



**T.C.
BAHÇEŞEHİR ÜNİVERSİTESİ**

**TÜRKİYE’DE YAZILIM MÜHENDİSLİĞİ VE
YAZILIM YÖNETİMİ UYGULAMALARI**

Yüksek Lisans Tezi

MERİÇ MERİH AYKOL

İSTANBUL, 2009



**T.C.
BAHÇEŞEHİR ÜNİVERSİTESİ**

**FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİ TEKNOLOJİLERİ PROGRAMI**

TÜRKİYE’DE YAZILIM MÜHENDİSLİĞİ VE YAZILIM YÖNETİMİ UYGULAMALARI

Yüksek Lisans Tezi

MERİÇ MERİH AYKOL

Tez Danışmanı: YRD. DOÇ.DR. ORHAN GÖKÇÖL

Ek Danışman: YRD. DOÇ.DR. MELİH KIRLIDOĞ

İSTANBUL, 2009

T.C
BAHÇEŞEHİR ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİ TEKNOLOJİLERİ PROGRAMI

Tezin Adı: Türkiye’de Yazılım Mühendisliği ve Yazılım Yönetimi Uygulamaları

Öğrencinin Adı Soyadı: Meriç Merih Aykol

Tez Savunma Tarihi: 22.01.2009

Bu tezin Yüksek Lisans tezi olarak gerekli şartları yerine getirmiş olduğu Enstitümüz tarafından onaylanmıştır.

Prof. Dr. A. Bülent ÖZGÜLER
Enstitü Müdürü
İmza

Bu tezin Yüksek Lisans tezi olarak gerekli şartları yerine getirmiş olduğunu onaylarım.

Yrd. Doç.Dr. Orhan GÖKÇÖL
Program Koordinatörü
İmza

Bu Tez tarafımızca okunmuş, nitelik ve içerik açısından bir Yüksek Lisans tezi olarak yeterli görülmüş ve kabul edilmiştir.

Jüri Üyeleri

İmzalar

Yrd. Doç. Dr. Orhan Gökçöl

Yrd. Doç. Dr. Melih Kırılıdoğ

Prof. Dr. Oya Kalıpsız

ÖZET

TÜRKİYE’DE YAZILIM MÜHENDİSLİĞİ VE YAZILIM YÖNETİMİ UYGULAMALARI

Aykol, Meriç Merih

Bilgi Teknolojileri

Tez Danışmanı: Yrd. Doç.Dr. Orhan Gökçöl

Ek Danışman: Yrd. Doç.Dr. Melih Kırılıdoğ

Ocak 2009, 163 Sayfa

Bu tez çalışması ile hedeflenen, bir anket çalışması ile Türkiye’de yazılım mühendisliği ve yazılım yönetimi uygulamalarının gelişimini ve mevcut olgunluk düzeyini değerlendirmek; bu konuda daha önce gerçekleştirilmiş araştırmalar ile kıyaslayarak, bu alanda öncelikleri belirlemektir.

Bu çalışmayı gerçekleştirirken yazılım sektörünün son yedi yıllık gelişimini de inceleyerek yorumlayabilmek ve gelişimi nesnel bir tabana dayandırabilmek için 2001 yılında Türkiye Kalite Derneği (KalDer) Yazılımda Toplam Kalite Yönetimi Uzmanlık Grubu tarafından gerçekleştirilen “Yazılım Sektörü Anketi” baz alınmıştır.

Anket, 10 konu başlığı altında toplam 100 çoktan seçmeli soru ve 804 seçenekten oluşmaktadır. Konu başlıkları “**Yazılım Mühendisliği**” (Yazılım gereksinimleri, Yazılım tasarımı, Yazılım gerçekleştirme, Yazılım sınama, Yazılım bakımı) ve “**Yazılım Yönetimi**” (Yazılım konfigürasyon yönetimi, Yazılım mühendisliği yönetimi, Yazılım süreç mühendisliği, Yazılım araç ve yöntemleri, Yazılım kalitesi) şeklinde iki temel disiplin altında toplanarak değerlendirilmiştir. Bunun dışında katılımcı profiline yönelik altı soru, erişim bilgilerini içeren isteğe bağlı yanıtlanan iki soru ve anketin değerlendirmesine yönelik olarak bir soru içermektedir.

2001 yılında yapılan çalışmada sağlıklı veri kümesini oluşturan 50 firmaya işaret eden 68 kişinin verileri üzerinde analizler gerçekleştirilmiştir.

2008 yılında, 2001 yılında olduğu gibi, verilerin değerlendirmesi üç boyutta yapılmıştır. Bu boyutlar içinde; 1. Kesit, “Görev”, 2. kesit “Kuruluş Türü”, 3. Kesit ise “Yazılım Kadrosu Çapı” olarak seçilmiştir. Kuruluş türü içinde katılımcılar; “yazılım alt yüklenicileri, kendi yazılımını geliştiren kuruluşlar ve paket yazılım üreticileri” şeklinde üç ayrı segment altında incelenmiş, katılımcıların kuruluşlarının yazılım kadrosu çapına (1-9 kişi, 10-49 kişi, 50-99 kişi, 100 kişi ve üstünde) ve görevlerine göre analizler gerçekleştirilmiştir.

Anahtar Kelimeler: Yazılım Mühendisliği, Yazılım Yönetimi, SWEBOK, Yazılım Süreçleri, Yazılım Süreç Olgunluk Düzeyi, Yazılım Kalitesi.

ABSTRACT

SOFTWARE ENGINEERING AND SOFTWARE MANAGEMENT APPLICATIONS IN TURKEY

Aykol, Meriç Merih

Information Technologies

Supervisor: Assist. Prof. Orhan Gökçöl
Co-Supervisor: Assist. Prof. Melih Kırılıdoğ

January, 2009, 163 Pages

The objective of this thesis is to prioritise improvement actions which carry the applications of software engineering and software management in Turkey, by performing a survey, and comparing the results with previous surveys.

“KalDer Software Sector Survey – 2001” conducted by the workgroup formed by Turkish Quality Association has been basis to this thesis to interpret development of software industry in the last seven years objectively.

The survey consisted of 100 multiple-choice questions on 10 SWEBOK knowledge areas, and 804 response options in total. These ten knowledge areas that were assessed are divided into two basic disciplines – Software Engineering (software requirements, software design, software development, software testing, software maintenance) and software management (Software configuration management, software engineering management, software process engineering, software tool and methods, software quality). In addition to 100 questions, six questions for the profile of participants, and optional two questions for communication information, and one question for the assessment the survey were asked.

Responses, as in 2001, were assessed by examining responses in three dimension. The dimensions were chosen as position of respondent in the organization, type of organization, size of software development group.

Respondents’ organizations were analysed under three different segments (subcontractor software companies, enterprises which develop its own software, companies which produce packaged software). And analyses were performed based on software development and maintenance work force of the organization (1-9; 10-49, 50-99; 100+), position of the respondent in the organization.

Keywords: Software Engineering, Software Management, SWEBOK, Software Process, Software Process Maturity Level, Software Quality.

İÇİNDEKİLER

TABLolar	vii
ŞEKİLLER	ix
KISALTMALAR	xiii
1. GİRİŞ	1
1.1 PROBLEMİN TANIMI VE ÇALIŞMANIN AMACI	1
2. GENEL BİLGİLER	3
2.1 CMM VE SPICE YAKLAŞIMLARI	4
2.2 LİTERATÜRDE YER ALAN ARAŞTIRMA ÇALIŞMALARI	5
2.3 PROBLEM İFADESİ VE METODLAR	15
3. ARAŞTIRMA ÇALIŞMASI	17
3.1 ARAŞTIRMA ARAÇLARI	17
3.2 ARAŞTIRMA YÖNTEMİ	18
3.3 VERİ TOPLAMA VE DEĞERLENDİRME YAKLAŞIMI	20
3.4 VERİLERİN GÜVENİLİRLİĞİ	23
4. ANALİZ VE SONUÇLAR	25
4.1 SONUÇLAR VE GELECEKTEKİ ÇALIŞMA	25
4.2 GENEL ANALİZLER	30
4.3 SÜREÇLER BAZINDA ANALİZ SONUÇLARI	35
4.3.1 Yazılım Gereksinimleri	35
4.3.2 Yazılım Tasarımı	40
4.3.3 Yazılım Gerçekleştirme	44
4.3.4 Yazılım Sınama	48
4.3.5 Yazılım Bakımı	53
4.3.6 Yazılım Konfigürasyon Yönetimi	58
4.3.7 Yazılım Mühendisliği Yönetimi	62
4.3.8 Yazılım Süreç Mühendisliği	70
4.3.9 Yazılım Mühendisliği Araç ve Yöntemleri	77
4.3.10 Yazılım Kalitesi	83
4.4 KURULUŞ TÜRÜNE GÖRE ANALİZLER	88
4.4.1 Genel Göstergeler	89

4.4.2 Yazılım Gereksinimleri	90
4.4.3 Yazılım Tasarımı	92
4.4.4 Yazılım Gerçekleştirme.....	94
4.4.5 Yazılım Sınama	95
4.4.6 Yazılım Bakımı	97
4.4.7 Yazılım Konfigürasyon Yönetimi	102
4.4.8 Yazılım Mühendisliği Yönetimi.....	106
4.4.9 Yazılım Süreç Mühendisliği.....	111
4.4.10 Yazılım Mühendisliği Araç ve Yöntemleri	114
4.4.11 Yazılım Kalitesi	115
5. SONUÇ VE ÖNERİLER	118
5.1 GENEL DEĞERLENDİRME	118
5.1.1 Kesit-1: Görev Tanımı	119
5.1.2 Kesit-2: Kuruluş Türü	120
5.1.3 Kesit-3: Yazılım Kadrosu Çapı	121
5.2 GİRDİ ve ÇIKTI BAZINDA DEĞERLENDİRMELER	123
5.2.1 GİRDİLER.....	123
5.2.2 ÇIKTILAR.....	125
5.3 ÖNERİLER	126
EKLER	127
EK 1 - Sorular	128
EK 2 - Katılımcı Profili	145
EK 3 - Tek Yanıtlı Sorular	146
EK 4 - Anket Sorularının Güvenilirlik Analizi	147
EK 5 - Anket Verileri – Ortalama Değerler	149
EK 6 - Süreç Özetleri.....	154
KAYNAKÇA	158
ÖZGEÇMİŞ	162

TABLolar

Tablo 3.1 : SPICE (ISO 15504) süreç düzeyleri	21
Tablo 3.2 : Örnek puan hesaplama tablosu	22
Tablo 4.1 : Görev yapılan firma türü	25
Tablo 4.2 : Firmanın hitap ettiği sektör	26
Tablo 4.3 : Kuruluş çapı.....	26
Tablo 4.4 : Yazılım kadrosu çapı	26
Tablo 4.5 : Görev tanımı	27
Tablo 4.6 : Deneyim süresi	27
Tablo 4.7 : Kuruluşunuzda yazılım sına senaryoları nasıl tanımlanıyor?	96
Tablo 4.8 : Hangileri yazılım bakım işlerinin en önemli kaynaklarıdır?	98
Tablo 4.9 : Yazılım bakım türlerinden hangisi en fazla kaynağı tüketiyor?	98
Tablo 4.10 : Yazılım ürününü geliştiren personel bakımında görev alıyor mu? ...	99
Tablo 4.11 : Yazılımla ilgili sorun ve yeni gereksinimleri kim topluyor?	99
Tablo 4.12 : Sorun/gereksinim değerlendirme/önceliklendirmesinde rol alanlar	100
Tablo 4.13 : Bakımı kolaylaştırmak için hangi önlemler uygulanıyor?	100
Tablo 4.14 : Bakım aşamasında ölçümlerden hangilerini kullanıyorsunuz?	101
Tablo 4.15 : Konfigürasyon yönetimi politika, prosedür ve standartları	103
Tablo 4.16 : Denetleme türlerinden hangileri projelerinizde uygulanıyor?	104
Tablo 4.17 : Dayanak tanımlama yaklaşımlarından hangileri kullanılıyor?	104
Tablo 4.18 : Projelerin konfigürasyon belirleme aşamasında hangileri yapılır? .	105
Tablo 4.19 : Konfigürasyon değişiklik kontrolü aşamasında neler uygulanıyor?	105
Tablo 4.20 : Yazılım mühendisliği politika, prosedür ve standartları	107
Tablo 4.21 : Yazılım organizasyon yapısı ile ilgili olarak hangisi geçerlidir?	107
Tablo 4.22 : Kuruluşunuzda yazılım Proje Planında hangileri mutlaka yer alır?	108
Tablo 4.23 : Ölçme uygulamalarının benimsenmesine hangileri engel oluyor? .	109
Tablo 4.24 : Yazılım ölçümlerinden hangileri kuruluşunuzda uygulanıyor?	110
Tablo 4.25 : Yazılım süreç standartlarından hangisini temel alıyorsunuz?	112
Tablo 4.26 : Kuruluşunuzda süreçlerle yönetim konusunda hangisi en doğru? ..	112
Tablo 4.27 : Yazılım mühendisliği bilgi birikimi nasıl kurumsatırılıyor?	113

Tablo 4.28 :	Yazılım mühendisliği araçları için hangi görüşlere katılıyorsunuz?	114
Tablo 4.29 :	Yazılım mühendisliği ve yönetimi araç seçimini kim yapıyor?	114
Tablo 4.30 :	Mühendislik yöntemlerinin seçiminde hangi kriterler belirleyici?..	115
Tablo 4.31 :	Çalıştığınız birimde hangileri ürün kalite kriterlerindedir?	116
Tablo 4.32 :	Ürün güvenilirliği verilerini hangi yöntemlerle elde ediyorsunuz?...	116
Tablo 4.33 :	Ürün kalitesi verileri ile ilgili hangisi mevcut durumu yansıtıyor?.	117
Tablo 5.1 :	Göreve göre karşılaştırma.....	119
Tablo 5.2 :	Kuruluş türüne göre karşılaştırma	121
Tablo 5.3 :	Yazılım çalışanı sayısına göre karşılaştırma	122

ŞEKİLLER

Şekil 2.1 : ESI 1997 anketi - katılım	6
Şekil 3.1 : SWEBOK – ilk beş bilgi alanı	18
Şekil 3.2 : Konular ve yeterlilik düzeyleri	22
Şekil 4.1 : Anket başarı değerlendirilmesi.....	29
Şekil 4.2 : Süreçler bazında olgunluk düzeyleri – 2008.....	30
Şekil 4.3 : Kurumsallaşma oranı – 2001	31
Şekil 4.4 : Kurumsallaşma oranı – 2008	31
Şekil 4.5 : Kurumsallaşmanın kadro çapına göre dağılımı – 2001.....	32
Şekil 4.6 : Kurumsallaşmanın kadro çapına göre dağılımı – 2008.....	32
Şekil 4.7 : Kurumsallaşmanın göreve göre dağılımı – 2001	33
Şekil 4.8 : Kurumsallaşmanın göreve göre dağılımı – 2008	33
Şekil 4.9 : Kurumsallaşmanın kuruluş türüne göre dağılımı – 2001	34
Şekil 4.10 : Kurumsallaşmanın kuruluş türüne göre dağılımı – 2008.....	34
Şekil 4.11 : Gereksinim belirleme teknikleri.....	35
Şekil 4.12 : Gereksinim tanımlama teknikleri.....	36
Şekil 4.13 : Gereksinim riskleri.....	37
Şekil 4.14 : Gereksinim analizi teknikleri	37
Şekil 4.15 : Yazılım gereksinimlerinin tanımlanması	38
Şekil 4.16 : Gereksinimlerin onaylanması	39
Şekil 4.17 : Gereksinim değişim yöntemleri	39
Şekil 4.18 : Gereksinimlerin yönetimi	40
Şekil 4.19 : Yazılım tasarımı	41
Şekil 4.20 : Yazılım teknolojileri	42
Şekil 4.21 : Tasarım nitelikleri	42
Şekil 4.22 : Tasarım kalite güvence yöntemleri	43
Şekil 4.23 : Formal tasarım teknikleri	44
Şekil 4.24 : Programlama yöntemleri	45
Şekil 4.25 : Kodlama standartları	45
Şekil 4.26 : Kodlama kriterleri	46

Şekil 4.27 : Tümlleştirme stratejisi.....	46
Şekil 4.28 : Kod kalite güvence yöntemleri	47
Şekil 4.29 : Kullanıcı belgeleri.....	47
Şekil 4.30 : Yazılım sınaama mühendisleri.....	48
Şekil 4.31 : Yazılım sınaama planlaması	49
Şekil 4.32 : Sınaama senaryoları	50
Şekil 4.33 : Sınaama teknikleri	51
Şekil 4.34 : Yükleme sınaamaları	51
Şekil 4.35 : Sınaama sonuçlarının yönetimi.....	52
Şekil 4.36 : Yazılım sınaama faaliyetleri	53
Şekil 4.37 : Yazılım bakım işlerinin kaynağı	54
Şekil 4.38 : Yazılım bakım türleri	54
Şekil 4.39 : Yazılım bakım aşamaları	55
Şekil 4.40 : Bakımı yapan çalışanlar	55
Şekil 4.41 : Yazılım bakımı.....	56
Şekil 4.42 : Yazılım bakımı – sorun ve yeni gereksinimler	57
Şekil 4.43 : Yazılım ile ilgili sorun ve gereksinimlerin değeriendirilmesi	57
Şekil 4.44 : Yazılım bakımını kolaylaştırmak.....	58
Şekil 4.45 : Konfigürasyon yönetimi politika, prosedür ve standartları.....	59
Şekil 4.46 : Konfigürasyon yönetim planı.....	59
Şekil 4.47 : Konfigürasyon öğeleri.....	60
Şekil 4.48 : Konfigürasyon dayanak tanımlama yaklaşımları.....	61
Şekil 4.49 : Konfigürasyon denetleme türleri.....	61
Şekil 4.50 : Yazılım mühendisliğı politika, prosedür ve standartları.....	62
Şekil 4.51 : Yazılım organizasyon yapısı	63
Şekil 4.52 : Yazılım mühendisliğı insan kaynakları yönetimi	64
Şekil 4.53 : Yazılım mühendisliğı faaliyetleri – İletişim mekanizmaları.....	65
Şekil 4.54 : Yazılım proje rolleri.....	66
Şekil 4.55 : Yazılım proje planı.....	67
Şekil 4.56 : Yazılım ölçümleri	68
Şekil 4.57 : Yazılım ölçme uygulamalarının benimsenmesi.....	70
Şekil 4.58 : Yazılım süreç standartları	71
Şekil 4.59 : Yazılım mühendisliğı ve yönetim süreçleri	72

Şekil 4.60 : Yazılım süreçlerinin tanımlanma ve iyileştirilmesi	72
Şekil 4.61 : Süreçlerle yönetim	73
Şekil 4.62 : Yazılım mühendisliği bilgi birikimi.....	74
Şekil 4.63 : Süreçlerle yönetim	74
Şekil 4.64 : Yazılım süreçlerinin uygulanması ve iyileştirilmesi.....	75
Şekil 4.65 : Projede başarısızlık durumu.....	76
Şekil 4.66 : Yazılım süreçlerinin genel performansı	76
Şekil 4.67 : Kullanılan yazılım mühendisliği araçları	78
Şekil 4.68 : Yazılım mühendisliği araçlarının kullanım durumu	79
Şekil 4.69 : Yazılım mühendisliği araçları hakkında görüşler	79
Şekil 4.70 : Kullanılan yazılım yönetim araçları.....	80
Şekil 4.71 : Yazılım yönetim araçlarının kullanım durumu	81
Şekil 4.72 : Yazılım mühendisliği ve yönetim araçlarının seçimi	81
Şekil 4.73 : Araç seçim kriterleri.....	82
Şekil 4.74 : Yazılım yaşam döngüsü modelleri.....	83
Şekil 4.75 : Müşteri tanımı	83
Şekil 4.76 : Ürün kalite kriterleri.....	84
Şekil 4.77 : Ürünün işlevsel kalitesi	85
Şekil 4.78 : Ürünün güvenilirliği.....	86
Şekil 4.79 : Yazılımın kullanım kolaylığı	87
Şekil 4.80 : Ürün kalitesi	87
Şekil 4.81 : Ürün kalite standartları ve ürün kalitesi ölçüm modelleri.....	88
Şekil 4.82 : Gereksinim belirleme teknikleri.....	90
Şekil 4.83 : Gereksinim riskleri ve risk azaltma yöntemleri	91
Şekil 4.84 : Gereksinim değişim yönetimi	91
Şekil 4.85 : Projelerde yazılım tasarımı	92
Şekil 4.86 : Tasarım riskleri ve risk azaltma yöntemleri.....	92
Şekil 4.87 : Tasarım kalite güvence yöntemleri	93
Şekil 4.88 : Yazılım tasarımı süreç profili.....	93
Şekil 4.89 : Yazılım geliştirme süreç profili	95
Şekil 4.90 : Yazılım sınaama süreç profili	97
Şekil 4.91 : Yazılım bakımı süreç profili	102
Şekil 4.92 : Yazılım konfigürasyon yönetimi süreç profili	106

Şekil 4.93 : Yazılım mühendisliği süreç profili – 2008.....	111
Şekil 4.94 : Yazılım süreç mühendisliği süreç profili – 2008	113
Şekil 4.95 : Yazılım mühendisliği araç ve yöntemleri süreç profili – 2008.....	115
Şekil 4.96 : Yazılım kalitesi süreç profili – 2008.....	117
Şekil 5.1 : Görev tanımı	119
Şekil 5.2 : Kuruluş türü	120
Şekil 5.3 : Yazılım kadrosu çapı.....	121

KISALTMALAR

Association for Computing Machinery	: ACM
Allied Quality Assurance Publications	: AQAP
Capability Maturity Model Integration	: CMMI
European Software Institute	: ESI
International Electrotechnical Commission	: IEC
Institute of Electrical and Electronics Engineer	: IEEE
International Organization for Standardization	: ISO
Joint Technical Committee 1	: JTC1
Software Process Improvement Capability detErmination	: SPICE
Software Engineering Body of Knowledge	: SWEBOK
Software Engineering Institute	: SEI
Standard CMMI Assesment Method for Process Improvement	: SCAMPI
Sub-Committee Seven is part of the JTC 1	: SC7
Türkiye Kalite Derneği	: KalDer
Unified Modeling Language	: UML

1. GİRİŞ

Yazılım sektöründe, diğer sektörlerle oranla geliştirenlerin ürüne etkileri daha fazladır. Kişilerden kaynaklanan değişkenliği azaltmak, standart yazılım geliştirme süreçlerinin kullanılmasını gerekli kılar. Ürünün kalitesini ağırlıklı olarak onu üreten sürecin kalitesi belirlediği için, süreç odaklı kalite yaklaşımı hakimdir.

