

Saldırgan Yazılım Geliştirme (Scrum Software Development)

Sadi Evren SEKER, İstanbul Medeniyet Üniversitesi, Yönetim Bilişim Sistemleri A.B.D. İşletme Bölümü
academic@sadievrenseker.com

Çevik yazılım geliştirme (agile software development) yöntemlerinden birisi olan saldırgan yazılım geliştirme (scrum software development) yönteminin en önemli özelliği, yazılım geliştirme sürecinde sürekli tekrarlanır (iterative) yapıda ve arttırımlı (incremental) özellikte çözüm üretmesidir. Çevik yazılım geliştirme yöntemindeki bütün yaklaşım özelliklerini benimsemiştir. Örneğin ihtiyaçların bir yayık gibi sallandığı (requirements churn) değişkenlik ortamında, tamamen değişken özelliklerde ekip yönetimi ve ürün geliştirme stratejilerinin kullanıldığı bir yöntemdir. Saldırgan yöntem (scrum method) göre bir yazılım problemi tam olarak anlaşılabilir ve tanımlanamaz, bu uğurda enerji harcamak yerine deneyimci (empirical) şekilde ihtiyaçlara hızlı bir şekilde cevap verilmesi daha doğrudur.

Tarihsel Süreç

Bilgi yönetimi (knowledge management) konusundaki çalışmaları ile de tanınan Ikujiro Nonaka ve Hirotaka Takeuchi [1] 1986 yılındaki “Yeni Yeni Ürün Geliştirme Oyunu” [2] başlıklı yazılarında geleneksel ve ardışık yaklaşıma karşı çıkmışlardır. Bu yazılarında önemli bir yaklaşım farkı aşağıdaki şekilde tercüme edilebilir:

“örgütsel bilgi üretimi, özellikle yeniliğin sürekli, artan ve spiral olarak getirilmesi ile iyi olmaktadır” [3]. O yıllarda henüz yazılım endüstrisinin bugünkü kadar gelişmiş olduğundan bahsedilemez bu yüzden yazarlar daha çok fotokopi makineleri, otomobil ve yazıcı endüstrileri üzerinden örnekler vermişler [4] ve bir çok amaçlı takımın birbiri ile kesişen aşamalarındaki proje süreçleri üzerinde takım çalışması ile bütün süreci çözmelerinden bahsetmişlerdir. Bu yaklaşımı da bütüncül (holistic) ve saldırgan bir oyun ismi olan ragbi (rugby) isimleri ile nitelmişlerdir. Rugby oyunu ayrıca “scrum” teriminin çıkışı için de önemlidir çünkü oyun içerisinde oyuncuların kafalarını aşağı eğerek top hakimiyeti elde etmeye çalıştıkları ve oyuncuların yakın durdukları bir dizilime de scrum ismi verilmektedir [5].

Saldırgan yazılım geliştirmenin Scrum (saldırgan) kelimesi ile tek başına anılması ilk defa 1990’lı yıllarda olmuştur ve

Bu yazı için tavsiye edilen atf (APA şekli):

Seker, S. E. (2015) *Saldırgan Yazılım Geliştirme (Scrum Software Development)*, YBS Ansiklopedi, v. 2, is. 1, pp. 12-15

Easel Şirketindeki gelişmiş geliştirme yöntemleri için kullanılmıştır[6]. Schwaber ve Sutherland, 1995 yılında bu yaklaşımı bir makale haline getirerek Austin Teksas’taki OOPSLA çalıştayında sunmuşlardır [7].

2001 yılında ise yine Schwaber tarafından yazılan ve başlığı “Scrum ile Çevik Yazılım Geliştirme” olarak Türkçeye çevrilebilecek kitapta [8] da Scrum’dan bahsedilmektedir ancak bu dönemde scrum kelimesinin anlamında da değişim olmuş ve ilk başlardaki anlamı yerine çoğu işletme tarafından artık farklı kelimelerin baş harflerinden oluşan SCRUM kısaltmasına dönüşmüştür.

Daha sonraları Schwaber tarafından Scrum Organizasyonu kurulmuş ve Scrum Master Sertifikaları bu kurum tarafından vermeye başlanmıştır. 2009 yılında ise bu organizasyondan ayrılan Ken tarafından Scrum.org sitesi açılarak Scrum yaklaşımı farklı bir yoldan da devam etmektedir [9].

