007-11-19
Visual Studio 2010 /Ultimate 2010	<i>Dev10/Rosario</i>	10.0	4.0	2010-04-12
Visual Studio 2012		11.0	4.5	2012-09-12

2.5. Veritabanı Sistemleri

2.5.1. Veri Nedir?

Veri (Data) ve Bilgi (Information) terimleri sık sık birbirinin yerine kullanılmaktadır. Veri kavramı kimi yerde olayları veya yerleri, kimi yerde insanları veya diğer nesnelere ilgilendiren gerçeklerdir. Dolayısıyla veri kavramı her alan için farklılıklar gösterebilir. Biz bilgisayar ortamında veri kavramını yerine göre, bilgisayarın bir manyetik disk üzerinde (örn. Hard disk), yarı iletken oluşturmuş bir hafıza biriminde (örn. RAM) veya bir veritabanı içerisinde işlenebilecek durumda bulunduran kayıtlar olarak kabul edeceğiz. Veri çeşitli şekillerde işlenmeye veya tasfiye edilmeye hazır durumda olan fakat ilk bakışta faydasız ve anlamsız gibi görünen bir sürü kaydı gösterir.

Bilgi kavramı işlenmiş ve kullanıcı için yararlı ve kullanışlı olan verilere denmektedir. Örneğin öğrenci bilgilerinin tutulduğu bir veritabanı düşünelim. Bu veritabanından “Bilgisayar Teknolojisi ve Programlama” bölümü öğrencileri içinden “Veri Tabanı Yönetim Sistemleri” dersinden bütünlemeye kalan öğrencilerin listesi, veritabanları dersini veren ders hocası için çok kullanışlı ve önemli bir bilgiyi göstermektedir.

2.5.2. Veri Tabanı Nedir ?

Veritabanı, birbiriyle ilişkisi olan verilerin tutulduğu, kullanım amacına uygun olarak düzenlenmiş veriler topluluğunun mantıksal ve fiziksel olarak tanımlarının olduğu bilgi depolarıdır. Veritabanları gerçekte var olan ve birbirleriyle ilişkileri olan nesnelere ve ilişkilerini modeller.

Veri tabanı; banka üniversite, okul, seyahat şirketi, hastane, devlet dairesi gibi bir kuruluşun çalışıp işleyebilmesi için gereken uygulama programlarının kullandığı operasyonel çok çeşitli verilerin toplamıdır. Ticari bir şirket için müşteri bilgileri, satış bilgileri, ürün bilgileri, ödeme bilgileri vb. Okul için öğrenci bilgileri, açılan dersler, okula kaydedilmiş öğrenciler, öğretmen bilgileri vb. Hastane için hasta bilgileri, doktor bilgileri, yatakların doluluk ve boşluk bilgileri, teşhis-tedavi bilgileri, mali bilgiler vb. Kullanılan çok çeşitli operasyonel verilere örnek olarak gösterilebilir.

Belirli bir konu hakkında toplanmış veriler bir veritabanı programı altında toplanırlar. Bu verilerden istenildiğinde; toplanılan bilgilerin tümü veya istenilen özelliklere uyanları görüntülenebilir, yazdırılabilir ve hatta bu bilgilerden yeni bilgiler üretilerek bunlar çeşitli amaçlarla kullanılabilir. Veri tabanı ile ilgili bazı tanımlamalar listelenmiştir.

2.5.3. Neden Veritabanı Kullanılır?

Bilgisayar ortamında verilerin tutulması, saklanması ve erişilmesinde günümüze kadar değişik yöntem ve yaklaşımlar kullanılmıştır. Bu yaklaşımlardan biri olan Geleneksel Yaklaşım verilerin ayrı ayrı dosyalarda gruplanma yaklaşımını kullanmaktadır. Veritabanı programlarından önce verileri saklamak için programlama dillerinde sıralı ve rastgele dosyalama sistemleri kullanılırdı. Bu sistem; birbiriyle ilgili olan ve aynı gruba dâhil olan verilerin bir dosyada tutulması yöntemine dayanmaktadır. Verilerin artması, verilere aynı anda erişilmesi ve aynı anda erişilen verilerin erişenlere göre düzenlenmesi gibi ihtiyaçlar arttıkça geleneksel yaklaşım yetersiz kalmıştır. Veri tabanı yaklaşımının yararları aşağıda sıralanmıştır.