Yazılım üretiminde, verimlilik ve kaliteyi arttırmak için uygulanması gereken metodlar bulunmaktadır. Yazılımın doğası nedeniyle diğer endüstriyel ürünlerden farklı özellikler göstermesi, kalite ile ilgili genel model ve standartlarının uygulamasını güçleştirir, özel yaklaşımlar gerektirir. Yazılım geliştirme faaliyetleri yapan firmalarda süreç ve metodolojilerin kullanımı sadece avantaj değil, zorunluluk olarak da öne çıkar.

Yazılım geliştirme süreçlerini iyileştirmek isteyen kuruluşlar, bu konuda kabul görmüş standart ve modeller ile başlangıç yapmalı, sürekli iyileştirmeyi sağlamak için, uluslararası standart bir çerçeve/model kullanılmalıdır.

1.1 PROBLEMİN TANIMI VE ÇALIŞMANIN AMACI

2001 yılında KalDer Yazılım Toplam Kalite Yönetimi Uzmanlık Grubunun gerçekleştirdiği “Yazılım Sektörü Anketinin” sonuçları, süreç standart ve modellerinin yazılım sektörümüzde referans alınması, uygun altyapı/araçlarla çalışma ve eğitim eksikliklerinin giderilmesi, organizasyonlardaki uzmanlık alanlarının çeşitlendirilmesi, mühendisler için rol esaslı kariyer geliştirme modellerinin uygulanması gereklerini ortaya koymaktadır.

Yazılım projelerinde küreselleşme ve gerektirdiği rekabet koşulları gözönüne alındığında, araştırma sonuçları, standartlaşma ve evrensel metodların kullanımı gibi kritik konuları net olarak ortaya koymaktadır.

Bir tez çalışması olarak 2008 yılında düzenlenen bu anketin amacı, 2001 yılında Türkiye Kalite Derneği (KalDer) Yazılımda Toplam Kalite Yönetimi Uzmanlık Grubu

tarafından gerekleřtirilen ‘‘Yazılım Sekt6r6 Anketi’’ni tekrar ederek, T6rkiye’de Yazılım M6hendislięi ve Yazılım Y6netimi uygulamalarının son yedi yıl ierisindeki geliřimini ve mevcut olgunluk d6zeyini deęerlendirmektedir.

Olgunluk d6zeyi aynı zamanda, yazılım geliřtiren ve bakımını yapan ulusal kuruluřların, yazılım m6hendislięi ve yazılım y6netimi alanlarında uluslararası iyi uygulamaları benimseme derecelerinin de bir g6stergesi olacaktır.

2. GENEL BİLGİLER

Yazılım üretimi diğer alanlara göre oldukça farklı özellikler göstermektedir. Yazılım sektöründe diğer sektörlere kıyasla tekrar oldukça az, geliştirenlerin ürüne etkileri daha fazladır.

Yazılımda üretim teknikleri çok fazla değişken olmadığı için ürünün kalitesini ağırlıkla onu üreten sürecin kalitesi belirler. Bu nedenle, yazılım sektöründe ürün odaklıdan çok, süreç odaklı kalite yaklaşımı hakimdir. Müşteriye sağlanan ürün ve hizmet, yönetilen süreçlerin çıktılarıdır. Süreç yönetimi temelli düşünce, yazılım geliştirme faaliyetleri bulunan firmalarda metodolojilerin kullanımını avantaj ve zorunluluk olarak öne çıkarmaktadır.

Yazılım projelerini başarılı kılan üç temel öge bulunmaktadır: insan, süreç ve teknoloji. İnsan kaynağı kaçınılmaz bir şekilde çok önemli olmakla birlikte süreç en kolay gözardı edilen öğedir. Başarılı sonuçlar için hem teknoloji, hem yetkin insan kaynağı hem de iyi süreçlere ihtiyaç bulunmaktadır.

Yazılım projelerinin büyümesiyle 1970'lerde ortaya çıkan kalite sorunu 1980'li yıllarda yazılım krizi (Glass 2000) olarak anılacak düzeye ulaşmıştır. 1980'lerin ortalarından bu yana yapılan çalışmalarla, belgelendirme, süreç iyileştirme ve yetenek belirleme amacı ile farklı modeller ortaya konmuştur.

Yazılıma özgü ilk süreç modeli, Carnegie Mellon Üniversitesinde ABD Savunma Bakanlığı desteği ile kurulan Yazılım Mühendisliği Enstitüsü'nde (SEI) geliştirilen Yetenek Olgunluk Modeli, CMM olmuştur. 1990'lı yılların başlamasıyla ağırlıkla Avrupa'da olmak üzere, ISO 9000 temelli yazılım kalite sistem sertifikasyonlarında önemli artış gözlenmiştir. Modellerin çoğalması, bir üst model/çerçevenin ortaya konması girişimine yol açmış ve 1993 yılında ISO/IEC JTC1/SC7 tarafından SPICE projesi 20 ülkeden kuruluşların katılımı ile başlayarak, ilk versiyonu 1995 yılında ISO 15504 teknik raporu olarak yayınlanmıştır. 1998 yılında üçüncü versiyonu, resmi bir ISO yazılım süreç değerlendirme standardı olarak yayınlanmıştır. (Aykol, İnce 2008)

2.1 CMM VE SPICE YAKLAŞIMLARI

CMM (Software Engineering Institute 2006) ve SPICE (ISO 15504) metodolojileri, süreçlerin değerlendirilmesine odaklıdır. CMM etkin bir yazılım sürecinin anahtar elemanlarını tanımlayan bir çerçeve modeldir. Yeteneğin olgunluğa bağlı olduğunu ortaya koyar ve olgun olmayan bir süreçten, disiplinli bir sürece giden bir yol çizer. Tanımladığı beş olgunluk düzeyi, bir kuruluşun yazılım sürecinin olgunluğunu değerlendirmek ve diğer organizasyonlarla karşılaştırmak için bir ölçüm aracıdır. Belirlenen düzey, edinilecek yazılımın kalitesi ve firma ile çalışmaktan kaynaklı riskler konusunda bir göstergedir. (Aykol, İnce 2008)

SPICE süreç değerlendirmesine kuruluşun değil, uygulanan sürecin olgunluk düzeyi olarak yaklaşır. Her süreç kendi başına değerlendirilerek bir yetenek notu alır.

Yazılım sektöründe yaygın bir süreç değerlendirme modeli olan CMMI, Carnegie Mellon Üniversitesi Yazılım Mühendisliği Enstitüsü (SEI) tarafından geniş kapsamlı bir katılımın görüşleri alınarak geliştirilmiştir. CMMI, SPICE ve CMM modelleri gibi, süreç tanımlama, süreç iyileştirme, risk yönetimi, yetkinlik değerlendirmesi için rehberlik sağlar.

CMMI, yazılım üreten kuruluşların evrensel standartlara göre değerlendirilmesi için bir yol ve iyileşme çalışmaları için bir çerçeve oluşturur. Ayrıca bir yazılım kuruluşu için o kuruluşun notuna göre güven, risk ve ürün kalitesi beklentisi oluşturur.

CMMI değerlendirmeleri tüm Dünyada SEI gözetiminde yapılır. SEI, gelen raporları sonuçları itibariyle ve istatistik olarak yayınlar.

Mart 2008 tarihli süreç olgunluk profili raporuna göre (Carnegie Mellon University 2008a) katılan 2652 firmanın yüzde 72'sini ticari yazılım ürünü üreten ve iç kullanım için yazılım üreten firmalar oluşturmaktadır. Sadece yüzde 5.1 oranında savunma sanayi ve kamu görülürken, geri kalan yüzde 23 ise kamu ve savunma sanayinin alt yüklenicisi olan firmalardan oluşmaktadır.

Mart 2008’de yayınlanan SEI’nin yaptığı en son araştırmaya göre (Carnegie Mellon University 2008a) SEI’ye rapor edilmiş 2674 şirkete ait değerlendirme sonuçlarında, CMMI düzeylerinin dağılımının III. düzeyde yoğunlaştığı görülmektedir. Süreç olgunluk profili içinde yer alan firma ölçekleri değerlendirildiği zaman geçmişteki CMM ve SPICE yapısındaki modellerin sadece büyük organizasyonlara uygun ağır modeller olduğu düşüncesinin (Wilkie, McFall, Mc Caffery 2005) hatalı olduğu açıkça görülmektedir. Değerlendirmeye katılan firmaların yarısından fazlası 100 kişiden az çalışanı olan firmalardır. Diğer taraftan coğrafi dağılım incelendiği zaman CMMI modelinin süratle yaygınlaştığı söylenebilir. Son yıllarda CMMI belgesi alan şirketler aralarında Norveç, Bengaldeş, Suudi Arabistan ve Macaristan’ın da bulunduğu geniş bir ülkeler yelpazesine dağılmıştır.

Ancak standart ve modellerin uygulanıyor olması da başarıyı garantilemez (Gefen, Zviran 2006), yazılım süreç iyileştirmesi, organizasyonun stratejik planları ve iş hedefleri, yapısı, kullandığı teknoloji ve sosyal kültürü ile bağlantılıdır.

2.2 LİTERATÜRDE YER ALAN ARAŞTIRMA ÇALIŞMALARI

Yazılım konusunda ülke/ülkeler çapında olgunluk düzeyi değerlendirmeleri yapılması çok yaygın bir yaklaşım olarak görülmemektedir. Bu nedenle bu konuda yapılan çalışmaların sayısı ve ele aldıkları kapsam oldukça sınırlıdır.

Bu bölümde, alt başlıklar ile daha önce yapılan süreç bazlı olgunluk değerlendirmeleri ve sonuçları özetlenmektedir.

Avrupa Yazılım Enstitüsü - En İyi Yazılım Pratikleri Anketi

Avrupa Yazılım Enstitüsü tarafından 1997 yılında düzenlenen bir araştırma (European Software Institute 1997), 2001 yılında yapılan Yazılım Sektör Araştırması gibi, bu çalışma için de referans oluşturmaktadır. Ancak Communications of the ACM (Dutta, Vanwassenhove, Kulandaiswamy 1998) ve IEEE Software (May/June 1999) dergilerinde yayımlanan bu araştırmanın 1997 yılı sonrasında tekrarı yapılmamıştır.

KalDer 2001 anketinin katılımcı sayısı, Şekil 2.1’de görülen ESI tarafından 1997 yılında firmalara yönelik olarak düzenlenen anket ile kıyaslanırsa, 20 ülke arasında üçüncü sırada, bu raporla açıklanan 2008 yılı çalışması ise ikinci sırada yer almaktadır.

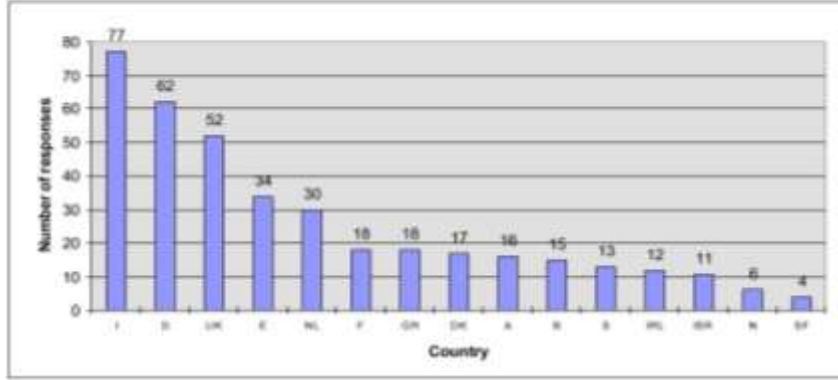


Figure 1: Number of valid responses by country

Legend

A - Austria	E - Spain	ISR - Israel	S - Sweden
B - Belgium	F - France	IRL - Ireland	SF - Finland
D - Germany	GR - Greece	N - Norway	UK - United Kingdom
DK - Denmark	I - Italy	NL - Netherlands	

Kaynak: Avrupa Yazılım Enstitüsü (ESI - European Software Institute) - 1997 Software Best Practice Questionnaire - Analysis of Results

Şekil 2.1 : ESI 1997 anketi - katılım

ESI anketinin sorularını en çok yanıtlayan ülke olan İtalya'dan 77 ve en az yanıtlayan ülke Finlandiya'dan dört veri seti alındığı değerlendirilerek bu tez kapsamında yapılan araştırmaya katılım oranının oldukça yüksek olduğu söylenebilir.

Bu tez kapsamında yapılan araştırmada, kapsam 10 konu ve 100 (ortalama 8 seçenekli) soru ile belirlenirken ESI tarafından yapılan kuruluşlara yönelik araştırma kapsamında "Evet/Hayır" türünde kapalı yanıtlar alınan 42 soru aşağıdaki kırılımlar ile yanıtlanmıştır.

- a) Organizasyonel Konular (8)
- b) Standartlar ve Prosedürler (13)
- c) Ölçümler (8)
- d) Geliştirme Sürecinin Denetimi (6)
- e) Araçlar ve Teknoloji (7)

Araştırma çalışmasında ülke ve alt sektörler göre kesitler vurgulanmıştır.

Bu tez kapsamında yapılan arařtırmada bulunan 804 seçeneęe karřılık, ESI arařtırmasında 84 seçenek yer almaktadır. Yapılan arařtırma çalıřması, ESI arařtırmasında yer alan 42 sorunun hemen hemen tümünü içeren seçeneklere sahiptir.

ESI tarafından yapılan arařtırma, yazılım yönetimi pratikleri/uygulamaları açısından arařtırmaya katılan Avrupa ülkeleri arasında önemli farklılıklar olduğunu göstermektedir. Bu konuda belirleyici olan standart ve prosedürlere uyum düzeyidir. Etkin proje yönetimi, birçok Avrupa ülkesinde anahtar bir güç unsuru olarak görülmektedir. İngiltere gibi uygulama konusunda başarı düzeyi yüksek ülkelerde, yazılım projelerinin organizasyonun hedefleri ile ilişkilendirilmesi ve paydařların planlama sürecine katılımı başarıyı arttırmaktadır. Ortak standartların kullanımı, prototip çalıřmaları ve tasarım notasyonları da başarıya yardımcı olmaktadır. En büyük uygulama farklılıkları ise; başlamadan önce projelerin risk ve yapılabilir olma deęerlendirmelerinin yapılması; kaynak, zaman ve maliyet tahmin prosedürleri; bağımsız kalite güvence fonksiyonları; devretme prosedürleri, önemli fazlar arasındaki gözden geçirme ve deęerlendirmeler; yazılım, gereksinim, tasarım ve dokümantasyon deęiřikliklerinin kontrolü gibi yönetim pratiklerinde yer almaktadır.

Kore Savunma Sanayinde Yazılım Test Olgunluęu Arařtırması

Kore savunma sanayinde test olgunluęu, test pratikleri ve yazılım geliřtirme karateristiklerini belirlemek amacıyla yapılan arařtırma çalıřmasında (Park, Ryu, Choi 2008), 38 kiřinin yanıtladıęı anketin bulguları ile savunma yazılımı organizasyonlarının karřı karřıya olduęu güçlü ve zayıf yönler ile konuya ilişkin önemli konular ortaya koyulmaktadır.

Sorular, yanıtlayanın demografik bilgileri, test olgunluęu, deneme uygulamaları ve yazılım karateristikleri olmak üzere dört bölümden oluşmaktadır. Test olgunluęunu ölçmek için tasarlanan sorular, test süreci iyileřtirme konusunun denetim noktalarından geliřtirilmiřtir.

Araştırmayı yanıtlayan 38 kişinin yüzde 73,7'si yazılım geliştirme uzmanı, yüzde 21'i proje yöneticisidir. Yanıtlayanlar bu alanda ortalama dokuz yıllık deneyime sahiptir. Organizasyonların araştırma kapsamındaki ana yazılım ürünleri, çoğunlukla (yüzde 81,57) savunma sistemlerine gömülü ve yüzde 15,57 oranında da komuta ve kontrol amaçlı yazılımlardır.

32 yanıt göz önünde bulundurularak (yüzde 84,21) yanıtlayanın organizasyonunda V-modeli en çok kullanılan test süreç modelidir. Yanıtlayanın katıldığı test tipleri çoğunlukla yüzde 36,90 oranıyla birim testi, yüzde 44,05 oranıyla entegrasyon testidir.

Organizasyonlarda yazılım testine katılanların yüzde dağılımı içinde; yazılım geliştiriciler - yüzde 27,13; geliştirme takımındaki testçiler - yüzde 5,43; projedeki test takımı - yüzde 25,58; organizasyondaki test takımı - yüzde 22,48, kendi organizasyonları dışındaki bağımsız organizasyonlar - yüzde 19,38 oranlarında yer almaktadır.

Araştırmada test tekniklerinin kullanımının etkinliği konusunda net bir bulgu elde edilememiştir. Test tekniklerini etkin olarak kullanamama nedenleri yüzde 42,86 oranla zaman ve maliyet yetersizliği; yüzde 33,33 oranla fayda elde edilememesinden kaynaklanmaktadır. Test için zaman olmaması yüzde 46,97'lik oranla yanıtlayanın karşılaştığı en yaygın problemdir. yüzde 18,18'lik oranla test kaynağı konusunda eksiklik bir diğer ana sorundur.