Roller

Saldırgan geliştirmedeki rolleri anlatmaya geçmeden önce, kavramların üzerine oturtulduğu bir fabl’dan bahsetmekte yarar var. Fabl bir domuz ve tavuk arasında geçmektedir ve İngilizce literatürde “pig and chicken fable” olarak bulunabilir. Buna göre bir tabakta yemek ikram etmek isteyen domuzun tek alternatifi kendi etini tabağa koymaktır, buna karşılık tavuk için yumurta ikram etme imkanı vardır. Bu iki yaklaşım arasındaki fark, birisinin kendisini ikrama adanması iken diğerinin sadece bu sürece katkı sağlamasıdır.

Bu fabl örneği yazılım geliştirme süreci için sıklıkla kullanılmaktadır. Yani yazılım geliştirme sürecinde kendisini sürece aday ve kendi varlığını bu sürecin bir parçası haline getiren kişilerin yanında, sadece sürece katkı veren ve bu katkının karşılığında, aslında çok da bedel ödemeyen rollerden bahsedilebilir.

Temel olarak saldırgan modelde 3 rol vardır :

- Ürünün sahibi: genelde müşteridir ve çoğu zaman yazılımın paydaşları olarak düşünülebilir. Müşterinin sesi (voice of customer) olarak da ifade edilebilir.
- Geliştirme Takımı: Yazılımı geliştirecek takımdır. Takım ruhu ile çalışması, 3-9 kişi arasında ve çok fonksiyonlu bireylerden oluşması, her dönemde (sprint) dönem sonu teslimlerini (PSI: potentially

shippable sprints) yapması beklenir.

- Scrum Master (Saldırı Yöneticisi, Saldırı Ustası): Sürecin yönetilmesinden sorumludur ancak klasik bir yönetici veya takım lideri olarak düşünülmemelidir. Daha çok saldırı takımının çalışmasını engelleyici bütün unsurların çözülmesi ve takımın daha başarılı şekilde sonuca ulaşmasından sorumludur. Scrum sürecinin sağlıklı ilerlemesinden sorumludur ve scrum yönteminin başarısı için yöntemin şartlarının doğru uygulanması için takımı ve paydaşları zorlamakla yükümlüdür.

Scrum yöntemindeki yöneticilik rolü daha çok liderlik çeşitlerinden “Hizmetkar Lider” (servant leader) çeşidine benzetilebilir. Yani takımın başarısı için takıma hizmet eden lider rolüdür.

Scrum yönteminde ayrıca ürün sahibinin de sorumlulukları bulunmaktadır. Temel olarak yazılımın paydaşları (kullanıcılar, kullanıcıların hizmet verdiği müşteriler, teknik destek ekibi, tedarikçiler vs.) ile yazılım ekibi arasında köprü görevi bulunmaktadır. Bunlar aşağıdaki şekilde sıralanabilir:

- üretilen çözümleri paydaşlara sunmak ve açıklamakla sorumludurlar
- Yeni sürümlerin duyurusunu yaparlar
- Takımın iletişiminden sorumludurlar
- Proje aşamalarının organizasyonundan sorumludurlar
- Paydaşların gelişim sürecindeki eğitiminden sorumludurlar
- Öncelikler, kapsam, bütçe ve zamanlama konularını tartışan makamdır

Scrum Olayları

Saldırgan yazılım geliştirme (scrum software development) yönteminde süreç boyunca olaylar ve bu olayların sıralaması önemli bir rol teşkil eder. Bu olayların isimleri aşağıdaki şekilde sıralanabilir.

- Çıkışlar (Sprint, iteration): Saldırgan yazılım geliştirme yönteminde koşullardaki hızlı çıkışlara (sprint) benzer şekilde çıkışlar bulunmaktadır. Aslında bu çıkışlar farklı aşamalarda teslimleri temsil etmektedir. Her çıkış belirli bir zaman kutusuna konulmuştur ve saldırgan yazılım geliştirme yönteminin temel yapı taşı olarak kabul edilebilir. Her çıkış, daha önceden tanımlanmış bir hedefe ulaşmak için yapılır ve çıkış sonunda hedefe ulaşıp ulaşılmadığı kontrole tabi tutulur. Başarılı her çıkışın teslim edilebilir bir ürün veya eklenti olarak yazılımın paydaşları tarafından kullanılacak halde teslim edilmesi, gerekli eğitim ve son kullanıcı dokümanlarını içermesi beklenir.
- Toplantılar: Saldırgan yazılım geliştirme yönteminde

toplantılar 3 grupta toplanabilir:

