Selsoft
İş ve İhtiyaç Analisti Yetiştirme/Geliştirme Programı

Akin Kaldiroglu

akin@selsoft.com.tr

Temmuz, 2014

Gündem

- Ön tanımlar:
  - İş analizi ve iş analisti
  - İhtiyaç analizi ve ihtiyaç analisti
- Analiz, Tasarım, Gerçekleştirme ve Test
- Gerçekler ve Belirtiler!
- Sonuç ve Çözüm
- Program Detayları ve Kazandırdıkları

(Sıkıntı,) çözümü görememeleri değildir.
(Sıkıntı,) problemi görememeleridir.

It isn't that they can't see the solution.
It is that they can't see the problem.

G. K. Chesterton,
The Point of a Pin in The Scandal of Father Brown

Kavramlar ve Roller: Kargaşa

- İş Analizi ve İş Analisti (Business Analysis and Analyst)
- İhtiyaç (Gereksinim) Analizi - İhtiyaç Analisti (Requirements Analysis and Analyst)
- Sistem Analizi - Sistem Analisti (System Analysis and Analyst)
- Analist-Programcı (Analyst Programmer)
- İsterler - İhtiyaçlar - Gereksinimler
- Analiz - Tasarım (Design) - Gerçekleştirme (Implementation) - Test

Meslek Standartı

- 5 Kasım 2013'de Resmi Gazete'de yayınlanan meslek standartına göre
- *BT İş Analisti (Seviye 6), İSG, çevre koruma, kalite kural ve yöntemleri çerçevesinde; yetkisi dâhilinde ve tanımlanmış görev talimatlarına göre; yazılım geliştirme projesinin ön hazırlığını yapan, projeyi tasarlayan, geliştirme çalışmalarını koordine eden, yazılımın uygulama ortamına uygunluğunu ve çalışırlığını test eden, yazılım dokümantasyonunu tamamlayan, yazılımın uygulamaya alma hazırlıklarını yapan ve bu çalışmaları koordine eden, yazılım iyileştirme çalışmaları yapan, mesleki gelişim faaliyetlerini takip eden nitelikli meslek elemanıdır.*
- *İşlemlerin yapılmasında iş talimatlarına uygun çalışır ve sorumluluk alanı dışında kalan arızaları ve hataları ilgili kişilere bildirir.*

İş Analizi Nedir?

- Bir işin süreçlerinin sistemli olarak çözümlenmesine iş analizi (business analysis) denir.
- İş analizi, işin nasıl yapıldığına odaklanır ve işin
  - Hedeflerini ve stratejilerini
  - Süreçlerini ve süreçlerdeki normal, alternatif ve sıra dışı akışları ve durumları
  - Süreçlerdek yapılan her türlü aktiviteyi,
  - Bu akışları yönlendiren iş kurallarını (hesaplama, kısıtlama, tetikleme ve şart ifadelerini),
  - Aktörleri ve dış etkileşimi: insan rolleri ve entegre olunan diğer sistemleri,
  - Ve kısıtlarını, ilgili standartlarını
- sistemli bir şekilde çözümler.

İş Analisti Nedir? I

- İş analizi yapan kişiye **iş analisti** (**business analyst**) denir.
- İş analisti, işi sadece bilmekle kalmaz, aynı zamanda işi analitik olarak **detaylandır**.
- İş analisti işin detaylarını
  - Bulur ve ortaya koyar (extraction),
  - Çözümler, bileşenlerine ayırır (analysis),
  - Sistemli olarak ifade eder, tanımlar (specification) ve
  - Formal olarak dökümante eder (documentation).
- İş analisti işin daha etkin olarak nasıl yapılacağı üzerine de kafa yorar ve iyileştirme önerilerinde bulunur.
- Hatta iş analisti yeni süreç ve iş kuralı tasarlar ve kullanıma alır.

