

Yazılım Ürün Hattı Projelerinde Kalite Maliyeti ve Hata Yoğunluğu Ölçüm Yaklaşımı

Sezen Erdem¹, Elif Ceylan²

¹ASELSAN A.Ş. SST-KKYTM P.K.1 06172, Yenimahalle/Ankara, Türkiye

²ASELSAN A.Ş. SST-KKYTM P.K.1 06172, Yenimahalle/Ankara, Türkiye

¹erdem@aselsan.com.tr, ²eerkul@aselsan.com.tr

Özet. Yazılım Kalite Maliyeti ve Hata Yoğunluğu yazılım geliştirme projelerinde etkinlik ve verimlilik ölçümleri için sıklıkla kullanılan yöntemlerdir. Kalite maliyetleri en uygun seviyeye çekilirken hata yoğunluğunun da en az seviyeye indirilmesi yazılım projelerinin başarısı için kritik olan zaman, maliyet, kalite dengesinin kurulması açısından önem taşımaktadır. Bu sebeple yazılım geliştirme projelerinde kalite maliyeti ve hata yoğunluğu değerlerinin ölçülmesi, değerlendirilmesi ve kontrol altına alınması gerekmektedir. Yazılım Ürün Hattı gibi yeniden kullanımın etkin olarak devrede olduğu projelerde kalite maliyeti ve hata yoğunluğu değerlerinin ölçüm yöntemlerinde özelleşmeye gitme ihtiyacı oluşmaktadır. Ölçümler yapılırken farklı projelerde kullanılan ortak varlıkların geliştirme döngüleri, doğrulama yöntemleri, ürün hazırlama prosedürleri gibi Yazılım Ürün Hattı yaklaşımının getirdiği farklılıklar dikkate alınmalıdır. Bu makalede Teknik Ateş Destek Sistemleri yazılımlarının yeniden kullanım temelli üretilmesi için geliştirilen TADES Yazılım Ürün Hattı projesi kapsamında geliştirilen yeniden kullanım varlıklarının bir araya getirildiği yazılım projelerinde kalite maliyeti ve hata yoğunluğu değerlerinin ölçümü için belirlenen yöntem anlatılmaktadır. Ayrıca, alınan ölçümlerin otomasyonunu sağlamak amacı ile TADES Yazılım Ürün Hattı Sürüm Otomasyon aracı olan ReleaseMaker yazılımına yapılan eklentiler ve işçilik bilgilerinin toplandığı Microsoft Team Foundation Server üzerinde yapılan iş planlama yaklaşımına ait detaylar verilmektedir.

Anahtar Kelimeler: Yazılım Kalite Maliyeti, Hata Yoğunluğu, Yazılım Ürün Hattı

1 Giriş

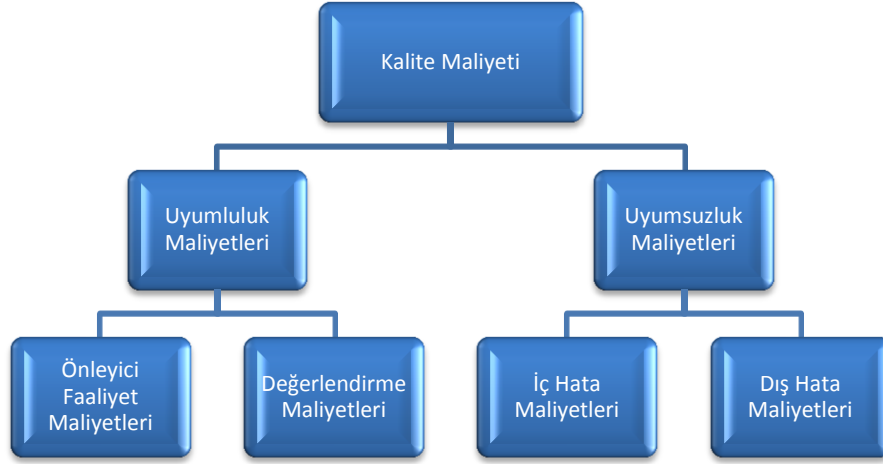
Bilişim teknolojilerindeki hızlı gelişimin kaçınılmaz sonucu olarak yazılımlar hayatın her alanına dahil olmuş durumdadır. Yazılımlar, insanoğlunun ihtiyaçlarını karşılamak için basit uygulamalar olmaktan çıkıp bir hayli karmaşık yapılar haline gelmiştir. Yazılım geliştirme organizasyonlarının varlıklarını sürdürdürebilmeleri için gelişen teknolojilere ve ihtiyaçlara uygun projeler geliştirebilmeleri gerekmektedir. Günümüzün rekabetçi ortamı göz önüne alındığında yazılım geliştirme organizasyonlarının zaman, bütçe, kalite dengesini doğru kurmaları yazılım