Yanıtlayanın deneyimlediği yazılım geliştirme başarısızlıkları; yüzde 11,63'lük oranla proje takvim ve bütçesindeki aşmalar ile yüzde 11,63'lük oranla beklentilerin karşılanamamasıdır. Bu başarısızlıklara neden olan faktörler; proje gereksinimlerinde aşırı oranda değişiklikler (yüzde 60) ve savunma kurumlarının ihtiyaç duyduğu yüksek miktardaki manuel müdahaledir (yüzde 15). Savunma yazılımları testinde odaklanılan kalite karakteristikleri; güvenilirlik, fonksiyonluluk/işlevsellik ve kullanım kolaylığıdır. Bu üç karakteristiğin her biri yüzde 88,90'lık toplam cevaplar içinde üçte birlik bir orana sahiptir.

Kore savunma sanayi yazılım endüstrisine ilişkin pilot araştırmanın sonuçları değerlendirildiğinde; öncelikle statik test teknikleri, metodolojinin kapsamı ve test süreç yönetiminin olgunluk ölçeğinin, 5'in altında olduğu belirlenmiştir. Katılım düzeyi, ölçüm göstergeleri, test araçları, özveri ve motivasyon, raporlama, test yazılımları değerlendirme ve pilot test yüksek olgunluk düzeyine sahip unsurlar olarak değerlendirilmemiştir.

Savunma yazılımı test uygulamalarında, organizasyonlar test tekniklerinin ve yeterli kaynak yönetiminin proaktif olarak daha iyi kullanımını talep etmektedir. Savunma yazılımı organizasyonları zaman ve insan kaynağı kısıtı nedeniyle savunma yazılımı sürecinde bir çok problemle karşı karşıya kalmakta ve gereksinimlerdeki değişiklik talepleri nedeniyle sorun yaşamaktadır.

Beş Asya Ülkesinde Yazılım Pratikleri - Keşif Çalışması

Hızla bilgi teknolojileri alanında dışkaynak kullanımı hizmeti sağlayan önemli bir bölge haline gelen Güneydoğu Asya'da, yazılım geliştirme alanında yayınlamış çalışma yok denecek kadar azdır. Bu araştırma (Sison ve dig. 2006) Malezya, Filipinler, Singapur, Tayland ve Vietnam'da faaliyet gösteren yazılım firmalarının yazılım uygulamalarına ilişkin bir keşif ve vaka çalışması niteliği taşımaktadır.

Çalışmanın sonuca yönelik olmaktan çok keşfe yönelik olması dört nedene dayanmaktadır. Asya ülkelerinde e-mail adresleri de dahil olmak üzere yazılım firmalarına ilişkin güvenilir veri bulmakta sorun bulunmaktadır. Ülkeler bazında ve Asya kıtası genelinde, yazılım uygulamalarının çeşitliliğine ilişkin yeterli bilgi bulunmamaktadır. Asya ülkelerindeki yazılım uygulamalarını irdelemek için hangi yazılım süreç modelini kullanmanın daha uygun olacağına karar vermek kolay değildir. Son olarak, yukarıda söz edilen çalışmaların hiçbirinin, kurumlar ve ülkeleri karşılaştırabilmek için ölçüm göstergesi olabileceği düşünülmemektedir.

Araştırmanın soru formunun birinci bölüm yanıtlayanın profili ve çalıştığı kuruma ilişkin bilgilere yönelik temel sorular, ikinci bölüm 55 sorudan oluşmaktadır. 52 soru, yanıtlayanın şirketinde yürütülen iyileştirme projelerinde yazılım uygulamalarının ne

ölçüde (hepsi, bir kısmı, hiçbiri) kullanıldığını anlamaya yöneliktir. Yazılım uygulamaları ve gruplamaları temel olarak SEI'nin SW-CMM 2. düzey olgunluk soru formu ve Microsoft ve Infosys'in bazı en iyi uygulamaları dikkate alınarak oluşturulmuştur. Geri kalan üç soru ise yanıtlayanın kendi organizasyonunun hangi süreç alanlarında daha kuvvetli ve daha zayıf olduğuna dair algısı ile akademik dünyanın mezunları hangi alanda eğitmesi yönünde görüşünü almaya yöneliktir.

Araştırma soruları beş ülkeden rassal olarak seçilen 1000 yazılım firmasına postalanmış ve Vietnam'da yazılım firmalarının listesi elde edilemediği için amaca yönelik bir örneklem oluşturulmuştur. Tüm çabalara rağmen dört ülkenin her birinden yüzde 13 mertebesinde geri dönüş alınmıştır. Konusunda uzman, nitelikli ve görel olarak düşük ücretle çalışan yazılım uzmanlarının sayısal çokluğu dikkate alındığında, bölgedeki yazılım endüstrisi hızlı bir büyüme potansiyeline sahiptir. Bu doğrultuda bölgedeki yazılım firmalarının uygulamalarına ilişkin bulgu/bilgiler çok önemlidir. Çalışma keşif amaçlı olduğu için bulgular sonuca ilişkin kesin yargılara varmaktan çok konuya ışık tutmaktadır.

Bulgular doğrultusunda, bu ülkelerde faaliyet gösteren yazılım firmaları “yazılım gereksinim yönetimi ve yazılım proje planlama” süreçlerinde, SW-CMM/CMMI'ın ikinci seviyesinde bulunan diğer süreçlere göre daha güçlüdür. Bölgede gelişmekte olan yeni yazılım firmalarının kullanımı için bölgedeki “yazılım gereksinim yönetimi ve yazılım proje planlama” alanlarındaki en iyi uygulamalar baz alınarak eğitim modülleri ve vaka çalışmaları geliştirilebilir. Bölgedeki yazılım firmalarının “proje planlama uygulamalarında tahminleme araçlarının kullanımı, gereksinim izlenebilirlik matrislerinin kullanımı, kantitatif kalite ölçü birimlerinin kullanımı, değişiklik kontrolü” konularında zayıf olabileceği düşünülmektedir. Bu uygulamaların, uygulanmamasının sebebi sadece maddi kaynak eksikliğine bağlı olmamakla beraber, her dört uygulamanın da yazılım ürün kalitesini etkilediği gözlenmektedir.

Bölgedeki yazılım firmalarının, bazı yazılım kalite güvence pratiklerinde zayıf olduğu görülmekle birlikte bu durum firmaların yazılım ürün kalitesi konusunda zayıf olduğu anlamına gelmemektedir. Bu konu, yerel müşterinin bu uygulamaların parasını

ödeyecek gücü olmaması ve benzeri nedenlerden dolayı firmanın bu alanda bilinçli olarak performans göstermemesinden kaynaklı olabileceği gibi kalite süreç tanımlaması veya uyumun zor olmasından da kaynaklı olabilmektedir.

Araştırmaya katılan firmalar arasında yazılım kalite güvence konusunda güçlü olanlar “kalite oryantasyonu, bağımsız test ekipleri ve yazılım mühendisliği süreç grupları, meslektaş gözden geçirmeleri ve kalite yönetim sistemleri” konusunda iyi uygulamalara sahiptir. Firmaların işe yeni aldığı çalışanlarının söz konusu alanlarda daha iyi eğitilmesi konusundaki taleplerinden, bölgedeki üniversite ve okullarda öğretilmekte olan yazılım kalite kavramları ve uygulamalarının geliştirilmesine ihtiyaç bulunduğu görülmektedir. Firmaların bilinen yazılım geliştirme metodları yerine çevik metodları uyguladığı görülmektedir. Bölgedeki çok sayıda küçük yazılım geliştirme firması ve çevik metodları alternatif bir yöntem olarak ortaya çıkması bilgileri ışığında Asya/Güneydoğu Asya bölgesinde yazılım geliştirmek için planlı ve dokümantasyona dayalı CMM/CMMI vb yaklaşımların uygunluğu değerlendirilmelidir. Asya’daki yazılım firmalarını CMMI-SW II. seviye ile tanıştırma ihtiyacı bulunmaktadır. Böyle bir orta seviye, bölgedeki yazılım firmalarının çok fazla maliyete yol açmaksızın, süreçlerinde kurumsallaşmasını kolaylaştırabilir.

Son olarak, firma boyutu ve yazılım kalite güvence uygulamaları arasındaki ilişki ile firmanın yazılım ihracatının boyutu ile yazılım kalite güvence uygulamaları arasında ilişkiler incelenmesi gereklidir.

Tayland’da Yazılım Süreç İyileştirme

Tayland’da son on yılda batılı yazılım süreç geliştirme modelleri ile yazılım kalitesini geliştirme çabaları olmakla birlikte, 380 şirketin sadece 17’si bilinen yazılım süreçlerini uygulayabilmiştir. Bir araştırma çalışması ile (Phongpaibul, Boehm 2005) başarısızlık sebepleri ve deneyimler analiz edilmiş, bulunan sonuç sorunun temel nedeninin kültürel farklılıklar olduğunu göstermiştir. Güncel yazılım süreç modellerinin çoğu, batı kültürlerine göre tasarlanmıştır.

Farklı toplumlardaki insan davranışını anlama çabasında, kültürel farklılıklar modeli sosyal bilimde yaygın olarak çalışılmıştır. Farklı ülkelerin vatandaşları arasında varolan düşünce ve sosyal aksiyon farklılığı 4 ana kültürel boyutta açıklanmıştır. **Güç mesafesi**, prestij, zenginlik ve gücün eşitsizlik derecesidir, farklı toplumlarda farklı ölçümlenir, ast-üst arasındaki baskınlık derecesini belirler. **Belirsizlikten kaçınma**, toplumdaki insanların belirsiz durumları tolere etme derecesidir. Belirsizlikten kaçınma derecesinin yüksek olduğu toplumlarda, belirsiz durumlardan korkulur ve resmi kurallar ile bu durumdan kaçınılır. **Bireysellik**, birey ve grup arasındaki ilişkiyi tanımlar. Bireyselliğin güçlü olduğu toplumlar kendine odaklıdır, kararlarını başarılarına dayanarak verir ve özel yaşamlarına saygı duyarlar. Kollektivizm grup yapısına dayanır, kararlar grup olarak alınır. **Dişilik ve erkeksilik** bir kültürün, insanlara, sosyal destek ve yaşam kalitesine karşılık baskınlığıdır. Erkeksi kültür, toplumdaki dişi ve erkek rollerden beklentileri ayırırken, dişi kültürlerde dişi ve erkek rolleri birbirine eşittir.

Bu çalışmada, tüm görüşme yapılanlar, Amerikan ve Tayland kültürleri arasındaki farklılıkların Tayland'daki yazılım süreç geliştirmesinde kilit başarı olduğu konusunda fikir birliğine varmıştır. Söz konusu kültürel farklılıkların daha genel göstergeleri de bulunmaktadır. Sonuç olarak kalitenin paydaş değerlerine göre tanımlanması, kalitenin süreç ya da uyum spesifikasyonuna göre tanımlanmasından daha başarılıdır.

Yazılım kalitesine ilişkin daha spesifik göstergelerden bazıları ise şöyledir:

- a) Doğruluk kontrat yönelimli kültürde önemli olmakla birlikte müşteri ve kullanıcı tatmininin daha önemli olduğu sosyalleşmiş kültürlere uyarlanamaz.
- b) Performans ve kullanılabilirlik teknik özellikleri gözden geçirerek belirlenmez, sistemlerin prototipleri ve hızlı ilk versiyonları ile belirlenir.
- c) Bakım süreci, kapsamlı dokümanlar sağlamak yerine bu süreçte yer alacak kişileri geliştirme sürecine katarak daha iyi garanti altına alınır.
- d) Küçük uygulamalar için birlikte işlerlik teknik dokümanlardan daha çok ilgili ekipler ile karşılıklı iletişimle garanti altına alınır. Büyük sistemlerde ise takım etkileşimi ve arayüz spesifikasyonlarını birlikte gerektirir.

- e) Diğer kültürlerle uygun olarak yerelleştirme farklı dillerde girdi ve çıktıların ifadesinden daha fazlasını gerektirmektedir. Örneğin pop-up pencereleri çok renkli kültürlerde istenirken, tek renkli kültürlerde arzu edilmez.

Tedarikçisi Yurtdışı Kaynaklı Geliştirme Projelerinin Risk Profili

Yapılan bir araştırma (IT Week 2004) yazılım geliştirmeyi yurtdışı tedarikçilere emanet eden her on firmadan sekizinin, yetersiz ön hazırlık ve başarısız yönetime bağlı olarak önemli problemler yaşadığını ortaya koymuştur. Yine de bilgi teknolojileri konusunda yurtdışı tedarikçi kullanımı, sağladığı ekonomik avantaj nedeniyle büyümektedir. Forrester Research araştırma şirketine göre (2007), iki yıl önce benzer organizasyonların sadece yüzde 45'i uygulama gelişimi projesi için yurtdışı tedarikçilerden yararlanırken, bugün (1000 ve üzeri çalışana sahip) Amerikalı ve Avrupalı girişimcilerin yüzde 65'i yazılım geliştirme için yurtdışı tedarikçileri kullanmakta, yüzde 13'ü ise yurtdışı tedarikçi kullanımını planlamaktadır.

Eğilimdeki artış ve yurtdışı tedarikçilerin daha etkin nasıl yönetilebileceğini anlama ihtiyacı sebebi ile bir araştırma (Iacovou, Nakatsu 2008) yapılmıştır. Risk faktörlerini belirlemek için tedarikçisi yurtdışında olan projelerin yönetimini bilen deneyimli proje yöneticileri/direktörlerinin bilgisine başvurulmuştur. Deneyimleri değerlendirilerek 57 yöneticinin içinden onbeşi panelist olmak üzere seçilmiştir. Seçilenlerin toplam 135 uluslararası proje, ortalama 17 yılın üzerinde BT ve 15 yılın üzerinde proje yönetimi deneyimi bulunmaktadır. Her biri, dokuzu uluslararası olmak üzere 51 proje yönetmiştir.

Çalışmada panelistler tarafından “önemli” ve “çok önemli” olarak değerlendirilen ilk 10 faktör irdelenmiş ve bulgular üç ana alanda gruplanmıştır;

Müşteri tedarikçi iletişimi: Bir çok uluslararası projenin, başlangıçtaki proje gereksinimlerinin değişmesi ve değişiklik taleplerinin sebep olduğu karışıklık nedeniyle eksik/yanlış iletişime yatkın olduğu görüşüdür. Dil, çevre ve diğer faktörler insanların konuları nasıl anladığını büyük ölçüde etkilemektedir. Çıktılar ne denli belirgin olursa olsun, uluslararası işbirliklerinde her zaman tahmin yürütecek kadar bir gri alan

oluşmaktadır. Yüksek risk faktörü ile değerlendirilen diğer konu proje iletişimde dil engelidir. Takım üyelerinin aynı dili konuştuğu durumlarda da, dil ve buna bağlı anlayışımız kültürel varsayımlara dayalı olduğu için ortaya çıkan yüksek orandaki yanlış anlama olasılığı sebebi ile sorun oluşmaktadır. İletişim kaynaklı son risk, değişiklik taleplerinin etkin olarak kontrol edilememesidir. Verimsiz kontroller, proje kapsamında sapmalara neden olabilmekte, iletişimle ilgili açıklanan sorunlar nedeniyle bu risk uluslararası projelerde daha büyük sorun oluşturmaktadır.

Müşterinin şirket içi yönetim konuları: Uzmanlar müşteri kökenli üç geleneksel risk faktörünü önemli bulmaktadır; Üst Yönetimin konuyu sahiplenmesinde zayıflık, yetersiz kullanıcı ilgisi ve son kullanıcı beklentilerinin yönetilmesinde güçlükler. Ek olarak tanımlanan diğer risk faktörleri ise proje yönetimi bilgisi konusundaki eksiklik ve olası tüm maliyetleri öngörme konusundaki yetersizliktir. Uluslararası olmasa da tüm teknoloji projeleri için “Üst Yönetimin konuyu sahiplenmesinde zayıflık” en yüksek risk olarak tanımlanmaktadır. Yetersiz kullanıcı ilgisi yine kritik önemde bir tehdit olarak değerlendirilmekte, kullanıcıların ilgisinin sağlanmasının sistemi kabullenme ve kullanma konusunda da önemli bir zorunluluk olduğu belirtilmektedir. Proje yönetimi bilgisi eksikliği ve son kullanıcı beklentilerinin yönetimi uluslararası olsun olmasın en önemli risk faktörleri arasında sıralanmaktadır. Uzak mesafe ve iletişimden kaynaklanan sorunlar nedeniyle beklentileri anlamak, yönlendirmek ve karşılamak oldukça zordur.

Tedarikçinin yeterlilikleri: Hizmet verilen şirkete ilişkin sektörel ve teknik bilgi eksikliği seçilen tedarikçinin vasıfları ile ilgilidir. Müşterinin ulusal düzeyde iş şartlarını bilen bir tedarikçi ile çalışılması faydalıdır. Bir tedarikçinin yüksek CMMI olgunluk düzeyine sahip olmasının, projede aktif olarak çalışacak ekibin sahip olduğu süreç olgunluğunun göstergesi olmadığı belirtilmiştir. Bazen de proje ekibinin sahip olduğu yetenek ve bilgi, tedarikçinin yönetim kadrosu tarafından yanlış yönlendirilebilmektedir. Uluslararası tedarikçinin doğasında var olan “karakutu” niteliği nedeniyle, teknik yeterliliğinin ölçümünde zorluk bulunmaktadır.

Sonuç olarak, uluslararası projeler doğaları gereği riskli taahhütlerdir. Bu tür projeler geleneksel projelerdeki risklerin yanısıra uluslararası çevreye özgü risklerin

kombinasyonu ile karşı karşıyadır. Temel proje yönetimi becerileri bu projeleri yürütmek için de gerekli olmakla beraber, uluslararası projeleri yönetmek için tek başına yetersizdir. Şirketler uluslararası projeleri yönetmek için uzaktan yönetime özgü zorluk ve fırsatları dikkate almadıkça, girişimleri, artan düzeyde yüksek başarısızlık riskinin tehdidi altında olacaktır.

2.3 PROBLEM İFADESİ VE METODLAR

2001 yılında 50 yerli şirketi temsil eden 68 kişinin katıldığı “Yazılım Sektörü Anketinin” sonuçlarına göre, ülkemizde yazılım projelerinin profesyonel bir şekilde planlanıp yönetilmediği ortaya çıkmıştır. Yine anket sonuçlarına göre yazılım üreten firmalarda süreç yaklaşımının olmadığı, özellikle iç kullanım amaçlı yazılım geliştiren firmaların genellikle hiçbir standardı temel almadığı görülmüştür. Kuruluşlarda yazılım mühendisliği bilgi birikiminin ağırlıklı olarak “usta-çırak” ilişkisi ile kurumsallaştırılmakta, eğitim önemli bir sorun olarak karşımıza çıkmaktadır.

Anketi yanıtlayan kişilerin beyanlarına göre, yazılım yönetimi yeterince araç desteği ile yapılmamakta, ölçme ve iş akış sistemlerinin bulunmaması ve yönetim zaafı bulunmaktadır.

Yazılım konusunda süreçlere verilen önem, eldeki kaynaklarla orantılıdır. Yazılım tasarımı yeterli kaynak varsa “tasarımcı”, kaynak yoksa “en yetkin proje elemanı” olarak genellikle Proje Yöneticisi tarafından yapılmaktadır. Projelerde yazılım, Proje Yöneticisi tarafından tümleştirilmektedir. Yazılım mühendisliği ve yönetim araçlarının seçimi, çoğunlukla Proje Yöneticisi tarafından yapılmaktadır. Araçları kullanacak mühendisler seçimde etkili olmadığı gibi teknik danışmanlar araç seçimine katılmamaktadır. Seçimlerde kurul kararı sınırlı orandadır, kurul/kural olmaması bu konuda kurumsallaşma olmadığının göstergesidir.

Yazılım proje planlarının; yüzde 52’inde proje süre/emek tahminleri, yüzde 28’inde proje hedefleri, yüzde 25’inde proje görev/zaman çizelgesi yer almamaktadır.