- Çıkış Planlama Toplantıları (Sprint Planning Meeting): her çıkış öncesinde çıkış ile ilgili planlama yapılır. Periyotları çıkış süreleri ile aynıdır. Örneğin ortalama ayda bir çıkış yapılan bir geliştirme modelinde aylık toplantılar yapılır. Her toplantıda bir kaynak planlaması da yapılır. Kaynak planlaması genel olarak zaman ve ekip ayrımı gibi detayları içerir.
- Günlük Saldırgan Toplantılar (Daily Scrum Meetings): Her gün gerçekleştirilen toplantılarda, her takım üyesi günlük olarak gelişmeleri içeren bir rapor sunar. Takım üyelerinden eksikler olsa bile daha önceden belirlenen zamanda mutlaka toplantı başlar. Toplantı sürekli aynı mekanda yapılır. Genelde herkesin katılımına ve konuşmasına açıktır ancak genelde çekirdek rollerdeki kişilerin konuşması beklenir. Toplantılar zaman sınırına sahiptir ve genelde 15 dakikalık bir süredir. Toplantının cevap aradığı 3 temel soru vardır ve bunlar “dün hedef için neler yapıldı?”, “bugün hedef için neler yapılacak?”, “hedefe ulaşmayı engelleyici herhangi bir gelişme var mı?”
- Sonuç toplantıları (Sprint Review, Sprint Retrospective): Çıkış sürecinin sonunda değerlendirme toplantısıdır. Geriye yönelik olarak (retrospective) çıkış (sprint) süresince yapılan işler değerlendirilir ve her takım üyesinin iki soruya cevap bulması istenir. Bunlar “iyi giden işler nelerdir?”, “iyileştirilmesi gereken işler nelerdir?” sorularıdır. Oturun yine Saldırı Ustası (Scrum Master) tarafından yönetilir ve genelde 3 saat gibi bir zaman limiti bulunur.
- Alternatifler (Extensions): Saldırgan yazılım geliştirme yaklaşımında alternatif yöntemlerden de bahsetmek mümkündür. Bu yöntemler genelde yaşanan tecrübelerle göre yöntemin farklı alt uygulamaları olarak görülebilir. Başlıca alternatifler aşağıda sıralanmıştır.
 - Kayıtların Tımar edilmesi (Backlog Refinement, Grooming): Bu alternatif yaklaşımda çıkışlarda kullanılacak içerik tanımlarının (backlog) üzerinden geçilerek bu içerikler tanımlarının anlaşılabilir, ulaşılabilir olması sağlanır. Genelde bu

amaca yönelik olarak içeri tanımlarının daha alt parçalara bölünmesi, müşteri ile yeniden görüşülerek hedeflerin açık hale getirilmesi veya teknik detayların araştırılarak açıklanması gibi işlemler yapılabilir.

- **Saldırıların saldırısı (scrum of scrums):** Birden fazla ekibin aynı proje üzerinde çalışması gibi durumlarda ihtiyaç duyulan bir alternatiftir. Genelde yazılımın farklı parçalarının uyum (integration) sürecinde devreye alınır ve farklı ekiplerin birbirine göre tutum alması sağlanır. Genelde takımların birbiri ile aynı protokol üzerinde çalışıyor olması iletişim ve uyum sorunlarını çözmek için yeterli olmaktadır ancak kaynak yönetimi ve projenin metrikleri açısından senkronize çalışmaları da büyük önem taşır, bu yüzden proje ekiplerinin birbirinin zamanlamasına uygun hale getirilmesi için takımlar arası toplantılar düzenlenerek geri kalan takımın belirlenmesi, geri kalma sebeplerinin belirlenmesi ve çözüm yollarının geliştirilmesi gibi konular üzerinde durulur.

Yapı Taşları

Saldırgan yazılım geliştirme yönteminde yapı taşı olarak kullanılan bazı kavramlar aşağıdaki şekilde sıralanabilir [10]:

İş Sonu Çizelgeleri (Burn-down Charts): Zamana göre kalan işin durumunu gösteren çizelgedir. İş sonu çizelgeleri zorunluluk olmamakla birlikte sürecin şeffaflığını göstermek için kullanılabilirler.

İş Başı Çizelgeleri (Burn-up Chart): Zamana göre ölçümlerdeki artışı göstermek için kullanılan çizelgelerdir. İş sonu çizelgelerine benzer şekilde zorunlu olmamakla birlikte şeffaflık amacıyla kullanılabilir.