İş Analisti Nedir? II

- Gittikçe karmaşıklaşan iş süreçlerini ve kurallarını bilgi yığını olarak ele almak, ölümcül bir hatadır.
- Çözümleyici olmak, işteki
  - Süreçler ve akışları ile kurallar ve aktörler arasındaki ilişkileri,
  - Sebep-sonuç, öncelik-sonralık bağımlılıklarını, karmaşıklık düzeylerini,
  - Sıra dışı durumları, envai çeşit “what if” senaryolarını,
  - Her türlü girdiyi, çıktıyı ve raporu,
  - Önemli bilgi öbeklerini (information set) ve ilgili kısıtları, sistemli bir şekilde bulup ortaya koymaktır.
- Bu anlamda iş analisti bir yazılım uzmanı değildir, işini yapmak için yazılımlardan faydalanan bir iş uzmanıdır!

İş Analisti ile Operasyonel Çalışan

- Operasyonel çalışanlar, o işin sadece kendine bakan taraflarını, **parçalı** bir şekilde, **senaryo** düzeyinde ve kendi bireysel yeteneklerinin izin verdiği ölçüde **gözlemlere dayanarak**, çoğu zaman **gelişi-güzel** ifade ederler, muhtemelen dokümante *edemezler.*
  - **Ufak tefek iyileştirme** önerilerinde bulunabilirler ama **yeni süreç** *kurgulayamazlar*.
- İş analisti ise işi **bütüncül** olarak, her türlü **detayıyla**, **sistemli** olarak ele alır, **çözümleyici** bir şekilde yaklaşır ve **formal** yöntemlerle **tarif** edip standart **doküman** olarak ifade eder.
  - Sistematik ölçümlerden faydalanarak iş süreçlerinde **iyileştirmeler** yapabildiği gibi **yeni süreçler** de kurgulayabilir.

İhtiyaç Analizi Nedir?

- Bir işin, bir yazılım sistemiyle otomatikleştirilebilecek şekilde detaylandırılmasına **yazılım ihtiyaçları analizi** (**software requirements analysis**) ya da kısaca **ihtiyaç analizi** denir.

- *Bir işin yazılım sistemiyle otomatikleştirilebilmesi için öncelikle o işin iş analizinin yapılmış olması gereklidir.*

- Genel olarak bir işin analizinin sorumluluğu o işi yapan kurumda, yazılım ihtiyaçlarının analizi ise yazılımı geliştirecek kurumda olmalıdır.
  - **İş bölümü** (**separation of concerns**)!

İhtiyaç Analisti Nedir? I

- Yazılım ihtiyaçlarını analiz eden kişiye de **ihtiyaç analisti** (**requirements analyst**) denir.

- İhtiyaç analisti, iş analizi yapılmış bir işin yazılıma konu olacak şekilde detaylarını
  - Bulur ve ortaya koyar (extraction),
  - Çözümler, bileşenlerine ayırır (analysis),
  - Sistemli olarak ifade eder, tanımlar (specification) ve
  - Formal olarak dökümante eder (documentation).

İş Analizi ve İhtiyaç Analizi

- İhtiyaç analizinin, iş analizinden farkı, işin bir yazılıma konu olabilecek şekilde ifade edilmesi, dolayısıyla iş analizinin yazılım ihtiyaçlarıyla donatılmasıdır:
  - Performans, ölçeklenirlik, güvenlik, kullanılabilirlik vb. mimari ihtiyaçlar,
  - Kullanıcı arayüzleri, insan-bilgisayar iletişimi (HCI) ihtiyaçları, girilen bilgiler ve kısıtları
  - Bilgi öbeklerinin detayları, tip, doğrulama ve kısıt bilgileri,
  - Artan kullanıcı çeşitliliğinin getirdiği kolay kullanım yetenekleri,
  - Entegrasyonlar, sistemin dışarıdan aldığı ve dışarıya verdiği bilgiler,
  - Sistem ihtiyaçları, gerekli donanım ve yazılım sistemleri,
  - Yazılım standartları, uyulması gerekenler şartlar vs.,
  - Yazılım kalitesi kriterleri, hata sayısı vb.
  - Projenin bütçe, zaman, çalışan, risk vb. faktörleri.