Risk yönetimi süreci; firmaların yüzde 25'inde tanımlı ve risk yönetimi araçlarının kullanım oranı sadece yüzde 8 oranındadır. Yazılım proje planları içinde; proje riskleri ve risk yönetim metodu yüzde 30 oranında yer almaktadır.

2001 yılının 2. çeyreğinde bir araştırma grubu tarafından gerçekleştirilen (Kulik, Weber 2001) dünya çapında Xerox, Motorola, Nasa gibi kuruluşların da arasında yer aldığı 268 yazılım firması üzerinde yapılan "Risk Yönetimi" uygulamaları konusundaki araştırmanın sonuçlarına göre; katılımcı firmaların yüzde 97'sinde riskleri tanımlamak ve değerlendirmek için prosedürler bulunmakta, yüzde 80'inde problemler tahmin edilmekte ve önlem alınmakta, yüzde 60'ında sürprizler önlenmektedir. Katılan kuruluşların yüzde 64'ünde proje yönetimi ofisi bulunmakta, yüzde 68 oranında proje yönetim süreçleri tanımlı, yüzde 70 oranında yazılım geliştirme süreçleri tanımlıdır.

2001 yılında yapılan araştırmanın sonuçları; yazılım projelerinde küreselleşme ve gerektirdiği rekabet koşulları gözönüne alındığında, standartlaşma ve evrensel metodların kullanımı konusunda üç kritik boyutu net olarak ortaya koymaktadır:

1. Yazılım üreten firmaların konuya gereken önemi vermesi ve gerekli kaynağı sağlayarak yazılım projelerinin tanımlı metod/standartlar ile geliştirilmesi gereklidir.
2. Üniversitelerin eğitim kapsamlarını, yazılım mühendisliği kapsamı ile kıyaslayarak gözden geçirme gereksinimi ve yazılım geliştirme konusunda tanımlı metod/standartların, eğitim kapsamında yer alması gereklidir.
3. Yazılım alımı yapan firmaların bilgi ve bilinç düzeyleri yükseltilerek, yazılım/yazılım firması seçiminde doğru tercihleri yapabilmelerine olanak sağlanması gereklidir.

3. ARAŞTIRMA ÇALIŞMASI

3.1 ARAŞTIRMA ARAÇLARI

Bu çalışma için 2001 yılında Türkiye Kalite Derneği (KalDer) Yazılımda Toplam Kalite Yönetimi Uzmanlık Grubu tarafından gerçekleştirilen “Yazılım Sektörü Anketi”nin soruları güncel hale getirilerek Türkiye Kalite Derneği (KalDer) izni ile kullanılmıştır.

Anket, SWEBOK’un 10 ana konusunda uygulamaları sorgulayan 10’ar adet çoktan seçmeli soru içeren alt anketlerden oluşmaktadır (Bkz. EK 1):

- A01. Yazılım gereksinimleri
- A02. Yazılım tasarımı
- A03. Yazılım gerçekleştirme
- A04. Yazılım sınama
- A05. Yazılım bakımı
- A06. Yazılım konfigürasyon yönetimi
- A07. Yazılım mühendisliği yönetimi
- A08. Yazılım mühendisliği araç ve yöntemleri
- A09. Yazılım süreç mühendisliği
- A10. Yazılım kalitesi

Soruların geneli çok seçenekli olup, tek seçenekli soru ve kategoriler EK 3’de verilmiştir (Bkz. EK 3). Katılımcılar anketin tümünü yanıtlayabileceği gibi, sadece seçtikleri konuların sorularını da yanıtlayabilmektedir.

Anketin İçeriği

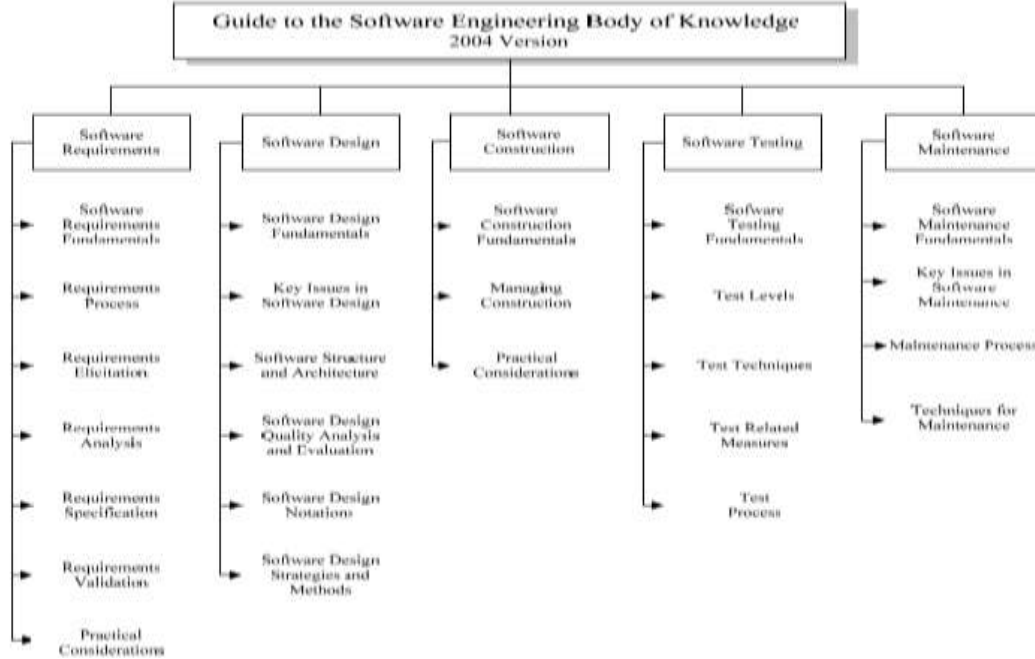
- a) Konular:
 - i. 10 konu
 - ii. 100 soru - 804 seçenek
- b) Katılımcı profili: 6 soru
- c) Erişim bilgileri (isteğe bağlı)
- d) Başarı değerlendirme: 1 soru

3.2 ARAŞTIRMA YÖNTEMİ

Referans model: Anket soruları IEEE ve ACM'in önderliğinde ve uluslararası katılımı geliştirilmekte olan Yazılım Mühendisliği Bilgi Dağarı Rehberi (*Guide to the Software Engineering Body of Knowledge - 2004*, www.swebok.org) kapsamında tanımlanan 10 konuyu ve bu konuların altbaşlıklarını temel almaktadır. Anket, SWEBOK'un ulusal çapta bir anket için kullanımına dünyada ilk örneği oluşturduğundan, 2001 yılında yapılan ilk araştırma bu girişimin yönetimi tarafından da ilgi ile izlenmiştir.

Yazılım mühendisliği disiplininin kapsam ve sınırlarını tanımlamak, yazılım mühendisliği literatürüne erişimi kolaylaştırmak, yazılım mühendisliği mesleğine tutarlı ve evrensel bir görünüm kazandırmak, ders programları ve sertifikasyon için temel oluşturmak, rehber oluşturma çalışmasının temel amaçları arasındadır. IEEE-CS/ACM Koordinasyon Komitesi Çalışma Grupları; Yazılım Mühendisliği Mesleki Ahlak Kuralları, Yazılım Mühendisliği Bilgi Dağarı, Yazılım Mühendisliği Eğitimi, Yazılım Mühendisleri için Performans Standartları şeklindedir.

Şekil 3.1'de ilk beşi örnek olarak görüldüğü gibi, SWEBOK on bilgi alanı tanımlamıştır.



Kaynak: Guide to the Software Engineering Body of Knowledge (SWEBOK), www.swebok.org, IEEE and ACM, 2004

Şekil 3.1 : SWEBOK – ilk beş bilgi alanı

Katılımcı kitlesi: Anket, yazılım mühendisliği ve yönetimi faaliyetlerinde görev alan çalışanlara yöneliktir ve anonim olarak uygulanmıştır. Katılımin 70 - 100 kişi arasında olacağı tahmin edilmiş, 75 kişiden yanıt alınmıştır. Yanıt alınan kişilerden 64'ü kimlik ve e_posta bilgilerini açıklamış, 11 kişi ise gizli bırakmıştır. Kimliğin yüzde 85 oranında biliniyor olması, beyanların mevcut durumdan daha iyi olması riskini beraberinde getirmekle birlikte, veri kalitesi açısından önem taşımaktadır.

Uygulama yöntemi: Marmara Üniversitesi'nin web ortamında araştırmacıların kullanımına açık olan anket sistemi aracılığı ile veri toplanmıştır. (<http://anket.marmara.edu.tr/v2/survey.php?sid=187>)

Anketin dili: Ankette kullanılan yazılım mühendisliği terimlerinin Türkçe karşılıkları için 2000 yılında Türkiye Bilişim Vakfı Standartlar Kurulu çalışmaları bağlamında hazırlanan *Yazılım Kalite Güvence Terimleri Sözlüğü V1.11* temel başvuru kaynağı olarak kullanılmıştır.

Anketin uygulanması: Anket, 14 Eylül – 10 Kasım 2008 tarihleri arasında web ortamında uygulanmış, toplanan veriler Ocak 2009 tarihine kadar değerlendirilmiştir.

Anket Cevaplama Süresi İstatistikleri:

Ortalama Tamamlama Süresi: 31dk 19sn (Min: 0dk 32sn, Max: 148dk 48sn)

Çıkıştan Önceki Ortalama Süre: 7dk 51sn

Anket değerlendirme yaklaşımı: Anket verilerinin değerlendirilmesinde aşağıdaki yaklaşımlar kullanılmıştır:

1. Genel yanıt dağılımı
2. Katılımcı demografik özellikleri bazında yanıt dağılımı
3. Odak konularda çapraz yanıt inceleme
4. 2001 yılı verileri ile kıyaslama ve değerlendirme
5. Uzmanlık komitesi ile birlikte sonuçların e-posta aracılığı ile gözden geçirilmesi

3.3 VERİ TOPLAMA VE DEĞERLENDİRME YAKLAŞIMI

Veriler üç boyutta incelenerek değerlendirilmiştir. Bu boyutlar içinde; 1. Kesit, “görev tanımı”, 2. kesit “kuruluş türü”, 3. Kesit ise “yazılım kadrosu çapı” olarak seçilmiştir.

Kesit-1: Görev Tanımı	2001	2008
Yazılım mühendisi	N=22	N=20
Kalite veya süreç yöneticisi	N=7	N=4
Diğer yönetim görevleri	N=37	N=51
Kesit-2: Kuruluş	2001	2008
Yazılım yüklenicisi	N=30	N=44
İç kullanım için yazılım geliştiren	N=24	N=28
Paket yazılım üreticisi	N=14	N=3
Kesit-3: Yazılım Kadrosu Çapı	2001	2008
1-9 kişi	N=20	N=30
10-49 kişi	N=19	N=22
50-99 kişi	N=16	N=9
100 kişi ve üstü	N=13	N=14

Bu yaklaşım ile ortaya koyulan veriler üç kesit altında toplam 10 kriter (Yazılım mühendisi, Kalite veya süreç yöneticisi, Diğer yönetim görevleri, Yazılım yüklenicisi, İç kullanım için yazılım geliştiren, Paket yazılım üreticisi, 1-9 kişi, 10-49 kişi, 50-99 kişi, 100+ kişi) ile 100 sorunun çarpımı kadar farklı göstergelere işaret etmektedir.

$$\text{Toplam} = (3 + 3 + 4) \times 100 = 1000$$

Değerlendirme sırasında beyan ve göstergelere dayalı olarak “Yazılım Süreçlerinin Yeterliliği” SPICE (ISO/IEC-TR15504) ve CMMI 1.2 süreç modelleri referans alınarak değerlendirilmiştir.

SPICE modeli içinde süreç düzeyleri aşağıdaki gibi (Tablo 3.1) ele alınır.

Tablo 3.1 : SPICE (ISO 15504) süreç düzeyleri

Düzyey	İsim	Nitelikler
0	Eksik	
1	Varolan	Temel uygulamaların varlığı
2	Yönetilen	Etkinliklerin başarı ile planlanması ve yönetilmesi
3	Kurumsallaşmış	Standart süreç tanımı ve uyarlama kurallarının varlığı
4	Niceliksel Yönetilen	Ürün ve süreç ölçümlerinin başarıyla kullanılması
5	Eniyilenen	Süreç değişiminin yönetilmesi

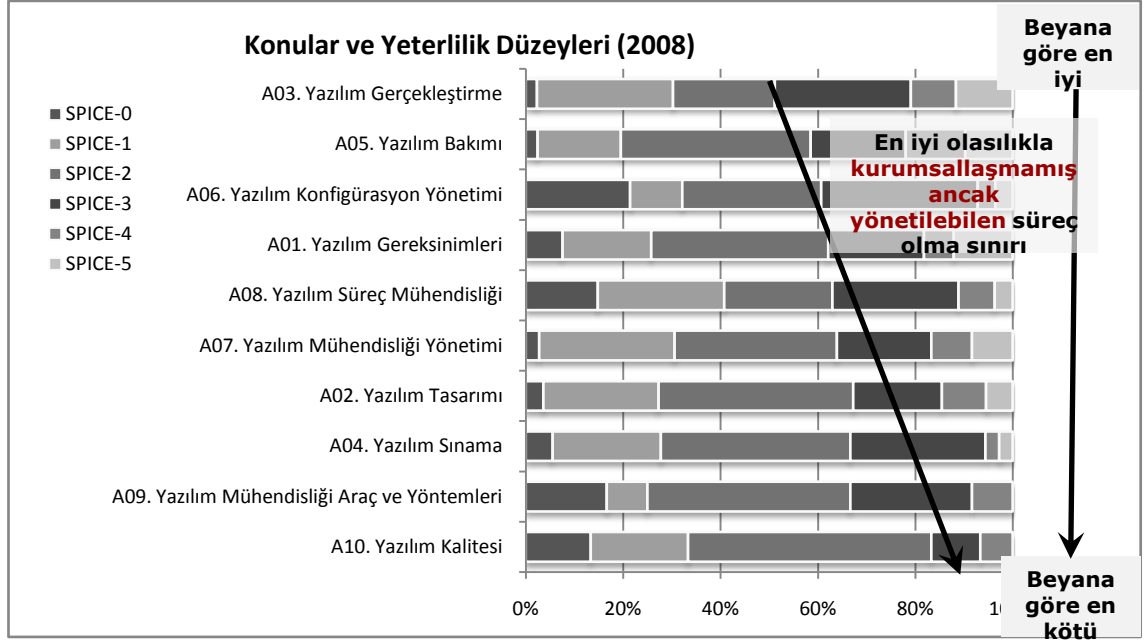
Kaynak: ISO/IEC TR 15504-1:1998(E)

Anket için göstergeler ile beyan arasındaki tutarlılığın sağlanması ise her kategorinin sonunda o kategoriye ilişkin olarak sorulan jenerik 10. sorular ile yapılmıştır.

Soru örneği;

- Kuruluşunuzda **gereksinimlerin yönetimi** ile ilgili genel bir değerlendirme yapmak gerekirse, aşağıdakilerden *hangisi* mevcut durumu en iyi yansıtıyor?
 - a. Başarıyla uyguladığımız söylenemez
 - b. Kişisel beceri ile yürütüyoruz
 - c. Kurumsal standart olmasa da planlı yürütülebilen bir sürecimiz var
 - d. Kurumsal düzeyde tanımlanmış ve uygulanan bir sürecimiz var
 - e. Tanımlı ve niceliksel denetim altında yürüyen bir sürecimiz var
 - f. Tanımlı, niceliksel denetim altında ve sürekli iyileştirilen bir sürecimiz var

Yanıtların özeti Şekil 3.2’de verilmiştir. Bu gösterimde, anketin 10 konusu “iyiden kötüye” doğru sıralanmıştır. Sıralamaya esas olan kriter, belirli bir konuda kendini 0-1-2 düzeylerinde gören katılımcıların oranıdır. Bu değerlendirmeye göre, bu oranın en düşük olduğu konu **A01. Yazılım Gerçekleştirme** uygulamaların en başarılı olduğu alan, bu oranın en yüksek olduğu konu **A10. Yazılım Kalitesi** ise en başarısız bulunduğu kabul edilen alan olarak ortaya çıkmaktadır.



Şekil 3.2 : Konular ve yeterlilik düzeyleri

2001 ve 2008 yılları arasındaki gelişimi açıklayabilmek için süreçler bazında olgunluk düzeylerinin sayısal ifadesi yöntemine başvurulmuştur.

Olgunluk düzeylerini sayısal olarak ifade edebilmek için her sürecin sonundaki düzeyi belirleyen 10. sorularda seçeneklerin (düzeylerin) karşısındaki yüzde oranları ile o seçeneğe ait düzey katsayısı çarpılarak ortaya çıkan sonuç rakamı detayı Tablo 3.2’de açıklandığı şekilde hesaplanarak karşılaştırma için kullanılmıştır. Aynı sonuçlar Minitab yazılımı kullanılarak yapılan ortalama hesapları ile de doğrulanmıştır. (Bkz. EK 5)

Tablo 3.2 : Örnek puan hesaplama tablosu

Seçenekler	2008	2001	Katsayı	2008 Puan	2001 Puan
0. Başarıyla uyguladığımız söylenemez	6%	2%	0	-	-
1. Kişisel beceri ile yürütüyoruz	19%	17%	1	0,19	0,17
2. Kurumsal standart olmasa da planlı yürütülebilen bir sürecimiz var	37%	31%	2	0,73	0,62
3. Kurumsal düzeyde tanımlanmış ve uygulanan bir sürecimiz var	21%	36%	3	0,62	1,07
4. Tanımlı ve niceliksel denetim altında yürüten bir sürecimiz var	6%	2%	4	0,25	0,10
5. Tanımlı, niceliksel denetim altında ve sürekli iyileştirilen bir sürecimiz var	11%	12%	5	0,56	0,60
Toplam Düzey Puanı				2,35	2,55

Bu örnekte ele alınan sürecin olgunluk düzeyi 2001 yılında yapılan arařtırmaya göre azalmıř durumdadır.

3.4 VERİLERİN GÜVENİLİRLİĐİ

Her iki arařtırmaya ait veriler içinde de “Yazılım MühendisliĐi Araç ve Yöntemleri, Yazılım Kalitesi, Yazılım Süreç MühendisliĐi, Yazılım Konfigürasyon Yönetimi ve Yazılım MühendisliĐi Yönetimi” az bilinen, az uygulanan ve arařtırma çalıřması içinde daha az yanıtlanan konular olmuřtur.

Arařtırma çalıřması içinde yanıtları bir ölçek içermeyen sorular bulunması sebebi ile anketteki tüm cevapları hesaba katarak; ortalama, standart sapma gibi istatistikleri hesaplamak ve bulguları bu açıdan yorumlamak, mümkün olmamıřtır. Arařtırmada yer alan on konu altında yer alan sorular göreceli olarak birbirinden baĐımsızdır. Ait olduĐu konu ile ilgili olmakla beraber, diĐer konulara ait sorulardan farklı bir bilgiyi konu edinmektedir. Bu nedenle analizler içinde her kategori altında yer alan dokuz soru o bölümün sonunda yer alan soru ile karşılaştırılarak saĐlaması yapılmıřtır. Her bölümün sonunda yer alan onuncu sorular, birbiri ile baĐlantılı ve benzer yapıda olduĐu için bu soruların yanıtları ile güvenilirlik analizleri gerçekleştirilmiřtir.

Güvenilirlik, baĐımsız ölçümler arasındaki kararlılıktır, korelasyon katsayısı (r) ile belirlenir. Korelasyon katsayısı, sıfır ile bir arasında deĐişen deĐerler alır ve bu deĐer bire yaklařtıkça güvenilirliĐin yüksek olduĐu kabul edilir.

Cronbach-Alpha katsayısı ise, soruların birbirleri ile yakınlıĐını ortaya koyar, n sorunun türdeř bir yapıyı açıklamak ya da sorgulamak üzere bir bütün oluşturup oluşturmadıĐını sorgulamayı saĐlar.