- Ortak verilerin tekrarının önlenmesi; verilerin merkezi denetiminin ve tutarlılığının sağlanması,
- Veri paylaşımının sağlanması,
- Fiziksel yapı ve erişim yöntemi karmaşıklıklarının, çok katmanlı mimarilerle kullanıcılardan gizlenmesi,
- Her kullanıcıya yalnız ilgilendiği verilerin, alışıktığı kolay, anlaşılır yapılarda sunulması,
- Sunulan çözümler, tasarım ve geliştirme araçları ile uygulama yazılımı geliştirmenin kolaylaşması,
- Veri bütünlüğü için gerekli olanakların sağlanması, mekanizmaların kurulması,
- Güvenlik ve gizliliğin istenilen düzeyde sağlanması
- Yedekleme, yeniden başlatma, onarma gibi işletim sorunlarına çözüm getirilmesi.

2.5.4. Veri Tabanı Yönetim Sistemi Nedir?

Veritabanı Yönetim Sistemi (VTYS), yeni bir veritabanı oluşturmak, veritabanını düzenlemek, geliştirmek ve bakımını yapmak gibi çeşitli karmaşık işlemlerin gerçekleştirildiği birden fazla programdan oluşmuş bir yazılım sistemidir. Veritabanı yönetim sistemi, kullanıcı ile veritabanı arasında bir arabirim oluşturur ve veritabanına her türlü erişimi sağlar. Veritabanı yönetim sistemi programları; fiziksel hafızayı ve veri tiplerini kullanıcılar adına şekillendirip denetleyen ve kullanıcılarına standart bir SQL ara yüzü sağlayarak onların dosya yapıları, veri yapısı, fiziksel hafıza gibi sorunlarla ilgilenmek yerine veri giriş-çıkışı için uygun ara yüzler geliştirmelerine olanak sağlayan yazılımlardır. VTYS'de kullanıcılar, roller ve gruplar vardır ve bunlar verileri tutmak üzere birçok türde nesne ve bu nesnelere

erişimleri düzenleme görevi yaparlar. Her bir kullanıcının veritabanı yöneticisi tarafından yapılan tanımlanmış belirli hakları vardır. Bu haklar verilebilir, verilmiş haklar arttırılabilir, kısıtlanabilir veya silinebilir.

2.5.5. Veri Tabanı Yönetim Sisteminin Sağladığı Yararlar

- Veri tek bir merkezde tutulur ve aynı veri her kullanılan değişik bilgisayarda tekrar tekrar tutulmaz.

- Eklenen, düzenlenen veya silinen bir bilgi merkezde değil de kişisel çalışılan bilgisayarlarda yapılırsa bu bilgi ile merkezdeki farklı olacaklarından veri tutarsızlığı oluşur. Aynı verinin değişik yerlerde birkaç kopyasının bulunması aynı zamanda bakım zorluğunu da beraberinde getirir.

- Veritabanı Yönetim Sistemi kullanılmayan sistemlerde veriye erişim sıralı yapılır ve birden çok kullanıcı aynı anda aynı veriye erişemezler. Erişmek istediklerinde ilk istekte bulunan erişir ve diğerleri onun işleminin bitmesini bekler. Hazırlanan programda sadece okumak amacıyla tüm kullanıcılara erişme hakkı verilebilse de hepsi işlem yapmak istediklerinde ilk kullanıcının işlemini yapmadan diğerlerine işlem yapma izni vermez. Bir VTYS“de ise verinin tutarlılığını ve bütünlüğünü bozmadan aynı veritabanlarına saniyede yüzlerce, binlerce erişim yapılabilir.

- Bir tabloda bir işlem yapıldığında buna bağlı diğer tablolarda da ilgili işlem yapılır. Örneğin bir okuldaki bir öğrencinin kaydı silinirse, öğrencinin not, harç ve diğer bilgileri de diğer tüm tablolardan silinmelidir.

- Verilerin kasıtlı veya yanlış kullanım sonucu bozulmasını önlemek için verilerin korunabildiği özellikler bulunur. Örneğin; veri tabanına girmek için kullanıcı adı ve şifre istenmesi, kişilerin sadece kendilerine verilen haklar ölçüsünde tablolar ya da tablo içinde belirli kolonlarla işlem yapabilmeleri gibi.

- Programcı, kullandığı verilerin yapısı, organizasyonu ve yönetimi ile ilgilenmeden veritabanının bunları kendinin koordine etmesi ve yönetmesidir. Veri bağımsızlığı, veritabanı yönetim sistemi programlarının en temel amaç ve özelliklerindedir.

