

Değişiklik Yönetimi Süreçlerinin Tanımlanması ve Ölçülmesi

Pınar Evrensel¹

Mesut Gözütok²

Halit Oğuztüzün³

^{1,2}Havelsan A.Ş., Ankara

³Bilgisayar Mühendisliği Bölümü, ODTÜ, Ankara

¹e-posta: pinar.evrensel@havelsan.com.tr

²e-posta: mgozutok@havelsan.com.tr

³e-posta: oguztuzn@ceng.metu.edu.tr

Özetçe

Sistem/yazılım yaşam döngüsü içinde değişikliklerin uygun süreçlerle yönetilmesi yazılım kalitesinin sağlanması ve korunması için gereklidir. Bu bildiriye, dağıtık bir simülasyon sistemi geliştirilmesi bağlamında, kavramsal modelin tanımlanmasından, sistemin teslimatına kadar ve teslimattan sonra bakım/idame sırasında, yapılması gereken değişikliklerin yönetimi ve metriklerinin tutulması için tanımlanan süreçler, kullanılan araçlar ve yöntemler tartışılmaktadır. Ayrıca sistemin tamamlanmasından sonra kaydedilen verilerin analiz edilerek sistem geliştirme süreçlerinin iyileştirilmesi kapsamında değerlendirilmesi konusu irdelenmektedir.

1. Giriş

Sistem/yazılım geliştirilirken olası hataları en aza indirmek, elde edilen verilerden maksimum kazanımı sağlamak için kalite standartları mevcuttur. Ancak, standartlara göre tanımlanmış kalite süreçlerine bağlı kalınıp, gerekli tüm dokümantasyon hazırlansa ve gözden geçirmeler yapılsın, Müşteri/Kullanıcı onayları alınsa da, süreç içinde değişiklik ihtiyaçları ile karşılaşılması kaçınılmazdır. Değişiklik ihtiyaçlarına rağmen riskin minimumda tutulması için proje yaşamı boyunca karşılaşılan değişikliklerin de kontrol altında olması, değişikliğe sebep olan hataların analiz edilerek tekrarlanmaması için ölçümlerinin ve kaydının tutulması gerekmektedir.

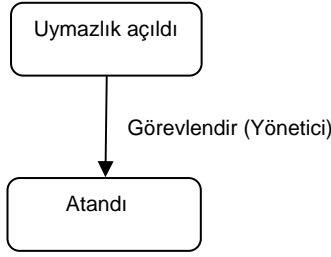
Bu bildiriye, tanımlanan değişiklik kategorileri ve örnek olarak bir kategorinin yaşam sürecinden bahsedilmektedir. Ayrıca değişiklik süreci kapsamında kaydedilen verinin analizi değerlendirilmektedir. Bildiriye konu alınan proje, aşağı yukarı 1300 sistem gereksinimi, 5000 yazılım gereksinimi ve 850,000 satır kaynak kod (C++ dilinde) içeren bir dağıtık simülasyon sistemi geliştirme projesidir. Bu çapta bir sistemde değişikliklerin sistematik bir süreç altında yapılmaması hem takvim, hem de maliyet açısından, proje ekibine ve şirkete yüksek oranlarda zarara sebep olabilir. Geliştirilen sistem büyüdükçe, tespit edilen hatalar, yapılan düzeltmeler tamamen kontrolden çıkıp, takip edilemez bir hale gelebilir. Bu sebeplerle belli bir büyüklüğün üzerindeki yazılım geliştirme projeleri kapsamında değişiklik yönetim sürecinin tanımlanması ve değişikliklerin sürece uygun olarak yönetilmesi kritiktir. Bu bildiri, değişiklik yönetim süreçleri konusunda mühendislik pratiği paylaşımını amaçlamaktadır.

2. Değişiklik Yönetimi

Sistem/Yazılım Mühendisliği kapsamında, Değişiklik Yönetimi Süreci; değişiklik talebi (İng. requesting), yapılabiliğin belirlenmesi (İng. determining attainability), planlama, gerçekleştirme (İng. implementing) ve değişikliklerin değerlendirilmesi (İng. evaluation of changes) işlemlerini kapsamaktadır [2, 3, 8, 9, 10].

