

Essence Süreç Modelleme Çerçevesi için Otomasyon Yazılımı Geliştirme Deneyimi

Yagup Macit¹, Çağdaş Üsfekes¹, Ahsen İkbal Aytekin¹, Eray Tüzün²
¹HAVELSAN Bilişim Hizmetleri Müdürlüğü
²HAVELSAN Teknoloji ve Akademi Direktörlüğü
06510 Ankara, Türkiye

Özet. Yazılım ve sistem geliştirme çalışmalarını desteklemek için tanımlanmış olan çok sayıda pratik, yöntem ve süreç bulunmaktadır. Bu geliştirme yönergeleri, genellikle ilerlemenin doküman üzerinden izlenmesi temeline dayanmaktadır. Ancak, başka endüstriler için otomasyon çözümleri üreten bilişim dünyasının kendi üretimlerini doküman ile yürütmesi beklenemez. Her bir geliştirme yönergelerinin otomasyona dönüştürme çalışması kendine özgü zorluklar barındırmaktadır. Essence Çerçevesi, yazılım geliştirme süreç, yöntem ve pratiklerini genel olarak ortaklamayı amaçlayan bir süreç modelleme dili ve genişletilebilir bir çerçeve sağlamaktadır. Bu ortaklama yaklaşımı, kurumlara proje/ürün geliştirme ve izleme için ortak bir zemin vaat etmektedir. Ancak bu ortak zeminin, yürütme aşamasını kapsamamakta ve işleyiş el ile yürütülebilen kartlar üzerinden sağlanmaktadır. Bu çalışmada, Essence dil ve çerçevesi temel alınarak, işletilebilir modelinin oluşturulması ve bir otomasyon uygulamasının yazılmasına dönük deneyimler aktarılmıştır.

Anahtar Kelimeler: Yazılım Geliştirme Yaşam Döngüsü Yönetimi, Yazılım Süreçleri, Süreç Otomasyonu, Yazılım Proje Yönetimi, Essence Çerçevesi

Development Experience in Automation Software for Essence Process Modeling Framework

Yagup Macit¹, Çağdaş Üsfekes¹, Ahsen İkbal Aytekin¹, Eray Tüzün²
¹HAVELSAN Bilişim Hizmetleri Müdürlüğü
²HAVELSAN Teknoloji ve Akademi Direktörlüğü
06510 Ankara, Türkiye

Abstract. In the literature, there are various number of practices, methods, and processes that are defined to support software and system development work. These development guidelines are usually document-based, where carrying out the work and tracking the progress is supported with the help of documents. In

today's information age, information systems which produces automation solutions for other industries are not expected to carry out their development activities with document-based only approaches. Transforming the development guidelines, some of which are well-defined with certain usage practices, into automation has its own unique challenges. The Essence Framework provides a process modeling language and an extensible framework that aims to give a common framework for various process, methods and practices of software development in general. This consolidation approach promises institutions a common ground for project / product development and monitoring. However, there is no direct-operable automation infrastructure for this common ground, and currently, the progress is monitored with the help of physical cards. In this study, we have reported on our experiences on creation of a software model based on the Essence framework's domain language to support a common process automation application.

Keywords: Software Development Lifecycle Management, Software Processes, Process Automation, Software Project Management, Essence Framework

1 Giriş

Bilgisayar uygulamalarının, gündelik mal ve hizmetler ile sosyal yaşamın üretilmesini sağlayan temel bileşenlerden biri olması, yazılım üretimi çabalarını endüstrinin önemli süreç alanlarından biri haline getirmektedir. Bir yazılım ürününün geliştirilmesi, iyileştirilmesi veya hizmet olarak sunulması için düzenlenmiş adımlar dizisi Yazılım Geliştirme Süreci olarak tanımlanmaktadır [1]. Yazılım üretimi, çok sayıda pratik, yöntem ve süreç [2] kullanılarak yapılabilmektedir. Çoğu birbirleri ile benzeşmeyen bu çalışma biçimlerinin öngörülemezliğinden ve ayrıksılığından kaynaklanan sorunlarına karşı 1960'lı yıllarda Hoiser [3] ve Royce [4] ile başlayan yanıt verme çabaları günümüzde de devam etmektedir.