projelerinin başarısı için çok büyük önem taşımaktadır. Bu kritik dengenin doğru kurulması için mevcut kaynakların etkin ve verimli kullanılması gerekmektedir. Yazılım Kalite Maliyeti ve Hata Yoğunluğu yazılım geliştirme projelerinde etkinlik ve verimlilik ölçümleri için sıklıkla kullanılan yöntemlerdir. Kalite maliyetleri en uygun seviyeye çekilirken hata yoğunluğunun da en az seviyeye indirilmesi yazılım projelerinin başarısı için kritik olan zaman, maliyet, kalite dengesinin kurulması açısından önem taşımaktadır. Bu sebeple yazılım geliştirme projelerinde kalite maliyeti ve hata yoğunluğu değerlerinin ölçülmesi, değerlendirilmesi ve kontrol altına alınması gerekmektedir. Ancak Yazılım Ürün Hattı gibi yeniden kullanımın etkin olarak devrede olduğu projelerde kalite maliyeti ve hata yoğunluğu değerlerinin ölçüm yöntemlerinde özelleşmeye gitme ihtiyacı oluşmaktadır. Ölçümler yapılırken farklı projelerde kullanılan ortak varlıkların geliştirme döngüleri, doğrulama yöntemleri, ürün hazırlama prosedürleri gibi Yazılım Ürün Hattı yaklaşımının getirdiği farklılıklar dikkate alınmalıdır. Bu çalışmada ASELSAN Savunma Sistem Teknolojileri Sektör Başkanlığı bünyesindeki Komuta Kontrol Yazılım Tasarım Bölümü'nde yürütülen ve Teknik Ateş Destek Sistemleri yazılımlarının yeniden kullanım temelli üretilmesi çalışması olan TADES Yazılım Ürün Hattı projesi kapsamında geliştirilen yazılım projelerinde kalite maliyeti ve hata yoğunluğu değerlerinin ölçümü için belirlenen yöntem anlatılmaktadır. Ayrıca, alınan ölçümlerin otomasyonunu sağlamak amacı ile TADES Yazılım Ürün Hattı Sürüm Otomasyon aracı olan ReleaseMaker yazılımına yapılan eklentiler ve işçilik bilgilerinin toplandığı Microsoft Team Foundation Server üzerinde yapılan iş planlama yaklaşımına ait detaylar verilmektedir.

Makalenin 2. bölümünde Yazılım Kalite Maliyeti ve Hata Yoğunluğu ölçüm yöntemleri, 3. bölümünde Yazılım Ürün Hattı yaklaşımında Kalite Maliyeti ve Hata Yoğunluğu ölçümlerinde yaşanabilecek sorunlar ve 4. bölümde de ortaya konulan çözüm ve uygulamalarından bahsedilecektir.

2 Yazılım Kalite Maliyeti ve Hata Yoğunluğu

Üretim alanında yaygın olarak kullanımı olan Kalite Maliyeti kavramının temelleri 1950'li yıllara dayanmaktadır. Juran, kalite maliyetini "kaçınılabilecek maliyetler ya da madendeki altın" olarak tanımlamıştır [1]. Kalite maliyeti bir işi ilk seferde tam ve hatasız yapamamanın maliyeti olarak değerlendirilebilir. Crosby, "Quality is Free" kitabında kalite maliyetlerini Uyumluluk Maliyetleri ve Uyumsuzluk Maliyetleri olmak üzere iki kategoriye ayırmıştır [2]. Uyumluluk Maliyetleri, gereksinimlere uygun ürün oluşturabilmek için yapılan önleyici ve değerlendirme maliyetlerini, Uyumsuzluk Maliyetleri ise oluşan iç ve dış hataların giderilmesine ait maliyetleri içermektedir (Şekil-1).



Şekil 1. Kalite Maliyeti Kategorileri

Toplam kalite maliyeti önleyici faaliyet, değerlendirme ve hata maliyetlerinin toplanması ile elde edilmektedir. Kalite maliyetleri süreç izleme, yönetim ve iyileştirme faaliyetlerinde sıklıkla kullanılan bir metriktir [1][2].

Üretim alanında yaygın kullanımı olan kalite maliyetlerinin yazılım geliştirme alanında da kullanımı değerlendirilmiş ve yazılım kalite maliyetlerinin hesaplanması ve değerlendirilmesi üzerine çalışmalar yapılmıştır [3][4]. Yazılım kalite maliyetlerini oluşturan temel kalemler kısaca şu şekilde tanımlanabilir [5].