Sistem geliştirme sürecinde değişiklik yönetimi ihtiyaçları ele alınmış, MIL-STD-498 [11], IEEE-12207 [12] ve CMMI [7] gibi standartlar kapsamında tanımlı olan değişiklik yönetimi süreçlerinin ihtiyaçları incelenmiş ve Unified Change Management (UCM) [6] gibi benzer süreçler incelenerek değişiklik yönetimi süreçleri tanımlanmıştır. Tanımlanan süreçler analiz amaçlı taktik hareket simülasyonunu hedefleyen bir proje özelinde uygulanmaktadır. Sistem geliştirme aşamasında ihtiyaç olan değişiklikler kapsamında yapılacak işlemlerin farklılaşması, sorumluların farklı olması ve metriklerin ayrı toplanma ihtiyacı gibi nedenlerle, değişiklikler 4 kategoriye ayrılmış ve her bir kategori için farklı süreçler tanımlanmıştır. Genel bakış itibarıyla süreçler birbirlerine benzerlik gösterse de, kategorilerine göre farklılıkları bulunmaktadır.

Değişiklik yönetim süreçlerinin tanımlanmasında kullanılacak diyagramlar araştırılmıştır. Bu kapsamda “aktivite diyagramı” (İng. activity diagram), “kulvar diyagramı” (İng. swimlane diagram) ve “durum diyagramı” (İng. state diagram) [4] incelenmiştir. Ancak hiç birinin ihtiyacımızı tam olarak karşılamadığı görülmüştür. Tanımlamaya çalıştığımız süreç durum diyagramı gibi durum tabanlı, ancak kulvar diyagramı gibi aktivitelerden sorumlu olan rollerin de tanımlanmasını gerektirmektedir. Kulvar diyagramlar kullanılarak tanımlanabilse de, süreç yönetimi bakımından odak noktası rollerden çok durumlar olduğundan, tercih edilmemiştir. Onun yerine durum geçişlerini sağlayacak aktiviteyi gerçekleştirecek olan rolleri gösterecek şekilde durum diyagramı uyarlanmıştır. Akışı gösteren ok üzerinde bulunan aktivitenin yanına aktiviteyi gerçekleştirmekten sorumlu ve gerçekleştirmeye hakkı olan rolün adı yazılmaktadır. Şekil 1’deki örnekte uymazlık Müşteri tarafından açıldıktan sonra iş akışı “Uymazlık açıldı” durumuna gelir. “Uymazlık açıldı” durumuna gelen iş akışı, Yönetici’nin uymazlığı giderecek olan Yazılım Mühendisi’ni görevlendirmesi ile “Atandı” durumuna gelir. “Görevlendir” aksiyonunu yalnız “Yönetici” haklarına sahip olan kullanıcılar yapabilecektir.



Şekil 1 – Uyarlanmış durum diyagramı

2.1. Değişiklik Kategorileri

Konfigürasyon yönetimi kapsamına alınan konfigürasyon elemanlarının (KE) sistem geliştirme sürecinin hangi aşamasında olduklarına bağlı olarak temel çizgileri (Ing. baseline) belirlenir. Örneğin gereksinim analizi tamamlanmış ve gereksinim spesifikasyonları Müşteri/Kullanıcı tarafından onaylanmış bir KE'nin "fonksiyonel temel çizgisi" belirlenmiş olur. Bundan sonra spesifikasyonlarda yapılması gereken değişikliklerin yine Müşteri/Kullanıcı onayına sunulması gerekmektedir. Ancak KE'de talep edilen değişiklik spesifikasyonların değişmesini gerektirmiyorsa Müşteri/Kullanıcı onayına gerek duyulmadan ilgili değişiklik yapılabilmektedir. Bu sebeple KE'lerde Müşteri/Kullanıcı/Proje mühendisi tarafından talep edilen değişikliklerin, KE'nin bulunduğu temel çizgiye bağlı olarak, farklı ele alınmaları ve yönetilmeleri gerekmektedir. Örneğin, bir KE'de yazılım testleri sırasında tespit edilen bir hatanın giderilmesi, eğer gereksinim ve tasarım değişikliği gerektirmiyorsa, Müşteri/Kullanıcı onayına gerek olmadan, yazılımı geliştiren kişi tarafından hatanın düzeltilmesi ve test mühendisi tarafından yeniden test edilmesi ile kapatılabilir. Ancak bulunan hatanın giderilmesi için, KE'de tasarım değişikliği yapılması gerekiyorsa, uygulanacak süreç farklı olmalıdır. Farklı süreçlerde yapılacak işler farklı olduğundan, işlerin sorumluları da farklı olabilmektedir. Projenin Sistem Mühendisliği Yönetimi kapsamında çeşitli aşamalarda karşılaşılabilecek değişiklikler dört ayrı kategoride sınıflandırılmıştır:

- Hata takibi (Uymazlık): YKE, YKE entegrasyon, sistem-alt sistem entegrasyonu ve sistem vasıflandırma testleri sırasında tespit edilen, test edilen sistemin tanımlı gereksinimlere uygunsuzluğu sebebiyle yapılması gereken değişikliklerdir. Uymazlıklar birim testlerinden başlayarak, bakım/idame aşamasında bile tespit edilebilir. Bu tip hatalar, sistemin ya da yazılımın gereksinim veya tasarımında değişiklik gerektirmemektedir.
- Değişiklik İstekleri: Kullanıcı / Müşteri isteği, yazılım geliştirme sırasında ortaya çıkan ihtiyaçlar, testler sırasında tespit edilen gerekler, vb. sebeplerle daha önceden onaylanmış kavramsal model, sistem/yazılım gereksinimleri ve/veya tasarımda değişiklik yapılması gerekebilir. Bu değişiklikler sözleşme değişikliğine de sebep olabilirler. Bu tip değişikliklerin; proje/şirket üst yönetimi, kalite sorumluları, Müşteri, Kullanıcı, sorumlu

proje mühendisleri, konfigürasyon yöneticisi, vb. kişilerden oluşan Konfigürasyon Kontrol Kurulu (KKK) tarafından onaylanması gerekmektedir. Değişiklikler gerekli onay alındıktan sonra yapılmalıdır.

- İyileştirme Önerileri: Bir hata olarak kabul edilmeyen ve/veya kavramsal model/gereksinim/tasarım değişikliği gerektirmeyen, ancak kullanım kolaylığı sağlayacak, görselliği zenginleştirecek vb. iyileştirme önerileri bu kategori kapsamında ele alınmaktadır. İyileştirme önerileri Kullanıcı/ Müşteri, proje geliştirme ekibi, test ekibi ya da yönetimden gelebilir. Bu sınıfa ait değişikliklerin, hata takibi sürecinden ayrı olarak takip edilmesinin sebebi, bu değişikliklerin sistemin kabulünü engellememesi, sistemin hatalı çalışmasına sebep olmaması, hata takibi sınıfı kapsamına giren değişikliklere göre öncelik ve önemlerinin daha az olması, ve hata sınıfındaki değişikliklerin analiz ve değerlendirmesinin ayrı olarak yapılması gerekliliğidir.
- İşlem maddeleri: Gerek Müşteri ile yapılan ve/veya proje grubu toplantılarında ortaya çıkan işlem maddeleri, gerekse şirket/grup kapsamında yapılması gereken idari/teknik her türlü aktivitenin planlanması ve takip edilmesi kapsamında bu süreç tanımlanmıştır. Bu kapsama dahil olan istekler, diğer kategorilerden farklı olarak, proje kapsamında ihtiyaç duyulan araştırma, mukayese etüdü, idari yazışmalar, vb. sistemin dokümantasyonunda ve/veya yazılımın kaynak kodunda değişikliğe sebep olmayan isteklerdir.

Farklı sınıftaki değişikliklerin metrikleri ayrı ayrı tutularak değişik analizler yapılabilmelidir. Hata takibi sınıfı kapsamındaki metrikler yazılımı geliştiren proje personelinin performans ölçümlerinde kullanılabilirken, değişiklik isteğindeki metrikler sistem analizi ve tasarımından sorumlu personelin performans değerlendirmelerinde kullanılabilir.

2.2. Süreçler

Her farklı kategori için farklı süreçler tanımlanmış ve süreçler bir değişikliklik yönetim aracı üzerinde uygulanmıştır. Sürece özel durumlar ve durum geçişleri ile ilgili sorumlu roller tanımlanmıştır. Süreç içinde ve sonrasında analiz edilmesi planlanan verilerin de kaydının tutulması sağlanmıştır. Bu bölümde Uymazlık Süreci örnek olarak anlatılmaktadır. Diğer süreçler ile ilgili detaylı bilgi bu kapsamda hazırlanan teknik rapor [5] dokümanında bulunabilir.