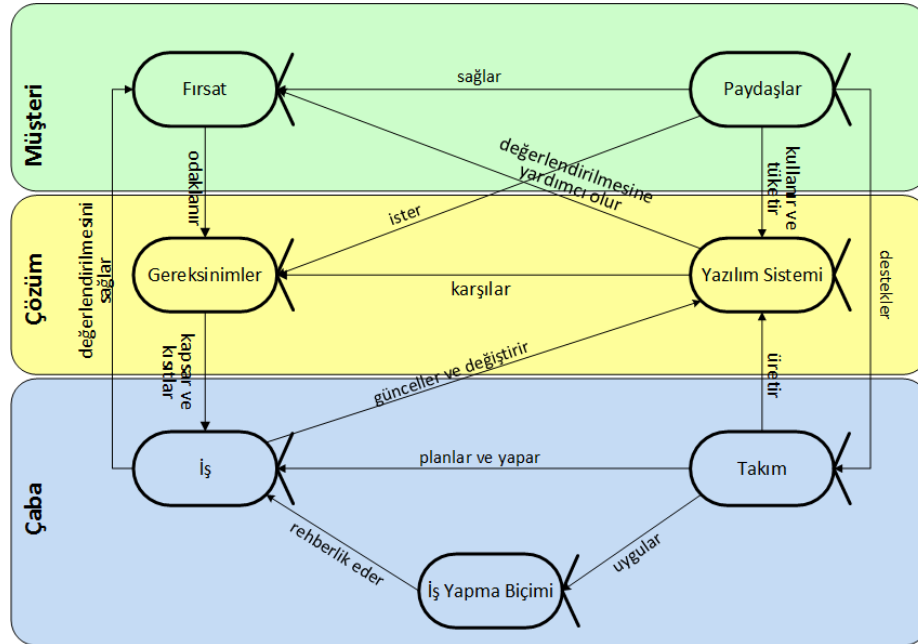
Endüstrinin diğer alanlarında olduğu gibi, bilişim alanında da üretim işlemlerinin otomasyon desteği ile yapılabilmesine olanak sağlayacak süreç standartlaştırma çalışmaları yürütülmektedir. Bu çalışmalarla elde edilen, alan dilleri ve modeller SPEM [5] ve Essence [6] belirtimleri şeklinde Object Management Group (OMG) tarafından yayınlanmaktadır. Her iki belirtim de Yazılım Mühendisliği üst (meta) modelini ve geliştirilmiş süreç modelini barındırmaktadır. Essence belirtimi, kartlar üzerinden el ile işletilebilen bir çekirdek sağladığından, otomasyon dönüşümü için seçilmiştir. Bu çalışmada, Essence belirtimi temel alınarak işletilebilir model oluşturulması ve otomasyon yazılımı geliştirilmesi, deneyimleri paylaşılmıştır.

Bildirinin ikinci bölümünde, Essence Çerçevesi ele alınmıştır. Üçüncü bölümde ise, otomasyon yazılım üretimini tetikleyen motivasyon kaynakları belirtilmiştir. Dördüncü bölümünde, otomasyon yazılımını geliştirme deneyimi aktarılmıştır. Beşinci bölümde ise sonuç ortaya konulmuştur.

2 Essence Çerçevesi

Software Engineering Method and Theory (SEMAT) tarafından geliştirilen Essence, benzer belirtilerden farklı olarak dil ve hazır bir çekirdekte oluşan çerçeve ile karşımıza çıkmaktadır. Çekirdek, Essence model dili kullanılarak ortak kullanım için üç ilgili alanında konumlandırılan yedi temel odağı¹ ve bu odakların aralarındaki ilişkiyi tanımlayan şekilde modellenmektedir. Bu model, Şekil 1’de görülmektedir.