- **Önleyici Faaliyet Maliyetleri:** Yazılım geliştirme sürecinin tüm aşamalarında hataların oluşmasını önlemeye yönelik olarak geliştirme aşamasında yapılan faaliyetleri içerir. Kalite planlarının hazırlanması, yazılım test planlarının hazırlanması, eğitimlerin alınması, prototip hazırlama faaliyetleri önleyici faaliyetler kapsamında değerlendirilebilir.
- **Değerlendirme Maliyetleri:** Yazılım kalitesinin değerlendirilmesine yönelik faaliyetleri içerir. Gözden geçirme, birim test, doğrulama ve geçerli kılma faaliyetleri değerlendirme maliyeti olarak sayılmaktadır.
- **İç Hata Maliyetleri:** Yazılım geliştirme ve doğrulama aşamalarında tespit edilen hataların giderilmesine yönelik harcanan efor maliyetidir.
- **Dış Hata Maliyetleri:** Kullanıcı tarafından tespit edilen hataların giderilmesine yönelik olarak harcanan efor maliyetidir.

Knox üretim alanı için geliştirilen kalite maliyeti modellerini yazılım geliştirme alanına uyarlamış ve organizasyonların olgunluk seviyeleri ile kalite maliyetleri arasındaki ilişkileri ortaya koymuştur [6]. Yazılım kalite maliyetleri proje performans ölçümlerinde ve süreç iyileştirme çalışmalarında yaygın olarak kullanılmaktadır [7][8].

Yazılım kalitesine yönelik bir başka ölçüm de hata yoğunluğudur. Hata yoğunluğu yazılımda bulunan hata sayılarının yazılım büyüklüğüne oranı olarak tanımlanmaktadır. Yazılım büyüklüğü olarak kod satır sayısı ya da yazılım işlevsel büyüklük değeri kullanılabilir. Hata yoğunluğu, tespit edilen hataların önem derecelerine, gerçekleşme ve tespit dönemlerine göre sınıflandırılabilir. Yazılım kalitesinin artırılması ve müşteri memnuniyetinin yükseltilmesi için hata yoğunluğu ölçümü önemli bir metriktir. Altı sigma ve benzeri kalite yaklaşımları hata yoğunluğunu en alt seviyelere çekme adına organizasyonlara yönlendirici adımlar sunmaktadır.

Organizasyonların Yazılım Kalite Maliyeti ve Hata Yoğunluğu ölçümlerini birlikte almaları ve analiz etmeleri yazılım kalitesinden taviz vermeden kalite maliyetlerini düşürmelerini sağlamaktadır. Bu açıdan bu iki ölçümün birlikte yapılması organizasyonların kaynaklarını hem daha verimli kullanmaları hem de daha kaliteli ürünler çıkarmaları konusunda yol gösterici olacaktır.

3 Yazılım Ürün Hattı Yaklaşımında Yazılım Kalite Maliyeti ve Hata Yoğunluğu Ölçümleri

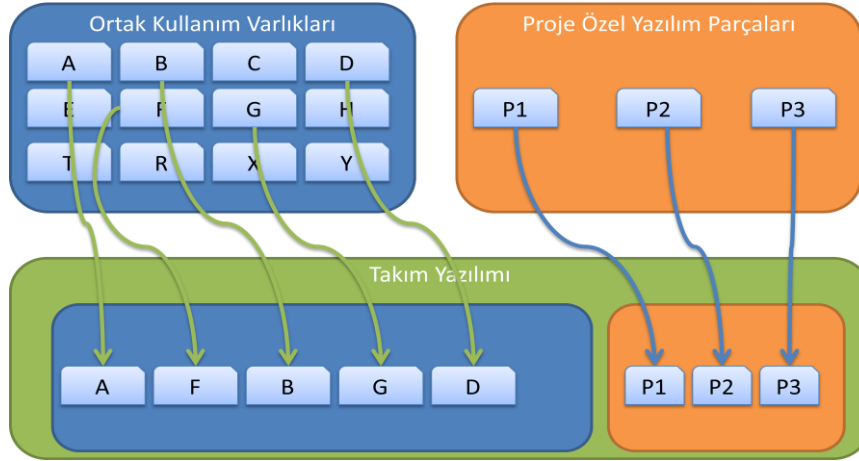
Yazılım Ürün Hattı, belirli bir pazarın özel ihtiyaçlarını karşılamak üzere, ortak ve yönetilen bir özellik grubunu destekleyen çekirdek bir varlık kümesinden, önceden tanımlanmış bir yöntem ile geliştirilen yazılım yoğun sistemlerdir[9]. Yazılım Ürün Hattı, yazılım ürünlerinin, ortak varlıkların doğrudan, değiştirilerek ya da ayarlanarak kullanılması yoluyla üretilmesi prensibine dayandığı için yeniden kullanılabilirliği üst düzeye çıkartmaktadır. Yeniden kullanımı en üst seviyelere çıkaran Yazılım Ürün Hattı yaklaşımı konfigürasyon kontrolü, değişiklik yönetimi gibi bir takım zorlukları da beraberinde getirmektedir. Bu zorluklardan kaynaklı olarak Yazılım Ürün Hattı yaklaşımının kullanıldığı projelerde Yazılım Kalite Maliyeti ve Hata Yoğunluğu gibi ölçümler için Yazılım Ürün Hattı yaklaşımının doğasına uygun yöntemler belirlemek gerekmektedir.