Günlük Saldırı (Daily Scrum): Günlük 15 dakikalık saldırı toplantılarıdır. Çıkışlar (Sprint) öncesinde geliştirme takımları ile yapılan planlamaları tutmaktadır. Güncellemeler saldırı kayıtlarında (Scrum Backlogs) tutulmaktadır.

Yapılan Tanımı (Definition of Done): Geliştirme takımı tarafından yönetilen ve projenin paydaşları tarafından aynı şeylerin anlaşılmasını sağlamak için kullanılan ortak bir sözlüktür.

Geliştirme Takımı (Development Team): Projenin geliştirilmesi, teslimler ve arttırmalardan sorumlu, bunları yöneten ve organize eden takımdır.

Ortaya Çıkarma (Emergence): Bir vakanın bir bilgiyi

ortaya çıkarması sürecidir. Yaşanan bir deneyim sonucunda bilgi elde edilmesidir. Farklı bir bakış açısında göre bilginin bir olay sonunda fark edilmesi olarak da görülebilir.

Empiryonizm (Empiricism): Deneyselciliktir ve bir kararın gözlem, tecrübe ve deneylere dayandırılmasıdır. 3 temel sütun üzerine inşa edilir ve bunlar şeffaflık (transparency), kontrol (inspection) ve uyarlamadır (adaptation).

Mühendislik standartları: Yazılımın kullanılabilir arttırmaları için geliştirme takımı tarafından uygulanan her türlü teknoloji standardını ifade eder.

Tahmin (Forecast): Çıkışlarda (sprint) kullanılmak üzere ürün kayıtlarından (product backlog) geliştirme takımının çıkardığı fonksiyonel beklentileri ifade eder.

Artırım (increment): Daha önceden üretilmiş yazılıma eklentiye ifade eder. Anlık olarak ürünün tamamı, o ana kadar yapılan arttırmaları ifade eder.

Ürün kaydı (Product Backlog): Bir yazılımın üretilmesi, sürdürülebilmesi ve bakımı için gerekli işlerin tutulduğu kayıttır. Ürün sahibi tarafından yönetilir.

Ürün kaydı arıtma (Product Backlog Refinement): Ürün kaydı üzerindeki çıkış konusu olacak parçanın ürün sahibi ve geliştirme ekibinin ortak kararıyla seçilmesi işlemidir.

Ürün Sahibi (Product Owner): Bir ürünün değerini azami seviyeye çıkarmak ve özellikle ürünün yönetimi ve iş süreçleri ve fonksiyonel beklentilerin geliştirme ekibine doğru aktarılmasından sorumludur.

Hazır durumu (Ready) : Çıkış planına geçirilen ve ürün kaydında bulunan tanımların ürün sahibi ve geliştirme takımı tarafından ortaklaşa olarak hazır olduğunun anlaşılması durumudur.

Hız (Velocity) : Genelde saldırgan yazılım geliştirme sürecinin gerekli adımlarından birisi olmamakla birlikte çoğu süreçte kullanılmaktadır ve ürün kayıtlarından (Product Backlog) arttırmalarla birer çıkışa dönüşme ve geliştirme takımı tarafından ele alınması ve saldırı ekibi tarafından takip edilmesi şekilden tanımlanabilecek sürecin tekrar hızını (frekansını) belirtmektedir.

Paydaş (Stakeholder): Basitçe saldırgan yazılım geliştirme sürecinden etkilenen herkesi ifade etmektedir ancak geliştirme ekibi (development team), ürün sahibi (product owner), saldırı ustası (scrum master) gibi bazı kişi ve gruplara özel olarak isim verildiği için bu grupların dışında kalan kişiler için kullanılmaktadır. Çoğu zaman da yazılımı kullanacak veya yazılımın etkisini hissedecek müşteri veya şirket çalışanlarını ifade eder.

Çıkış değerlendirilmesi (Sprint review): Çıkışların değerlendirilmesi için (örneğin aylık olarak) zaman kısıtlaması bulunan (örneğin 4 saatlik) toplantıların ismidir.

Çıkışta yapılan ürünlerdeki değişimlerin genel resme etkisinin incelendiği ve ürün kayıtlarının güncellendiği toplantılar olarak düşünülebilir.

Çıkışa geri bakışlar (Sprint retrospective): Genelde 3 saat ile sınırlı tutulan ve çıkışların sonunda yapılan toplantılardır. Bir çıkış sürecinin değerlendirilmesi yapılır ve gelecek çıkıştaki geliştirmeler planlanır.