Alpha katsayıları ölçekte yer alan k sorunun varyansları toplamının genel varyansa oranlanması ile bulunan bir aĐırlıklı standart deĐişim ortalamasıdır. 0 ile 1 arasında deĐişim gösterir. Sorular standardize edilmiřse, Cronbach-Alpha katsayısı, soruların ortalama korelasyonuna ya da kovaryansına dayanarak hesaplanır.

2008 yılında, tüm kategorilerde soruları yanıtlayan 17 kişinin verileri ile gerçekleştirilen güvenlik ölçümünde (nokta tahmini ile) Cronbach's Alpha = 0,9666 değeri bulunmaktadır.

2001 yılında tüm kategorilerde soruları yanıtlayan 27 kişinin verileri analiz edilerek Cronbach's Alpha = 0,9009 değeri bulunmaktadır. (Bkz. EK 4)

Verilerin güvenilir olması için bu değer 0,7 rakamının üzerinde olması önerilmektedir. (Hair ve diğ. 1998) Minitab yazılımı ile gerçekleştirilen güvenilirlik analizlerinde, bulunan alfa sayılarının 0,7 rakamının üzerinde olması, araştırma verilerinin güvenilir olduğunu göstermektedir.

2008 yılı verilerine göre en yüksek korelasyon (0,946), "Yazılım Gereksinimleri" ve "Yazılım Mühendisliği Yönetimi" soru kategorileri arasındadır. Bunu "Yazılım Tasarımı" ve "Yazılım Gerçekleştirme" izlemektedir (0,914). (Bkz. EK 4)

2008 yılı verilerine göre en düşük korelasyon (0,448), "Yazılım Konfigürasyon Yönetimi" ve "Yazılım Mühendisliği Araç ve Yöntemleri" arasındadır. Bunu "Yazılım Konfigürasyon Yönetimi" ve "Yazılım Süreç Mühendisliği" arasındaki düşük korelasyon izlemektedir (0,560). (Bkz. EK 4)

2008 yılı verilerine göre en yüksek korelasyon (0,733), "Yazılım Sınama" ve "Yazılım Mühendisliği Yönetimi" soru kategorileri arasındadır. Bunu "Yazılım Tasarımı" ve "Yazılım Mühendisliği Yönetimi" izlemektedir (0,716). (Bkz. EK 4)

2008 yılı verilerine göre en düşük korelasyon (0,297), "Yazılım Gereksinimleri" ve "Yazılım Kalitesi" arasındadır. Bunu "Yazılım Tasarımı" ve "Yazılım Kalitesi" arasındaki düşük korelasyon izlemektedir (0,301). (Bkz. EK 4)

4. ANALİZ VE SONUÇLAR

4.1 SONUÇLAR VE GELECEKTEKİ ÇALIŞMA

Anketin yayınlandığı web sitesine, toplam 81 giriş yapılmıştır. Yanıtları olmadığı belirlenen girişler elendikten sonra, sağlıklı veri kümesini oluşturan 75 kişinin verileri üzerinde analizler gerçekleştirilmiştir. Bu sayı kimliğini beyan eden katılımcıların beyanı doğrultusunda en az 62 firmaya işaret etmektedir. 2001 yılında KalDer tarafından gerçekleştirilen çalışmada ise bu sayılar 68 kişi ve 50 firma olarak gerçekleşmiştir.

Oldukça uzun soru ve seçenekler içeren (100 soru, 804 seçenek) bir değerlendirme olması, soru ve seçeneklerin olgunluk düzeyi düşük firmalar tarafından anlaşılabilmesi ve/veya bu firmaların olgunluk düzeylerini açıklamak istememesi gibi sebeplerle bu sayının daha yukarı çıkmadığı çalışmalar sırasında gözlenmiştir. Değerlendirmenin uzun olduğu bilinmekle birlikte tam bir kıyaslama yapabilmek için 2001 yılı ile aynı soru setini kullanma yaklaşımı tercih edilmiştir.

2001 yılında katılımcılar arasında; küçük yazılım yüklenicileri, kendi yazılımını geliştiren büyük kuruluşlar ve orta büyüklükte paket yazılım şirketlerinin dağılımı itibarı ile temsil niteliği yüksek bir örnekleme alınmışken 2008 yılında Tablo 4.1’de detayı görüldüğü şekilde sadece üç paket yazılım üreticisi firmadan yanıt alınabilmiştir.

Tablo 4.1 : Görev yapılan firma türü

Firma	2008	2001
Proje bazlı yazılım yüklenicisi	%59	%44
İç kullanım için yazılım geliştiren	%37	%35
Paket yazılım üreticisi	%4	%21
	N=75	N=68

Detayı Tablo 4.2’de yer alan veriler itibarı ile, en büyük müşteri grubunun endüstriden hizmete doğru kaydığı, en küçük müşteri grubunun yüzde 11 oranla kamu olduğu ve örneklenen firmaların ağırlıklı Ankara dışındaki yazılım firmaları olduğu görülmektedir.

Tablo 4.2 : Firmanın hitap ettiği sektör

Firmanın Hitap Ettiği Sektör	2008		2001	
Endüstri	25	%33	34	%50
Finans	16	%19	14	%21
Hizmet	28	%37	14	%21
Kamu	8	%10	6	%9
N	77		68	

Detayı Tablo 4.3’te yer alan katılımcıların dağılımı, kuruluş çapına göre incelendiğinde: 2001 yılı verilerinde farklı ölçekte firmalar arasında yaklaşık eşit bir dağılım görülürken, 2008 yılında değerlendirme yapanların yarısı çalışan sayısı 100 kişiden az firmalara işaret etmektedir.

Tablo 4.3 : Kuruluş çapı

Kuruluş Çapı	2008		2001	
1-99	38	%51	22	%32
100-499	20	%25	24	%35
500+	19	%24	22	%32
N	75		68	

Katılımcıların kuruluşlarının “Yazılım Kadrosu Çapı”na göre dağılımı ise Tablo 4.4’te görüldüğü gibi küçük yazılım firmalarını ağırlıklı temsil edecek yapıdadır.

Tablo 4.4 : Yazılım kadrosu çapı

Yazılım Kadrosu Çapı	2008		2001	
1-9	30	%40	20	%29
10-49	22	%29	19	%28
50-99	11	%12	16	%24
100+	14	%19	13	%19
N	75		68	

Diğer yönetim görevlerinin alt açılımları da dikkate alınarak Tablo 4.5’te mühendislerin ağırlığı dikkat çekmektedir. Yanıtlayanlar arasında 18 üst yöneticinin (yüzde 24) yer

alması da yanıtlayan küçük yazılım firmalarının çokluğu sebebi ile ortaya çıkan bir sonuç olmuştur.

Tablo 4.5 : Görev tanımı

Görev Tanımı	2008		2001	
Yazılım mühendisi	20	%27	22	%32
Kalite veya süreç yöneticisi	4	%5	7	%10
Diğer yönetim görevleri	53	%68	37	%58
N	75		68	

Katılımcıların deneyim süreleri incelendiğinde (Tablo 4.6), 2008 yılında deneyimli katılımcıların sayısının arttığı gözlenmektedir. 20 yıl ve üstü deneyime sahip katılımcıların deneyim süresi 20 yıl kabul edilerek “sınıf aralıkları” metodu ile ortalama hesaplandığı zaman, 2001 yılında en az 8,7 yıl olan deneyim ortalamasının 2008 yılında en az 11,06 yıl olarak gerçekleştiği görülmektedir.

Tablo 4.6 : Deneyim süresi

Deneyim Süresi	2008		2001	
0-1 yıl	1	%1	5	%7
2-4 yıl	12	%16	16	%24
5-9 yıl	25	%32	23	%34
10-19 yıl	27	%36	20	%29
20 yıl ve üstü	12	%15	4	%6
N	75		68	

Verilen Aralık İçin Yıl Ortalaması	2008 Yılı Yanıtlayan Kişi Adedi	Toplam Yıl	
0,5 x	1	=	0,5
3 x	12	=	36
7 x	24	=	168
15 x	27	=	405
20 x	11	=	220
Toplam	75	829,5	Ortalama Deneyim → 829,5/75 = 11,06

Verilen Aralık İçin Yıl Ortalaması	2001 Yılı Yanıtlayan Kişi Adedi	Toplam Yıl	
0,5 x	5	=	2,5
3 x	16	=	48
7 x	23	=	161
15 x	20	=	300
20 x	4	=	80
Toplam	68	591,5	Ortalama Yıl → 591,5/68 = 8,70

2008 yılı değerlendirmesinde ağırlık noktasını küçük şirketler oluşturmaktadır. Türkiye yazılım sektörü içinde süreç olgunluk düzeyi daha yüksek olan savunma sanayine yazılım geliştiren firmaların ve 2008 yılında araştırmaya katılımı mümkün olmamış, paket yazılımcıların ise katılımı sadece üç firma düzeyinde kalmıştır. Katılımcılar içinde CMMI vb sertifikasyona sahip firmaların da katılımı düşük düzeydedir. Katılımcıların araştırma içinde temsil edildiği oranlar, sektör içindeki dağılım oranlarını da temsil ettiği için katılımcılar iyi bir örnek grubu oluşturmaktadır. 2008 yılında paket yazılım üreticilerinin yerini daha çok yazılım yüklenicisi olan firmalar almıştır.

2001 yılı değerlendirmesinde katılımcılar endüstri ağırlıklıyken, 2008 yılı değerlendirmesi içinde göreceli olarak daha homojen bir katılım görülmektedir.

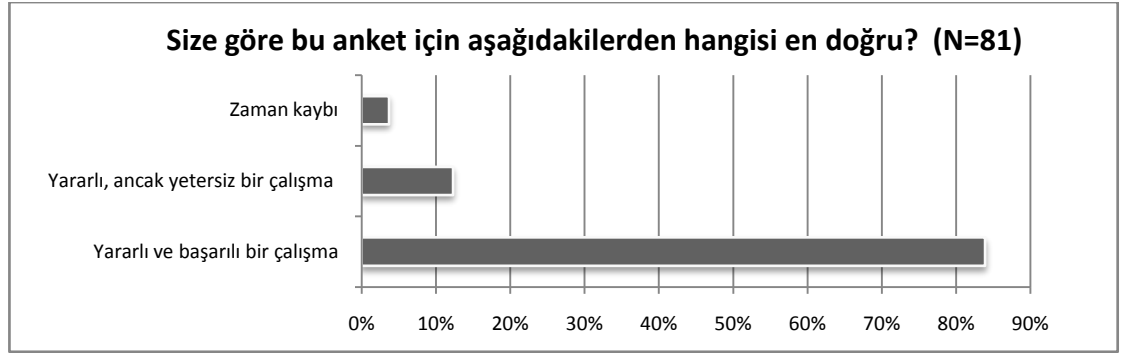
Savunma sanayi firmalarının ve büyük ölçekli firmaların katılımının 2008 yılında daha az olmasının doğal bir sonucu olarak, yanıtlayanlar içinde Kalite veya Süreç Yöneticilerinin katılımı yarı yarıya azalmıştır.

2008 yılı katılımcıları içinde diğer yöneticilerin katılımı önemli ölçüde artmıştır. Özellikle bu grup içinde 2001 yılında yüzde 15 olan üst yönetici katılımı yüzde 23'lere çıkmıştır. Yöneticilerin katılımındaki artış hem araştırmanın duyurulduğu kanallar ile hem de yönetim düzeyinde bu konudaki farkındalığın artmış olması ile ilgilidir.

Yönetim düzeylerindeki katılımın artışı ile paralel olarak, yanıtlayanların yazılım alanındaki deneyim süreleri de 2008 yılı araştırmasında 2001 yılına göre artış göstermiştir.

Her iki arařtırmaya ait veriler içinde de “Yazılım Mühendisliđi Araç ve Yöntemleri, Yazılım Kalitesi, Yazılım Süreç Mühendisliđi, Yazılım Konfigürasyon Yönetimi ve Yazılım Mühendisliđi Yönetimi” az bilinen, az uygulanan ve arařtırma çalıřması içinde daha az yanıtlanan konular olmuřtur.

Arařtırma çalıřmasının son sorusu bu çalıřmanın řekil 4.1 ile açıklanan başarı deđerlendirmesine aittir. 2001 ve 2008 yılları arasında arařtırmanın başarısı da (oransal olarak) tümü ile aynı deđerlendirme sonuçlarına sahiptir.



řekil 4.1 : Anket başarı deđerlendirmesi

Sonraki çalıřmalara girdi oluřturmak üzere arařtırmanın yetersiz ya da zaman kaybı olduđunu düřünen katılımcılar ile e_posta aracılıđı iletiřim kurulmuř ancak iyileřtirme konusunda önerilerini almak mümkün olmamıřtır.

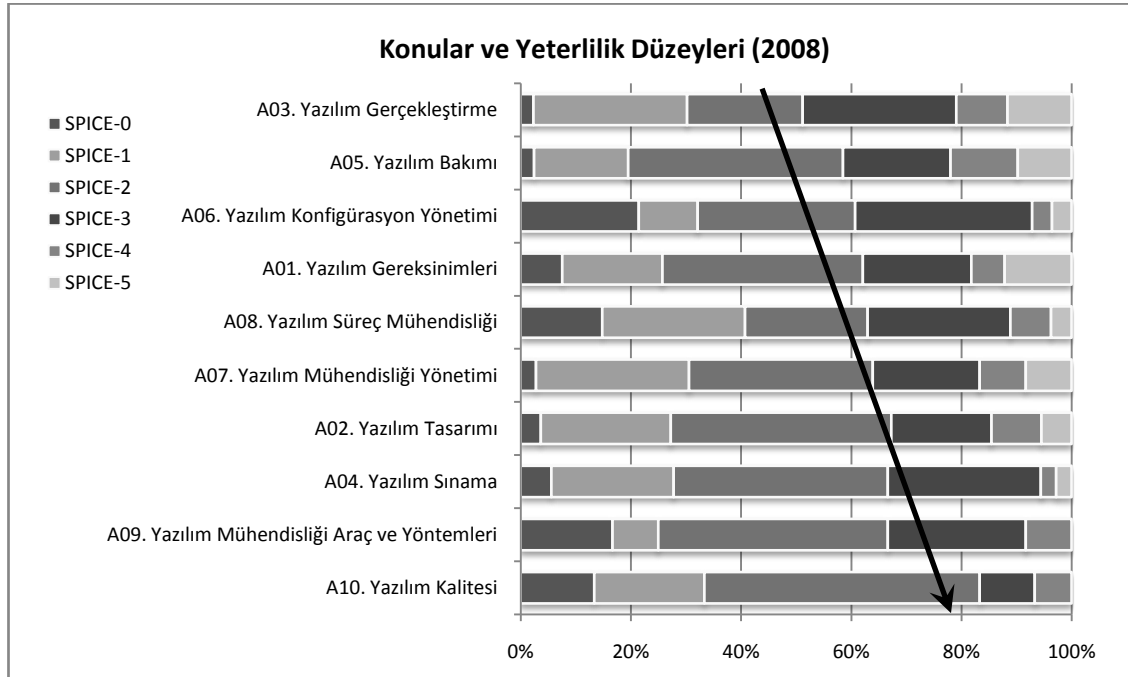
4.2 GENEL ANALİZLER

Anket için göstergeler ile beyan arasındaki tutarlılığın sağlanması, her kategorinin sonunda o kategoriye ilişkin olarak sorulan jenerik 10. sorular ile yapılmıştır. Bu sorulara alınan tüm yanıtlar on kategori için EK 6 ile verilmiştir. (Bkz EK 6).

2008 yılına ait tüm yanıtlarının özeti Şekil 4.2 ile verilmiştir. Bu gösterimde, anketin on konusu “iyiden kötüye” doğru (o konuda kendini 0-1-2 düzeylerinde gören katılımcıların oranına göre) sıralanmıştır. En iyi olasılıkla kurumsallaşmamış ancak yönetilebilir süreç olma sınırı Şekil 4.2 üzerinde çizgi ile gösterilmiştir.

Bu değerlendirmeye göre, 2001 bu oranın en düşük olduğu konu **A03. Yazılım Gerçekleştirme** uygulamaların en başarılı olduğu alan, bu oranın en yüksek olduğu konu **A10. Yazılım Kalitesi** ise en başarısız olduğu kabul edilen alan olarak ortaya çıkmaktadır.

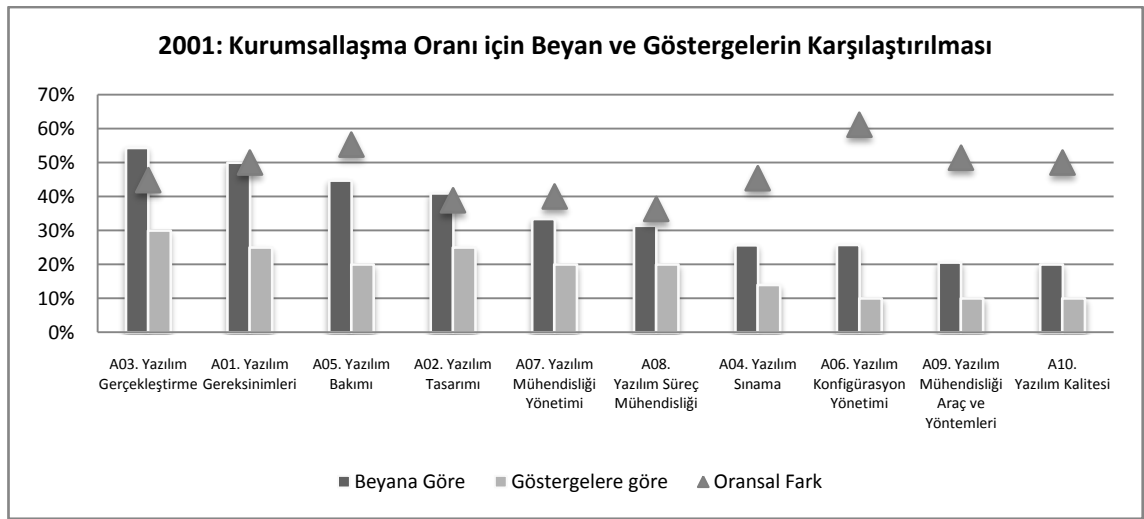
Genel görünümüne bakınca, yazılım mühendisliği uygulamalarının olgunluğu veya kurumsallaşma başarısı yazılım yönetimi konularına göre daha ileride görülmektedir.



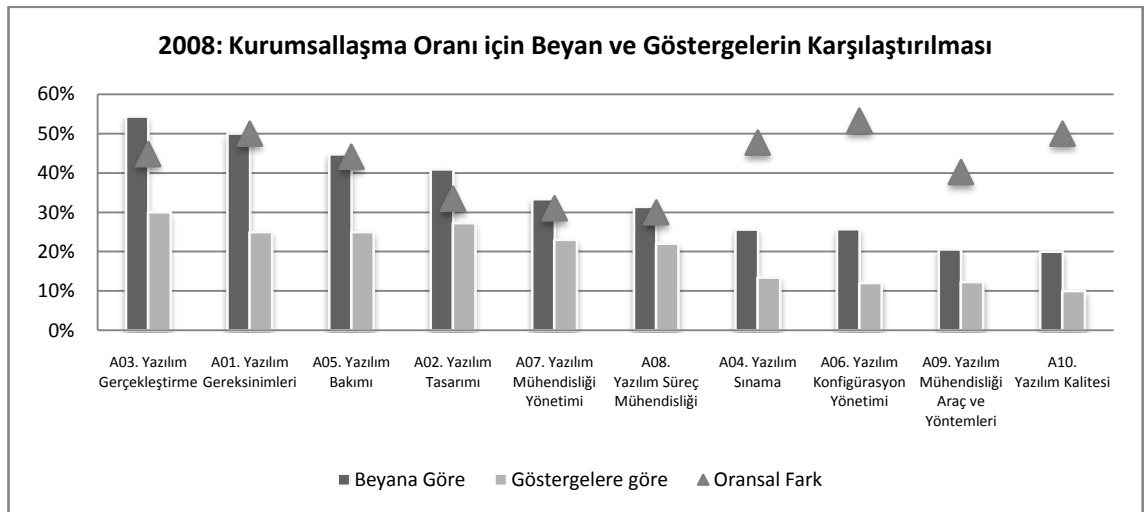
Şekil 4.2 : Süreçler bazında olgunluk düzeyleri – 2008

Kurumsallaşma oranı için beyan ve göstergelerin karşılaştırılması 2001 yılı verileri için Şekil 4.3, 2008 yılı verileri için Şekil 4.4'te yer almaktadır. Bu analiz içinde beyan edilen düzey hesabı her kategorinin sonunda yer alan 10. sorular karşılığındaki düzey alınarak hesaplanmıştır.