2.2.1. Uymazlık Süreci

YKE testleri, YKE entegrasyon testleri, sistem-alt sistem entegrasyon testleri ve sistem vasıflandırma testleri sırasında tespit edilen hatalar bu süreç ile takip edilmektedir. Hatayı tespit eden kişi değişiklik yönetim aracı üzerinde tanımlanan Uymazlık formunu kullanarak yeni bir uymazlık açar. Uymazlıklar, açan kişi tarafından ilgili YKE'nin sorumlusu olan yazılım mühendisine atanır. İlgili mühendis, kendisine atanan hata bildirimini inceler, hatayı giderir ve yine araç üzerinde bu uymazlığı "tamamlandı" durumuna getirerek teste alınmasını sağlar. Test edilen hata bildirimini, testten başarıyla geçiyorsa kapatılır, geçmiyorsa testi yapan kişi tarafından yeniden sorumlu mühendise atanır. Hata takibi ile ilgili akış şeması Şekil 2'de verilmiştir.

Akış şemasında verilen sürecin detaylı açıklamaları, hangi durumdan hangi duruma ne zaman, hangi görev ile ve kim tarafından geçilebileceği bilgileri için tablolar hazırlanmıştır. Örnek olması açısından yalnız Uymazlık Süreci ile ilgili tablo Tablo 1'de verilmiştir. Diğer süreçler için de benzer tablolar bulunmaktadır [5].