Yazılım geliştirme sürecinde yer alan etkinlikler ve varlıklar toplanarak Müşteri, Çözüm ve Çaba adıyla üç ilgi alanı tanımlanmıştır. Müşteri ilgi alanında, bir yazılım sistemini geliştirmeye veya değiştirmeye uygun kılan şartlar Fırsat (1), bu sistemini etkileyen veya etkilenen insanlar, gruplar veya kuruluşlar ise Paydaş (2), olarak tanımlanmaktadır. Çözüm ilgi alanında, Fırsatı değerlendirmek ve Paydaşların ihtiyacını karşılamak için yazılım sisteminin barındıracağı özellikler Gereksinim (3), yazılım, donanım ve verilerden oluşan ana üretim değeri ise Yazılım Sistemi (4), olarak tanımlanmaktadır. Çaba ilgi alanında, sonuç elde etmek için yapılan zihinsel veya fiziksel çabanın yer aldığı faaliyetler İş (5), Yazılım Sisteminin geliştirilmesi, bakımı, teslim edilmesi veya desteklenmesi için aktif olarak çalışanlar grubu Takım (6), Takım tarafından yapılan işlere rehberlik eden özelleştirilmiş süreç ve araç setini İş Yapma Biçimi (7), olarak tanımlanmaktadır.



Şekil 1. Essence Çekirdeği ve Odak Noktaları [6] [7]

¹ Odak: ALPHA (Abstract-Level Progress Health Attribute) olarak tanımlanmaktadır.

Essence Çekirdeğini oluşturan her bir odak, farklı yaşam döngüsü basamağı ve basamaklar arasındaki geçiş denetimleri ile tüm sürecin sağlığı hakkında soyut seviyede bilgi sağlamaktadır.

Essence Çerçevesi, sağlamış olduğu dil ve hazır çekirdek ile yazılım geliştirme çabalarında kullanılan farklı pratik ve yöntemlerin tek bir ortak zemin üzerinde birleştirme yeteneği sağladığı gibi dil özelliklerini kullanarak çekirdeğin genişletilebilmesi olanağını da sağlamaktadır. Bu genişletme olanağı ile farklı yazılım geliştirme pratikleri ve yöntemleri aynı ortak zemin altında tanımlanabilmektedir. Tanımlanan yöntemlerin, proje veya ürünler için yürütülmesini gerçekleştirecek elektronik bir araç ise Essence Çerçevesi tarafından sağlanmamaktadır.

3 Motivasyon

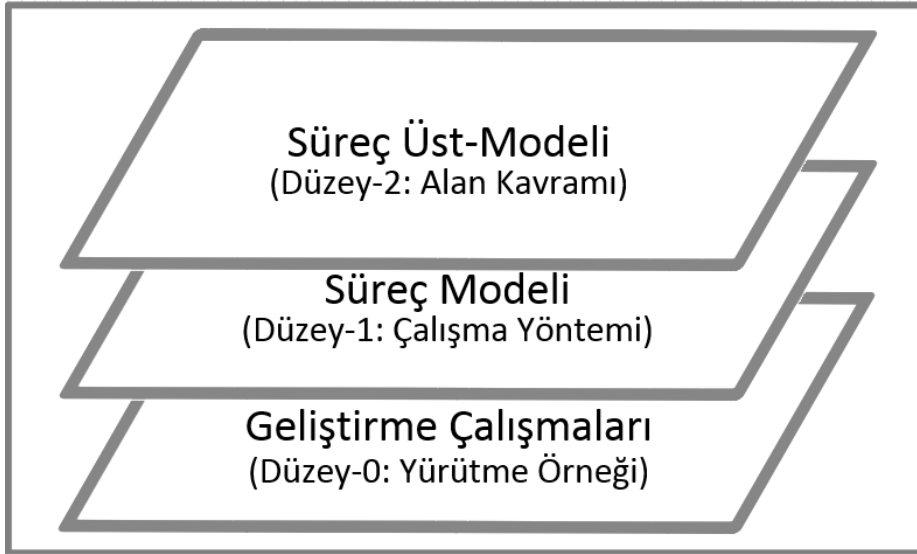
Proje, birbirinden farklı olarak nitelendirilen bir ürün ya da hizmeti başlatma, planlama, yürütme, izleme ve kapatma süreçlerinden geçirerek ortaya çıkarmak için gösterilen çabalar bütünü olarak tanımlanmaktadır [8]. Yazılım projeleri de çaba ve çıktılar açısından diğer projeler gibi birbirlerinden farklılık göstermektedir. Her biri kendine özgü olan yazılım projelerinin yönetiminde başarıyı elde etmek için bu süreçleri gerçekleyen yöntemler kullanılmaktadır. Proje yönetim sürecinde tercih edilen yöntemler değişiklik gösterse de her biri için temel süreci karşılayan ortak yönler bulunmaktadır. Şelale sürecinden çevik yaklaşımlara ve Scrum pratiğine kadar kullanılan bu yönergeler [4] [9] [10], projenin yürütülmesi ve izlenmesi için temel unsurlar olarak karşımıza çıkmaktadır. Birden çok projenin yürütülme olduğu kurumlarda, farklı yöntem ve pratiklerin tercih edilmesi, projelerin ortak bir paydada takip edilmesini zorlaştırmaktadır. Paydaşların iş birliğini arttırmak amacıyla bu yöntem ve pratiklerin herkes tarafından aynı dilde okunabilmesi bir gereklilik olarak görülmektedir [11].

