

Model Geliřtirmede Deęerlendirilmesi Gereken Unsurlar

Özlem Demirci

Jönköping University Information Engineering, Jönköping/İsveç

demirci.ozlem@gmail.com

Özetçe

Yazılım sektöründe, modele dayalı yazılım geliştirme uygulamalarının kullanımı gün geçtikçe artmaktadır. Artık yazılım geliştirme sürecinde modeller ana yapı olarak kullanılmakta ve diğer yapı taşları bu modellerden elde edilmektedir. Fakat bu uygulama sektöründe yeni olduğundan ötürü organizasyonlar bu uygulamaları kullanırlarken çeşitli zorluklar ve problemler yaşamaktadırlar. Bu durum böyle bir araştırmanın yapılması gerektiği yönünde güdüleyici olmuştur. Dolayısıyla bu yazıda, modele dayalı yazılım geliştirme uygulamaları yapılırken hangi etkenlerin, unsurların dikkate alınması gerektiği araştırılmıştır.

Abstract

The modeling approaches used in the software industry has widespread for over the years. The models have become the main artifacts in the development process and the rest of the artifacts have derived from those models.

However, applying these new approaches is not easy as it sounds because it is a new concept in the industry. There exist many and variety of problems and difficulties while applying these modeling approaches within the organization. Therefore these issues motivated to make this research regarding finding the important aspects needed be considered.

1. Giriş

Modelleme uygulamaları yazılım endüstrisinde geniş çapta kullanılmasına rağmen, halen modellemenin nasıl uygulanması ve başarılı, kaliteli bir uygulama süreci yönetilmesi ile ilgili arařtırmalar devam etmektedir. Bu durum da dikkate alınması gereken unsurların neler olduğu yönünde araştırma yapılması yönünde güdüleyici olmuştur.

Son zamanlarda, yazılım sektöründeki küçük, orta veya büyük ölçekli organizasyonlar modele dayalı yazılım geliştirme uygulamalarını yazılım geliştirme süreçlerinde uygulamaya başlamıştır. P. Mohagheghi , V. Dehlen, and T. Neple

“Definitions and approaches to model quality in model-based software development [1]” isimli makalelerinde, modelleme uygulamalarının endüstride ilk olarak 3. Kişiler ile iletişimi sağlamak amacıyla basit taslaklar geliştirerek uygulanmaya başladığını belirtmişlerdir. Daha sonrasında ise modelleme uygulamaları birçok alanda olduğu gibi birçok farkı amaçla uygulanmaya başlamıştır. Örneğin, günümüzde geliştirilen modellerle otomatik olarak kod geliştirmek bir amaç haline gelmiştir. Dolayısıyla, organizasyonların bu uygulamaları kullanmadaki amaçlarına göre yapmaları ve öngörmeleri gereken aşamalar değişmektedir.

Modele dayalı yazılım geliştirme ile ilgili araştırmalar ve endüstriyel faaliyetler incelendiğinde UML (Unified Modeling Language)’in en yaygın olarak kullanılan modelleme dili olduğu görülmektedir. Bu nedenle de bu araştırmada modele dayalı yazılım geliştirme ve UML birlikte ele alınmıştır.

2. Araştırma Metodu ve Önceki Araştırmalar

Bu konudaki araştırma, yayınlanmış makalelerin ve deneyim yazılarının incelenmesinden elde edilmiştir. Böylelikle hem araştırmacıların bakış açıları hem de gerçek hayattaki uygulamalardan elde edilen gözlemler ele alınmıştır. Özellikle durum çalışmalarının incelenmesi ile ne tür unsurların modelleme sürecinde etkin olduğu elde edilebilmektedir.

