

**İSTANBUL TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ**

PROJECT TEAMWORK

Bitirme Ödevi

**Aykut AKINCI
040020388**

**Selçuk Özdoğan
040020340**

**Bölüm : Bilgisayar Mühendisliği
Anabilim Dalı: Bilgisayar Bilimleri**

Danışman : Yrd. Doç. Dr. Feza Buzluca

Mayıs 2006

**İSTANBUL TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ**

PROJECT TEAMWORK

Bitirme Ödevi

**Aykut AKINCI
040020388**

**Selçuk Özdoğan
040020340**

**Bölüm : Bilgisayar Mühendisliği
Anabilim Dalı : Bilgisayar Bilimleri**

Danışman : Yrd. Doç. Dr. Feza Buzluca

Mayıs 2006

Özgünlük Bildirisi

1. Bu çalışmada, başka kaynaklardan yapılan tüm alıntılarını, ilgili kaynaklar referans gösterilerek açıkça belirtildiğini,
2. Alıntılar dışındaki bölümlerin, özellikle projenin ana konusunu oluşturan teorik çalışmalarını ve yazılım/donanımını benim tarafımdan yapıldığını bildiririm.

İstanbul, 22 Mayıs 2006

Aykut Akıncı

Selçuk Özdoğan

İmza

İmza

PROJECT TEAMWORK

(ÖZET)

Yazılım alanında son on ila on beş yıl içindeki gelişmeler ki bilhassa nesneye dayalı programlamanın yaygın bir biçimde kullanılmaya başlaması, her geçen gün daha güçlü bilgisayarların yapılması ve iş dünyasının bilgisayarlarla giderek daha da kaynaşması artık eskisinden çok daha büyük yazılım projelerinin yapılmasını zorunlu kılmıştır. Eskiden yazılım projeleri genellikle tek kişi veya iki kişi tarafından gerçekleştirirken bu gün yüzlerce kişinin çalıştığı yazılım projeleri bulunmaktadır. Bu durum yazılım projelerinin yapılmaya başlanmadan önce iyice planlanmasını gerektirmiş, yazılım mühendisliği dalının oluşmasına sebebiyet vermiştir. Günümüzde yazılım projeleri yazılım mühendisliğinin de yardımıyla bina planlaması yapılmış gibi planlanmaktadır. İşte Project Teamwork de tam bu noktada yazılım mühendislerinin ihtiyacını karşılamak üzere geliştirilmiş bir programdır.

Project Teamwork yardımıyla bir yazılım projesi baştan sona planlanabilir. Görevler gantt chart da düzenlenebilir. Görevler arasındaki bütün ikili bağlar kurulabilir. Bunlar Start to Start, Start to Finish, Finish to Start ve Finish to Finish' tir. Bu görevlere istenen proje elemanlarının istenen kadar zamanı ayırması sağlanabilir. Yine benzer şekilde istenen göreve istenen bir kaynak istenen miktarda atanabilir. Atanan bu kaynakların ve insanların olandan fazla atanıp atanmamış olduğu kontrol edilebilir. Critical Path görsel ve dinamik olarak gözlenebilir. Görevin ne kadarının tamamlanmış olduğu gene görsel olarak gösterilebilir. Bütün bu özellikler Project Teamwork'ü planlama aşamasından projenin bitimine kadar kullanılabilecek bir yazılım yapmaktadır.

Bütün bu özelliklerinin yanında Project Teamwork izin yapısıyla proje içindeki elemanların da istenen modüllere ulaşmasını sağlayabilir. Böylelikle proje içindeki elemanlar da görevlerini görebilir, diğer proje elemanlarıyla mail yoluyla haberleşebilirler. Proje yönetimine projenin elemanlarının da katılması sağlanmış olur. Böylelikle takım çalışmasına tam uyum sağlanmıştır.

PROJECT TEAMWORK

(SUMMARY)

The great development in computers and in programming idea, especially the idea of object oriented programming has lead programmers to write better and longer programs everyday. Twenty years before almost every program was written by only one programmer. In those days the programmer was planning his project and writing the code himself and there was no one else. If the programmers wanted to change somewhere in his program he could simply change that part. But today's programs are often much more bigger than those programs. Object oriented idea created the possibility for writing programs made up of millions of lines of code. For writing these millions lines of code there was need for teams, planning and management, namely the software engineering. Project Teamwork is the software to automate the tasks for these processes.