Yazılım Geliştirme Yaşam Döngüsü (YGY) evrelerinde bazı çıktılar, CASE [12] araç desteği ile üretilebilmektedir. Ancak bu destek, süreç otomasyonundan çok işlem otomasyonu seviyesinde kalmaktadır. Ayrıca, kurumsal YGY için takım halinde geliştirmeyi destekleyen ortak altyapılar kurulabilmektedir [13]. Bu tür kurumsal altyapılar, barındırılan projelerin/ürünlerin çalışmalarını belirli odak noktaları üzerinden ortak bir zeminde yönetme ve izleme ihtiyacını belirgin hale getirmektedir.

Farklı süreçlerin modellenmesi için kullanılan bu ortak dilin, kurumsal seviyede yürütülen projelerin takip edilebilmesi için işletilebilir bir altyapıya sahip olması gerekmektedir. Geliştirilen otomasyon yazılımı ile izlenen projeler için modellenmiş olan göstergelerin, işlemlerin yürütüldüğü YGY araçlarından otomatik olarak beslenmesi hedeflenmektedir. Bu yetenek ile proje çalışanı ve yöneticilerinin birbirlerine ayrıca bilgi aktarmalarına gerek olmaksızın odaklar, etkinlikler ve iş ürünleri ile ilgili bilgileri izleyip, verimlilik için değerlendirebilecekleri bir işbirliği ortamı sağlanmış olacaktır.

4 Geliştirme Deneyimi

Süreç modellemesi, temel olarak Şekil 2’de görülen üç düzey ile tanımlanmaktadır [14]. Düzey-2’de, alana özgü dil ve yapı taşlarının tanımlandığı üst model yer almaktadır. Düzey-1’de, söz konusu dil ve yapı taşlarının etkin kullanımı için önerilen bileşim ile ortaya çıkan yöntem yer almaktadır. Düzey-0’da ise ilgili yöntemin, yürütmesinin yapıldığı çalışma zamanı örnekleme bulunmaktadır.



Şekil 2. Süreç Modeli Düzeyleri [14]

Essence Çerçevesi, düzey yapılarına göre oluşturulan ve aşağıda detayları verilen bileşeler ile genişletilmeye ve otomasyon yazılımına dönüştürülmeye çalışılmıştır.

Dil Bileşeni: Düzey-2 için Essence Metamodel XMI şeması, alan dil belirtimi olarak devralınmış ve korunmuştur.

Çekirdek Bileşeni: Düzey-1 için Essence belirtimi tarafından kullanım önerisi olarak sunulan odaklar, etkinlik alanları ve yetkinlikler yer almıştır. Kurumsal kullanım yönelimleri göz önünde bulundurularak bu düzey temel soyutlama düzeyi olarak kurgulanmıştır

Ontoloji (Pratikler) Bileşeni: Scrum gibi endüstriyel pratiklerin metin kaynaklarından yola çıkılarak elde edilen odak, etkinlik, rol, iş ürünü gibi ontolojik kavramların sistematik eşleme [15] ile işlendiği ve çekirdeğin genişletildiği düzeydir.

Yöntem Bileşeni: Ontoloji bileşeninde girilmiş farklı pratiklerin, Proje/Ürün çalışma alanlarını kapsayacak yöntemler şeklinde organize edildiği düzeydir. Bu düzeyde, Scrum ve Sürekli Tümlleştirme gibi pratikler birleştirilerek Çevik yöntem elde edilebilir.