2.6 Microsoft SQL Server

2.6.1. Tarihi

Doğuşu

Versiyon 7.0'dan önce "kod tabanı" Sybase SQL Server tarafından Microsoft'a satıldı, ve bu Microsoft'un kurumsal seviyede veritabanı pazarına girişi oldu. Sybase SQL Server 3.0 ile esasen aynı olan ilk versiyon, SQL Server 1.0'ı yaratmak ve pazarlamak adına Microsoft, Sybase ve Ashton-Tate ile takım oluşturdu. 1992'de Microsoft SQL Server 4.2 sevkedildi. Daha sonra Windows NT 3.1 ile aynı zamanda Microsoft SQL Server 4.21 piyasaya sunuldu. Microsoft SQL Server 6.0 Windows NT için dizayn edilmiş ilk versiyon olmasıyla birlikte Sybase'den talimat alınmaksızın piyasaya çıkartıldı.

SQL Server 7.0, miras Sybase kodu ile yazılan bir "rewrite" versiyonu oldu, ve yerine SQL Server 2000 çıkartıldı. SQL Server 2000, IA-64 mimarisinden farklı olarak yazılan ilk sürüm oldu.

SQL Server 2000 sürümünden 10 yıl sonra performansta artışlar görüldü, IDE araçlarını ve tamamlayıcı başka sistemleri içeren SQL Server 2005 piyasaya sunuldu.

SQL Server 2005

SQL Server 2005 (Kod Adı Yukon), Ekim 2005 tarihinde piyasaya sunuldu. İlişkisel dataya ek olarak, XML datayı düzenlemeye destek içerdi. Bu amaçla, veritabanı sütunlarında bir veri tipi ya da sorgularda sabitler (literals) olarak kullanılacak bir XML veri tipi tanımladı. Özelleştirilmiş indeksleme, XML veri için kullanılabilir hale geldi. XML veri, XQuery kullanılarak sorgulanma imkanı sunuldu, bu sayede kişinin CLR (Common Language Runtime) ile Yönetilen Kod (Managed Code) olarak SQL kodu yazmasını mümkün kıldı. SQL Server 2005 ayrıca T-SQL içerisine XQuery sorgularını yerleştirmek için T-SQL diline eklemeler yaptı. Ek olarak, XQuery'ye, XML veriye sorgulama bazlı modifikasyonları mümkün kılan "XML DML" adı verilen yeni bir eklenti tanımladı.

SQL Server 2005, yeni indeksleme algoritması ve daha iyi hata geri dönüştürme (error recovery) gibi yeni özellikler ile geldi. İzinler ve giriş kontrolü daha granüler hale getirildi ve sorguların eş zamanlı olarak sorgulama işlemcisi (query processor) tarafından yürütülmesi daha etkin hale getirildi. Tablo üzerindeki bölümlenmeler ve indeksler desteklendiğinden, veritabanı kümeleme kolaylaştırıldı. .NET Framework ile birleşimi için SQL Server 2005 ile birlikte SQL CLR tanıtıldı.

SQL Server 2005, çoklu amaç için veritabanı kullanımını sağlayan "MARS"ı (Multiple Active Results Sets) tanıttı.

SQL Server 2005, "Server instance" sađlıđının kontrol edilmesi, problem teřhis etme ve performas ayarı gibi bilgilerin edinilmesini sađlayan "DMVs" (Dynamic Management Views) teknolojisini tanıttı.

SQL Server 2008

SQL Server'ın bir sonraki versiyonu, SQL Server 2008, 6 Ađustos 2008 tarihinde piyasaya sũrũldũ. Neredeyse sıfır kapalı kalma sũresi, veri y¶netimini kendi kendine ayarlanabilen, organize olabilen ve devamlılıđını sađlayan bir hale getirmek SQL Server 2008'in getirdiđi yeniliklerden bazılarıydı. SQL Server 2008 ayrıca, resim, video ve diđer multimedya formatlarını da i¶eren, yapılandırılmıř ve yarı-yapılandırılmıř veriler i¶in destek sundu.

Mekana bađımlı (location-dependent) veri i¶in ¶zelleřtirilmıř tarih ve zaman tipleri, ve Mekansal veri (Spatial data) yeni veri tipleri arasında yerini aldı. Dosya sistemindeki herhangi bir dosyanın referans edilmesi i¶in yapılandırılmamıř ve yarı-yapılandırılmıř verilere daha iyi destek, yeni veri tipi "FILESTREAM" ile sađlandı.