Dolayısıyla bu araştırma niteliksel bir araştırma olup, veri elde edimi deneyim yazıları, yayınlanmış makaleler ve yapılmış durum çalışmaları gibi ikincil kaynaklardan sağlanmıştır. Ayrıca kaynak olarak aşağıda belirtilen yayın kanalları araştırılmıştır;

- ACM Digital Portal

- Jönköping University Online Library (JULIA ve DIVA)
- SpringerLink Online Veritabanı

Bu kaynaklar aracılığı ile yapılan araştırmada, yapılan araştırmaların çoğunlukla model kalitesi ile ilgili olduğu elde edilmiştir. Model kalitesinin nelere bağlı olduğu, nasıl kaliteli modeller elde edilebileceği ve modellerin kalite ölçümü ilgili yeterli bilgi edinilmiştir. Örneğin, [1] ve [2] makalelerde, yazarlar kaliteli modelleri açıkladıkları gibi ne tür unsurların kaliteli model geliştirmek için gerekli olduklarına değinmişlerdir. Ayrıca [1] makalenin yazarları, 6 farklı kalite amacı belirleyip bunların model geliştirme de ele alınıp elde edilmesini öngörmüşlerdir.

Bu makalede UML ana modelleme dili olarak ele alındığından ötürü, UML’ in model geliştirmede ki etkileri de incelenmiştir. Örneğin, David Harel and Bernhard Rumpe bir makalelerinde [3], UML’ in sözdizimsel ve semantik yapısının modellerdeki önemine değinmişlerdir çünkü UML’ in semantik yapısında halen eksikler bulunmaktadır ve bu da model kalitesini ciddi anlamada etkilemektedir. Ayrıca bir model farklı UML diyagramlarından oluştuğundan ötürü, diyagramların dizaynının ve düzenlemesinin de önemli bir unsur olduğu yapılan araştırmalara göre söylenebilir.

3. Modele Dayalı Yazılım Geliştirme ve UML

Modele dayalı yazılım geliştirmeyi açıklamadan önce modelin ne olduğunu anlamak gerekir. Modeller bir sistemi temsil eden öğelerdir. Modeller bir sistemi diyagramlar yardımıyla görsel olarak ve sistem detaylarına girmeyerek gösterirler. Bir sistemin çok özel bir unsuru, görevi ile ilgili bilgi model aracılığı ile gösterilir.

Modele dayalı yazılım geliştirme sürecinde sistem modeller aracılığı ile soyut bir şekilde açıklanır. Bu kapsamda modele dayalı yazılım geliştirme geleneksel yazılım geliştirmeden farklılık gösterir çünkü burada modeller ana yapı taşı olarak kullanılır. Yapılan literatür taramasına dayanarak modele dayalı yazılım geliştirmenin birçok makalede tanımlandığı gibi soyut bir seviyede sistemi açıklamaktan daha çok sistemle ilgili detaylı bilgiyi de modellemek olduğu söylenebilir. Böylelikle geliştirilen modeller aracılığı ile otomatik olarak kod elde etme hedeflenmeye başlamıştır.

Önceden de bahsedildiği üzere, modele dayalı yazılım geliştirmede modeller ana yapı taşlarıdır ve modeller bir modelleme dili aracılığı ile yaratılır. UML bir modelleme dili olarak, gerekli işaret ve sembollerle gösterim için kurallar sağlar. Örneğin; kalıtım gösterimi ortak bir sembol ile gösterilir. Böylelikle farklı kişiler tarafından modeller arasında aynı tanımlamalar, aynı gösterimler elde edilebilir. Bu da yaratılan modellerin istikrarlı ve herkes tarafından aynı şekilde anlaşılabilirliğini sağlamaktadır.

Bir model çeşitli diyagramlardan oluşabilir. Örneğin, UML 2.0 ile 13 farklı diyagram çeşidi geliştirilmiştir ve her diyagramın kullanım amacı farklıdır. Buna bağlı olarak da bir sistemin bir spesifik özelliği farklı şekillerde temsil edilebilir. Örneğin, UML sınıf diyagramlar aracılığı ile sınıflar özellikleri ve diğer sınıflar ile olan ilişkileri ile tanımlanır. Bunun yanı sıra farklı objelerin birbiri arasındaki ilişkiyi ve iletişimi göstermek için ise UML sıra diyagramları kullanılır. UML diyagramları ve onların amaçlarını kavramak ve bilmek önemlidir çünkü bu çeşitli diyagramlar birbirini bir şekilde tamamlar ve daha net ve uyumlu bir model elde edilmesini sağlar. Russell Miles ve Kim Hamilton [4] "Learning UML 2.0" adlı kitaplarında Philippe Kruchten' nın "4+1

view model" i ile UML 2.0 diyagramlarını ve amaçlarını aşağıdaki şekilde görüldüğü gibi açıklamıştır.