Göstergelere göre olgunluk seviyesinin belirlenmesi için ise o konuda diğer dokuz soruya verilen yanıtlar genel bir bakışla değerlendirilerek özellikle 1. ve 2. seviye için gerekli görülen iş ürünlerinin mevcudiyet durumuna bakılmıştır.



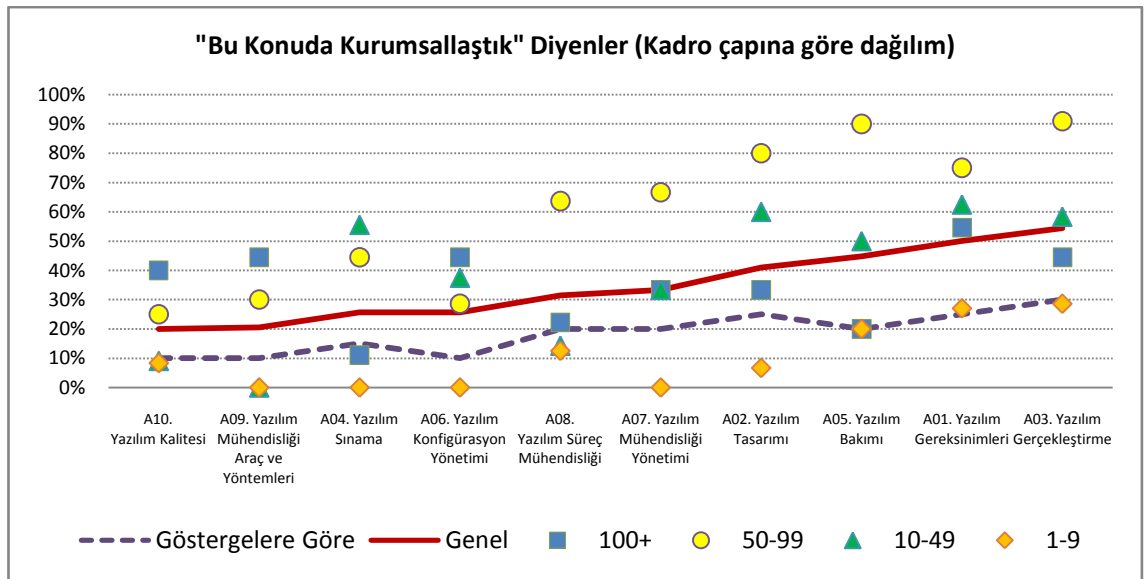
Şekil 4.3 : Kurumsallaşma oranı – 2001



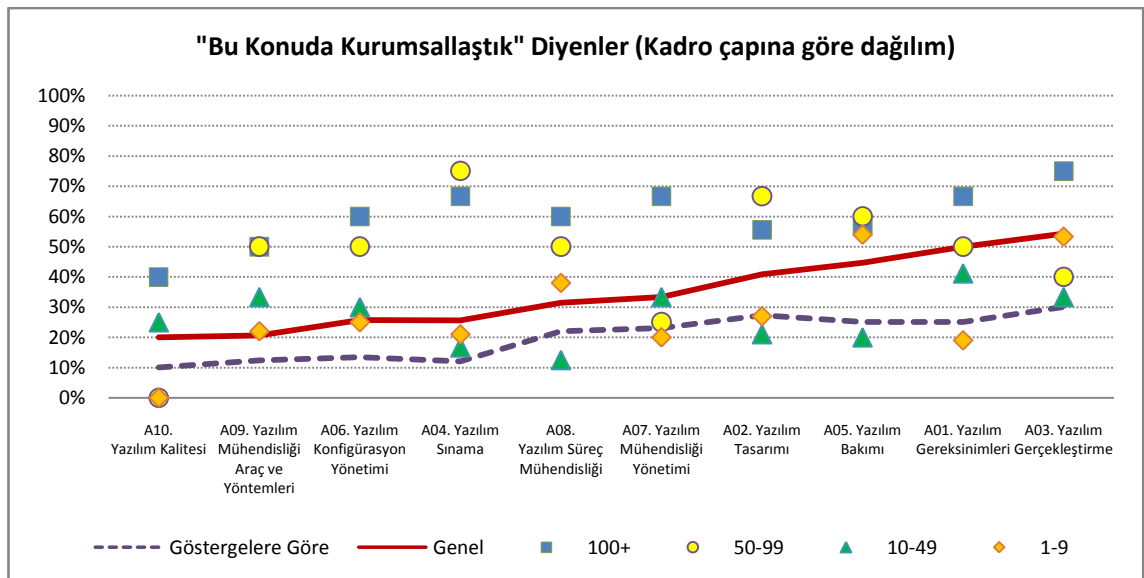
Şekil 4.4 : Kurumsallaşma oranı – 2008

2001 ve 2008 yılları için kurumsallaşma oranının karakteristik yapısı benzer olmakla beraber 2008 yılında oransal fark ya da sapma daha düşüktür. (Ortalama yüzde 47 → yüzde 42'ye inmiştir) Bu durum 2008 yılı göstergeleri ile beyanlarının 2001 yılına göre az da olsa daha tutarlı olduğunu göstermektedir.

Şekil 4.5 ve Şekil 4.6 ile kurumsallaşmanın kadro çapına göre dağılımı incelendiği zaman özellikle belli süreçlerde (tasarım, süreç mühendisliği, araç ve yöntemler) sürece verilen önemin eldeki yazılım kaynakları ile orantılı olduğu görülmektedir.

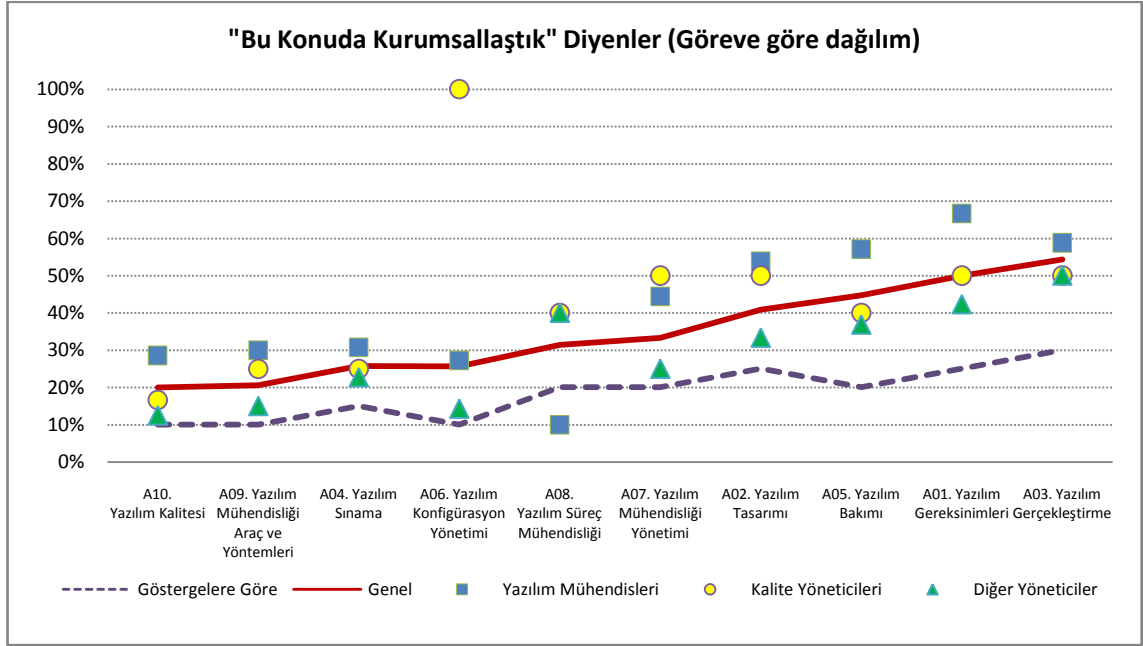


Şekil 4.5 : Kurumsallaşmanın kadro çapına göre dağılımı – 2001

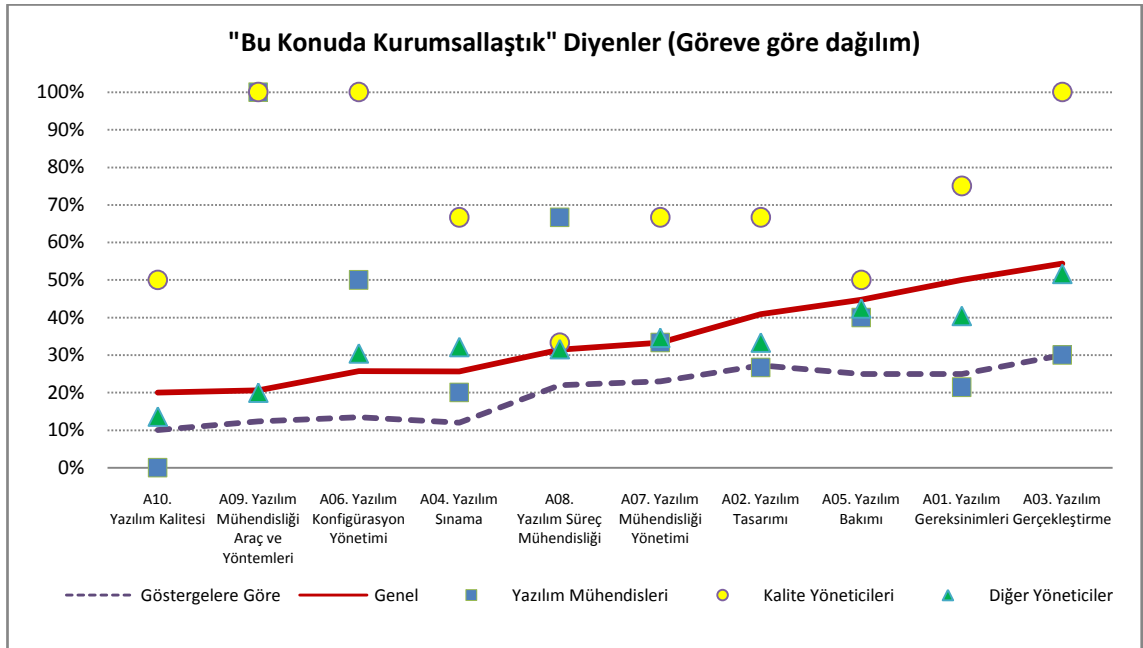


Şekil 4.6 : Kurumsallaşmanın kadro çapına göre dağılımı – 2008

Şekil 4.7 incelendiği zaman, 2001 yılında görev bakış açısına göre yeterlilik algılamasında önemli fark bulunmadığı görülmektedir. 2008 yılı verilerinde ise (Şekil 4.8) Kalite Yöneticileri diğer görevlere göre daha iyimser bir değerlendirme yapmıştır. Kalite Yöneticilerini, diğer yöneticiler ve yazılım mühendisleri izlemektedir.

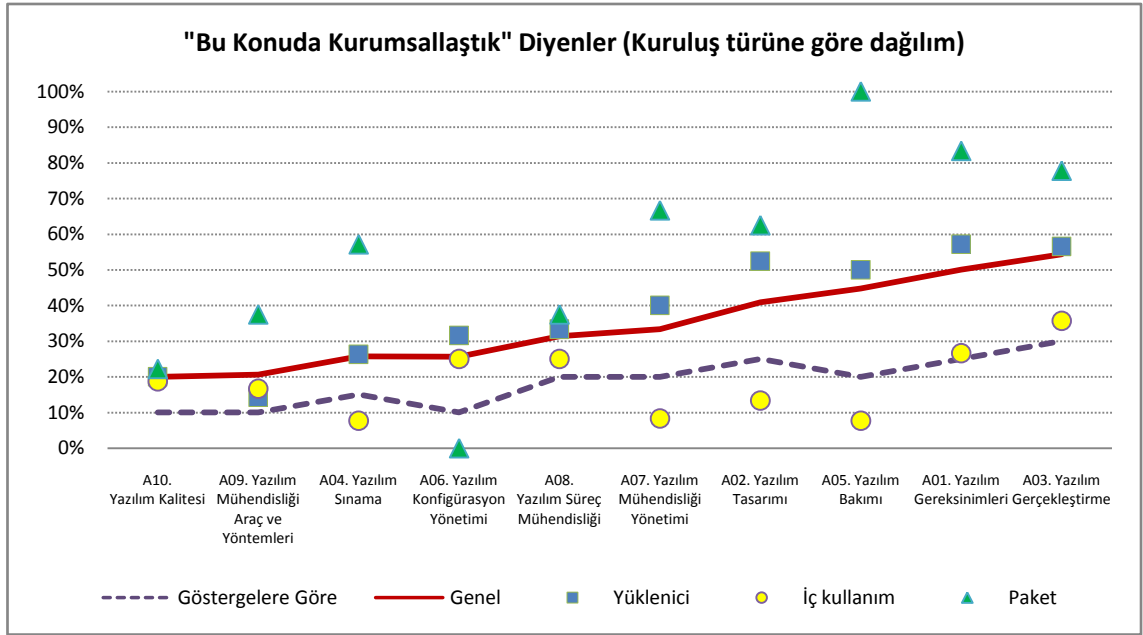


Şekil 4.7 : Kurumsallaşmanın göreve göre dağılımı – 2001

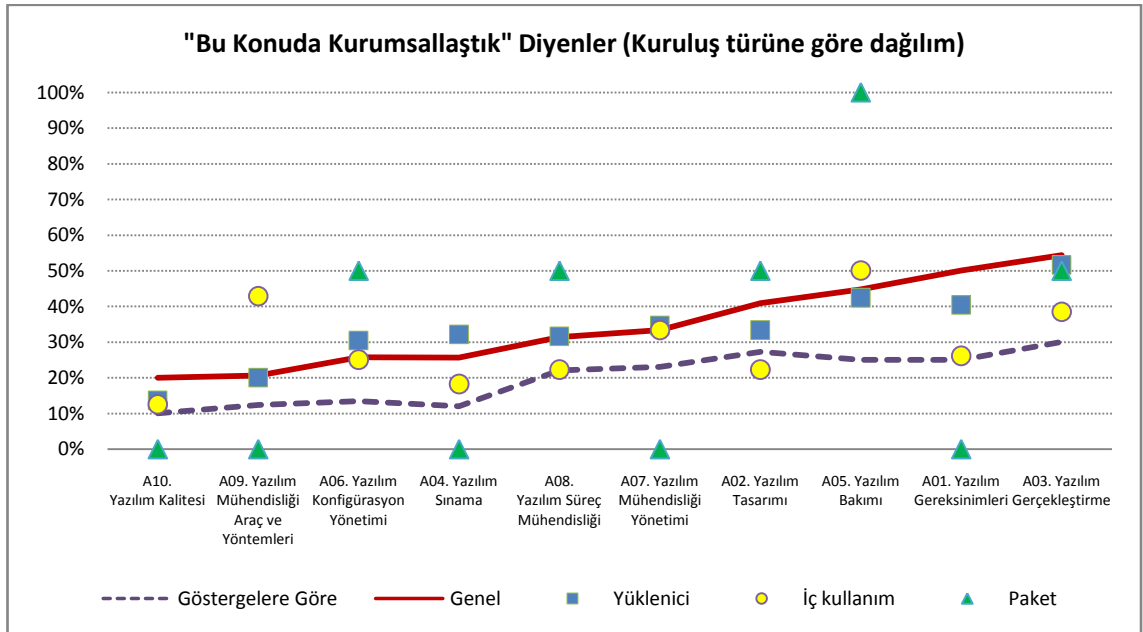


Şekil 4.8 : Kurumsallaşmanın göreve göre dağılımı – 2008

Veriler kuruluş türüne göre segmente edildiği zaman proje bazlı yazılım üreticilerinin süreç olgunluk düzeyinin, iç kullanım için yazılım üreten firmalara göre 2001 yılında da (Şekil 4.9) 2008 yılında da (Şekil 4.10) daha iyi olduğu görülmektedir. 2008 yılında araştırmaya katılan paket yazılım üreticilerinin sadece üç tane olması ve genel temsil etmede yetersiz kalacağı düşüncesi ile bu örnekler yorumlanmamıştır.



Şekil 4.9 : Kurumsallaşmanın kuruluş türüne göre dağılımı – 2001



Şekil 4.10 : Kurumsallaşmanın kuruluş türüne göre dağılımı – 2008

4.3 SÜREÇLER BAZINDA ANALİZ SONUÇLARI

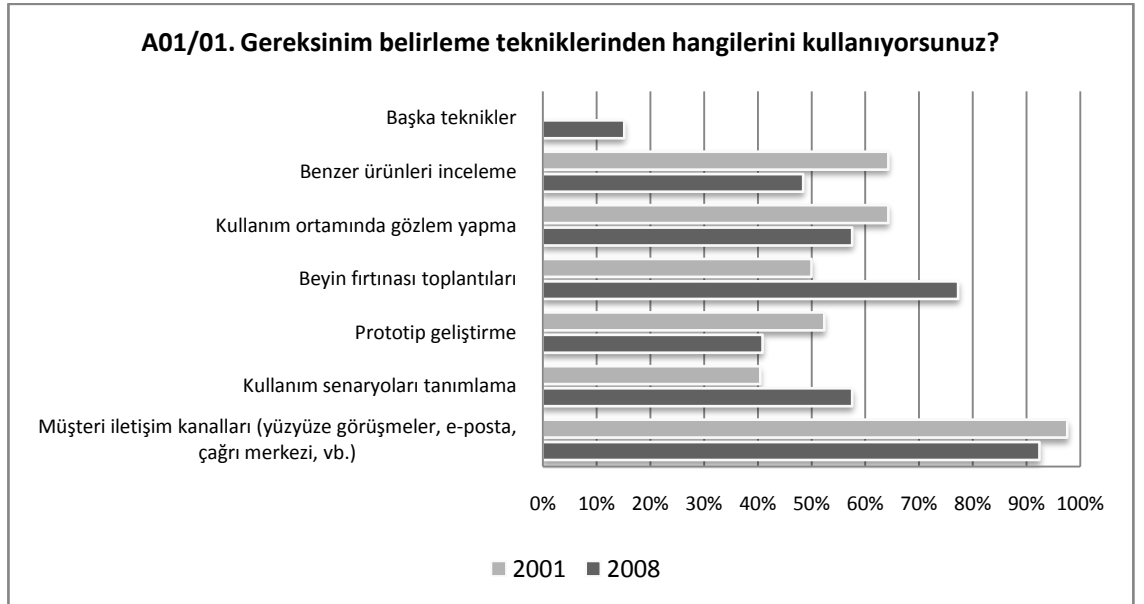
4.3.1 Yazılım Gereksinimleri

Yazılım gereksinimleri, bir yazılım ürününün amaçlarına erişmesi için istenen özellikleridir.

Detayı Şekil 4.11’de yer alan sonuçlar içinde, 2001 yılında gereksinim belirleme tekniklerine yönelik olarak; prototip geliştirmenin kullanılmakla beraber henüz ağırlığı olmayan bir uygulama olduğu görülmektedir. Müşteri iletişimi, gereksinim belirleme için en çok kullanılan yöntem olmakla beraber müşteri iletişiminin ağırlığı yazılım yüklenicisi türündeki katılımcılardan kaynaklanmaktadır.

2008 yılı sonuçları içinde ise, proaktif olarak müşteriyi daha anlamaya yönelik interaktif yöntemlerin ağırlığının arttığı görülmektedir.

