

Savunma Projelerinde Çevik Metodolojiler

Burcu Nalbant

Mert Bıçakçı

burcun@ayesas.com
AYESAŞ, Ankara

mertb@ayesas.com
AYESAŞ, Ankara

Özet. Günümüz Savunma Sanayi yazılım projeleri uzun takvimlerde, uluslararası standartlara uygun ve kısıtlı kaynaklar kullanılarak gerçekleştirilmektedir. Genellikle Savunma Sanayi projeleri geliştirilirken kontratsal olarak Şelale modeli uygulanmasına karar verilmektedir. Ancak Şelale modeli izlemek, projenin son safhasında yapılan değişikliklerin pahalıya mal olması, risklerin zamanında öngörülememesi, büyük takımlar içerisindeki iletişimin sağlıklı kurulamaması gibi sonuçlara yol açmaktadır. Bütün bu etkenler, Savunma Sanayi firmalarını, proje geliştirirken yeni yazılım geliştirme süreçleri arayışına sürüklemektedir.

Savunma projelerindeki mevcut problemleri en aza indirebilmek için şirketimiz AYESAŞ'ta Çevik yöntemler uygulanmıştır. Örneğin, büyük takımlar arasındaki iletişim kopukluğunu azaltmak için yapılan işlerdeki bilgi aktarımı günlük yapılan kısa süreli toplantılarla sağlanmıştır. Çevik metodlarının kullanılması, takım üyelerinin Sprint (koşu) boyunca yapacağı işleri görebilmesi açısından da fayda sağlamıştır. Böylece proje yönetimi, planlanan ve gerçekleşen eforun farkını minimuma indirip takvime uyabilmiştir. Ayrıca, riskler zamanında öngörülmüş ve gerekli önlemler alınabilmiştir.

AYESAŞ, CMMI Seviye 3 uyumlu yazılım süreçlerine sahip bir firmadır. Kontratsal ve süreçsel gereklerden ötürü Çevik metodları mevcut yazılım süreçlerine uygularken bazı uyarlamalar gerçekleştirilmiştir. Bu uyarlamalar sayesinde hem CMMI Seviye 3 uyumlu süreç kriterleri sağlanmış, hem de Çevik metodların prensipleri korunmuştur. Bu uyarlamalara örnek olarak, dokümantasyon gereksinimlerinin Sprint hedefine (Sprint Goal) dahil edilmesi, "bitti tanımı" (definition of done) ve Sprint hedefinin Sprint sonucu çıkacak ürüne/ürün parçasına göre belirlenmesi (Örneğin Sprint ürünü bir doküman setiyse, bitti tanımı ve Sprint hedefi bu doküman setinin eş gözden geçirilmiş olarak yayınlanması olabilir), organizasyonel yapının değişmeden Scrum rollerinin tanımlanması verilebilir.

Bu makalede, AYESAŞ'ta savunma projeleri geliştirirken kullanılan Çevik metodolojiler ve bunların CMMI Seviye 3 uyumlu AYESAŞ yazılım süreçlerine uyarlanması anlatılmaktadır.

1 Giriş

Savunma Sanayinde projelerin genel özellikleri olarak, projelerin uzun takvimlerinin olması (bir seneden fazla), bütçe ve kaynak ihtiyaçlarının yüksek olması, müşterinin ihtiyaçlarının net olmaması ve değişebilmesi, müşteri ile iletişimin az olması, proje süresi boyunca belirli kilometre taşlarının olması (gereksinim, detaylı

ve kritik tasarım, test hazırlık gözden geçirmeleri gibi) sayılabilir. Savunma Sanayi projelerinin genel özellikleri yazılım geliştirme aşamasında Şelale modelinin [5] kullanımını öne çıkarmaktadır. Öte yandan, gün geçtikçe Savunma Sanayi şirketleri ekiplerinden daha başarılı projeler yapmalarını beklemektedir. Proje başarısı ise proje harcamalarının azaltılması, çıkan ürün kalitesinin ve müşteri memnuniyetinin artması, proje takvimin azaltılması ile ölçülmektedir. Bu sebeplerden ötürü, proje ekiplerinin daha kompleks projeleri daha az bütçeyle daha az zamanda ve daha yüksek kalitede ürün çıkartarak yapmaları beklenmektedir.