Y¶netimi kolaylařtıran ve performansı arttıran Tam Metin Arama (Full-Text Search) ¶zelliđi SQL Server 2008 ile birlikte tanıtıldı.

SQL Server 2008, ¶l¶eklendirmeye ve indeksleme algoritmalarına da katkıda bulunan ve "filtrelenmiř indeksler" (filtered indexes) kavramıyla a¶ıđa ¶ıkan, daha iyi sıkıřtırma ¶zellikleri de getirdi.

SQL Server 2008 ayrıca veritabanlarını "Cmdlets" olarak Windows PowerShell sađlayıcıları üzerinden kullanılabilir hale getirdi, bu sayede sunucu ve ¶alıřan tũm oluřumlar Windows PowerShell üzerinden yũrũtũlebilir duruma geldi.

SQL Server 2008 R2

SQL Server 2008 R2 (eski Kod Adı "Kilimanjaro") TechEd 2009'da duyuruldu ve 21 Nisan 2010 tarihinde piyasaya sunuldu. SQL Server 2008 R2, SQL Server 2008'e ek olarak ana veri birimleri ve hiyerarřileri y¶netimi sađlayan "Master Data Services" olarak adı ge¶en yeni bir ana veri y¶netim sistemi ekledi. Ayrıca ¶oklu SQL Server oluřumlarını y¶netmeye yarayan ¶oklu Sunucu Y¶netimi (Multi Server Management), Raporlama Servisi (Reporting Service), Analiz ve Entegrasyon Servisleri (Analysis & Integration Services), Excel ve SharePoint i¶in "PowerPivot", "StreamInsight", eklenen yeni ¶zellikler arasında olduđu g¶rũldũ.

2.6.2. Versiyonlar

Microsoft, SQL Server'ı farklı versiyonlar ile sunmaktadır. Bu versiyonlar, farklı özellik setleri içerdiği gibi, farklı kullanıcıları da hedef alır.

2.6.2.1. Temel (Mainstream) Versiyonlar

Datacenter

SQL Server 2008 R2 Datacenter, yüksek seviye uygulama desteği ve ölçeklenebilirlik ihtiyaçlarını gideren, verimerkezleri (datacenters) için tasarlanmıştır. 256 mantıksal işlemciyi ve neredeyse sınırsız hafıza desteklemektedir. StreamInsight Premium versiyonuyla birlikte gelmektedir.

Enterprise

SQL Server Enterprise versiyonu, SQL Server kümelerini yaratmak ve düzenlemek için gerekli araçları içeren versiyon olup, çekirdek veritabanı motoru ve add-on servislerini desteklemektedir. 524 petabayta kadar veritabanı yönetebilmekte, 2 terabayt hafıza içermekte ve 8 fiziksel işlemci desteklemektedir.

Standard

SQL Server Standart versiyonu, çekirdek veritabanı motorunu, bağımsız servislerle (stand-alone services) birlikte içermektedir. Daha az aktif instance (kümedeki ağ sayısı) desteklemesi, hot-add memory (sunucu çalışırken hafıza eklenebilmesi) gibi yüksek süreklilik fonksiyonları ve paralel indeksleri içermemesi, SQL Server Enterprise versiyonundan farklı olduğu noktalar olmaktadır.

Web

SQL Server Web versiyonu, Web hosting için "low-TCO" (toplamda sahip olma maliyeti) bir seçenektir.

Workgroup

SQL Server Workgroup versiyonu, çekirdek veritabanı fonksiyonlarını içermekte ancak ek servisleri içermemektedir.

Express

SQL Server Express versiyonu, çekirdek veritabanı motorunu içeren, ücretsiz bir versiyondur. Veritabanı ve kullanıcı sayısında bir sınırlama yoktur, ancak 1 işlemci, 1 GB hafıza ve 4 GB veritabanı dosya limiti bulunmaktadır.

2.6.2.2. Özelleştirilmiş (Specialized) Versiyonlar

Azure

Microsoft SQL Azure Veritabanı, Microsoft SQL Server'ın bulut bazlı (cloud-based) versiyonudur. Azure Servisleri Platform'unda "software as a service" (servis olarak yazılım) olarak sunulmuştur.