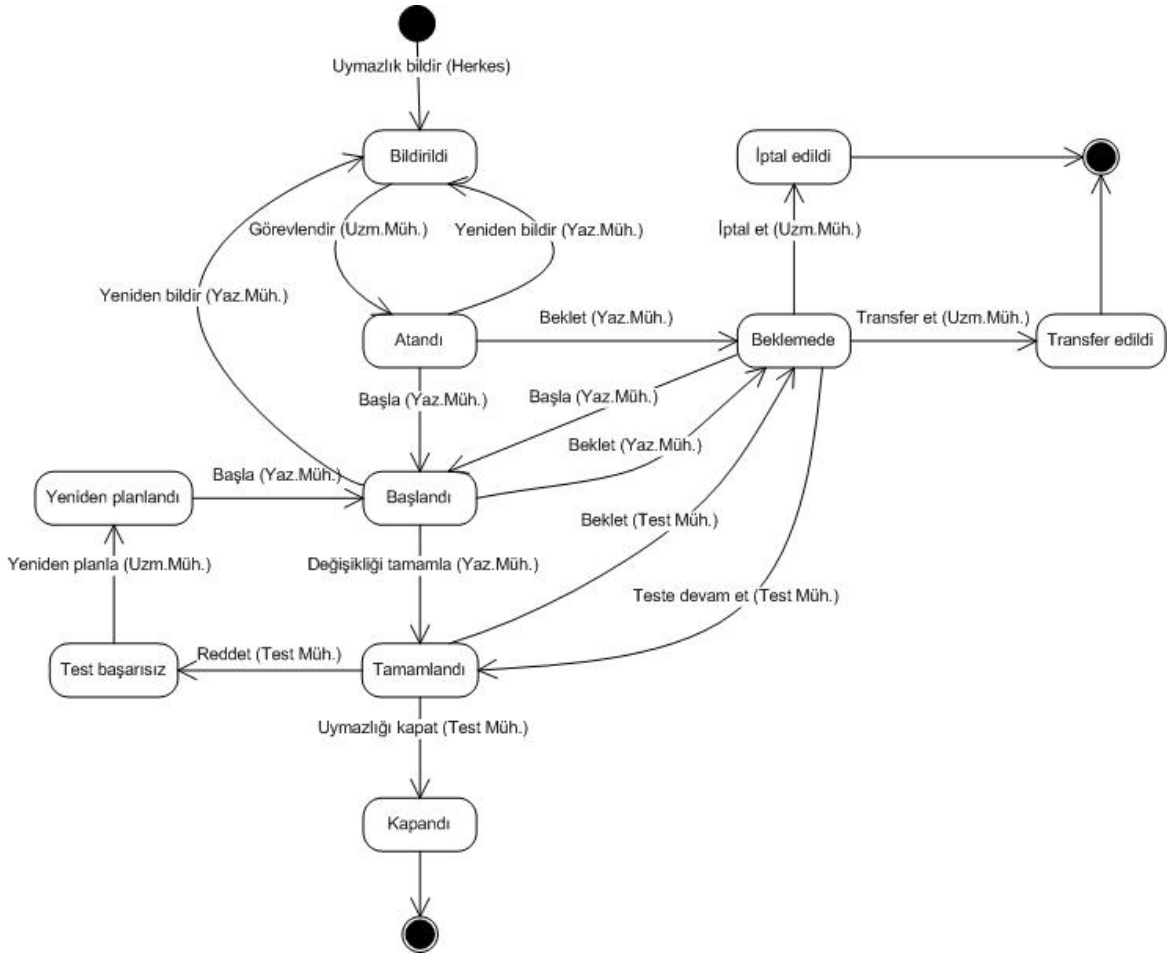
2.2.2. Verinin Analizi

Değişiklik Yönetimi kapsamında süreçlerin tanımlanması ve bir araç üzerinden takip edilmesi, sistemin geliştirilmesi

sırasında gecikmelerin ve kişiler üzerindeki iş yükünün görülebilmesi açısından önem taşımaktadır. Proje Yöneticisi devam eden değişiklik isteklerini değerlendirerek proje takvimini takip edebilmektedir.

Değişiklik süreçleri, proje devam ederken veya tamamlandıktan sonra, proje çalışanlarının performanslarının değerlendirilmesi kapsamında da kullanılmaktadır. Proje çalışanlarının performans değerlendirilmesinde kullanılan verilerden bazı örnekler aşağıda listelenmiştir:

- Planlanan zamanlamaya ne kadar uydukları ("Görevlendirme" yapılırken değişikliğin tamamlanması gereken zaman bilgisi ile işin tamamlandığı zamanın karşılaştırılması),
- Planlanan efora göre sorunların ne kadar efor harcanarak çözümlendiği ("Görevlendirme" yapılırken belirlenen değişikliğin yapılabileceği adam-saat bilgisinin, gerçekleşen adam-saat bilgisi ile karşılaştırılması),
- Personelin geliştirmekten sorumlu olduğu kısım ile ilgili gelen uymazlık sayısı
- Yapılan çözümün testten ne kadar sürede geçebildiği ("Test başarısız" durumuna kaç defa geldiği)



Şekil 2 – Hata Takibi Süreci Akış Şeması

4. ULUSAL YAZILIM MÜHENDİSLİĞİ SEMPOZYUMU - UYMS'09

Tablo 1: Uymazlık Süreci Durum Geçiş Tablosu

Hangi Durumdan:	Hangi Görev İle:	Hangi Duruma:	Kimin Tarafından:	Ne Zaman:
Bildirildi	Görevlendir	Atandı	Uzman Müh.	Belirtilen uymazlığın kimin tarafından yapılacağına karar verilip bir yaz. müh.e atanması için
Atandı	Atamayı beklet	Beklemede	Yaz.Müh.	* Eğer sistem birden fazla yapıda geliştiriliyorsa, ilgili uymazlığın daha sonraki sürümlerde yapılmasına karar verildiğinde * Değişikliğin yapılabilmesi için Kullanıcı/Müşteriden bilgi beklendiğinde
	Yeniden bildir	Bildirildi	Yaz.Müh.	* Atamanın yapıldığı kişi ilgili hatayı düzeltecek kişinin kendisi değil de, bir başkası olduğunu ve uymazlığın kendisine yanlışlıkla atandığını düşünüyorsa, hatanın ilgili kişiye yönlendirilmesi için
	Başla	Başlandı	Yaz.Müh.	Uymazlığı yapacak kişi değişikliği yapmaya başladığı zaman
Beklemede	İptal et	İptal edildi	Atamayı yapan kişi	Bekleyen uymazlığın yapılmasına gerek kalmadığı/ yapılmaması gerektiğine karar verildiği zaman
	Transfer et	Transfer edildi	Atamayı yapan kişi	Uymazlığın iyileştirme veya gereksinim/tasarım değişikliği olduğu değerlendirildiğinde, bu süreçten çıkıp Değişiklik İsteği veya İyileştirme sürecinden devam etmesi gerektiği zaman.
	Teste devam et	Tamamlandı	Test Müh.	Bir görev "Tamamlandı" olduktan sonra bir nedenle "Beklemede" durumuna getirildiyse, yeniden "Tamamlandı" durumuna geri dönmek için
	Başla	Başlandı	Yaz.Müh.	Bekleyen uymazlığın yapılmasına karar verildiğinde / zamanı geldiğinde / beklenen bilgi tamamlandığında
Başlandı	Beklet	Beklemede	Yaz.Müh.	* Daha sonraki sürümlerde yapılmasına karar verildiğinde * Değişikliğin yapılabilmesi için Kullanıcı/Müşteriden bilgi beklendiğinde
	Yeniden bildir	Bildirildi	Yaz.Müh.	İşe başladıktan sonra bu işi başkasının yapması gerektiğini düşünüyorsa, başka Yaz.Müh.e atamak için
	Değişikliği tamamla	Tamamlandı	Yaz.Müh.	Uymazlık yazılım mühendisi tarafından giderildiğinde
Test_Başarısız	Yeniden planla	Yeniden planlandı	Atamayı yapan kişi	Testte başarısız olan işe yeniden başlamak için
Yeniden planlandı	Başla	Başlandı	Yaz.Müh.	Yeniden planlanan işin yapılmasına başladığında
Tamamlandı	Reddet	Test_Başarısız	Test Müh.	Testlerde başarısız olduğu zaman
	Beklet	Beklemede	Test Müh.	Test edebilmek için bir veri beklendiğinde
	Uymazlığı kapat	Kapandı	Test Müh.	Yapılan uymazlık testlerden başarılı olarak geçtiği zaman