Savunma Sanayi projelerinde sıklıkla kullanılmakta olan Şelale modelinin günümüz proje dinamiklerine uymayan birçok dezavantajı vardır. Bu dezavantajlar arasında aşağıdakileri sayabiliriz:

- Kompleks olmayan ve küçük projelerde başarılı olma şansı yüksektir. Proje zorlaştıkça ve büyüdükçe başarılı olma şansı düşer.
- Kısa takvimlerde daha başarılıdır. Takvim uzadıkça başarı şansı düşer.
- Proje başlarken proje gereksinimlerinin proje ekibi tarafından çok iyi tanımlanmış olması gerekmektedir. Muğlak gereksinimler, Şelale modeli kullanılan projelerde başarı şansını azaltmaktadır.
- Belirlenen gereksinimlerin proje hayatı boyunca çok az değişikliğe uğraması gerekmektedir. Projenin test aşaması gibi son aşamalarında oluşacak gereksinim değişiklikleri bütçe ve kaynak açısından pahalıya mal olmaktadır.
- Projede sistem entegrasyonunun geç safhada gerçekleşmesi ortaya çıkacak sorunların geç fark edilmesine, dolayısıyla bu sorunların büyümesine yol açmaktadır.

Ancak Savunma Sanayi projelerinin bazı özellikleri, Şelale modelinin kullanımını zorunlu hale getirmese de tercih edilmesini sağlamaktadır. Savunma Sanayi projelerinde genellikle bazı teknik gözden geçirmeler gibi kilometre taşları mevcuttur. Bu kilometre taşlarının varlığı, Şelale modelindeki gibi gereksinim analizi, ön tasarım, detaylı tasarım, yazılım geliştirme, entegrasyon ve test aşamalarının varlığını zorunlu kılmaktadır. Savunma projelerinde takvim sırasıyla gereksinim, ön tasarım, detaylı tasarım, teste hazırlık gibi gözden geçirmeler gerçekleşmektedir. Bu gözden geçirmeleri gerçekleştirebilmek için Şelale modelindeki gibi önce gereksinimleri analiz etmek, sonra ön ve detaylı tasarım yapmak, daha sonra yazılım geliştirmek, yazılım kod parçalarını entegre etmek ve en son test yapmak gerekmektedir. Dolayısıyla, bu kilometre taşlarının varlığı yazılım geliştirme modeli olarak Şelale modelini seçmeyi daha olası kılmaktadır.

Yazılım dünyasında Çevik metodolojilerin kullanımı hızla artmakta ve Çevik metotları örnek alarak yeni metodolojiler ortaya çıkmaktadır. Popüler Çevik metodolojiler olarak Crystal, Unified Process, Scrum, Extreme Programming (XP), Test Driven Development (TDD) metodolojileri sayılabilir. Bütün bu metodolojilerin amacı, yazılım geliştirirken daha çevik davranma becerisini proje ekibine aktarmaktır. Ancak daha önce de bahsettiğimiz gibi Çevik metodolojileri yazılım geliştirme süreçlerine adapte ederken Savunma Sanayi projelerinin özelliklerini de göz önünde bulundurmak gerekir.

2 Çevik Metotlar ve Savunma Projeleri

Çevik metotlar, yazılım geliştirme süreçlerinin verimliliğini arttırmak, süreçleri daha pratik hale getirmek ve hedefe yönelik çalışma sağlamak için uzun zamandır yazılım sektöründe uygulanmaktadır. Çevik yaklaşımlar, klasik olarak uygulanan Şelale modelinden farklı olarak yinelemeli metodolojiyi benimser. Böylece, alınan geri bildirimlerle sürekli olarak isteklere cevap verebilen ve değişime adapte olan ürünlerin oluşabilmesini sağlar.