Tablo 1. Kruchten'nın 4+1 view model

Görüş Şekli	Tanım	UML Diyagram
Mantıksal Görüş (Logical View)	Sistemin parçaları ve birbiri ile olan ilişkileri tanımlanır.	Sınıf (Class), Obje (Object), Sonlu Otomat (State Machine), ve Etkileşim (Interaction)Diyagramları
Süreç Görüşü (Process View)	Sistemin işleyiş sürecinin basamakları açıklanır.	Aktivite (Activity)Diyagramlar
Geliştirme Görüşü (Development View)	Sistemin modülleri ve bileşenleri tanımlanır.	Paket (Package) ve Bileşen (Component) Diyagramları
Fiziksel Görüş (Physical View)	Modelin son hali bir önceki 3 diyagramda belirtilen bilgilere göre tanımlanır.	Konuşlanma (Deployment) Diyagram
Kullanım Durumu Görüşü (Use Case View)	Sistemin işleyiş dışarıdan bir görüşe göre açıklanır.	Kullanım Durumu (Use Case), Betimleme (Description), ve Gözden Geçirme (Overview) Diyagramları

Modellemenin çeşitli amaçlar için kullanıldığından çok az da olsa bahsetmişim. Şimdi detaylı olarak bu konuyu ele aldığımızda görüyoruz ki modeller farklı alanlar için, farklı amaçlar ve farklı şekillerde kullanılmaktadır. Russell Miles ve Kim Hamilton "Learning UML 2.0" [4] adlı kitaplarında Martin Fowler'ın UML kullanımı ile ilgili olan 3 aşamasını incelemiştir. Bu aşamalar;

Taslak olarak (Sketch) UML: Organizasyon modellemeyi en basit şekilde taslak olarak yapmayı hedefler. Böylelikle sistem ile detaylı bilgi örneğin sistemin teknik özellikleri belirtilmez. Modeller sadece iletişimi temsil etmek için kullanılır. Bu nedenle kullanıcılar tarafından anlaşılabilir olması gerekir.

Detaylı tasarım olarak (Blueprint) UML: Organizasyon daha detaylı bir şekilde sistemi model ile açıklar. Böylelikle modele dayanarak kod elde edilir. Bu yüzden model tamamlanmış, sistem ve kod detayı içeren, tutarlı ve somut olması gerekir.

Programlama Dili olarak (Programming Language)UML: Geliştirilen modeller aynı zamanda otomatik olarak koda çevrilebilmelidir. Bu otomatik kod elde etme işlemi için otomatik kod üretme araçlarının kullanılması gerekir.

Fowler'ın bu tanımlamaları göz önüne alınarak, her farklı kullanım şekli için farklı unsurların göz önünde bulundurulması gerektiği belirtilebilir. Böylelikle organizasyonlar ne tür bir amaca göre ne gibi aktivite ve görev yapmaları gerektiğine karar verip, buna bağlı olarak model geliştirebilirler.

4. Model Kalite Hedefleri

Literatür taramasına dayanarak, model kalitesinin modele dayalı yazılım geliştirmede önemli olduğuna dair bir sonuç elde edilebilir. Bu konuda yapılan birçok araştırma yer almaktadır. Özellikle P. Mohagheghi, V. Dehlen, and T. Neple "Definitions and approaches to model quality in model-based software development [1]" isimli makalelerinde bu konu ile ilgili yayınlanmış birçok araştırmaları inceleyerek model kalitesi hedeflerini 6 ana başlık altında açıklamışlardır. Bu 6 ana model kalite hedefleri aşağıda ki tablo da belirtilmiştir.