Ortak kullanım varlıklarının bir ürün ailesinin ihtiyaçlarını karşılamak adına birden çok proje kapsamında kullanımının olması sebebi ile kalite maliyeti ölçümlerinin hangi seviyede yapılacağı konusunda bir yöntem belirlemek gerekmektedir. Ortak kullanım varlıklarına yönelik yapılan geliştirme faaliyetlerinin proje kalite maliyetleri hesaplarına nasıl yansıtılacağı, ortak kullanım varlıklarında yapılan güncellemeler sonucunda ürünlere yansıyan değişikliklerinin kalite maliyetine etkileri, ortak kullanım varlıklarının konfigürasyon kontrolü ve sürüm yönetimi konuları değerlendirmeye alınmalıdır. Hata yoğunluğu ölçümlerinde ise ortak kullanım varlıklarında tespit edilen hataların ürün hata yoğunluğu değerlerine etkisi, hata yoğunluğu hesaplamalarında yazılım büyüklüğü değerinin bulunmasında yeniden kullanımın dikkate alınması gibi konular göz önünde bulundurulması gereken hususlar olarak görünmektedir. Ortaya konulacak olan ölçüm yönteminin Yazılım Ürün Hattı yaklaşımının getirdiği farklılıkları dikkate alacak şekilde geliştirilmesi gerekmektedir.

4 TADES'te Yazılım Kalite Maliyeti ve Hata Yoğunluğu Ölçümleri

TADES, ASELSAN Savunma Sistem Teknolojileri Sektör Başkanlığı bünyesindeki Komuta Kontrol Yazılım Tasarım Bölümü'nde yürütülen Teknik Ateş Destek Sistemleri yazılımlarının yeniden kullanım temelli üretilmesi çalışmasıdır[10]. TADES kapsamında yapılan alan analizi sonucunda 30 adet ortak kullanım varlığı geliştirilmiştir. Geliştirilen ortak varlıklar kullanılarak TADES Yazılım Ürün Hattı'ndan bugüne kadar 9 farklı proje kapsamında 19 yazılım ürünü geliştirilmiştir. CMMI Seviye 3 belgelendirmesinin ardından ASELSAN Savunma Sistem Teknolojileri grubu bünyesindeki Yazılım Tasarım Bölümleri'nde yazılım geliştirme alanında CMMI yüksek olgunluk seviyelerine yönelik olarak çalışmalar başlatılmıştır. Bu çalışma kapsamında belirlenen hedefler sonucunda Yazılım Kalite Maliyeti ve Hata Yoğunluğu ölçümleri istatistiksel kontrol altına alınmaya çalışılmaktadır. TADES Yazılım Ürün Hattı kullanılan projelerde yazılım kalite maliyeti ve hata yoğunluğu ölçümlerinin alınmasında 3. bölümde bahsedilen konulara çözüm bulunması amacı ile bir ölçüm yöntemi belirleme çalışması yapılmıştır.

TADES Yazılım Ürün Hattı yaklaşımı, organizasyonun mevcut süreçlerinde tanımlı olan Takım Yazılım kavramını kullanmaktadır. Takım yazılımı, ortak kullanım varlıkları ile projeye özel geliştirilen yazılım parçalarını uygun konfigürasyon ayarları ile bir araya getirerek müşteriye teslim edilecek olan nihai yazılım ürününü oluşturmaktadır (Şekil-2).