Compact (SQL CE)

SQL Server Compact versiyonu, gömülü bir veritabanı motorudur. Diğer SQL Server versiyonlarının aksine, SQL CE motoru, SQL Mobile (başlangıçta mobil cihazlar için tasarlanmıştır) üzerine kuruludur ve aynı ikili değerleri paylaşmamaktadır. Boyutunun küçüklüğüne (1 MB DLL footprint) ek olarak, özellik setleri belirgin olarak azaltılmıştır. Windows Service olarak çalıştırılmayacağı gibi, 4 GB veritabanı limiti bulunmaktadır.

Developer

SQL Server Developer versiyonu, SQL Server Datacenter versiyonuyla aynı özellikleri içermektedir, ancak sadece geliştirme (development) ve test sistemi olarak kullanılabilir.

Embedded (SSEE)

SQL Server Embedded versiyonu, SQL Server Express veritabanı motorunun, sadece belli Windows Servisleri tarafından ulaşılabilen, özel olarak düzenlenmiş bir ismidir.

Evaluation

SQL Server Evaluation versiyonu, Deneme Sürümü (Trial Edition) olarak bilinen, Enterprise versiyonunun tüm özelliklerini içeren, sadece 180 güne sınırlı olan versiyonudur.

Fast Track

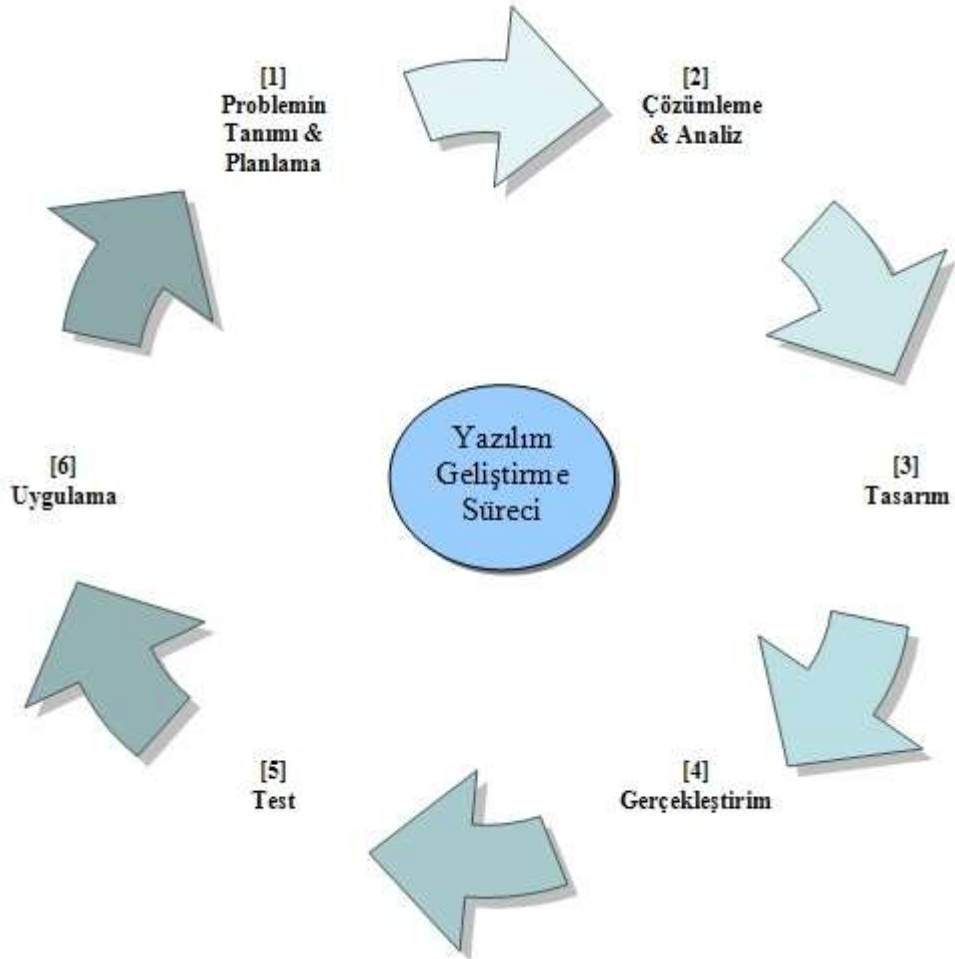
SQL Server Fast Track, özel olarak, ticari kurum bazında, veri depolama ve zeka işleme (intelligence processing) işlemlerinde kullanılan versiyondur.

Parallel Data Warehouse (PDW)

Parallel Data Warehouse versiyonu, yüzlerce terabaytlık veri depolama işlemleri için optimize edilmiş versiyondur.