Yürütme Bileşeni: Yöntem bileşeninde elde edilen yöntemin herhangi bir proje/ürün çalışması için çalışma zamanı örneğinin yürütmesini sağlayan katmadır. Bu bileşen için sağlanan eklenti desteği ile proje/ürün işlemlerinin gerçek zamanlı olarak

çalışıldığı gereksinim, tasarım, geliştirme, test ve yayın/dağıtım gibi YGY araçlarından her aşamada bilgi çekilebilmekte ve sonuçlar iş ürünü veya odağa aktarılabilir. Yazılım Bileşenleri; düzey, çalışma konusu ve örnek nesne açısından **Tablo 1**'de özetlenmiştir.

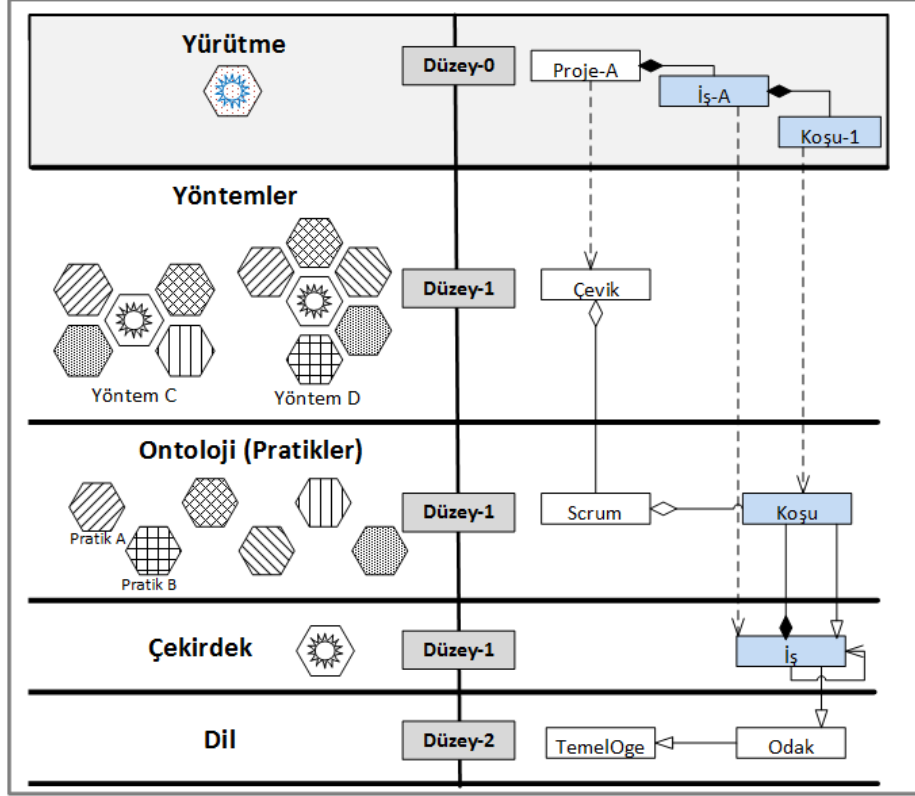
Tablo 1. Otomasyon Yazılımı Düzey Örneklemesi

Düzey	Bileşen	Çalışma Konusu	Örnek Nesne
2	Dil	Essence Metamodel XMI şeması	Odak
1	Çekirdek	Essence Belirtilim Çekirdeği	İş
1	Ontoloji (Pratikler)	Scrum Guide gibi pratik belirtilimleri	Koşu
1	Yöntem	Yöntem kalıbı	Koşu
0	Yürütme	Proje/Ürün	Koşu-1

Essence çekirdeğinin, dil belirtilimindeki sınıflardan türemiş olması ve kalıt aktarımı nedeniyle yazılım geliştirme için Nesne Yönelimli Yazılım Mühendisliği [16] yaklaşımı tercih edilmiştir. Bu tercihin, Ontoloji bileşeni için nesne genelleştirme, yöntem bileşeni için pratiklerin çakışma alanları dikkate alınarak birleştirilmesi ve yürütme bileşeni için istenmeyen özelliklerin hariç tutulması ile YGY araç eklentilerinin yönetimi açısından katkıları görülmüştür.