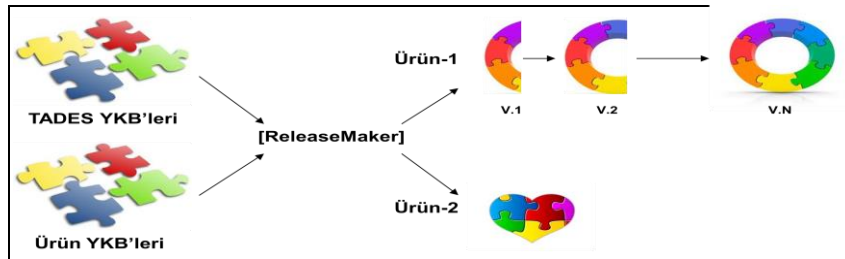
Şekil 2. Takım Yazılımı

Takım yazılımı oluşturan yazılım parçalarının her biri için yazılım geliştirme süreci işletilmekte ve her bir parçanın bağımsız doğrulama faaliyetleri yürütülmektedir. Doğrulama faaliyetleri tamamlanan yazılım parçaları bir araya getirilerek nihai yazılım ürünü olan takım yazılımı oluşturulduktan sonra doğrulama faaliyetlerine takım yazılımı ile devam edilerek son ürün hazır hale getirilmektedir. Yazılım

parçalarının bir araya getirilmesi için TADES yazılım geliştirme ekibi tarafından geliştirilen ReleaseMaker: Yazılım Ürün Hattı Sürüm Otomasyon Aracı [11] kullanılmaktadır (Şekil-3, Şekil-4). ReleaseMaker aracı projeler, projeler kapsamında geliştirilen takım yazılımları ve takım yazılımlarına ait sürüm bilgilerini içermenin yanı sıra her bir takım yazılım sürümünde kullanılan ortak kullanım varlıklarının ve proje özel yazılım parçalarının sürümleri arasındaki ilişkiyi de tutmaktadır. Örneğin Şekil 3'te AA projesi kapsamında geliştirilen ürünlerden birisi olan BAD-AA Takım Yazılımı'nın 05.00.0002.201211.130000 sürümünü oluşturan yazılım parçaları ve sürüm bilgileri gösterilmiştir. Kullanılan yazılım parçalarından bir tanesi bu projeye özel geliştirilmiş, diğer yazılım parçaları ise TADES Yazılım Ürün Hattı kapsamında geliştirilen ortak kullanım varlıkları kütüphanesinden kullanılmıştır.

YKB Name	YKB Version No	ArtifactNo	YKBType
a	00.30.0002.231011.140000		TADES
e	00.26.0002.231011.140000		TADES
i	00.20.0002.231011.140000		TADES
j	00.14.0001.130511.164500-(AC)		TADES
l	00.12.0002.231011.140000		TADES
m	00.23.0002.231011.140000		TADES
n	00.21.0001.231011.140000		TADES
o	00.19.0001.140611.160000		TADES
r	00.27.0001.050911.080000		TADES
x	00.02.0002.231011.140000		TADES
v	00.02.0002.311011.080000		PRODUCT
y	00.21.0001.050911.080000		TADES
z	00.24.0002.231011.140000		TADES
zz	00.17.0002.231011.140000		TADES
ff	00.14.0002.231011.140000		TADES
kk	00.26.0002.231011.140000		TADES
oo	00.13.0002.231011.140000		TADES
uu	00.24.0002.231011.140000		TADES
jj	00.13.0001.180711.130000		TADES

Şekil 3. ReleaseMaker Sürüm Otomasyon Aracı



Şekil 4. ReleaseMaker ile Takım Yazılım Hazırlama

TADES Yazılım Ürün Hattı yaklaşımında Yazılım Kalite Maliyeti ve Hata Yoğunluğu ölçümleri için belirlenen yöntem, takım yazılım ve ortak varlıkların Microsoft Team Foundation Server (MS TFS) aracında ölçümlere uygun şekilde tanımlanması ve ReleaseMaker aracına ölçüm otomasyonu adına yetenekler eklenmesi ile hayata geçirilmiştir.

4.1 Yazılım Kalite Maliyeti Ölçümü

Yazılım Tasarım Bölümleri, yazılım projeleri işçilik verilerini toplamak için MS TFS aracını kullanmaktadır. Projeler MS TFS aracında tanımlanmakta ve yazılım geliştirme ekipleri iş kırılım ağaçlarını MS TFS üzerinde tanımlayarak projelerinin takibini yapmaktadır. Yazılım geliştirme ekipleri planlamalarını TFS üzerinde “task” yapılarını kullanarak yapmaktadır. TFS üzerinde tanımlanan her bir task için “Geliştirme”, “Değerlendirme”, “Önleyici Faaliyet”, “İç Hata” ve “Dış Hata” değerlerinden birisinin seçildiği kalite maliyeti alanı bulunmaktadır (Şekil-5). MS TFS’ten toplanan işçilik verileri kullanılarak Yazılım Kalite Maliyeti hesaplanmaktadır. Elde edilen kalite maliyeti değerleri proje performans takip çizelgelerine yansıtılmakta ve kalite maliyetine etki eden faktörler değerlendirilerek sonuçlar süreç iyileştirme faaliyetlerinde kullanılmaktadır.