Sistem Analizi ve Analisti Nedir?

- Sistem analizi ise sistemin donanım ve yazılımını geliştirmeye kadar girmese bile, sistemin fonksiyonel çözümlemesi yanında fonksiyonel mimarisinin de tasarlanmasına verilen isimdir.
- Bu anlamda sistem analisti, veri tabanı tasarımı, fonksiyonel bileşenler gibi konularda tasarım yanında sistem testini hatta sistem kurulumunu (deployment) da yapar.
  - Bu sebeple sistem analisti yazılım geliştirme dilleri ve ortamlarına aşina olmalıdır.
  - Bu anlamda sistem analisti 80'li ve 90'li yılların pozisyonudur.
- Modern zamanlarda sistem analistleri, tasarım sorumluluklarını bırakarak daha fazla müşteri ihtiyacına odaklı analist olmaktadırlar.

Analist Programcı

- Analist programcı ise, programladığı işin analistliğini de yapan programcıdır.
- Geleneksel yazılım geliştirme kültürünün en önemli rolüdür.
- Hem müsterinin ihtiyaçlarını algılar, hem de algıyı yazılım olarak gerçekleştirir.
- Daha küçük, karmaşıklığı nispeten daha az iş mantığı ve/ veya yazılım evleri ve birimlerinde uygulanan yaklaşımdır.
- Karmaşık ve derin iş mantığını bütüncül olarak ele alamamak ve problemli müşteri ilişkileri en temel sıkıntısıdır.

Nasıl Bir Analist?

- Aslında aslolan, bir proje yaparken ya da ürün geliştiriken iş ve ürün bilgisinin tekrar kullanıma el verecek şekilde yakalanmasıdır.
  - İş birimlerinin formal iş bilgisine sahip olma dereceleri
  - Developerların legacy sistemlerle çalışmalarından dolayı sahip oldukları iş bilgisi ve tecrübeleri
  - Analistlerin o sektördeki hatta o proje ya da üründeki tecrübeleri
- Nasıl bir analiz sürecinin olması gerektiğini belirleyen temel etmenlerdendir.

Analiz, Tasarım, Gerçekleştirme ve Test I

- **Analiz**, olanı ve/veya olması gerekeni ortaya koymaktır.
  - Dolayısıyla analiz "**Ne**?" ("**What**?") sorusunun cevabıdır.
- **Tasarım** (**design**) ise olması gerekenin, nasıl gerçekleştirileceğinin analizidir.
  - Dolayısıyla tasarım "**Nasıl**?" ("**How**?") sorusunun cevabıdır, olması gerekenin yani çözümün soyut olarak ifade edilmesidir.
- **Gerçekleştirme** (**implementation**) ise tasarımda ortaya konan soyut çözümün yapılması, gerçekleştirilmesi, yerine getirilmesidir.
- **Test** ise gerçekleştirilen çözümün, analizine ve tasarımına olan bağlılığını ve muhtemel sapmalarını tespit etmektir.

Analiz, Tasarım, Gerçekleştirme ve Test II

- Bu anlamda analiz, diğer safhaları besleyen, yanlış veya eksik yapıldığında düzeltme maliyeti en yüksek olan safhadır.
- Diğer safhaların varlık sebebi, analizi yapılmış bir ihtiyacın olmasıdır.
- Dolayısıyla sağlıklı bir analiz yoksa, projenizde
  - Neyi tasarlayıp gerçekleştireceğiniz dolayısıyla da neyi test edeceğiniz,
  - Nereden başlayıp, nasıl ilerleyeceğiniz ve ne zaman bitireceğiniz,
  - Plan, bütçe, insan kaynakları vb. faktörler,
  - bilinemez.