Otomasyon yazılımı için geliştirilen mimarinin Essence Model Düzeyleri ile olan ilişkisi **Şekil 3**'te görülmektedir. Essence Model yapısı için genişletme, Düzey-1 olarak etiketlenen Çekirdek ile onun türevleri olan Ontoloji (Pratikler) ve Yöntem bileşenlerinde yapılabilmektedir.

Düzey geçişlerindeki nesne evrimi; Düzey-2 Dil bileşeninde, *Temel Öge*'den devralınan *Odak* nesnesinin, Düzey-1 Çekirdek bileşeninde *İş* odağına, Düzey-1 Ontoloji bileşeninde, *Scrum* pratiğinin *Koşu* (sprint) alt odağına, Düzey-0 Yürütme bileşeninde ise *Proje-A* için *İş-A* odağının *Koşu-A* alt odağına, dönüşümü şeklinde izlenebilmektedir.

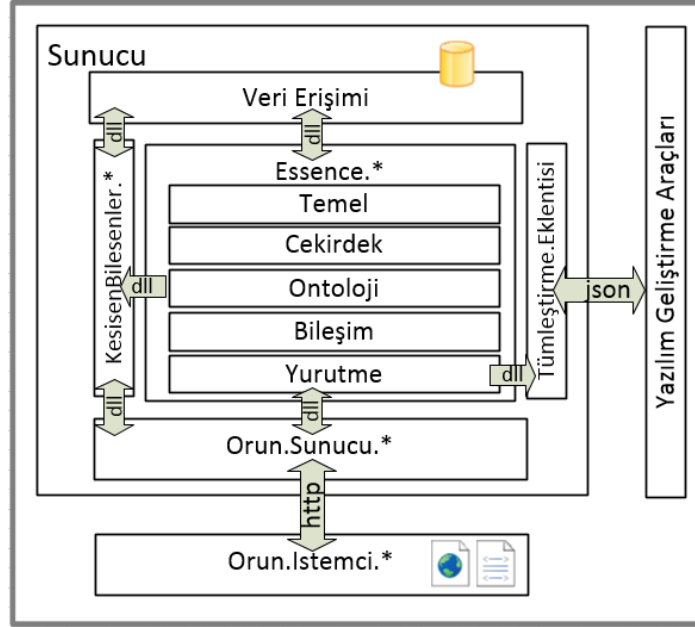


Şekil 3. Essence Model Düzeyleri ve Mimari İzdüşümü

Essence Çerçevesi üzerinde Düzyey-1 genişletmeleri ve Düzyey-0 eklemesi sonucunda elde edilen yeni çerçeve, hem yeni bileşenler ile zenginleşmiş hem de operasyonel nitelik kazanmıştır. Yeni çerçeve içerisindeki bileşenlerin hizmet verebilmesi için ihtiyaç duyacağı veri erişimi ve güvenlik özellikleri, ürün (web) mimari konumlanması ile ürün sunucusu üzerinden sağlanmıştır. Yazılım Konumlanma Mimarisi Şekil 4'te görülmektedir. Kurumsal politikalara göre ürün sunucusu ve veri tabanı sunucusu ayrı konumlandırılabilir.

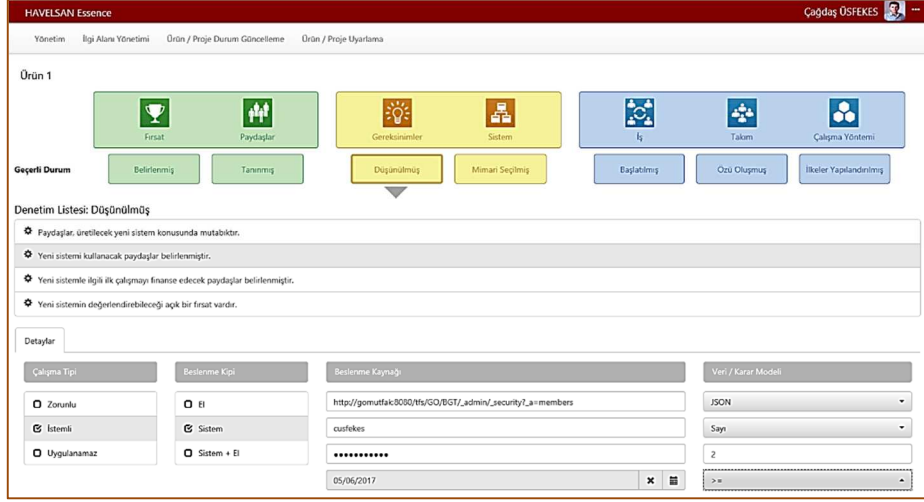
İletişim mimarisi açısından, genel amaçlı ürün istemcisi ile uygulama sunucusu arasında istemci-sunucu mimarisi uygulanmıştır. Yine, yürütme bileşeninin tümleştirme eklentileri ve diğer YGY araçlarının sorgulanmasında da istemci-sunucu mimarisi doğal yönelim olarak ortaya çıkmıştır. Ürün sunucusunun iç bileşenleri arasındaki iletişim, dll (dynamic link library) teknolojisinin kullanımı ile sağlanmıştır.