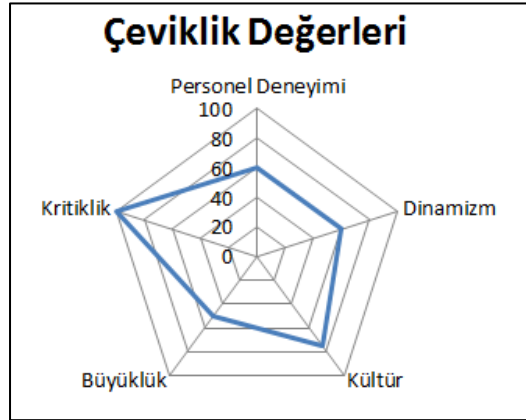
Çevik metotların en çok uygulananlarından biri olan Scrum [4], yalın ve oturmuş çerçevesiyle yazılım projelerinde katma değer bir performans artışı sağlamaktadır. Scrum'ın en belirgin özelliklerinden biri "Sprint" adı verilen birkaç haftalık periyotlardan oluşması ve her Sprint'in kendi içerisinde tüm yazılım yaşam döngüsünü barındırmasıdır. Sprint sonunda, "bitmiş" ve sevk edilebilecek hale gelmiş ürün veya ürün parçası oluşturmak hedeflenir. Diğer bir deyişle, oluşan çıktı, geliştirilmiş, test edilmiş ve hatalarından arınmıştır. Savunma sanayi projelerinde genellikle uygulanan Şelale modeline göre ise uzun bir geliştirme fazını test fazı takip eder. Zamanında tespit edilemeyen ve birbirini bloke eden hatalar, test sürecinin sancılı ve tahminlerin çok üzerinde eforlarla gerçekleştirilmesini sağlar. Çevik metodolojiler sayesinde her Sprint'te geliştirilen yazılım Sprint içerisinde test edildiği ve hatalar zamanında giderildiği için Savunma projelerindeki test fazına gelindiğinde "bitti tanımı"na uyan, çalışan ve minimum hataya sahip olan, dolayısıyla daha kaliteli bir yazılım çok daha az eforla entegre edilebilir/kullanılabilir.

Çevik metodolojilerde ve özellikle Scrum'da, takım çalışması ön plandadır. Adını Rugby sporundaki bir hücum taktiğinden alan Scrum, planlanan işin takım olarak benimsenmesini ve hedeflere ulaşmak için kendi kendini organize edebilen, motive ve sürekli iletişim halinde çalışan takımlar oluşmasını sağlar. Savunma sanayi projelerinin genellikle büyük ölçekli olmasından ötürü farklı ve dağıtık bireylerden oluşan kalabalık takımlar kurulabilmektedir. Sprint başında yapılan planlama toplantıları sayesinde bu takımların ortak hedefe yönelmeleri, günlük toplantılarla devamlı iletişimde olmaları, Sprint sonunda planlananlar gerçekleştikçe ise takım olarak motivasyonlarının artması yine Çevik metotlarla sağlanmaktadır. Ayrıca, günlük toplantılar ve sürekli takip edilen Scrum tahtası (Scrum Board) sayesinde, Çevik yaklaşımların ana prensiplerinden biri olan Şeffaflık (Transparency) [4] uygulanarak büyük ölçekli projelerde bilgi akışındaki aksaklıklar giderilebilir. Böylece genellikle uzun takvimlere sahip olan Savunma Sanayi projelerinde riskler zamanında öngörülerek gerekli önlemler alınabilir. Bu açıdan Çevik yöntemler, projenin son safhalarında yeni fark edilen ve artık alınabilecek bir aksiyon kalmayan veya yüksek miktarda para/zaman kaybına neden olan kritik durumların da önüne geçebilmektedir.

Bilindiği gibi, Savunma Sanayi projeleri birden fazla sistemin entegrasyonunu gerektirebildiği gibi birçok iş kırılımından oluşur ve karmaşıklık dereceleri yüksektir. Bu açıdan, projenin başında planlar ne kadar detaylı olursa olsun analiz ve tasarım faaliyetleri ilerledikçe yeni iş kırılımları eklenebilir, mevcut olanlar değiştirilebilir ya da silinebilir. Proje belli bir olgunluğa gelip taraflar tüm gereksinimlerini detaylandırırsa da işin yapılışıyla ilgili günlük hatta saatlik planları aylar önce

kestirmek mümkün değildir. Çevik yöntemler sayesinde, hedeflenen işler Sprint başlangıcında planlanarak en küçük parçalara kadar kırılır. Bu sayede, Sprint süresince tamamlanacak tüm görevler belirlenerek etkin bir zaman yönetimi gerçekleştirilir ve projenin sonuna doğru yaşanacak takvimsel sıkışıklıklar önlenir. Ayrıca, proje küçük parçalara ayrıldıkça karmaşıklık derecesi de en düşük seviyeye indirilir.