Tablo 2. 6C Model Kalite Hedefleri [1]

Model Kalite Hedefleri	Tanım
Doğruluk (Correctness)	Bir model, doğru faktörleri ve birbirleri arasında ki ilişkileri doğru bir şekilde açıklamalıdır. Ayrıca sistem ve alan ile ilgili doğru bilgiler yer almalıdır.
Eksiksizlik, Tamamlık (Completeness)	Bir model, gerekli olan bütün bilgiyi yeterince detay vererek açıklamalıdır.
Tutarlılık (Consistency)	Model de herhangi bir çelişki yer almamalıdır.
Anlaşılabilirlik (Comprehensibility)	Gerek kullanılan araçlar gerekse hedeflenen model kullanıcıları geliştirilen modeli anlayabilmelidir.
Kısıtlama (Confinement)	Burada kısıtlama ile kasıt, geliştirilen model hedeflenen modele hedeflenen soyut seviyesinde bağlı kalmalıdır. İlgili bilgiler hedeflenen soyut seviyesinde açıklanmalıdır.
Değişebilirlik (Changeability)	Geliştirilen modeller herhangi bir değişiklik ihtiyacı söz konusunda kolayca değiştirebilmeyi, güncellenmeyi desteklemelidir. Böylelikle sürekli bir model gelişimi elde edilebilir.

Model kalite hedeflerinin haricinde, yapılan literatür taramasından elde edilen verilere göre, UML' in karmaşık yapısı ve semantik (semantic) yapısında ki eksik tanımlamalar ve kurallardan

ötürü, model kalitesinde bir düşüş olduğu öngörülmektedir. Örneğin, [1], [5], [6], [7], [3], [8] makalelerde de bahsedildiği gibi UML' in sözdizimsel ve semantik yapısındaki eksiklikler geliştirilen modellerin kalitesinde büyük etken oluşturmaktadır. Ayrıca bu makalelerde UML' in sözdizimsel ve semantik yapısının haricinde çok çeşitli diyagramlı yapısının da geliştirilen modellerde istikrarı ve anlaşılabilirliği olumsuz yönde etkilediği belirtilmiştir.

Bütün bunlar göze alındığında hem model kalitesinin hem de UML' in yapısının geliştirilen modellerde etkili olduğu ve organizasyonların bu yeni modelleme uygulamalarını uyarlarken ve uygularken bu etkenleri göz önünde bulundurmaları gerektiği yargısı edinilebilir.

5. Model Kalitesini Değerlendirme Unsurları

Bölüm 4 de belirtilen model kalite hedeflerine ulaşmak ve UML' in eksik yapısal problemlerini çözümlemek için, organizasyonların modelleme yaklaşımlarını uygulamaları süreci sırasında göz önünde bulundurmaları gereken unsurlar yer almaktadır.

Modelleme sürecinde, değerlendirilmesi gereken bu unsurlar, ileride daha başarılı ve gelişmiş modelleme uygulamalarını yapmada etkili rol alırlar. Örneğin, CMM (Capability Maturity Model) ele alındığında, geleneksel yazılım geliştirme sürecinin ne kadar olgun bir şekilde gerçekleştirildiği yönünde organizasyonlara fikir verir. Bu makalede incelenen unsurlar da olası bir olgunluk, performans ve kalite değerlendirilmesi yapılması gerektiğinde ele alınması öngörülen noktalardır.

Modele dayalı yazılım geliştirme yeni bir kavram ve metot olduğundan ötürü henüz bu modelleme süreci ile ilgili tanımlanmış, standart

haline gelmiş bir süreç izleme prensibi yer almamaktadır. Ayrıca geleneksel yazılım geliştirme sürecinden farklı bir süreç olduğundan ötürü, kesinlikle bu sürecin tanımlanması gerekmektedir.