Yeni çerçevenin ürün sunucusunda hizmete alınmasıyla, hem genel amaçlı ürün istemcisi için MVC, hem de olası diğer erişim kanalları için REST API yeteneği elde edilmiştir.



Şekil 4. Yazılım Konumlanma ve İletişim Mimarisi

Kurumsal kullanıcıların yönetim, yapılandırma, ontoloji kitaplığı tanımlama ve proje/ürün yürütme işlemleri için Şekil 5’te görülen ürün istemcisi geliştirilmiştir. Örnek ekran görüntüsünde odak öğeleri ve seçili odağa ait olan denetim öğeleri listelenmiştir. Denetim öğeleri arasındaki geçişi sağlayabilmek için gerekli veri kaynağının adresi, kullanıcı adı, şifresi, veri tipi gibi alanlar bu ekran üzerinden belirlenir. İstemci mimarisi ajax üzerinde REST API istemcisi olarak çalışıp HTML denetimlerini çalışma zamanında oluşturan javascript üzerine kurgulanmıştır. REST API sayesinde her bir ürün istemcisi metodu servis gibi kullanabilmektedir ve platform bağımsız olarak farklı uygulamalar ile kolay entegrasyon sağlanabilmektedir.



Şekil 5. Essence Örün İstemcisi Yürütme Eklentisi Ekranı

Essence çerçevesi süreç otomasyon yazılımı, istemci düzlem bağımsızlığı ve tümleşme yetenekleri nedeniyle ürün tabanlı teknolojiler ile geliştirilmiştir. Geliştirme ortamı için Visual Studio 2017 kullanılmıştır. Sunucu bileşenleri, ASP.NET üzerinde nesne yönelimli programlama kullanılarak C# dili ile istemci bileşenleri ise HTML, CSS, JavaScript dilleri ve Bootstrap çerçevesi [17] ile geliştirilmiştir. Veri katmanında, programlama için T-SQL, sunucu hizmetleri için Microsoft SQL Server 2012 kullanılmıştır. İstemci sunucu iletişimde ise RESTAPI üzerinden AJAX teknolojisi ve JSON veri biçimi kullanılmıştır.

5 Sonuçlar

Yazılım ve sistem geliştiren gruplar, farklı ve çok sayıda pratik ve yöntem kullanabildiği için soyutlama katmanından faydalanılarak tüm pratik, yöntem ve modeller karşılanmaya çalışılmıştır. Bu şekilde, aynı kurum içerisinde farklı veya aynı ürün/proje için çalışan grupların kendi çalışma alanlarında farklı pratik ve yöntemler kullanabilmesi Essence Çerçevesinin temel hedefi doğrultusunda desteklenmiştir.

Tüm ürün ve proje çalışmalarını ortak bir altyapı ile iş birliğine açmak, yönetmek ve izlemek için kullanılan pratik ve yöntemlere ontoloji tabanlı yaklaşılması desteklenmiştir. Bu amaçla, tespit edilecek olan rol, etkinlik, iş ürünü ve denetim dizilerinin uygulamaya öğretilebileceği ontoloji kitaplığı sağlanmıştır.