New Task 1: Field 'Title' cannot be empty. × Close

Copy template URL

Tags Add...

<Enter title here>

STATUS	CLASSIFICATION	EFFORT (HOURS)	SCHEDULE
Assigned To: Sezen ERDEM	Area: YMM2015	Original Estimate	Start Date
State: Proposed	Iteration: YMM2015\CMMI HM	Remaining work	Finish Date
Reason: New	Discipline:	Completed work	PLANNING
	Component:	Quality Cost	Priority: 2

DESCRIPTION IMPLEMENTATION OTHER HISTORY ATTACHMENTS ALL LINKS DOORS LINK DOORS TFS SCHEDULING

DISCUSSION ONLY ALL CHANGES

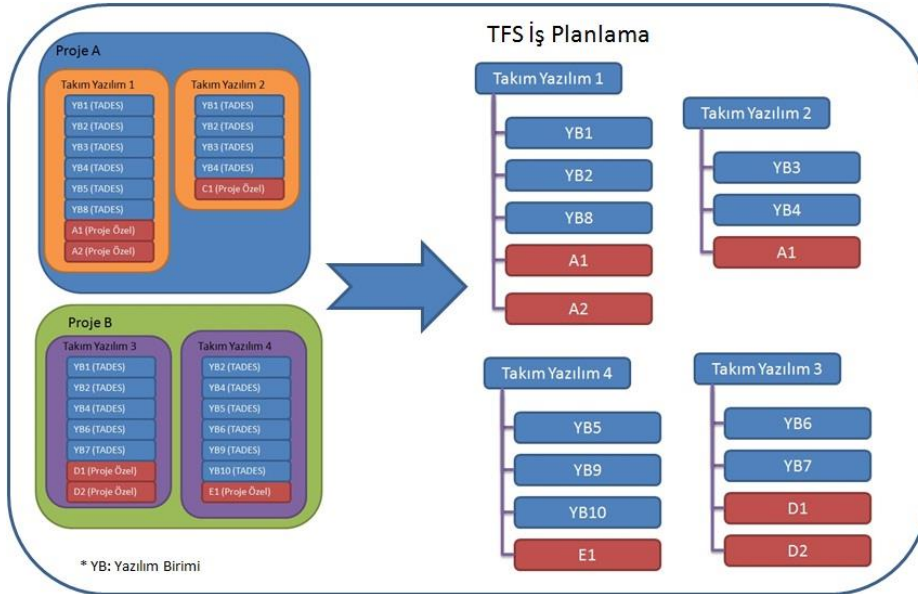
Save Save and close Cancel

Şekil 5. MS TFS İş Tanımlama

Ortak kullanım varlıklarının ayrı ayrı kalite maliyetlerinin hesaplanması projeler içerisinde geliştirilen yazılım ürünlerinin kalite maliyeti analizlerinde kullanım zorlukları yaratmaktadır. Tanımlanacak bir yöntem ile ortak kullanım varlıklarının kalite maliyetlerinin kendi başlarına değerlendirilmesi yerine proje ve ürün kalite maliyet analizlerine dahil edilebilmesi gerekmektedir. Ayrıca ortak kullanım

varlıklarının kalite maliyet değerleri ortak kullanım varlıklarının bir araya getirilerek takım yazılım oluşturulması ve bu aşamadan sonra yürütülen doğrulama faaliyetleri efor bilgilerini içermektedir. Bu sorunları aşmak ve ortak kullanım varlıkları ve proje özel geliştirilen yazılım parçalarının kalite maliyetlerinin proje kalite maliyeti analizlerine dahil edilebilmesi için “Sponsor Proje” ve “Sponsor Takım Yazılımı” yaklaşımı belirlenmiştir.

TADES Yazılım Ürün Hattının kullanıldığı projelerin MS TFS iş dağılım ağacı proje kapsamında geliştirilecek olan Takım Yazılımları içerecek şekilde oluşturulur. TADES Yazılım Ürün Hattı projesinde yıllık periyotlarla potansiyel projelerden ortak kullanım varlıklarına yönelik istekler toplanmaktadır. Toplanan isteklerin değerlendirilmesi sonucunda geliştirilmesine ya da güncellenmesine karar verilen ortak kullanım varlıklarında yapılacak işler MS TFS üzerinde tanımlanan iş dağılım ağacında ortak kullanım varlığına en fazla istek gönderen proje altındaki takım yazılımlarında tutulacak şekilde tanımlanır. Ortak kullanım varlığı için bu proje “Sponsor Proje” ve bu proje kapsamında üretilen takım yazılımları da “Sponsor Takım Yazılımı” olarak adlandırılır. Ortak kullanım varlıklarına harcanan efor bilgileri bu şekilde takım yazılım kalite maliyet hesaplarına dahil edilmiş olur.