In the beginning software engineering was made with pen and paper. Charts or diagrams were drawn to paper and people had to obey the dates on those charts. However these charts that are drawn over the paper were far from the needs of the planning in the ways that we are now going to explain. First of all these charts can not be changed too many times after the first creation. However creating a software is implementing a new technology and can take various time. In addition to this you can not control your resources that you give for these tasks. So if you make these charts by hand it can give you no dynamism after the first creation and it would simply be a plan for the first days of the project. After the first month you have to make a new plan. On the other hand if you use software for creating these charts then you can change them dynamically while making the project and it will still remain the plan of the project after the first month...

The aim of the project teamwork is to automate the tasks of the project manager. We also wanted our program to be an easy to use, user friendly software. After inspecting almost ten project management software we decided that our software should include the following features:

- A module to show the tasks in the gantt chart. So that every task can be managed seperately.
- Implementing all the relations between these tasks. So that the software can be more easily used in the implementing process too, because that changing a task's duration would effect other tasks recursively to create the new plan.
- Visually inspectable percentages on tasks to make the software a full management software to use in implementing and in planning.
- Adding people and resources to the project.
- Employing people and giving resources to tasks. After these employments the project manager should also be aware if he has used more resources or more people than he has.
- Visually inspectable dynamic Critical Path.
- Adding project members to the project planning/inspecting process.

For implementing these features we decided to use .NET technology. The most familiar programming language was C# in the .NET framework for us so we used C#. This was an easy decision for us. The difficult was to choose the database engine.

When we thought about the engines and the software we wanted to make, we came across three different solutions.

- The first idea was not to use any database engine. In a project management software we did not expect that there would be tens of thousands of entries in a such database. So the project could easily be made with a xml based save engine. When we want a project saved this engine would xml-serialize the project class containing all the information about the project. Then it would save it as xml. When we wanted to load the project then we would have to xml-deserialize the class to read the project from the xml file. This would make the project independent from any database. While a company generally would not have thousands of projects this solution would not be a slow solution in anyway. In fact we also think that other programs about project management used a system like this one. We did not use this solution because we wanted our program to have personalized and more secure data like login info etc.
- The second idea was to use a database engine like Access that is smaller than those great database engines like Oracle or MS SQL. This seemed enough for making those logins secure enough. But when we were in the planning process we thought to make the project reachable over the web. When we decided to use the web access the realized that all the project created with the program would be in a single database. So this database had to be more powerful and more secure (Access has a great security leakage over the net that if you can write the url of the *.mdb file to your browser you can download the Access database.) than Access.
- The third idea was to use a strong database like MS SQL. This would solve all the problems over the net. The MS SQL database engine is more strong and more reliable than any other database like Access. In addition to this making a project with MS SQL would be a great experience. The only problem was that we never used MS SQL before. We did not know how difficult or easy it could be or what kind of problems we could encounter while using the MS SQL engine.

The best choice was to choose the MS SQL engine. We decided to use MS SQL 2005 then. After this decision we planned the modules that we are going to make. There had be a module to create a new project without getting deeper into the program. A person module to manage people. A module for managing resources. A module to show tasks. And a module to estimate the project at the beginning of the project.

We wanted our software to be able to work over the internet from desktop machines, from the browser and mobile machines like PDA.

After implementing our project we came up with the following modules.

- **Create Project Interface:** This was the easiest way to create a project in the programs we examined. This part uses the step by step method to create a new project. First the project details are entered here. Then goals, goal tasks, risks and

risk tasks are given. Then calendar details for the company is entered and this part is done.

- **Estimator:** This is the place where you use the methods of the software engineering to estimate the costs of the project. Basic COCOMO, Intermediate COCOMO and Function Point approach are included. The project manager can pre-estimate the costs of his project from this module to decide whether to start this project or not.
- **Person Module:** To add people to your project this module is used. Here you can add people and change their properties. Here you can change the wage, email address, phone number and etc.
- **Resources Module:** Just like the person module this module is used to manage the resources of the project. There are two types of resources. First type is the fixed cost which means that the cost is a fixed cost and can not be assigned to a task. For example buying a some computers is generally a fixed cost. The cost of the resource does not depend on the duration of the task. If the resource is not a fixed cost then it can be assigned to any task. When a resource is assigned to a task, the duration of the task effects the cost of the resource.
- **Permission Module:** This is the place where you set the permissions for the people you create. Every person you create in the person module may have a user id and password for logging into the project. After logging in a user can change the things that he/she is allowed to. Also in some modules restricted users will have a more limited functionality than the project manager.
- **Tasks Module:** This is the place where you set the tasks for your project. This is the most detailed part of the project so we must get deeper into it. The tasks module is made up of two components. In the left you see the tasks in a datagridview and in the right you see the tasks visually. For the component to the right a new component is created from the scratch. The component to the right gets the values from the datagrid and visually shows the tasks which is in fact a gantt chart. When the user changes something in the datagrid the gantt control is dynamically updated. In the same way when you change something in the gantt chart visually the datagrid is updated via events. For this purpose events are written to the newly created component. In addition to this the gantt chart had to support the predecessor property. Every task in the gantt chart may have a predecessor. That is another task can precede a task in four ways. This is implemented via some recursive algorithm. So that when a project's dates are changed because of its predecessors, the tasks that this task precede would be able to change. There is another property of the gantt chart that we had to prevent circular predecessors. That is when a task A precedes a task B and task B precedes another task C, then task C can not precede task A. Moreover any other tasks that task C precedes can not precede task A. This is implemented in the gantt module too.
- **Mail Sender:** This module is created to allow users to send project members email easily.
- **Utility module:** This module is made to be able make the project usable over the internet. By this module with a small modification we can make the functions accessible through the web.
- **Main module:** This is the parent form for the modules above. Almost all the modules are shown as child modules of this main form.