Ayrıca [9] makalede belirtildiği üzere, modelleme süreç geliştirme ve uygulamada ki bu tanım ve standart eksikliği haricinde diyagramlarda ki dizayn eksikliği ve kalite kontrol faaliyetlerinin tanımlanmamış olması organizasyonun modelleme amaçlarına erişimi başarısını düşürmektedir.

Parastoo Mohagheghi and Vegard Dehlen "Developing a Quality Framework for Model-Driven Engineering" [2] makalelerinde model geliştiricilerinin (developers) bilgilerinin ve tecrübelerinin, kullanılan modelleme araçlarının ve kullanılan modelleme dilinin de model geliştirmede önemli faktörler olduğunu belirtmişlerdir. Eğer geliştirici kullanılan modelleme diline ve araçlarına ilişkin eksik bilgiye ve tecrübeye sahip ise üretilen modellerde büyük eksiklikler görülebilmektedir.

Eğer modelleme süreci iyi ve açık tanımlanmamış ise, özellikle de büyük ölçekli sistemleri modellemeye çalışırken düzensizlik ve takipsizlik kaçınılmazdır. Bunu takiben, diyagramların dizaynı eksiksiz, istikrarlı ve anlaşılabilir olmalıdır ki bu süreçte yer alan tüm kullanıcılar ve geliştiriciler doğru bilgiyi doğru şekilde aktarabilmeli ve anlayabilmelidir.

Bölüm 4'de belirtilen hedefler de göz önüne alındığında 3 numaralı tabloda belirtilen unsurların organizasyon tarafından değerlendirilebilmesi gerekmektedir. Böylelikle amaca uygun, tam, eksiksiz ve istikrarlı model geliştirme işlemi uygulanabilir ve

organizasyonlar neyi nasıl yaptıklarını değerlendirebilirler.

Tablo 3. Modelleme Sürecinde Değerlendirilmesi Gereken Unsurlar

Unsurlar	Tanım
Modelleme Süreci (Modeling Process)	Organizasyonun amaç ve hedefleri doğrultusunda yazılım geliştirme süreci eksiz ve bütün katılımcıları göz önünde bulundurarak tanımlanmalıdır. Genellikle bu süreç planlama, gereksinim tanımlamaları gibi aşamalardan oluşur.
Kalite Güvence Teknikleri (Quality Assurance Techniques)	Geliştirilen yazılımın ve sürecin kalite standartlarına uygunluğunu değerlendirmek için yapılan aktivitelerdir. Böylelikle projenin organizasyonun hedeflerini ve beklentilerini karşılayıp karşılamadığı ölçülebilir.
Eğitim (Training)	Katılımcılar gerek görüldüğü ve ihtiyaç duyulduğunda gerek yeni araçlar gerekse yeni modelleme dili ile ilgili eğitim almalıdır. Ayrıca katılımcıların motivasyonları, tecrübeleri, bilgi birikimleri başarılı bir proje geliştirme süreci için önemli yer taşır.
Araçlar (Tools)	Kullanılacak araçlar organizasyonun amaçlarına uyumlu olmalı ve de kullanılan alanın gerektiği

koşulları sağlayabilmelidir. Örneğin, savunma alanında bir modelleme gerçekleştiriliyorsa kullanılan araçların sertifikalı olması ve güvenli olması gerekmektedir.

Dizayn (Design)
Teknik bilgiye sahip olan katılımcılar olduğu gibi bu bilgiye sahip olmayan katılımcılarda yer aldığı için diyagramların dizaynı anlaşılabilir, belli standartlarda ve istikrarlı olmalıdır. Örneğin, birbirini kesen çizgiler anlaşılabilirliği olumsuz yönde etkiler.

Sözdizimsel ve Semantik (Syntax and Semantic)
UML' in bu yapıdaki eksikliklerinden ötürü, organizasyon sözdizimsel ve semantiği belli modelleme konvansiyonlarına göre tanımlanmalıdır. sözdizimsel bir dili kullanma ilgili kurallar dizisidir; semantik ise o dilin ne anlam ifade ettiğidir. Dolayısıyla bu iki kavramın tam ve net şekilde tanımlanmış olması gerekir. Böylelikle farklı kişiler tarafından geliştirilen model veya diyagramlarda bir bütünlük sağlanabilir.