Şekil 6. TFS İş Dağılım Ağacı Yapısı

Şekil 6’da örnek bir iş kırılım ağacı yapısının oluşturulma mantığı gösterilmektedir. Şekilde farklı projeler kapsamında geliştirilen 4 ayrı takım yazılım ürününün yapısı gösterilmektedir. A ve B projeleri sponsor projeler ve bu projeler altındaki takım yazılımları da altlarında tanımlanan ortak kullanım varlıkları için potansiyel sponsor takım yazılımlarıdır. Takım yazılımları TFS iş ağacına aktarılırken bir ortak kullanım varlığını en fazla kullanan takım yazılım bu ortak kullanım varlığı

için sponsor takım yazılım olarak seçilir. A projesi altındaki TY1 takım yazılımı YB1, YB2 ve YB8 TADES ortak kullanım varlıkları için sponsor takım yazılım olup, bu ortak kullanım varlıkları için oluşturulacak tüm TFS “task”ları TY1 iş dalı altına tanımlanacaktır ve harcanan tüm efor verisi TY1 takım yazılımı kalite maliyeti analizlerine dahil edilecektir. TY1 takım yazılımı sponsorluğuna giren TADES ortak kullanım varlıklarında diğer takım yazılımlar ya da projeler kapsamında da güncellemeler yapılabilir. Ancak bu güncelleme faaliyetlerinin efor verileri de TY1 takım yazılımı kalite maliyet değerlerine yansıtılır. Bu açıdan bakıldığında iş kırılım ağacı oluşturulurken ortak kullanım varlığı için en fazla güncelleme isteği sağlayan proje ve takım yazılımın sponsor olarak seçilmesi analizlerin doğruluğu açısından önem taşımaktadır.

4.2 Hata Yoğunluğu Ölçümü

Ortak kullanım varlıklarının geliştirilmesi ve güncellenmesi sırasında mevcut yazılım geliştirme süreci adımları izlenmektedir. Her bir ortak kullanım varlığının kendine ait geliştirme ve doğrulama adımları bulunmaktadır. Takım yazılım oluşturma aşamasında doğrulama faaliyetleri tamamlanmış olan ortak kullanım varlıklarının sürümleri kullanılmaktadır. Ortak kullanım varlıklarının doğrulama aşamalarındaki hata kayıtları da sürece uygun şekilde kayıt altına alınmakta ve hata yoğunluğu analizleri ortak kullanım varlıklarının özelinde de yapılabilmektedir. Ancak takım yazılımlar seviyesinden bakıldığında doğrulama faaliyetleri ortak kullanım varlıklarının doğrulanması ile tamamlanmış olmamaktadır. Takım yazılımları kullanılarak yapılan entegrasyon ve sistem testleri ile doğrulama faaliyetlerine devam edilmektedir. Bu yüzden hem ortak kullanım varlıklarının özelindeki doğrulama faaliyetlerinde ortaya çıkan hataları hem de takım yazılım seviyesindeki hataları hata yoğunluğu ölçümlerine dahil edebilmek için bir yöntem belirlenmiş ve bu yöntemin işletilmesini sağlamak için ReleaseMaker aracında geliştirmeler yapılmıştır.

Ortak kullanım varlıklarının kullanıldığı takım yazılımlar birbirlerinden bağımsız ve farklı zamanlarda test edilebilir. Bu şekilde farklı takım yazılımlar içerisinde ortak kullanım varlıklarının işlevlerine yönelik testler gerçekleştirilmiş olmaktadır. Bu testlerden çıkan hatalar ortak kullanım varlıklarından, proje özel geliştirilen yazılım parçalarından ya da bu parçaların bir araya getirildiği takım yazılım konfigürasyonlarından kaynaklı olabilir. Bir takım yazılımında tespit edilen ortak kullanım varlığındaki hata, ortak kullanım varlığının bu sürümünün kullanıldığı tüm takım yazılımlarında da tespit edilmiş olur ve takım yazılımların hata yoğunluğu hesaplarına otomatik olarak dahil edilir.