Projeler tamamlandıktan sonra, projeler bazında tutulan, değişikliklerin gerçekleştirilmesi ile ilgili metrikler analiz edilerek, sistem geliştirme süreçlerinin iyileştirilmesi kapsamında kullanılmaktadır. Bu kapsamda incelenen verilerden bazıları aşağıda listelenmiştir:

- Tamamlanan projelerde yapılan değişikliklerin sebeplerinin analiz edilerek diğer projelerde tekrarlanmaması için önlemler alınması

- Projede değişiklikler kapsamında harcanan zamanın analiz edilerek diğer projelerin zaman planı yapılırken değişiklikler için gerekli zamanın dikkate alınması
- Değişikliklerin gerçekleştirildiği süreler ve testlerden geçiş süreleri ile değişiklikleri gerçekleştiren personelin uzmanlıkları analiz edilerek personel deneyiminin sistemin geliştirilmesinde etkisinin belirlenmesi,
- Projeler geliştirilirken takip edilen geliştirme, test (Örn. Birim test sayısı, test araçlarının kullanılması), dokümantasyon yöntemlerine ve detay seviyeleri ile

proje süresince ortaya çıkan toplam değişiklik sayıları ve bu değişiklikleri düzeltmek için geçen sürelerin analiz edilmesi,

- Tüm metriklerin geliştirilen sistemin tipine (askeri sistem projesi, bilgi sistemi projesi, bakım projesi, vb.) bağlı olarak sonuçların farklılaşp farklılaşmadığının analizi.

2.2.3. Genel

Her üç süreç kapsamında, süreç işlemeye başladıktan sonra işin aslında farklı bir süreç ile takip edilmesi gerektiği fark edilirse; aynı iş, takip edilmesi gereken yeni süreçte tanımlanır; eski süreçte form üzerine yeni tanımlanan işin numarası girilerek "Transfer edildi" durumuna alınır. Örneğin, iyileştirme isteği kapsamında açılan bir işin aslında gereksinim değişikliği gerektirdiği fark edildiğinde bu yöntem kullanılabilir.

Ayrıca işler devam ederken, ilgili mühendis tarafından "Beklemede" durumuna alınabilir. Eğer sistem birden fazla yapıda geliştiriliyorsa ve hatanın daha sonraki yapılarda giderilmesine karar verilmesi, hatanın giderilebilmesi için kullanıcı/ müşteriden bilgi beklenmesi ya da yönetim/KKK tarafından incelenmesi gerekli görüldüğü durumlarda işler beklemeye alınabilir.