Gerçekler I

- İş analizi ve ihtiyaç analizi ayırımı dolayısıyla da iş bölümü (separation of concerns) pratikte yoktur!
  - Her ikisini de ya BT bölümleri ya da yazılım evleri yapıyor.
  - Yani işi yapan birimin ya da kurumun, ne yaptığını tarif etmekten uzak olduğu bir durum çok yaygın olarak görülmektedir.
- Dolayısıyla da iş analisti ve ihtiyaç analisti ayırımı yerine, sadece analist var!
  - Hatta sistem analisti var.
  - Çoğu zaman ara eleman, joker oyuncu, development dışındaki şeyelrden sorumlu!

Gerçekler II

- Analiz ve tasarım iç içe girmiş durumda,
  - Veri tabanı ve kullanıcı arayüzü tasarımı ile tablo yapıları ya da sorgular, prosedürler, arayüz renkleri, butonların yeri gibi gerçekleştirmenin parçası olan detaylar analizin içinde ele alınıyor!
  - İşe yani işin süreçlerine ve kurallarına odaklanmak yerine, çoğu zaman ekran, sorgu vb. odaklı analizler söz konusu.

- Dolayısıyla da analist girmemesi gereken ve kendisinin de yetkin olmadığı tasarım, hatta geliştirme detaylarında boğulurken pek çok süreç ve iş kuralı detayını atlıyor!

Gerçekler III

- Süreç yerine “ekran” kavramı üzerinden konuşmak, hem işi anlamada hem de sağlıklı bir yazılım geliştirme süreci kurgulamada son derece sağlıksız bir durum oluşturuyor:
  - Bütünsel anlayış yerine parçalı anlayış, süreç yerine ekran kavramı,
  - İş ihtiyaçlarına odaklanmak yerine ekran ihtiyaçlarına odaklanmak,
  - Ekran odaklı proje planı, geliştirme ve test ile ilerlemek => Google arama ☺

- Ekrana odaklanmak, sadece son kullanıcının operasyonel iş yapışına odaklı ve parçalı dolayısıyla da tutarsız bir yapıyı, kaotik bir süreçle geliştirmemize sebep oluyor.

Gerçekler IV

- Analist gerçeği: Yaptığı işi tanımlamaktan uzak, analitik olmak yerine yaptığı *postacılığa* daha yakın görünen, ne müşteriye ne de proje ekibine ve özellikle de yazılım geliştiricilere yaranamayan ara eleman!
- Buna rağmen analistler, yazılım projesindeki roller açısından bakıldığında, iş tanımları ve iş yapış şekilleri en belirsiz ve analitik olmaktan en uzak çalışan kişilerdir.
- Halbuki **analistler, "işi bilmek" açısından bir yazılım projesi için en kritik roldedirler**.

Belirtiler I

- İş uzmanları tarafından "**aynı dili konuşamamak**" ile suçlanmak,
  - Terim ve nüanslarda zorlanmak,
  - Sorunları çözememek, karmaşıklığı basitleştirememek.
- İstekten ihtiyaca, ihtiyaçtan, asıl kök probleme gidememek,
- Süreçler ve iş kuralları arasındaki ilişkileri çıkaramamak, önceliklendirememek ve karmaşıklığını algılayamamak,
- İhtiyaç duyulan noktalarda müşteri için yeni süreçler ve iş kuralları teklif edip bunları kurgulayamamak

Belirtiler II

- Yazılımcılar tarafından “**işi bilmemekle**”, “**yeterince detay vermemekle**”, “**geliştirme sırasında çıkan sorulara cevap verememekle**” suçlanmak,
  - **Postacılık sendromu**; çünkü analist sadece aktarımda bulunuyordur, hiç bir analitik katkısı yoktur.
- Yazılımcıların işine yaramayan, okunmayan, boş, yavan ya da çok karışık, organizasyondan ve analitiklikten uzak hantal, aynı zamanda analiz, tasarım ve kodun birbirine geçtiği dokümanlar üretmek,

Belirtiler III

- Sonuçta yazılımcının insiyatifine kalmak, kullanıcının ihtiyaçları atlanırken, yazılımcının (teknolojik) ihtiyaçlarına ve yazılımcının, kullanıcının ihtiyacı olduğunu düşündüğü noktalara odaklanmak.