3. PROJE

3.1. Yazılım Geliştirme Sürecinde Geliştirme Aşamaları



3.2. Yazılım Geliştirme Süreci

İlk olarak yazılımdan ne beklendiğinin belirlenmesi gerekir. Yani çözümün tanımlanmasıyla başlanır yazılım projelerine. Ardından çözümün geliştirilmesi adımı gelir. Son olarak da çözüm müşteriye kurulur ve bakım/destek süreci başlar.

1. Tanımla ve planla
2. Çözümü geliştir
3. Yazılımı kur ve destekle

3.2.1. Birinci Aşama: Tanımlama/Planlama

Ortada bir müşteri varsa yazılım projesinden bahsedilir. Bir müşteri varsa, mutlaka bir beklenti de vardır!

Müşterinin ne istediğini öğrenip, ayırdığı bütçeyle belirlediği zaman dâhilinde gerçekte nelere ulaşabileceğinin belirlendiği aşamadır bu aşama. Yazılımın “ne yapacağı”, bu aşamanın yazılım geliştirme sürecindeki payıdır. Bu aşamanın birde her bir yazılım geliştirme sürecinin bir proje oluşu nedeniyle planlama, bütçeleme, ekip oluşturma vb. yönü de olacaktır.

3.2.2. İkinci Aşama: Geliştirme

“Ne istendiği” tam olarak anlaşıldıktan sonra, sıra bunu yapmaya gelir. Öncelikle istenenin hangi teknolojileri, programlama dillerini, veri tabanı altyapısını kullanarak, nasıl sağlanacağı belirlenir. Çözümün tasarımı adımı olarak bilinen bu adımdan sonra ancak kodlamaya başlanır. Kodlama sırasında ve sonrasında da yapılanın isteneni karşılayıp karşılamadığı sürekli test edilir.

3.2.3. Üçüncü Aşama: Bakım/Destek

Yazılımlar canlı varlıklardır. Yazılımların yaşam döngüsü iki temel süreçten oluşur. Geliştirilme ve bakım.

Müşterinin istekleri ve kaynaklar çerçevesinde geliştirilen yazılım, müşteri yerine yüklendikten ve kullanıcılara eğitimleri verildikten sonra gerçek anlamda yaşamaya başlamıştır. Bu noktadan itibaren kullanıcı ihtiyaçlarının farklılaşması, işin yapılış biçiminin değişmesi, donanım yükseltmesi, yeni teknolojilerin gelişmesi gibi birçok farklı nedenden dolayı yazılımlarda değişiklik yapılması gerekecektir. İşte bu süreç bakım sürecidir.

Sebepten ya da şekil ne olursa olsun, yazılım geliştirme sürecinin en önemli adımı “ne yapılacağı” doğru belirlenmesi ve doğru planlanmasıdır. Ne yapılacağını doğru anlamadıysanız, ne kadar kaliteli bir tasarım çıkartıp, ne kadar

dođru bir şekilde kodlarsanız kodlayın sonuç hatalı olacaktır. Tabii ki sonrasında yapacağınız şeyi nasıl yapacağınızı (tasarım) dođru belirlemeniz ve bunu dođru kodlamanız (geliştirme) önem kazanacaktır. Ama neticede ne kadar iyi bir uygulama geliştirmiş olursanız olun, müşterinin beklentisini yerine getiremeyen bir uygulama, başarısız bir uygulama olacaktır.

Genel hatlarıyla 3 bakım türü vardır:

1. Düzeltici bakım: Tespit edilen hataların giderilmesi işlemidir. Kodlama hatalarını düzeltmek genelde az maliyetlidir. Tasarımdan kaynaklı hataların giderilmesi ise bazı sistem bileşenlerinin baştan yazılmasını vb. gerektirebilir ve nispeten yüksek maliyetlidir.

2. Uyarlayıcı (Adaptif) bakım: Yazılımın yeni bir çalışma ortamına uyarlanmasıdır. Bu bir donanım platformu değişikliği olabileceđi gibi (32 bitten 64bite geçiş gibi) farklı bir işletim sistemine uyarlama şeklinde de olabilir (kodun Windows'tan Linux'a taşınması gibi). Ayrıca, veri tabanı sistemi değişikliği de bu türden bir bakım olarak görülebilir (MS SQL Server bağımlı kodların Oracle'a uyarlanması gibi).

3. İyileştirici bakım: Sisteme yeni işlev ve özelliklerin eklenmesi, performansın artırılması gibi bakım çalışmalarıdır.