In addition to these modules three functions are implemented. The first one is showing the critical path. If you tick this option from the main menu then you can see the critical path

of your project in the gantt chart dynamically. That is when you change the duration of a task in any moment then the critical path is recalculated in that moment to show the new critical path. The second is for managing people. When assigning people to their tasks you can use percentage. So a person in your project can work more than one task at a time. When assigning a person more than one task you can assign more than 100 percent of his/her working time. If you choose people management from the main menu then you can see whether you made that mistake or not. If you made that mistake a form shows up and you can see where the mistake is. The third function is for resources. This is just like the one for the people. You can not use more resources than you have. So if your tasks use more than you have then it is shown to you. This is shown when you choose resource management from the main menu.

By implementing all these functions and modules we believe that we accomplished to make all the major needs of the project managers. With all its features and functions Project Teamwork can be used from the beginning of the project planning to the end of the project.

İÇİNDEKİLER

PROJECT TEAMWORK.....	1
PROJECT TEAMWORK.....	2
PROJECT TEAMWORK.....	4
PROJECT TEAMWORK.....	5
İÇİNDEKİLER.....	9
1 GİRİŞ.....	10
2 PROJENİN TANIMI VE PLANI.....	13
3 KURAMSAL BİLGİLER.....	16
4 ANALİZ VE MODELLEME.....	18
5 TASARIM, GERÇEKLEME VE TEST.....	31
6 DENEYSEL SONUÇLAR.....	58
7 SONUÇ ve ÖNERİLER.....	60
8 KAYNAKLAR.....	62

1 GİRİŞ

Özet bölümünde de değindiğimiz gibi yazılım projelerinin günümüzdeki çok büyük boyutlara ulaşması bunların planlanmasını artık zorunlu hale getirmiştir. Daha önceden elle kağıtlar üzerinde yapılan bu planlama işlemi günümüzde çeşitli yazılımlarla sağlanmaktadır. Bunlardan en önemlilerini burada sıralamak gerekirse kuşkusuz Microsoft Project[1] ve Rational[2] programları en önde gelenleri olacaktır. Gantt Project[3] ve Project Kickstart[4] da bu alandaki önemli yazılımlar arasındadır. Fakat bu tip yazılımların en büyük eskikliği takım çalışmasına uyum noktasında olmaktadır. Bu tip yazılımların bir çoğunda proje planını tek kişi hazırlar ve projenin takip aşamasında bu planı tek başına düzenlemek zorunda kalır. Oysaki yüzlerce görevin bulunduğu günümüzün yazılım projelerinde bu görevi bir kişinin tek başına yürütmesi gün geçtikçe daha da zorlaşmaktadır. Bu durumda kullanılacak en akıllıca yol kodu yazan elemanı da proje planlamasının ve yönetiminin içine sokmaktan geçer. Project Teamwork diğer adı geçen programların bir takım özelliklerini barındırmanın yanında proje elemanlarını da planın içine katarak diğerlerinden farklılaşacaktır.