Savunma Sanayi projelerinde Çevik metodolojilerinin uygulanıp uygulanamayacağı belirli faktörler çerçevesinde çeviklik analizi yapılarak değerlendirilebilir [1]. Bu faktörler kritiklik, büyüklük, dinamizm, kültür ve personel deneyimi olarak tanımlanmaktadır. Savunma projeleri için bu faktörleri gösteren örnek bir grafik Error! Reference source not found.'de verilmiştir. Bu faktörler YESAŞ'ta hali hazırda sürdürülmekte olan bir projeye ait değerlerdir. Şekilde faktörler merkeze yaklaştıkça projenin çevik metodolojiye uygunluğu artmaktadır. Savunma projelerinde çeviklik analizinden de anlaşılacağı gibi Çevik metodolojiler kullanmak mümkündür ancak Savunma Sanayi projelerinin dinamiklerini de göz önüne almak ve çeviklik ile plan odaklı yazılım geliştirme arasındaki dengeyi sağlamak önemlidir. Bu açıdan, Savunma Sanayi şirketleri Çevik metodolojileri kendi süreçlerine uygularken şirketlerin uyarlamaları daha etkili sonuçlar doğuracaktır.



Şekil 1 - Örnek bir Savunma Sanayi projesi için Çeviklik analizi

3 Çevik Metotlara AYESAŞ Yaklaşımı

AYESAŞ, CMMI Seviye 3 [3] uyumlu yazılım geliştirme süreçlerine sahip bir firmadır. Her ne kadar Çevik Yazılım Geliştirme Manifestosu'nda [2] kapsamlı dokümantasyon ve süreç ve araçlara daha az önem verildiği belirtilse de, CMMI gerekleri uygulanırken Çevik metodolojilerden de faydalanmak mümkündür. CMMI ve Çevik metodolojiler genel kanının aksine birbirleriyle çelişen süreçleri tanımlamaz [322]. Bu kapsamda, AYESAŞ yazılım geliştirme süreçlerine Çevik metotları adapte ederken bazı uyarlamalar yapılarak Çevik metodolojilerin avantajlı yanları alınırken CMMI uyumlu olmanın gerektirdiği kurallar da korunmuştur. Bu uyarlamalar aşağıda verilmiştir:

- Bazı Sprint’lerde sevk edilebilecek üründen ziyade doküman ürün olarak Sprint hedefine dahil edilmiştir.
- “Bitti tanımı”, Sprint sonu oluşturulacak ürüne göre belirlenmiştir. Örneğin, yazılım için “bitti tanımı”, “yazılımın doğrulanmış olması” iken, doküman için “bitti tanımı”, “dokümanla ilgili eş gözden geçirmenin tamamlanması” olarak kabul edilmiştir.
- Sprint planlama, Sprint kapanış, Sprint değerlendirme, günlük Scrum toplantılarının yanı sıra belirli periyotlarla düzenlenen IPR (Internal Project Review) toplantıları da gerçekleştirilmiştir.

Literatürde, Scrum takımlarının kişi sayısının 5-9 arasında olması önerilmektedir [4]. Ancak AYESAŞ Savunma Sanayi projelerinin doğası gereği takımlar daha fazla kişi içerebilmektedir. Bu yüzden, takım içi koordinasyonu aksatmamak açısından Scrum of Scrums (yani projede birden fazla Scrum takımının yer alması ve bu takımların birleşerek üst seviyede bir Scrum takımı oluşturarak çalışması) uygulanmıştır.

AYESAŞ organizasyonel yapısını korumak için Scrum rollerinde de uyarlamalar gerçekleştirilmiştir. Örneğin, Product Owner (Ürün Sahibi) rolünü Proje Yöneticisi üstlenmiştir. Bu noktada, Sprint sonunda yapılan demo’lar proje yöneticisine sunularak gereksinimleri karşılayıp karşılamadığı belirlenmiştir. Ayrıca, takımı en iyi tanıyan takım liderleri de Scrum Master olarak atanmıştır. Yine günlük Scrum toplantıları takım liderleri tarafından yürütülüp olası engeller ortadan kaldırılmıştır. Scrum of Scrums uygulanan projelerde ise teknik lider takımların Product Owner rolünü üstlenirken, ana Product Owner Proje Yöneticisi olarak atanmıştır.

AYESAŞ Scrum metodolojisi, projelere aşağıdaki özellikler göz önünde bulundurularak uygulanmıştır:

- 1 Hikaye Puanı (Story Point), 1 saatlik efor olarak kabul edilmiştir.
- İki-dört hafta arasında olması tavsiye edilen Sprint süresi, üç hafta olarak uygulanarak sürenin planlamaya geçecek kadar uzun, motivasyonu kaybetmeyecek kadar da kısa olması sağlanmıştır.
- Üç haftalık periyot için takım üyesi başına 135 saat olan çalışma süresi, 100 saatlik net iş olarak planlanmış, 35 saat planlanmayan diğer işler için tampon zaman olarak bırakılmıştır.
- Scrum metodolojisinde takım içerisinde üyeler rolden bağımsız olarak her işi yapabilirken, AYESAŞ’ta organizasyon yapısı gereği takım üyeleri sabit olan rollerini (test ve yazılım geliştirme gibi) yapmaya devam etmişlerdir.