80'li ve 90'lı yıllarda yapılan araştırmalar göstermiştir ki bakım çalışmalarının %65'i iyileştirici, %18'i uyarlayıcı ve %17'si düzeltici bakım şeklinde gerçekleşmektedir. Bu rakamları günümüzde de dođru kabul edebiliriz.

Genellikle bir sisteme sistem çalışmaya başladıktan sonra yeni bir işlev eklemek, aynı işlevin henüz geliştirme sürecindeyken eklenmesine göre çok daha maliyetlidir.

3.3. Projemizin Geliştirme Aşaması

3.3.1. Birinci Aşama: Tanımlama/Planlama

Hastane Otomasyonu

Hastaneler birden çok fonksiyonun bir arada götürüldüğü kompleks işletmelerdir. Öncelikle hastane bir işletmedir, daha sonra tıbbi hizmet veren bir kuruluştur, ayrıca otelcilik hizmeti verirler. Binlerce kalemi bulan ilaç ve tıbbi malzemenin yönetilmesi bile başlı başına ayrı bir hizmet olarak alınabilir. Bu karmaşık yapı içerisinde bilgisayar kullanımı birçok sorunu çözebileceđi gibi beraberinde birçok sorunu da gündeme getirebilir.

Hastanelerde kaynakların yerinde kullanılması ve israfın önlenmesi, oluşturulmaya çalışılan kalite ve standartların tesis edilmesi ve korunması, hastanelerde sunulan sağlık hizmetinin en iyi şekilde verilebilmesi, gelir ve giderlerin izlenmesi, gelir kaçaklarının önlenmesi, hastane yönetimine verilecek önemli kararlarda bilgi desteğinin sağlanması, sağlanan bilgi desteğiyle ileriye yönelik doğru hedef belirlenmesi ve doğru kaynak yönetimi, eksiksiz tıbbi kayıt, muhasebe, depo, ambar, demirbaş kayıtları tutabilmek amacıyla kullanılan bilgi sistemleridir.

Projedeki modüller: Projede olması düşünülen modüller aşağıdaki gibidir,

- Login modülü
- Patient modülü
- Appointment giriş modülü
- Survey modülü
- Labratory modülü
- Radiological modülü
- Cashier's-Office modülü
- Stocks modülü

3.3.2. İkinci Aşama: Geliştirme

“Ne istendiği” tam olarak anlaşıldıktan sonra, sıra bunu yapmaya gelir.

3.4. Projede Kullanılacak Teknolojiler

Proje de programlama dili olarak Microsoft firması tarafından geliştirilen CSharp(C#) dili kullanılacaktır. C# ile yazılımlar daha esnek, geliştirilebilir ve temiz bir kodlama standarttı ile kodlanmış olur. Bu da maliyet ve hız açısından çok büyük bir getirdir.

Projemizde Sql Server 2008 Database kullanılacaktır. Bu veritabanı Microsoft firması tarafından geliştirilmiş bir veritabanıdır. Büyük projelerde hiç çekinmeden kullanılabilecek bir veritabanıdır. Özellikle Visual Studio ile olan iletişimi gerçekten çok harikadır. Bu sayede veri tabanımızı geliştirme IDE miz üzerinden de rahatlıkla kullanabiliriz.

Editör olarak Visual Studio 2012 kullanılacaktır. Çok gelişmiş bir editör olarak karşımıza çıkmaktadır. Editörü <http://www.microsoft.com/visualstudio/eng/downloads#d-2012-express> adresinden indirebiliriz.

3.4.1. Üçüncü Aşama: Bakım/Destek

Zaman içerisinde deęişim ihtiyacı duyulmayacak bir yazılım sistemi düşünülemez. Kullanıcı ya da müşterilerin ihtiyaçlarındaki deęişimlerin sisteme yansıtılması gerekir. Ayrıca, yeni bir donanım ya da yazılım altyapısı nedeniyle sistemin çalışma koşulları deęişebilir. Tabii ki testler sırasında fark edilmeyen hatalar tespit edilebilir ve giderilmesi gerekir.

Yazılımın dağıtılması ve kullanıma başlanmasından sonra yazılımda yapılacak deęişiklikler yazılımın bakımı (software maintenance) olarak adlandırılır. Bu deęişiklikler basit kodlama hatalarının düzeltilmesi (bug-fixes) şeklinde olabileceęi gibi tasarımdan kaynaklanan hataların giderilmesi gibi daha kapsamlı deęişiklikler şeklinde de olabilir. Yazılımın bakımı aslında yazılımın evrimleşmesidir. Yazılımın yaşamına devam edebilmesi için gerekli deęişikliklerin uygulanmasıdır.