Bu süreçler bir değişiklik yönetim aracı [1] kullanılarak takip edilmektedir. Değişiklik yönetim araçları diyagramlarda verilen akışların tanımlanmasına ve gerekli verilerin toplanması için formların tasarlanmasına olanak sağlamaktadır. Proje kapsamında değişiklik yönetim aracı kullanılarak yukarıda bahsedilen süreçler tanımlanmıştır ve proje kapsamında kullanılmaktadır. Tanımlanan durum geçiş diyagramlarının değişiklik yönetim aracına entegrasyonu kapsamında yapılan faaliyetler aşağıda listelenmiştir:

- Durum geçiş diyagramlarında tanımlanan durumlar ve durumlar arası geçişleri sağlayan aksiyonlar ilgili aracın "durum-geçiş matrisi"nde (İng. state-transition matrix) tanımlanmıştır.
- Uygulamayı kullanacak olan tüm kullanıcılar ve gruplar araç üzerinde tanımlanmıştır.
- Durum geçişlerini sağlayacak aksiyonlara, tanımlanan kullanıcı grupları bazında haklar tanımlanmıştır.
- Metrikler, açıklayıcı bilgi, vb. sebeplerle doldurulması gereken formlar tasarlanmıştır. Formlar üzerinde zorunlu olan/olmayan alanlar belirlenmiş ve tanımlanmıştır. Özellikle toplanması hedeflenen metrik bilgileri kapsamında olan alanlar zorunlu alan olarak tanımlanmıştır.

3. Sonuç

Sistemlerin geliştirme ve bakım idame süreçleri kapsamında, değişik sebeplerle yapılması ihtiyacı ortaya çıkan değişikliklerin yönetilebilmesi için süreçler tanımlanmıştır. Bu süreçler bir değişiklik yönetim aracı ile takip edilmektedir. Amaç, hem değişiklikler yapılırken gecikme olup olmadığının takip edilmesi, hem de sonrasında analizlerin

yapılabilmesidir. Tutulan veriler kullanılarak proje kapsamında yapılan toplam değişiklik sayıları, değişikliklerin yapılması için harcanan toplam efor, projenin hangi aşamasında hangi tip hatalar ile ne sıklıkla karşılaşıldığı, mühendislerin değişiklikleri yaparken gösterdikleri performans, gibi raporların alınabilmesi hedeflenmektedir.

Bu bildiriye konu alınan değişiklik yönetimi süreci, eğitim ve analiz maksatlı kullanımları olan, artırılmış (İng. incremental) olarak geliştirilen dağıtık simülasyon sistemlerinin yazılım değişiklik yönetimi kapsamında tasarlanmış olmakla birlikte, benzer karmaşıklık düzeyinde, farklı alanlardaki uygulamalar için de etkili olabileceği değerlendirilmektedir. Çevik (İng. Agile) yazılım geliştirme süreçlerinde, değişiklik yönetimi yalnızca hatalar ve değişiklikler kapsamında değil, sistemin geliştirilmesi kapsamında da gerekmektedir. Bu bildiriye tanımlanan değişiklik yönetimi sürecinin çevik süreçlere özel gereksinimler kapsamında uyarlanması ileriye yönelik bir çalışma olarak düşünülmektedir.

4. Kaynakça

- [1] ClearQuest Değişiklik Yönetim Aracı, <http://www.rational.com>.
- [2] Sommerville, I., "Software Engineering", *Pearson Education Ltd.*, 8th Edition, 2007.
- [3] Stackpole, B.; Hanrion, P.; "Software Deployment, Updating, and Patching", *Auerbach Publications*, 2008.
- [4] UML 2.0, Object Management Group, <http://www.uml.org>, Temmuz 2005.
- [5] "Değişiklik Yönetim Süreçleri Uygulaması", MGKMOS-TR-01, *HAVELSAN*, Kasım 2008.
- [6] "Software Configuration Management: A Clear Case for IBM Rational ClearCase and ClearQuest UCM", *IBM Redbooks publication*, Aralık 2004.
- [7] Capability Maturity Model Integration (CMMI), Version 1.2, Software Engineering Institute, Carnegie Mellon University, Ağustos 2006.
- [8] Keyes, J., "Software Configuration Management", *Auerbach Publications*, 2004.
- [9] Berczuk, S.P., Appleton, B., "Software Configuration Management Patterns", *Addison Wesley*, 2003.
- [10] Buckley, F.J., "Implementing Configuration Management: Hardware, Software and Firmware", *IEEE Computer Society Press*, 1996.
- [11] MIL-STD-498, United States Department of Defense, Kasım 1994.
- [12] IEEE/EIA 12207.0, "Standard for Information Technology-Software Life Cycle Processes", IEEE.