4 Sonuç ve Gelecek Çalışmalar

AYESAŞ’ta uyguladığımız Çevik yöntemlerin proje ve takım performanslarına etkilerini ölçmek amacıyla birçok farklı açıdan bakış sağlayan metrikler tanımlanmış ve toplanan veriler anlamlandırılmıştır. Pilot projelerde gerçekleştirilen detaylı analizlerle Çevik yöntemler uygulanmadan öncesi ve sonrası değerlendirildiğinde günlük yazılan ortalama kod satır sayısı (Source Lines of Code) %129 artarken haftalık olarak proje ilerlemesinde kazanılan değer (Earned Value) ortalaması %61 oranında iyileşmiştir. Ayrıca, planlanan ve gerçekleşen saat bazındaki eforlar arasındaki ortalama fark yaklaşık %69 oranında azalmıştır. Çevik yöntemlerin

uygulanmaya başlanmasından şu ana kadar ise takım hızı (Velocity) yaklaşık %23'lük bir trendle artmaya devam etmektedir.

Sonuç olarak, AYESAŞ'ta uyguladığımız Çevik yöntemler sayesinde Savunma Sanayi projelerinin ve takımların performanslarında sayısal olarak ölçülen ve sözlü olarak bildirilen iyileşmeler gözlemlenmiştir. Bu iyileşmeler arasında takımın verimliliğinin, motivasyonunun ve takım içi iletişiminin artması, planlanan ile gerçekleşen eforların arasındaki farkın azalması, projenin anlık durumunun daha objektif ve net gözlenebilmesi, takım hızının artması, risklerin önceden görülebilmesi ve gerekli aksiyonların alınması sayılabilir.

Çevik yöntemler uygulanmadan önceki ve sonrasındaki performansları karşılaştıran metrikler periyodik olarak analiz edilmeye devam edilmektedir. Ayrıca, sürekli iyileştirmeyi sağlamak amacıyla çevik yöntemlerin başlangıcından itibaren Sprint performanslarını ölçmek için tanımlanan çevik metrikler de analiz edilerek karar destek sistemi olarak kullanılmaktadır. Bu metriklere örnek olarak Çevik CPI (Agile Cost Performance Index), Çevik SPI (Agile Schedule Performance Index), takım memnuniyeti anketi, hikaye döngü zamanı (Story Cycle Time), hız (Velocity), hız değişimi (Variation in Velocity), yapılmakta olan işler (Work in Progress), planlanmamış değişiklikler (Unplanned Changes) verilebilir.

İleriki safhalarda, bahsedilen analizler sayesinde var olan AYESAŞ yazılım geliştirme süreçlerine hangi Çevik metod özellikleri ekleneceği belirlenecek ve AYESAŞ proje yazılım yaşam döngüsü seçimine Çevik Metotlar da eklenecektir. Ayrıca, ileride yapılacak Savunma Sanayi projelerinde kontratsal olarak Çevik metotların kullanılması da hedeflenmektedir.

Kaynaklar

1. B. Boehm and R. Turner. *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison-Wesley, 2003.
2. K. Beck, J.Grenning, R. Martin, M. Beedle, J. Highsmith, S. Mellor, A. v. Bennekum, A. Hunt, K. Schwaber, A. Cockburn, R. Jeffries, J. Sutherland, W. Cunningham, J. Kern, D. Thomas, M. Fowler, and B. M. *Manifesto for agile software development*. <http://www.agilemanifesto.org>, 2001.
3. Software Engineering Institute (SEI) CMMI Product Team. *CMMI for Development Version 1.3*. <http://www.sei.cmu.edu/reports/10tr033.pdf>, Carnegie Mellon. November 2010.
4. J. Sutherland, K. Schwaber. *Scrum Kılavuzu*. Scrum.org. <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-TR.pdf#zoom=100>, Temmuz 2013.
5. Bell, Thomas E., and T. A. Thayer. *Software requirements: Are they really a problem? Proceedings of the 2nd international conference on Software engineering*. IEEE Computer Society Press, 1976.
6. C. Northern, Dr. K. Mayfield, R. Benito and M. Casagni. *Handbook for Implementing Agile in Department of Defense Information Technology Acquisition*. The MITRE Corporation, 2010.