Analiz

- Bir yazılım projesinin en zor safhası “neyin yapılacağına karar vermektir”.
  - Bilinmeyen işin ne tasarımı, ne gerçekleştirmesi, ne testi ne de proje planı söz konusudur.
- Geliştirilecek yazılımın ihtiyaçlarının, kullanıcı arayüzleri ve entegre olunacak sistemlerle olan haberleşmesine kadar her türlü ihtiyacının detaylı ve kesin olarak belirlenmesi, teorik olarak mümün değildir.
- Yazılım projelerinin başarısız olmasının en temel sebebi, kullanıcı ihtiyaçlarının düzgün bir şekilde belirlenip yönetilmemesidir.
  - Örneğin Standish grubunun kaos raporları

Sonuç I

- Yanlış, eksik ve tutarsız yazılımlar,
- Unutulan önemli özelliklere karşın, fazladan eklenmiş önemsiz özellikler, yazılımcı işgüzarlığı ya da **ne vereyim abime?** sendromu ☺
- Atlanan analitik yaklaşım ve tasarım sonucunda tutarsız çözümler,
- Çok hatalı dolayısıyla da çok tekrarlı yazılım geliştirme süreci,
- Gerçekçi olamayan proje planları, durmadan genişleyen kapsam ve bitmeyen projeler.

Sonuç II

- Yazılım geliştiricilerin, iş birimleriyle doğrudan iletişimi dolayısıyla daha da berbatlaşan bir dil problemi!
- Sağlıklı işlemeyen test süreci, çünkü olması gereken detaylı olarak tarif edilmemiştir.
- Yönetilemeyen bakım süreci ve
- Ürünleşememiş, her müşteride ayrı ve farklı bir kopyası olan yazılımlar.
- **Azalan gelir ve kaybedilen BT saygınlığı!**

Çözüm?

- Selsoft'un **İş ve İhtiyaç Analisti Yetiştirme/Geliştirme Programı**'nın amacı, ülkemiz BT projelerinin ihtiyaçlarına cevap verecek şekilde hem iş, hem analitik hem de kişisel yetkinliklerle donatılmış iş ve ihtiyaç analistleri yetiştirmektir.
- Dolayısıyla program
  - Sağlıklı bir **teorik altyapı** yanında **yoğun uygulamalı** içeriğe sahiptir ve
  - Modern yaklaşımları bünyesinde barındırmaktadır.
- Program, sadece teknik yetkinlikleri değil, aynı zamanda sorgulama, iletişim gibi temel yetkinlikleri de geliştirmeyi hedef edinmiştir.

Analist Yetkinlikleri

- Analistin temel üç yetkinlik boyutu vardır:
  - İş bilgisi (business/domain knowledge)
  - Formal iş ve yazılım ihtiyaçları analizi bilgisi
  - Analitik düşünme, güçlü iletişim, sorgulayıcı olma gibi bireysel ve temel yetenekler
- Genelde algımız, ilk boyut üzerine yoğunlaşır, dolayısıyla iş süreçlerini bilmek yeterlidir diye düşünürüz,
- İkinci boyut ya toptan unutulur ya da yeterince vurgulanmaz,
- Üçüncü boyut ise daha çok müşteri bağlamında düşünülür; yazılım geliştirici, PM vb. ile iletişim problemlidir.

Program

- Bu program temelde, iş ve ihtiyaç analizinin nasıl yapılacağına odaklanmaktadır.

- Dolayısıyla program, analisti, gerekli **formal analiz ve kişisel yetkinliklerle** donatmak hedefine sahiptir.

- Temel kavramları ve teknikleri tamamen uygulamalı olarak analiste kazandırmak amacındadır.