Hata yoğunluğu ölçümünün otomasyonunu sağlamak amacı ile ReleaseMaker sürüm otomasyon aracına eklenti geliştirilmiştir. Geliştirilen eklenti hata takip aracı olarak kullanılan ClearQuest veri tabanında hata kaynaklarından (ortak kullanım varlığı + proje özel yazılım parçaları + takım yazılımı) gelebilecek hataları sorgular. Takım Yazılım seviyesinde tespit edilen hatalar ClearQuest aracına hatanın tespit edildiği takım yazılımı ve sürüm bilgisi kullanılarak girilir. Girilen hatalar yazılım geliştirme ekibi tarafından analiz edilerek hatanın kaynağı olan ortak kullanım

Hata yoğunluğu hesaplaması için gerekli olan bir diğer veri olan kaynak kod satır sayısı olarak takım yazılımların toplam kaynak kod satır sayısının bulunması için bir yöntem belirlenmiştir. Belirlenen yöntem ile takım yazılım kaynak kod satır sayısı olarak ortak kullanım varlıklarının kaynak kod satır sayıları ile proje özel geliştirilen yazılım parçalarının kaynak kod satır sayılarının toplamı kullanılmaktadır. Bu amaçla ReleaseMaker aracına belirli bir takım yazılım sürümünün kaynak kod satır sayısının ölçülmesine yönelik bir eklenti yapılmıştır. Bu eklenti takım yazılımı oluşturan ortak kullanım varlıklarının ve projeye özel geliştirilen yazılım parçalarının ilgili sürümlerine ait kaynak kodlarını konfigürasyon kontrolü kaynak kod saklama alanından yerel bir dizine aktarır ve Source Monitor kod analiz aracını kullanarak kaynak kodların mantıksal kod satır sayısının elde edilmesini sağlar.

Elde edilen hata sayısı ve kaynak kod satır sayısı değerleri kullanılarak takım yazılım sürümlerine ait hata yoğunluğu değerleri elde edilmiş olur.

5 Sonuç

Yazılım Kalite Maliyeti ve Hata Yoğunluğu ölçümleri organizasyonlara zaman – kalite – maliyet dengesinin doğru kurulabilmesi adına yol gösterici metriklerdir. Yazılım Ürün Hattı yaklaşımının kullanıldığı, yeniden kullanımın yoğun olduğu yazılım projelerinde yazılım kalite maliyeti ve hata yoğunluğu ölçümlerinin nasıl yapılacağı konusunda dikkat edilmesi gereken bazı noktalar bulunmaktadır. Bu çalışmada ASELSAN Savunma Sistem Teknolojileri Sektör Başkanlığı bünyesindeki Komuta Kontrol Yazılım Tasarım Bölümü'nde yürütülen TADES Yazılım Ürün Hattı çalışmasının kullanıldığı yazılım geliştirme projelerinde yazılım kalite maliyeti ve hata yoğunluğu ölçümlerinin yapılması için belirlenen yöntem ve bu yöntem kullanılarak alınan ölçümlerin otomasyonunun sağlanması adına yapılan çalışmalar anlatılmıştır. Bu çalışmada anlatılan yöntemin Yazılım Ürün Hattı yaklaşımı ile geliştirilen projelerin yazılım kalite maliyeti ve hata yoğunluğu ölçümlerinde kullanılabileceği değerlendirilmektedir

Kaynaklar

1. Juran, J.M., "Quality Control Handbook". 1st ed. 1951, New York, NY: McGraw-Hill
2. Crosby, P.B., "Quality is Free". 1979, New York, NY: McGraw-Hill
3. Houston D. "Cost of Software Quality: Selling Software Process Improvement to Managers". Software Quality Journal,
4. Mandewille W.A., "Software Cost of Quality", IEEE Journal on Selected Areas in Communications, Vol.8, No:2, 1990, February.
5. Demirors, O.; Yildiz, O.; Selcuk Guceglioglu, A., "Using cost of software quality for a process improvement initiative," Euromicro Conference 2000. Proceedings of the 26th , vol.2, no., pp.286,291 vol.2, 2000
6. Knox, S.T. "Modeling the Cost of Software Quality", Digital Technical Journal, Vol.5, NO:4, pp 9-16, 1993
7. Dion, Raymond. "Process Improvement and the Corporate Balance Sheet", IEEE Software (July 1993) 28-35.

8. Price Waterhouse.. "Software Quality Standards: The Costs and Benefits. A review for the Department of Trade and Industry" London: Price Waterhouse Management Consultancy, 1988
9. Clements P., Northrop L., "Software Product Lines Practices and Patterns", Addison Wesley, 2002 ISBN: 0201703327
10. Barak E., Erdem S., Yılmaz H., "TADES: Komuta Kontrol Alanında bir Yazılım Ürün Hattı Çalışması", Ulusal Yazılım Mühendisliği Konferansı (UYMK), 2010
11. Barak E., Erdem S., Topçu H. H., "RELEASEMAKER: Yazılım Ürün Hattı Sürüm Otomasyon Aracı", Savunma Teknolojileri Sempozyumu (SAVTEK), 2012