Endüstride çok sayıda pratik ve bunlara ait rol, etkinlik, iş ürünü bulunması nedeniyle bu katmanda yapılan nesne genelleştirme çalışması ile yöntem katmanında bu pratiklere ait nesnelere ait çıkarışmasının yönetimi için özel bir çaba harcanması gerekmektedir.

Otomasyon yazılımı, YGY araçlarından veri/durum çekme ve çekilen verileri çekirdek üzerinde birleştirme yeteneği ile süreç eşgüdümü veya Tümlleştirme Göbeği² gibi konulara evrilme ufku açmıştır.

Otomasyon yazılımı üzerinde, tüm projelerin durumlarının birbirleri ile ilişkisel olarak izlenebilmesi sonucunda kurumsal portföy yönetimi için ufuk açılmıştır.

Geliştirme gruplarının iş birliğinin doğrudan artırılabilmesi ve farklı araç setine dönük ek tümlleştirme sorun ve maliyetlerinin azaltılması için Takım Geliştirme Sistemlerinin ortak kullanımı önemli görülmektedir.

Teşekkür. Yazarlar HAVELSAN yönetimine çalışmaya verdiği destek için teşekkürler ederler.

Referanslar

1. P. Feiler and W. Humphrey, "Software Process Development and Enactment: Concepts and Definitions," Software Engineering Institute, 1992.
2. M. Kenney, SDLC 3.0: Beyond a Tacit Understanding of Agile, Fourth Medium Press, 2010.
3. W. A. Hosier, "Pitfalls and Safeguards in Real-Time Digital Systems with Emphasis on Programming," *IRE Transactions on Engineering Management*, Vols. EM-8, no. 2, pp. 99-115, 1961.
4. W.W.Royce, "Managing the Development of Large Software Systems," *Proceedings of IEEE WESCON 26 (August): 1-9*, pp. pp.328-338, 1970.
5. O. M. Group, "Software & Systems Process Engineering Metamodel™ (SPEM™)," [Online]. Available: <http://www.omg.org/spec/SPEM/>. [Accessed 05 06 2017].
6. O. M. Group, "Essence™ - Kernel And Language For Software Engineering Methods (Essence)," [Online]. Available: <http://www.omg.org/spec/Essence/>. [Accessed 05 06 2017].
7. G. Giray, "Yazılım Geliştirme Süreci Uyarılama İçin Bir Çerçeve Önerisi," in *Ulusal Yazılım Mühendisliği Sempozyumu*, İzmir, 2015.
8. Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK Guide), Newtown Square: Project Management Institute, 2013.
9. J. S. Reel, "Critical success factors in software projects," *IEEE Software*, vol. 16, no. 3, pp. 18-23, 1999.
10. K. Schwaber and J. Sutherland, "The Scrum Guide," 2014.
11. A. McDonough, "Munich Re and ESSENCE – Kernel and Language for Software Engineering Methods: A Case Study," Object Management Group, 2014.
12. D. Teichrow and E. A. Hershey, "PSL/PSA a computer-aided technique for structured documentation and analysis of information processing systems," in *Proceedings of the 2nd international conference on Software engineering*, 1976.

² Tümlleştirme Göbeği: Farklı Yazılım Geliştirme işlevlerine dönük çalışan uygulamalarının veri değişimi yapabilmelerini sağlayan merkezi araçlara verilen isimdir. (ing. Hub).

13. Y.Macit, E.Tüzün, K.Ince, A.I.Aytekin, «Büyük Ölçekli Bir Organizasyonda Uygulama Yaşam Döngüsü Yönetimi Uygulama Deneyimi,» *Proceedings of the 8th Turkish National Software Engineering Symposium*, 2014.
14. C. Rolland, "Modeling the Requirements Engineering Process," in *3rd European-Japanese Seminar on Information Modelling and Knowledge Bases*, 1993.
15. G. Giray, E. Tüzün, B. Tekinerdogan and Y. Macit, "Systematic approach for mapping software development methods to the essence framework," in *Proceedings of the 5th International Workshop on Theory-Oriented Software Engineering*, 2016.
16. I. Jacobson and e. al., *Object Oriented Software Engineering*, Addison-Wesley, 1992.
17. "Bootstrap v.3.3.7," [Online]. Available: <http://getbootstrap.com/getting-started/>. [Accessed 13 Haziran 2017].