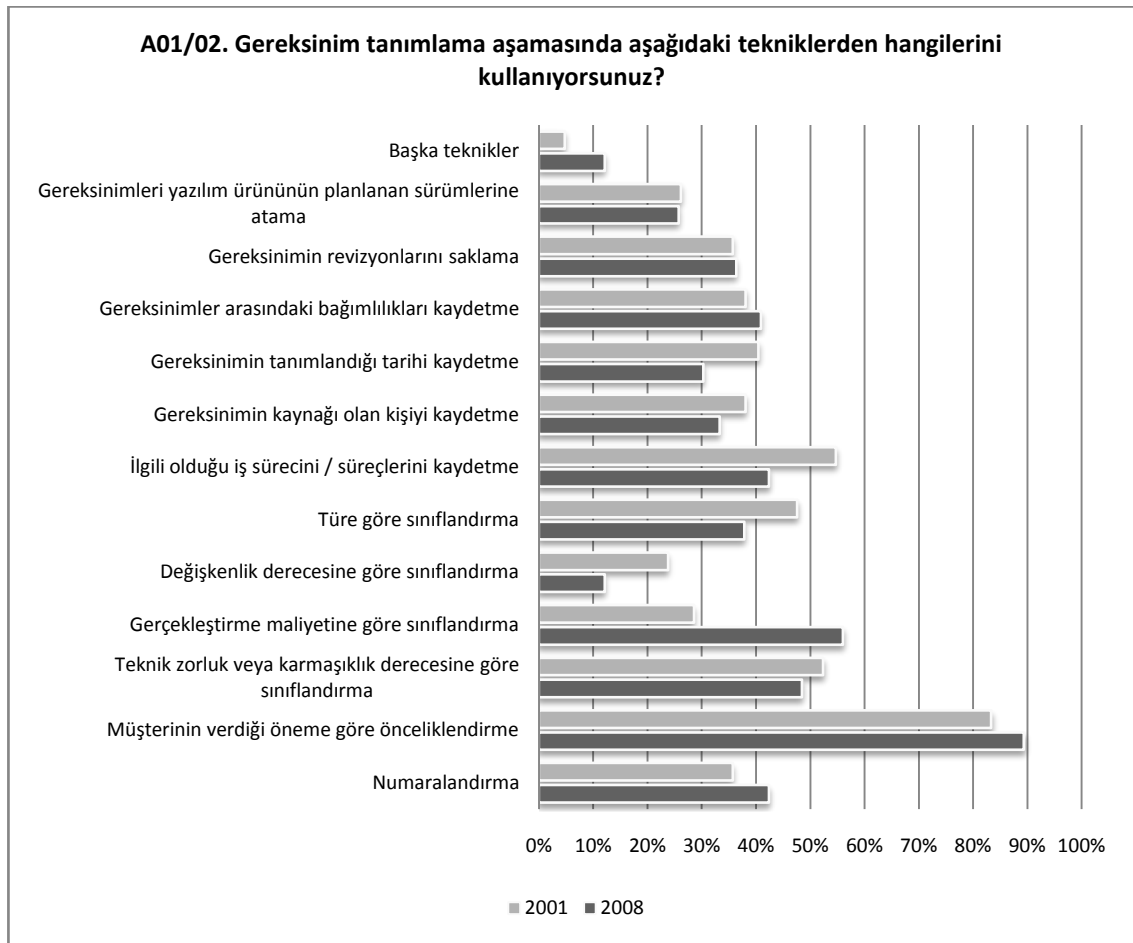
2008 yılında da devam eden prototip geliştirme uygulamasının azlığı, döngüsel geliştirme yönteminin ağırlıklı olmadığı izlenimini vermektedir.



Şekil 4.11 : Gereksinim belirleme teknikleri

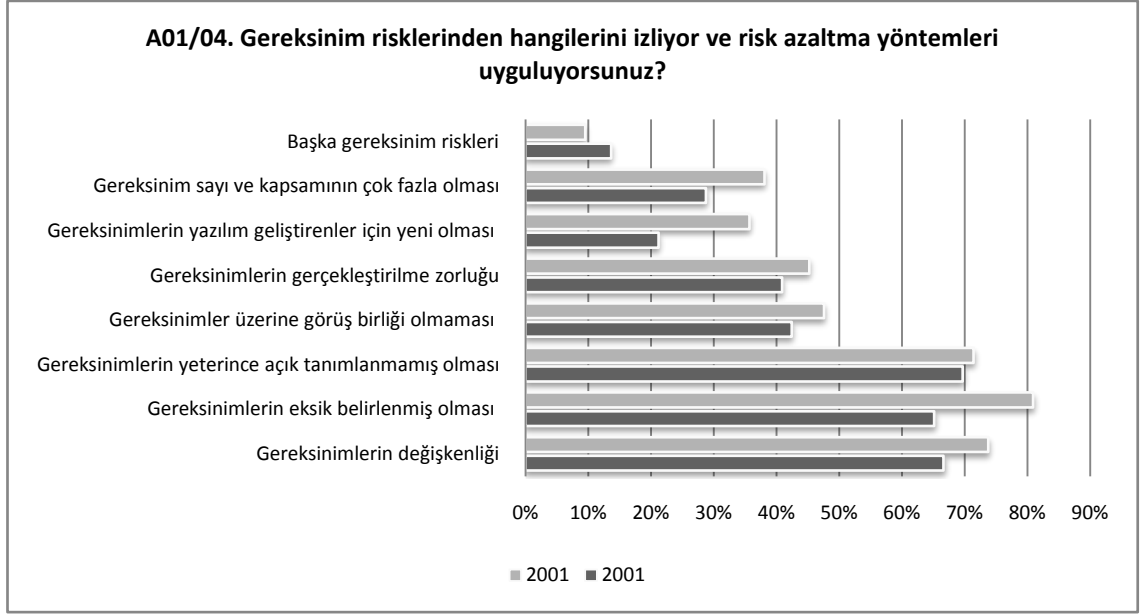
2001 yılı sonuçları içinde; gereksinim belirleme tekniklerine yönelik olarak (Şekil 4.12); gereksinimleri numaralandırma, yazılım sürümleri ile eşleme gibi “İzlenebilirlik” açısından kritik kabul edilebilecek uygulamalar arka sıralarda, müşterinin yönlendirmesi daha ön planda yer almaktadır.

2008 yılında ise tüm kriterler 2001 yılı ile paralellik arzederken, “gerçekleştirme maliyetine göre sınıflandırma” yönteminin müşteri önceliğinin hemen ardında ikinci sırada yer alması dikkat çekicidir.



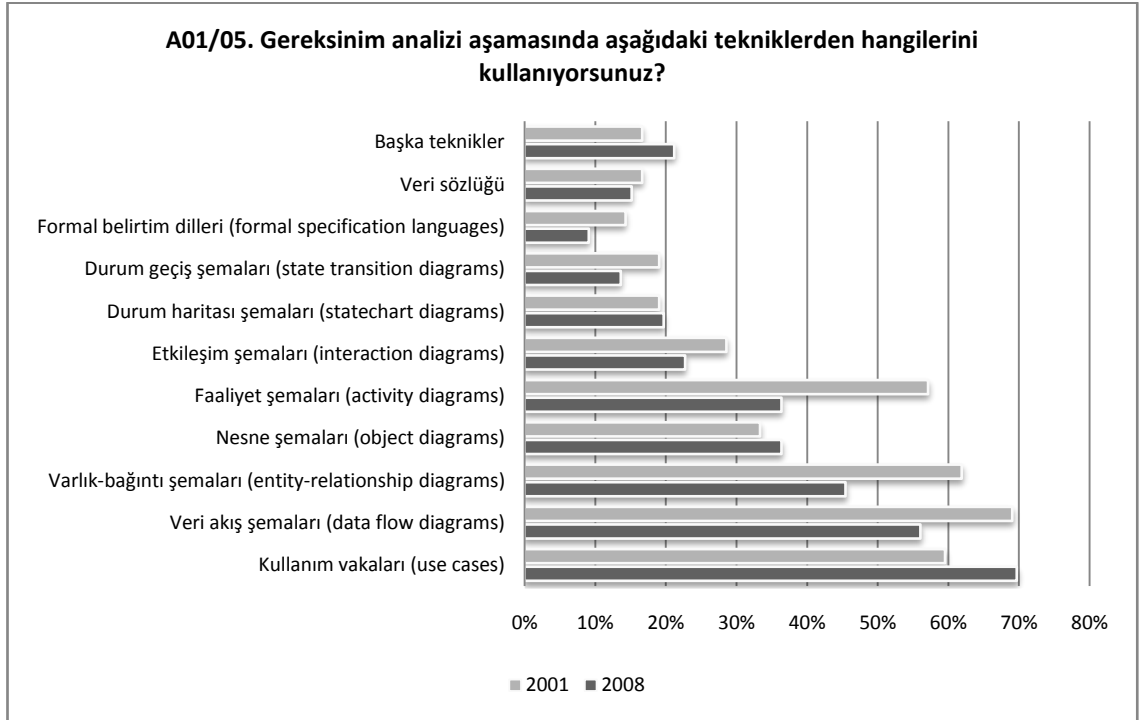
Şekil 4.12 : Gereksinim tanımlama teknikleri

2008 yılında da 2001 yılı sonuçları içinde olduğu gibi; gereksinim risklerinin izlenmesi ve önlem alınması kapsamında “gereksinimlerin eksikliği, açık tanımlanmamış olması ve değişkenliği” gibi evrensel sorunların öne çıktığı Şekil 4.13’te gözlenmektedir.



Şekil 4.13 : Gereksinim riskleri

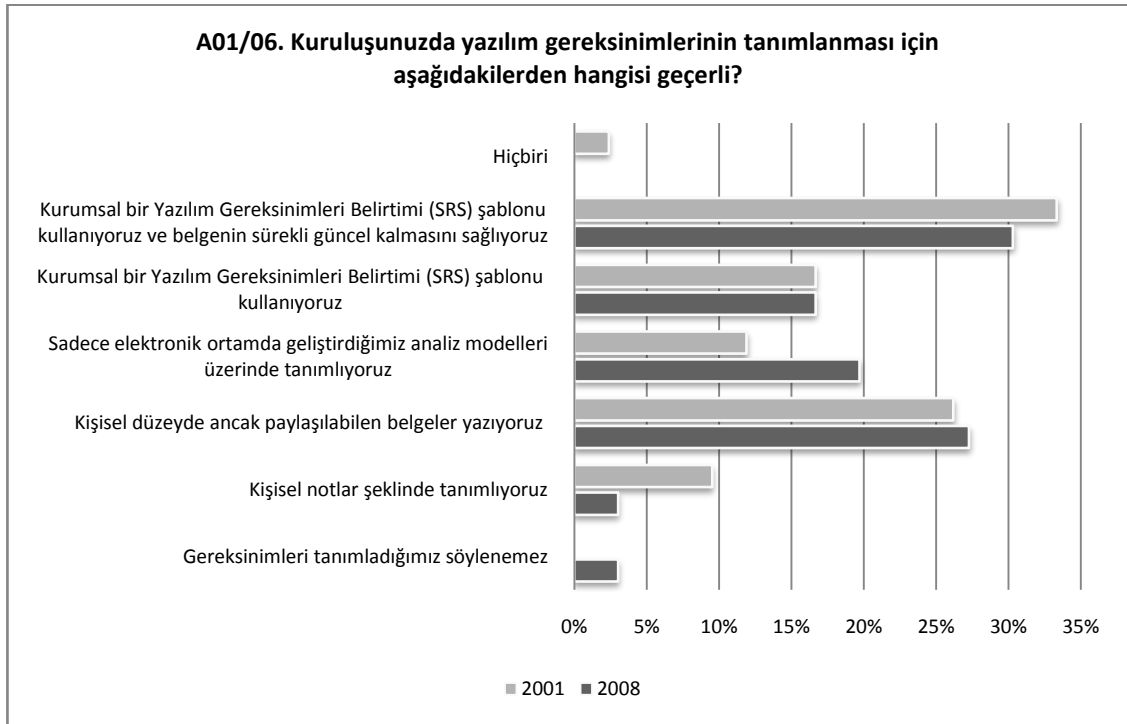
Şekil 4.14’te görüldüğü şekilde, 2001 ve 2008 yılı sonuçları içinde gereksinim analizi aşamasında kullanılan teknikler arasında, veri sözlüğünün düşük bilinirliği/kullanımı dikkat çekicidir. İzleyen geliştirme süreçlerinde veriler ve işlevler arasındaki ilişkilerin kontrollü tasarlanmasına olumsuz etkileri olacaktır.



Şekil 4.14 : Gereksinim analizi teknikleri

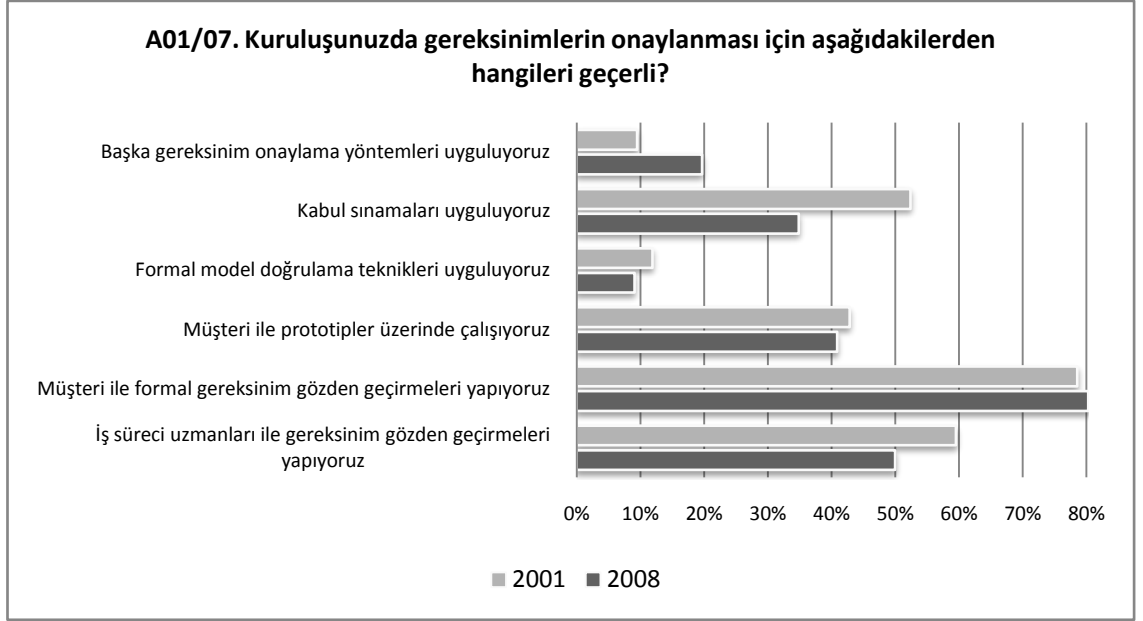
Yazılım gereksinimlerinin tanımlanması için kurumsal bir şablon kullanan ve güncel kalmasını sağlayan firmaların oranının hala yüzde 30 oranında olması sonraki süreçlerin performansı açısından düşündürücüdür (Şekil 4.15).

Geçen zaman süresince elektronik ortamlar üzerinde tanımlama oranında artış görülmekle birlikte, gereksinim tanımlama sürecinin kurumsallaşma oranı, kişisel düzeyde yapılan uygulamalar ve oranı az da olsa gereksinimleri tanımlamayan firmalar bulunması dikkat çekmektedir.



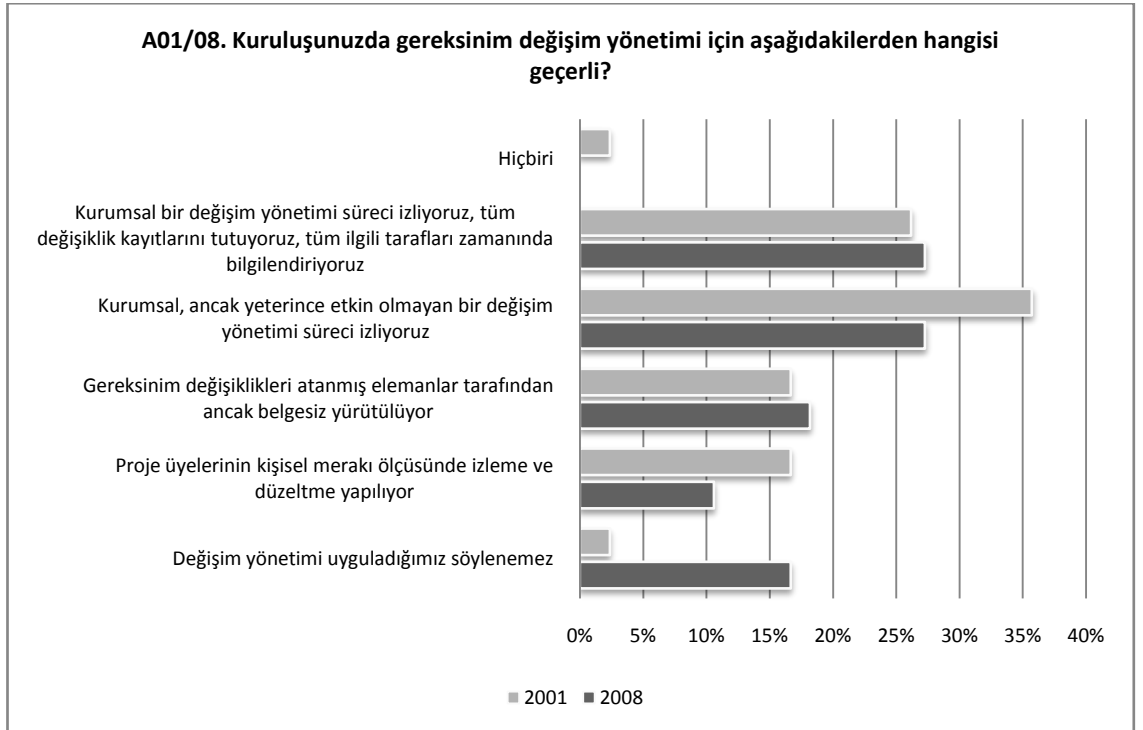
Şekil 4.15 : Yazılım gereksinimlerinin tanımlanması

Gereksinimlerin onaylanması konusunda da sonuçlar Şekil 4.16'da görüldüğü gibi 2001 yılı ile önemli ölçüde paralellik arz etmektedir. Bu konuda başka onaylama yöntemleri uyguluyoruz seçeneğindeki yüzde 10 düzeyindeki artışın arkasındaki görünmeyen sebebin, gereksinimlerin onaylanması konusunda bir yaklaşım bulunmaması olması mümkündür. Sonuçlar içinde formal tekniklerin çok az uygulanıyor olması bu konuda üzerinde durulması gereken bir başka önemli noktadır.



Şekil 4.16 : Gereksinimlerin onaylanması

Şekil 4.17’de, gereksinimler için değişim yönetimi uygulamadığını beyan eden firmalardaki artış dikkat çekicidir.



Şekil 4.17 : Gereksinim değişim yöntemleri

Şekil 4.18’de, gereksinimlerin yönetimi üzerine beyan edilen görüşler incelendiği zaman, 2008 yılında da 2001 yılında olduğu gibi, sürece “inançsız” katılımcıların azımsanamayacak oranda olması, hatta 2008 yılında artmış olması dikkat çekmektedir. Müşterilerin sürece katılımının yetersiz olması, sürecin yazılım geliştirenler tarafından yönlendirildiğini ve hataya açık olduğu izlenimini vermektedir. Önemli bir çoğunluğun “gereksinim derleme aracı” gerekliliğine inandığı gözlenmektedir. Veriler, firmaların yazılım çalışanı sayısına göre ayrıştırıldığında, gereksinim sürecine verilen önemin eldeki kaynaklarla orantılı olduğu yorumu yapılabilmektedir.