Dönüşüm (Transformation)
Dönüşüm ile burada ki kasıt, geliştirilen diyagramların otomatik olarak koda dönüştürülebilme

becerisidir. Eđer organizasyonun modelleme amacı bu ise geliştirilen modeller bu gerekliliđi karşılamalıdır.

Yukarıdaki tabloda belirtilen unsurlar, tüm modelleme süreci sırasında ilgili kişiler tarafından geliştirilmesi ve değerlendirilmesi öngörülebilir. Böylelikle modelleme yapmadaki amaca ulaşıp ulaşılmadığı ya da ne kadar ulaşıldığı ölçülebilir.

6. Sonuç

Sonuç olarak modele dayalı yazılım geliştirme gün geçtikçe daha da popüler olmaya başlamış ve birçok organizasyon tarafından kullanılmaya başlamıştır. Modelleme farklı alanlar için farklı amaçlarla uygulanmaktadır. Bu doğrultuda geliştirilen modellerden beklentiler de farklılık gösterilir. Ayrıca yeni bir kavram olduğundan ötürü halen bu alanda halen tartışmalar ve araştırmalar sürmektedir. Standart eksiklikleri organizasyonların modellemeyi uygularken nelere dikkat etmesi gerektiğini, neyi ne ölçüde yaptığını ölçmesi gerektiğini engellemektedir.

Bunların yanı sıra modelleme ile modelleme dili de bir bütün olduğundan ve UML' in sözdizimsel ve semantik yapısında ki eksikliklerden ötürü modelleme sürecinde ve geliştirilen modellerde aksaklıklar ve kusurlar oluşması neredeyse kaçınılmazdır.

Bütün bunlar göz önüne alınarak 7 farklı unsurun bu süreçte göz önünde bulundurulup, ilgili aktivite ve görevlerin yapılmasını öngördüm. Böylelikle organizasyonlar bu yeni uygulamada neyi nasıl yaptıklarını değerlendirebilecekleri gibi ne tür görevlerin

yerine getirilmesi gerektiğinin bilincinde olabileceklerdir.

REFERANSLAR

[1] Definitions and approaches to model quality in model-based software development– A review of literature, Parastoo Mohagheghi , Vegard Dehlen, Tor Neple, Elsevier B.V., April 2009

[2] Developing a Quality Framework for Model-Driven Engineering, Parastoo Mohagheghi and Vegard Dehlen, Springer-Verlag Berlin Heidelberg 2008

[3] Modeling Languages Syntax, Semantics and All That Stuff, David Harel and Bernhard Rumpe, yayın yeri CiteSeer, 2000

[4] Learning UML 2.0., Russell Miles, Kim Hamilton, O'Reilly Media, 2006

[5] Software Maintenance Maturity Model (SMmm): The software maintenance process model, Alain April, Jane Huffman Hayes, Alain Abran, and Reiner Dumke, John Wiley & Sons, Ltd, 2004

[6] An Experimental Investigation of UML Modeling Conventions, Christian F.J. Lange, Bart DuBois, Michel R.V. Chaudron, and Serge Demeyer, Springer-Verlag Berlin Heidelberg, 2006

[7] Improving the Quality of UML Models in Practice, Christian F.J. Lange, International Conference on Software Engineering Proceedings of the 28th international conference on Software engineering and yayın yeri ACM, 2006

[8] Meta Modeling Semantics of UML, Andy Evans, Robert France, Kevin Lano, Bernhard Rumpe, yayın yeri CiteSeer, 1999

[9] Defects in Industrial UML Models – A Multiple Case Study –Christian F.J. Lange, Michel R.V. Chaudron, the ACM/IEEE 10th International Conference on Model Driven Engineering Languages and Systems, 2007