3.6. Örnek Modül Görünüm ve Açıklamaları

- **Login Page**

Programa sadece yetkili kişilerin erişimini, diğer kişilerin program erişimini engellemek için LoginPage sayfamız hazırlanmıştır.



- **Main Page**

Programın bu kısmı çeşitli ihtiyaçlarımıza cevap vermesi için hazırlanmış formlara ulaşmak için tasarlanmıştır.



- **Hasta Kayıt**

Bu forum sistemimize yeni hasta kaydı, mevcut hasta kaydı düzenlemesi ve mevcut hasta kaydı silme işlemleri için tasarlanmıştır.



- **Randevu**

Sisteme kayıtlı hastalarımız için hasta randevu formu tasarlanmış olup bu form hastalara randevu vermek için tasarlanmıştır. Bu form sisteme hasta kayıt etmemize yarayan formdur.



- **Muayene Kayıtları**

Bu form mevcut hastaların muayene kayıtlarına ulaşmak için tasarlanmıştır. Burada hasta muayene bilgileri güncellemeleri tectik istekleri ve reçete bilgileri işlenebilmektedir.

Filtreleme Seçenekleri
Toplam Bulunan Sayısı

Başlangıç Tarihi
06 Haziran 2013 Pazartesi

Bitiş Tarihi
06 Haziran 2013 Pazartesi

T.C. Kimlik No/Nüfus Kayıt No

İsmler
Yeni Ekran Görü Listeye Yazdır İptal

Randevular

operasyonDate	appointmentTime	patientName
---------------	-----------------	-------------

Hasta Bilgileri Muayene Bulguları Laboratuvar Tetkikleri Radyoloji Tetkikleri Tedaviler Reçete

TC Kimlik No
İsim
Soyisim
Cinsiyet
İl
İlçe
Mahalle
Sokak
Öz Geçmiş

Baba Adı
Anne Adı
Doğum Tarihi 06.06.2013
Kan Grubu
Cep Tel
Evliş Tel
E-mail
Soy Geçmiş
Ameliyatlar

Tamam

- **Vezne İşlemleri**

Bu form mevcut hastaların muayene kayıtlarına ulaşmak ve hastaların muayene, tetkik vb. işlemlerini faturalandırmak, ayrıca faturalandırılmış bu hizmetlerin ücretlerinin tahsil edilmesi için tasarlanmıştır.



- **Stok İşlemleri**

Bu form mevcut ilaç stoklarıyla ilgili düzenlemeleri yapmak, hastane stoklarına yeni ilaçlar eklemek, silmek, stoklardaki ilaçları depolar arasında transfer etmek gibi işlemler için tasarlanmıştır.

Ana Depo

Stok Giriş  Sipariş Malzemeni Depoya Aktar  Sipariş Malzemeni Depodan Çıkart   

medicineID	medicineName	medicineNumber	medicinePrice
1	ASPIRIN	24	12
5	VERMIDON	10	10
4	NOVALGIN	10	10

Alt Depolar

Aşağıda Bilgileri Listelenen İçin Bir Depo Seçiniz 1 Depo

storeID	storeName	medicineName	medicineNumber
15	1. Depo	ASPIRIN	120
16	1. Depo	VERMIDON	10
17	2. Depo	NOVALGIN	40

KAYNAKLAR

- [1] Kernighan, Brian W.; Ritchie, Dennis M. (Şubat 1978). C Programlama Dili (1. bas.). Englewood Cliffs, NJ: Prentice Hall. ISBN 0-13-110163-3.
- [2] Kernighan, Brian W.; Ritchie, Dennis M. (Mart 1988). C Programlama Dili (2. bas.). Englewood Cliffs, NJ: Prentice Hall. ISBN 0-13-110362-8.
- [3] www.bilisimogrenci.com
- [4] www.zerrinayvazreis.com.tr
- [5] www.CppReference.com (İngilizce)
- [6] www.Cplusplus.com (İngilizce)
- [7] www.parashift.com (İngilizce)
- [8] Take Visual Studio 2010 For a Test Drive
- [9] Visual Studio Editions". Microsoft. Erişim tarihi: 2010-06-22.
- [10] Visual Studio Editions". Microsoft. Erişim tarihi: 2010-06-22.
- [11] "Computer Language Benchmarks Game". 2007-09-08.