İçerik I

- Program temelde şu başlıklardan oluşmaktadır:
  - Yazılım geliştirme süreci ve modern yaklaşımlar
  - İş ve yazılım ihtiyaçları kavramlarına giriş
  - Yazılım ihtiyaçları mühendisliği ve yönetimi
  - İhtiyaçları bulup çıkarmak, müşteri ve kullanıcı ilişkilerini yönetmek
  - Toplantı yönetimi ve sorgulama
  - İhtiyaçları analiz etme: süreç, kural, kullanıcı arayüzleri, kısıtlar vs.
    - Kullanım şekilleri (use-cases), senaryoları ve diyagramları ile kullanıcı hikayeleri (user stories)

İçerik II

- Yeni proje, gap analizi, bakım vb. farklı proje tipleri için farklı yaklaşımlar
- UML ile yazılı ve görsel modelleme
  - Use case, activity, class, state diagramları
- Prototyping
- Kullanıcı arayüz ihtiyaçları analizi ve tasarımı
- Gözden geçirme, doğrulama ile ihtiyaç analizi kalite kontrolü
- Formal olarak tanımlama ve belgeleme/dokümentasyon
- Karmaşıklık ve önceliklendirme
- Fonksiyonel olmayan ihtiyaçlar

İçerik III

- İhtiyaçları tasarlamaya ve kodlamaya geçiş
- Nesne-merkezli analiz (OOA): Nesneleri bulmak, detaylandırmak, ve modellemek, sınıf/nesne, activity diyagramları ile domain modelling
- Entityler ve ilişkileri: E-R diyagramları ile domain modelling
- Test caseleri ile teste geçiş
- Agile metodolojilerde analiz
- Ve **workshop**

Workshop I

- Katılımcılar gruplar halinde, derste öğrenilen teknikleri kullanarak hayali bir ATM sisteminin ihtiyaçlarını çıkarıp modellemek ve Yazılım İhtiyaç Belgesini (SRS)'ini oluşturacak şekilde bir analiz projesi gerçekleştirirler.

- Arzu edilirse müşteri kendi iş alanından, sınırları iyi tanımlanmış bir problemi ya da ihtiyacı da bu projede workshop malzemesi olarak kullanabilir.

Workshop II

- Workshop katılımcılara gerçek proje için yarı-gerçek bir uygulama alanı sunmaktadır. Dolayısıyla workshopta
  - Toplantı yönetimi ve toplantı çıktıları
  - Use-case tabanlı modelleme, iş kuralları, fonksiyonel olan ve olmayan ihtiyaçlar, sistem ihtiyaçları vs. bulunması
  - Nesne ve E-R diyagramları
  - Test senaryoları ile teste geçiş
  - Dokümentasyon
- ile tüm analiz süreci kontollü bir şekilde tecrübe edilecektir.

Değerlendirme Projesi

- Program sonunda katılımcılar, öğrendiklerini ölçmek amacıyla bir değerlendirme projesi almaktadırlar.
- Bu proje, verilen bir iş problemini, programda öğrenilen teknikler doğrultusunda analiz etmeyi içermektedir.
- Proje sınıfta alınabileceği gibi evde de yapılabilir.

Programın Çıktısı

- Program sonunda Selsoft, isteğinize göre size şu değerlendirmeleri sağlayabilecektir:
  - Her katılımcı için sınav sonucunu da içeren bireysel analist yetkinlikleri değerlendirmesi,
  - Kurumun analiz süreçleri ve ilgili dokümanlarıyla ilgili elde edilen bulgular ve değerlendirmeler,

Program, Analiste Ne Kazandıracak? I

- Program bir analiste aşağıdaki konularda yetişmesini sağlayacaktır:
  - **İstek => İhtiyaç => Kök problem** yaklaşımı ile gerçek probleme odaklanma,
  - **İhtiyaç kategorizasyonu**: Fonksiyonel olan ve olmayan ihtiyaçlar ayrımı. Ayrıca, use case merkezli kullanıcı süreçleri ve ihtiyaçları, iş kuralları, sistem ihtiyaçları, kalite ihtiyaçları, kısıtlar, vs. ayrımlarına tabi bir analiz süreci
  - **Yapısal toplantı ve workshop yönetimi**: Daha planlı ve yapısal bir toplantı ve workshop yönetimi ile daha verimli bir süreç.