Proje yapılmaya başlanmadan önce daha önce yapılmış olan benzer yazılımlar derinlemesine incelenmiştir. Bu konudaki incelediğimiz yazılımları kısaca bir sıralarsak:

- Microsoft Project
- Minuteman Plus
- Gantt Project
- Project Kickstart
- Dotproject
- Open workbench
- Tasse Estimator 2006

Bunlardan en göze batanları Gantt Project, Microsoft Project ve Project Kickstart yazılımları olmuştur. Bunlar projemize ana referans olarak seçilmiştir. Yani ayrıntıların nasıl yapılacağına karar verilirken bu yazılımlarda işin nasıl yapıldığına bakılmıştır. Bu üç yazılımın referans olarak seçilmesinin sebeplerini de burada kısaca anlatmakta fayda var. Project Kickstart yazılımı kolay kullanımı ve projelerin en kolay bir biçimde sade bir arayüzde planlanmasına olanak vermesiyle beğenimizi kazandı. Bizim yapacağımız yazılım da bunun gibi kolay bir arayüze sahip olmalıydı. Bu yazılımda kullanılan adım adım proje oluşturma kolaylığını kendi projemizde de kullanmaya karar verdik. Gantt Project bütün yazılımlarda ortak özellik olarak gördüğümüz gantt chart yapılması üzerine özelleşmiş açık kaynaklı bir projeydi. Her ne kadar açık kaynaklı olsa da kodlar yerine biz daha çok sistemin nasıl gerçekleştirildiğiyle ilgilendik. Açık kaynaklı bir sistemin son teknolojiye çok daha fazla yakın duracağını düşündük. Microsoft Project ise yazılım dünyasının standartlarını koyan Microsoft ailesinin bir ürünü olduğu için referans olarak alındı. Diğer projelerde gördüğümüz mantığını anlayamadığımız yerlerde ve yapılıp yapılmamasına karar veremediğimiz özellikleri yapıp yapmama kararını Microsoft Project'e bakarak aldık.

Referans aldığımız veya beğendiğimiz yazılımların hepsinde gantt diyagramı bulunmaktaydı. Görevlerin tarihleri görsel bir alanda görülüp gene aynı alandaki sürüklenme-bırakma

işlemleriyle değiştirilebiliyordu. Görevlere eleman ya da kaynak eklenebiliyordu. Görevler arasında bağlar kurulup koparılabilirdi. Gelişmiş raporlama özellikleri vardı. Ama bu konuda yapılmış olan yazılım projelerinin çoğu bu görev tablosu ve bunlara eklenen eleman kaynak ilişkilerinin etrafında dönüyordu. Aldığımız ilk kararlardan biri belki de bu tip görsel bir gantt diyagramını yazılımımıza eklememiz gerektiği oldu.

Yazılımlar dışında geçen dönem aldığımız Software Engineering dersinin notlarından da faydalandık.[5] Bu notlarda geçen BASIC, INTERMEDIATE COCOMO gibi hesaplama yöntemlerini de inceledik ve yazılım projemizde kullanmaya karar verdik. Work Breakdown Structure konusunu ise yedeğe aldık. Yani vakit yeterse yapacaklarımız kısmına.

Bunlara ek olarak yazılım mühendisliği biliminin güncel konularını da inceledik. Extreme Programming[6] gibi güncel konuları yazılımımıza ekleyip ekleyemeyeceğimizi tartıştık.

Sonunda Project Teamwork'ün amacının yazılım projelerinin planlanabilmesi ve projenin yapım aşamasında en kolay bir biçimde gözlemlenebilmesi ve takibi için en uygun ortamı yaratmak olması gerektiğine karar verdik. Bunun için de aşağıdaki amaçların gerçekleşmesi gerektiğine inandık:

- Proje özelliklerinin kolay ve sade bir ekranda girilerek projenin ilk oluşturulması aşamasında kullanıcıya kolaylık sağlanması
- Görevlerin gantt chart da gösterilmesi.
- Görevler üzerinde her çeşit (ki bunlar toplam 4 çeşittir. Start to Start, Start to Finish, Finish to Start, Finish to Finish) bağlantının kurulabilmesi ve görevlerin buna göre dinamik olarak tarihlendirilmesi
- Görevlerin ne kadarının tamamlandığının görsel bir ekranda görülebilmesi.
- Projeye kaynak ve eleman eklenebilmesi. Eklenen kaynak ve elemanların görevlere atanabilmesi. Atamaların uygunluğunun kontrol edilebilmesi.
- Critical Path'in görevler üzerinde görülebilmesi.
- Proje elemanlarının projenin izin verilen kısımlarını görebilmeleri.

Projemizi Visual Studio.NET 2005 de C# dilini kullanarak yapmaya karar vermiştik. Bunun yanında hangi teknolojileri kullanabileceğimizi bitirme hocamız sayın Feza Buzluca'yla da tartıştık. Aslında burada önümüzde 3 seçenek vardı.

Birincisi hiç bir veritabanı yönetim sistemi kullanmamaktı. Eğer projeyi böyle yapmaya karar verseydik veritabanına kaydedilecek bütün verileri yani projenin bütün özelliklerini programımızda bir sınıfta tutacaktık. Kayıt sırasında da bu sınıfı XML serialize teknolojisiyle serialize edip kaydedecektik. Geri yükleme esnasında ise bu serialize edilmiş text i alarak deserialize ederek sınıfımıza aktaracaktık.