Çıkış Planlama (Sprint Planning): Bir çıkışın başlamasından önceki planlama aşamasıdır ve genelde 1 gün ile sınırlıdır. Saldırı takımı (scrum team) tarafından ürün geçmişi (product backlog) üzerinde yapılan bir çalışma ile bir sonraki çıkışta (sprint) yapılması planlanan değişikliklerin üzerinden geçilir.

Çıkış Hedefi (Sprint Goal): Çıkışın hedefini anlatan kısa bir tanımdır. Genelde iş sürecindeki bir probleme atıfta bulunur. Çıkış hedefine ulaşmak için genelde çıkışın fonksiyonelliğinin ayarlanması gerekebilir.

Çıkış Kaydı (Sprint backlog): Çıkış hedefine ulaşmak için yapılan geliştirme çalışmalarının genel görüntüsünü sunar. Hedefe ulaşmak için gerekli olan fonksiyonellik, tahminler, ve işlerin listesi gibi içerikleri bulundurmaz. Geliştirme takımı tarafından yönetilir.

Çıkış (Sprint): Genelde 30 günlük geliştirme sürecidir. Çıkışlar genelde ardışık olarak ve aralarında boşluk bırakmadan yapılır. Bir çıkış, daha sonraki çıkışlara tecrübe ve iş başlıklarını da aktarmaktadır.

Kendini Yönetme (Self Organization): Takımların otonom olarak kendilerini yönetmesi anlamına gelir. Buradaki kendini yönetme tam özgürlük anlamından çok, takımların verilen hedefe ulaşmak için gerekli tercihleri kendilerinin yaptığı ve sınırları olan bir özgürlüktür. Takım, takım dışındakiler tarafından yönlendirilmek yerine, kendi kararlarını kendisi vererek işi tamamlamak için gereken adımları kendisi seçer.

Saldırgan Geliştirme Değerleri (Scrum Values): Saldırgan yazılım geliştirme yaklaşımının temel değerlerini içerir. Bunlar genel olarak adanmışlık, odaklanma, açıklık ve cesaret şeklinde sayılabilir.

Saldırı Takımı (Scrum Team): Kendini yöneten bir takımdır ve Ürün sahibi (product owner), geliştirme takımı (Development team) ve saldırı ustası (scrum master)'dan oluşmaktadır.

Saldırı Tahtası (Scrum Board): Genelde çıkış kayıtlarını tutmak için saldırı takımı tarafından kullanılan tahtadır. Saldırgan yazılım geliştirme sürecinde zorunlu değişimler ancak sürecin açıklığının sağlamak için Kan-ban [11] benzeri bir şekilde herkesin görebileceği bir ortamda bilginin tutulması anlamına gelmektedir.

Kaynaklar

[1] Sadi Evren SEKER (2014), Bilgi Yönetimi (Knowledge

Management), YBSAnsiklopedi, v.1, is.2, pp. 8 - 14

[2] "New Product Development Game". *Harvard Business Review* 86116:137–146, 1986. January 1, 1986. Retrieved March 12, 2013.

[3] *The Knowledge Creating Company*. Oxford University Press. 1995. p. 3. ISBN 9780199762330. Retrieved March 12, 2013.

[4] Takeuchi, Hirotaka; Nonaka, Ikujiro (January–February 1986). "The New Product Development Game" (PDF). *Harvard Business Review*. Retrieved June 9, 2010.

[5] *Scrum, scrimage* teriminin kısaltılması olarak, *Oxford English Dictionary Online*.

[6] Sutherland, Jeff (October 2004). "Agile Development: Lessons learned from the first Scrum" (PDF). Tarama 23 Ocak 2014.

[7] Sutherland, Jeffrey Victor; Schwaber, Ken (1995). *Business object design and implementation: OOPSLA '95 workshop proceedings*. The University of Michigan. p. 118. ISBN 3-540-76096-2.

[8] Schwaber, Ken; Beedle, Mike (2002). *Agile software development with Scrum*. Prentice Hall. ISBN 0-13-067634-9.

[9] Schwaber, Ken (February 1, 2004). *Agile Project Management with Scrum*. Microsoft Press. ISBN 978-0-7356-1993-7.

[10] Scrum.org, tarama 25 Ocak 2015

[11] Seker, S. E. (2014) KANBAN, YBS Ansiklopedi, v. 1, is. 3, pp. 21-25