Program, Analiste Ne Kazandıracak? II

- **Use case modeli**: Use case diyagramı ve use caselerin detaylı aktivite (activity) diyagramları ile UML tabanlı daha görsel ve matematiksel sunumla paydaşların özellikle de tasarım ve geliştirmeye rahat geçiş.

- **Nesne ve entity modelleri**: Nesne-merkezli ve database tasarımına geçiş için sınıf/nesne analizi ve diyagramları yanında karşılık gelen E-R modellemesi sayesinde nesne ve veri merkezli geliştirmeye rahat geçiş.

- **Standart ve yapısal dokümantasyon**: Farklı tipte ihtiyaçlar ve bunlara yönelik modellerle zenginleştirilmiş, ihtiyaçlarınıza uygun olarak standartlaştırılmış ve yapısal dokümantasyon.

Program, Kuruma Ne Kazandıracak? I

- Program bir kuruma aşağıdaki konularda katkıda bulunacaktır:
  - **Güçlü analistler**: İşini analitik olarak bilen ve özellikle de müşteri ve yazılım geliştiriciler karşısında ezilmeyen, güçlü iletişim ve dokümantasyon yeteneklerini haiz analistler.
  - **Matematiksel ve yapısal ihtiyaçlar**: Yapısal, matematiksel olarak modellenmiş farklı tipte paydaşlara farklı şekillerde hitap eden yazılım ihtiyaçları analizi.
  - **Doğru detayda ihtiyaçlar:** Doğru detayda iş ve ihtiyaç analizi
  - **Kurumsal iş bilgisi birikimi**: Kurumun iş bilgisinin tekrar kullanımını sağlayacak şekilde analitik formda biriktirilmesi ve paylaşılması.
  - **Daha az risk**: Yazılım ihtiyaçlarından kaynaklanan riskleri çok daha az ve kontrol edilebilir projeler ve ürünler.

Program, Kuruma Ne Kazandıracak? II

- Dolayısıyla bu program size
  - Daha az hatalı,
  - Daha çok detaylı,
  - Daha matematiksel ve kesin,
  - Daha yapısal ve görsel,
  - Daha rahat planlanabilir ve öngörülebilir,

- projeler ve ürünler geliştirebilmek için gerekli analiz alt yapısını verecektir.

Program Süresi

- Programın süresi 6 gündür.
  - 5 - 5,5 gün uygulamalı eğitim,
  - 0,5 - 1 gün workshop çalışmasıdır.
- Programı aynı anda 12 kişiden fazlasına uygulamak programın verimini düşürecektir.
- Program, kurumun ve analistlerinin ihtiyacına göre özelleştirilebilir.

IIBA CCBA ve CBAP Sertifikaları

- International Institute of Business Analysis'in Business Analysis Book of Knowledge (BABOK) içeriğine uygundur.
- Certification of Competency in Business Analysis (CCBA) ve Certified Business Analysis Professional (CBAP) sınavlarına hazırlanmak amacıyla alınabilir

Kullanılacak Kaynaklar

- Wiegers, Karl E. 2003. **Software Requirements**, 2d Ed. Washington: Microsoft Press
- Wiegers, Karl E. 2010. **More About Software Requirements**
- Fowler, Martin 2003. **UML Distilled**, 3rd Ed., Addison Wesley
- Leffingwell, Dean, and Don Widrig 2003. **Managing Software Requirements**: A Use Case Approach, 2nd Ed. Reading, Mass.: Addison Wesley
- Gottesdiener, Ellen. 2002. **Requirements by Collaboration: Workshops for Defining Needs**. Boston: Addison-Wesley
- BABOK Guide, IIBA

Teşekkür ederim.