Böylelikle proje hiçbir veritabanını kullanmıyor olacaktı ve bunun masrafından kaçınılmış olacaktı. Yani eğer bu sistem kullanılmazsa kullanıcı Access veya MS SQL programlarına da sahip olmak zorunda olacaktı ve bu da ek masraf demektir.

Buna ek olarak yazılımda çok fazla proje tutulmuyor olacağı için hızlı veritabanı işlemlerine de gerek yoktu. Bu nedenle de bu seçenek cazip görünüyordu.

Fakat eğer bu seçenek seçilmiş olsaydı proje elemanlarının projeye giriş yapması yani login olayını düzgün bir biçimde gerçekleştiremeyeceğimizi düşündük. Bu nedenle en azından Access gibi bir veritabanı kullanmamız gerektiğinde karar kıldık.

İkinci seçenek Access veya bu tarzda ucuz veya ücretsiz bir veritabanı kullanmak idi. Bu sayede hem veritabanı işlemlerimiz daha kolay olacaktı hem de çoklu kullanıcıya izin verilebilecekti. Fakat burada da peojeyi internette kullanma isteğimiz biraz ağır bastı. Yani bizim istediğimiz eğer vaktimiz olursa projeye bir web arayüzü de eklemek olacaktı. Böyle bir durumda da Access çok güvenliksiz bir yapı sunuyordu. Çünkü access veritabanları eğer url doğru yazılırsa internetten indirilebilir ve biz de bu riski almak istemedik.

Üçüncü ve son seçenek de MS SQL 2005 veritabanını kullanmaktı. Böylelikle web arayüzünün de daha güvenli olması sağlanacaktı. MS SQL 2005 (Yukon) kullanmak hem veritabanını çok da işlevsel yapacaktı hem de bize en üst düzey bir veritabanını kullanmanın tecrübesini kazandıracaktı. Yukon'un da dezavantajları yok değildi. Programı masaüstünde kullanmak üzere yüklemek isteyen her kullanıcının MS SQL'e sahip olması gerekliliği projenin masrafını arttırıyordu. Buna ek olarak MS SQL hiç kullanmamış olduğumuz için ne gibi zorluklarla karşılaşacağımızı bilmiyorduk. Ama gene de MS SQL 2005 önümüzdeki en mantıklı seçenek olarak duruyordu.

Bunun kararını da verdikten sonra projemize hangi arayüzleri yapacağımıza karar vermemiz gerekiyordu. Bilindiği gibi günümüz yazılımlarında Desktop arayüzünden başka web ve mobil arayüzleri de bulunabilmektedir. Bitirme hocamız sayın Feza Buzluca bize bu konuda çok yardımcı oldu. Öncelikle desktop arayüzünü bitirmemiz gerektiğini, eğer vakit kalırsa bunlarla ilgilenmemiz gerektiğini söyledi. Microsoft'un bu alandaki yeni özelliği click-once[7] incelendi. Aslında bu teknoloji javanın yıllardır var olan applet teknolojisine benziyordu. Bu arayüzleri zamanı geldiğinde kolaylıkla yapabilmek için yazılımımızı buna uygun geliştirmeye karar verdik. Bunun için bağlantı sınıflarımızı gerektiğinde webden de bağlantı kurabilecek şekilde webservice teknolojisine uyumlu olarak yazmaya karar verdik.

Bu rapor yazılırken projenin geniş tanımı ve işe başlamadan önce yapılan planı 2. kısımda verildi. Proje yapılırken kullandığımız teorik bilgileri 3. kısımda anlatmaya çalıştık. Bunlar hem programlama dilinin özelliklerini hem de genel windows programlama tekniklerini içermektedir. 4. kısım 2. kısımdaki planın biraz daha derinlemesine inilmiş hali oldu. Bunun yanında problemin bir analizi de 4. bölümde yer aldı. 5. bölümde yapılan programın tasarımı ve gerçekleşmesinin nasıl yapıldığı anlatıldı. Buna ek olarak 5. bölüm yazdığımız kodları test etmek için yaptıklarımızı da içerir. 6. bölüm programı yaparken elde ettiğimiz çıkarımları içeriyor. 7. bölümde ise program bittikten sonra yaptığımız çıkarımları, programı nasıl geliştirebileceğimizi ve bundan sonra benzer çalışmalar yürütecek arkadaşlarımıza önerilerimizi içeriyor.