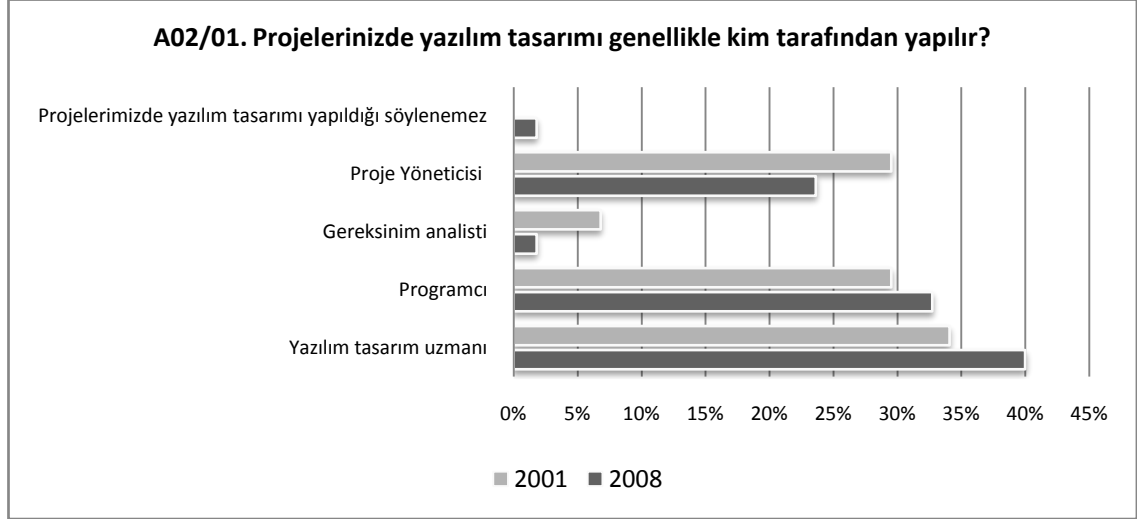
Şekil 4.18 : Gereksinimlerin yönetimi

4.3.2 Yazılım Tasarımı

Yazılım tasarımı, yazılım gereksinimlerini karşılayan bir ürünün gerçekleştirilmesine temel oluşturan yapıyı ve arayüzleri gerekli ve yeterli ayrıntı düzeyinde tanımlayan mühendislik çözümdür.

Şekil 4.19’da görüldüğü gibi, projelerde yazılım tasarımını 2001 yılında yeterli kaynak varsa “tasarımcı”, kaynak yoksa “en yetkin proje elemanı” olarak Proje Yöneticisi yapmaktadır. 2008 yılında Proje Yöneticisinin üzerindeki rol belli bir oranda tasarım

uzmanlarına kayarken oranı az da olsa yazılım tasarımı yapmadığını beyan eden firma bulunması dikkat çekici sonuçlardan biridir.



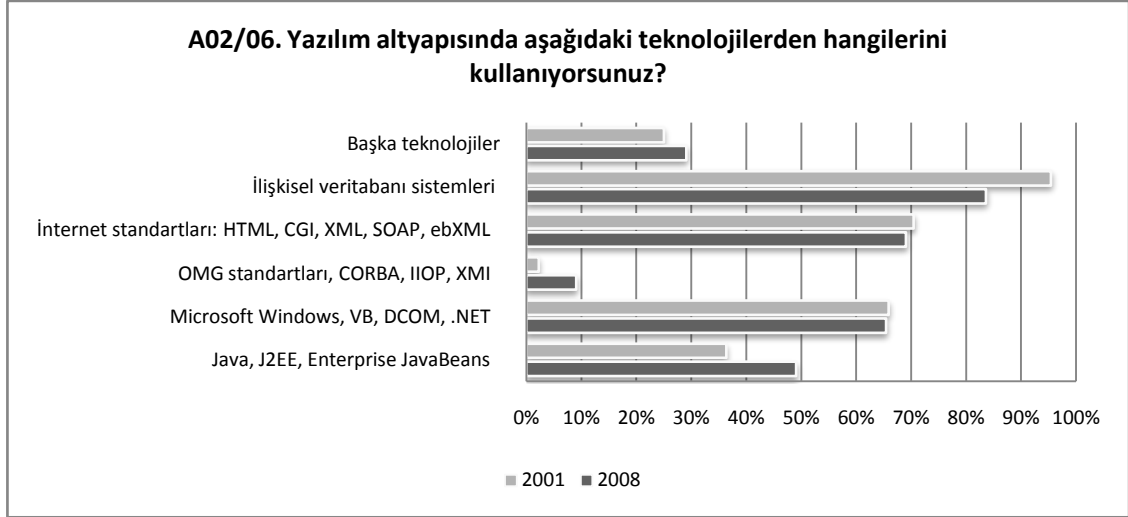
Şekil 4.19 : Yazılım tasarımı

2001 yılında yazılım tasarımı strateji ve yöntemleri içinde (Şekil 4.20); yaygınlaşma potansiyeli en yüksek yöntemlerden biri olarak kabul edilen UML (Unified Modeling Language) kullanımının kısıtlı olduğu, temel tasarım yöntemleri/tekniklerinin kullanılmakta olduğu incelenmiştir.

2008 yılında yöntemler içinde nesne yönelimli tasarım ilk sırayı alırken UML ve işleve yönelik tasarım sonraki sıraları almıştır.

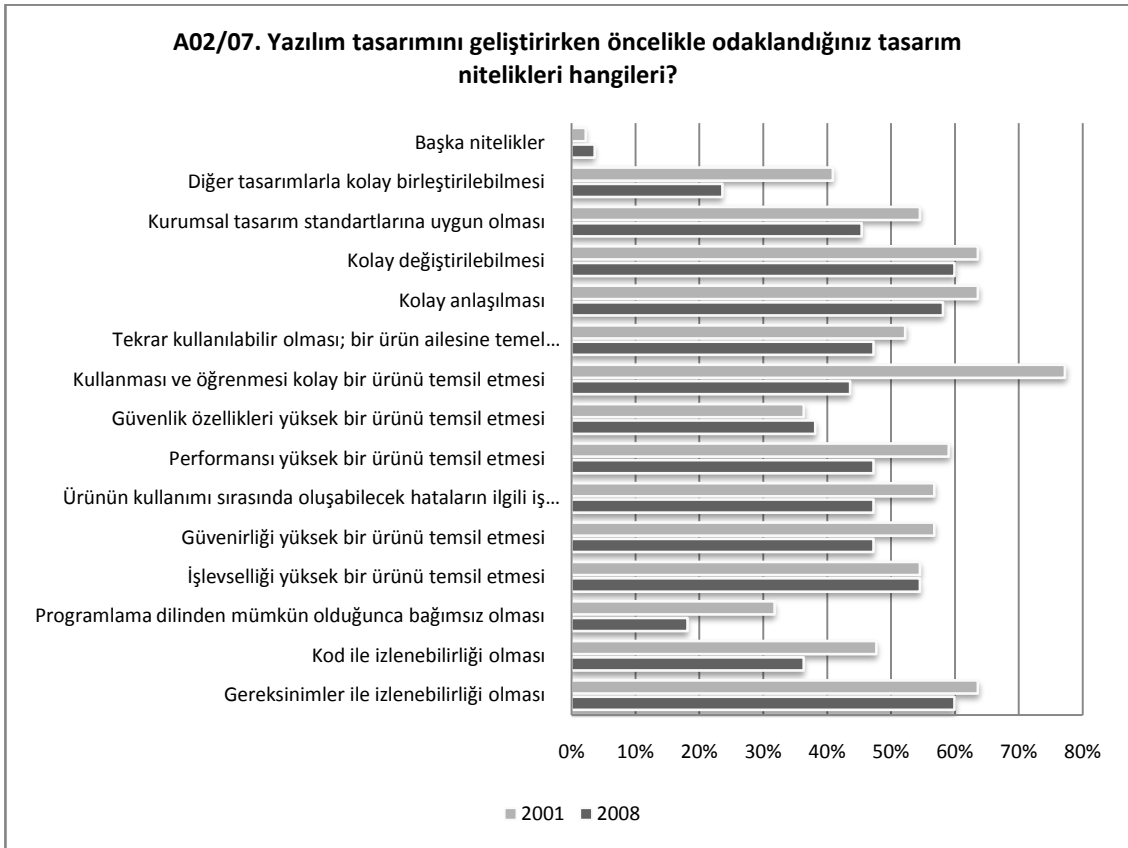
Anket sonuçları 2001 yılında yazılım altyapısında kullanılan teknolojiler itibarı ile; internet teknolojileri merkezli tasarımın ve dolayısı ile internet merkezli projelerin yaygınlığını vurgulamaktadır. En yüksek bilinirlik/kullanım oranı “veritabanları için ya da veritabanı merkezli tasarım” yönteminde ortaya çıkmaktadır.

2008 yılında da bu konuda sonuçlar 2001 yılı ile önemli ölçüde paralellik arzetymekte, sadece Java kullanımında artış görülmektedir.



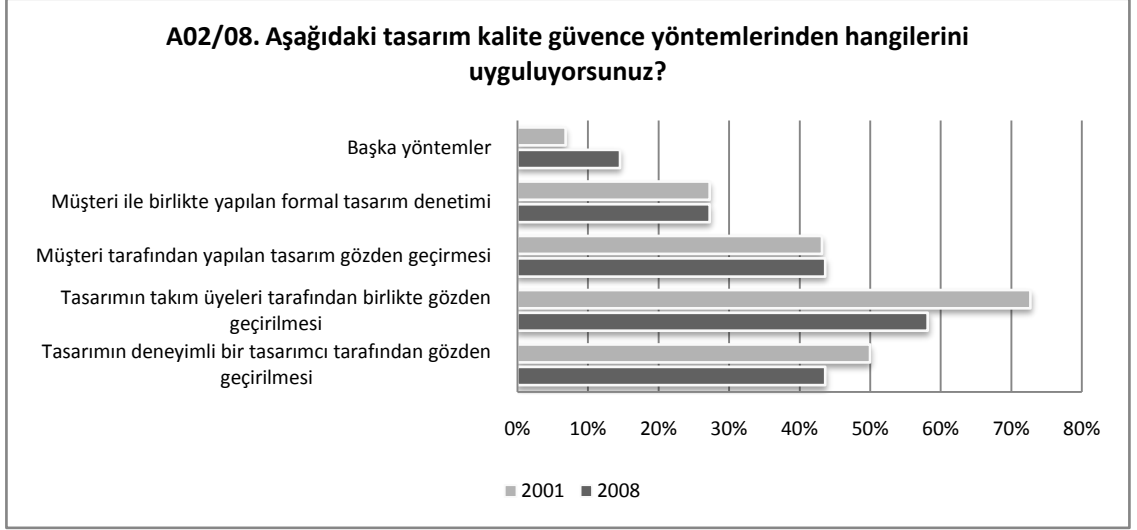
Şekil 4.20 : Yazılım teknolojileri

Odaklanılan tasarım nitelikleri itibarı ile de sonuçlar önemli ölçüde paralellik arz etmekle birlikte Şekil 4.21’de detayı görüldüğü şekilde, kullanım, öğrenme ve entegrasyon kolaylığı, kod ile izlenebilirlik gibi niteliklere odaklanma azalmıştır.



Şekil 4.21 : Tasarım nitelikleri

Şekil 4.22’de, tasarım kalite güvence yöntemleri içinde doğrulama tekniklerinin daha az uygulanır olduğu görülmektedir.

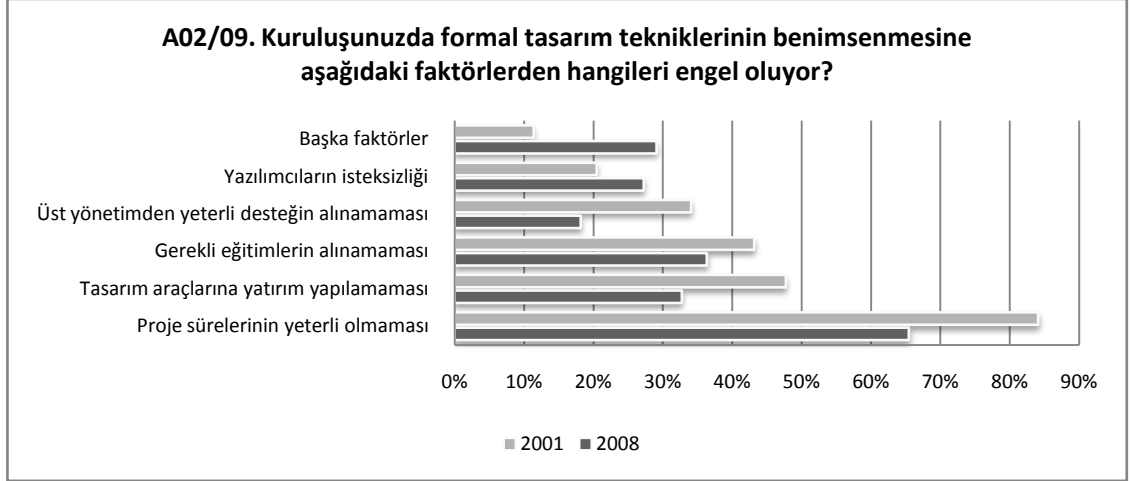


Şekil 4.22 : Tasarım kalite güvence yöntemleri

2001 yılında kuruluşlarda formal tasarım tekniklerinin benimsenme durumu incelendiğinde (Şekil 4.23), yüzde 85 oranda proje sürelerinin yetersiz olmasının tasarıma engel olduğu görülmektedir.

Bu konuda II. sorun tasarım araçlarına yatırım yapılmaması olarak beyan edilmiştir. Bu doğrultuda tasarım ile ilgili olarak; tasarımın uygun araçlar elde edilmeden ve eğitimler alınmadan kolay uygulanabilecek bir süreç olmadığı ve tasarım sürecinin gereksinim/analiz ve kodlama süreçleri arasına sıkıştığından ihmal edilme oranının yüksek olduğu söylenebilir.

Oranlarda değişiklikler olsa da 2008 yılında da sorunlar benzer yapıdadır.



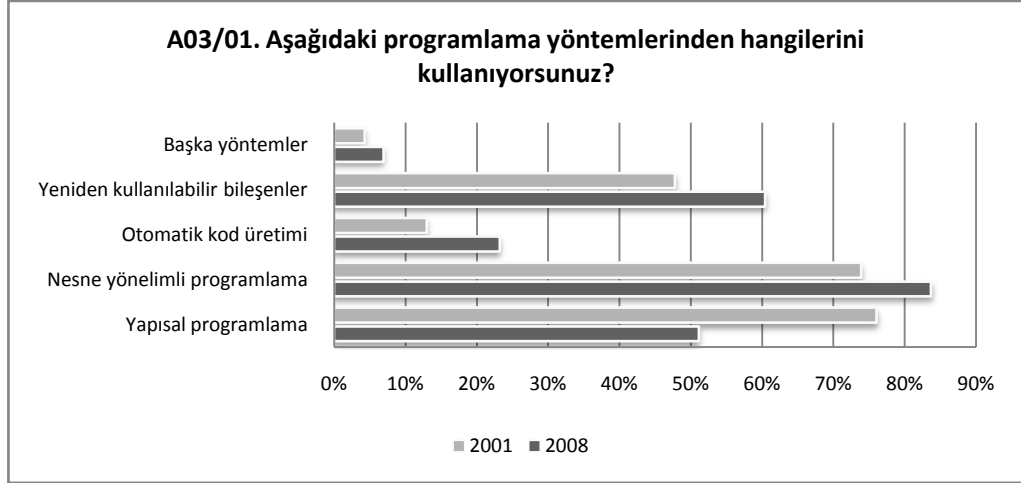
Şekil 4.23 : Formal tasarım teknikleri

Değerlendirme sonuçlarında, tasarım sürecinde de sürece verilen önemin yazılım kaynakları ile orantılı olduğu sonucu ortaya çıkmaktadır.

4.3.3 Yazılım Gerçekleştirme

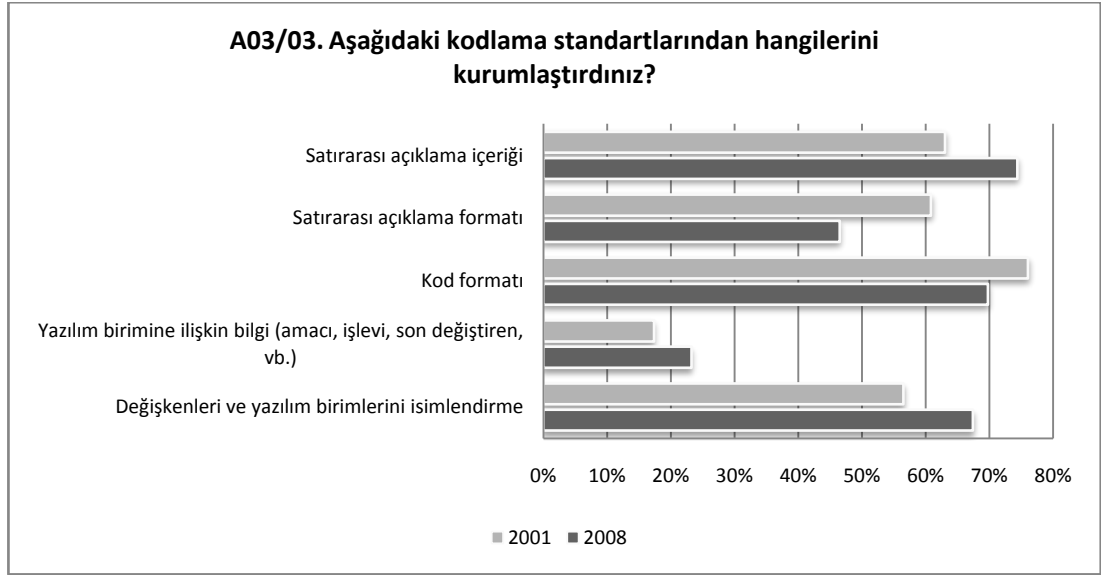
Yazılım gerçekleştirme, tasarımın öngördüğü yazılım birimlerinin üretilmesini ve sınanmasını, aşamalı olarak tümleştirilmesini ve müşteriye teslim edilecek ürün belgelerinin yazılmasını kapsar.

Programlama yöntemleri arasında “Otomatik kod üretimi” en az kullanılan yöntemdir (Şekil 4.24). En çok kullanılan yöntem nesne yönelimli programlama olup, yeniden kullanılabilir bileşenler artarken, yapısal programlama kullanımında azalma görülmektedir.



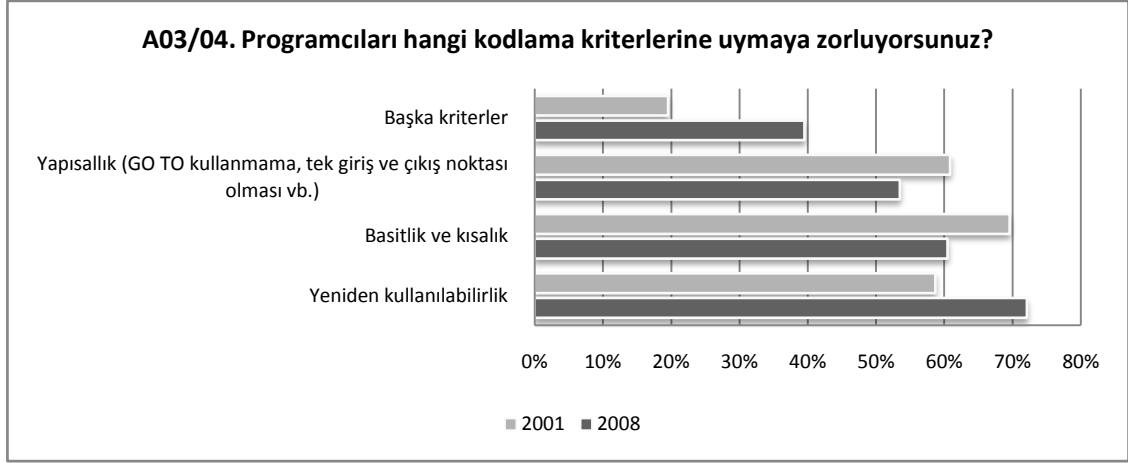
Şekil 4.24 : Programlama yöntemleri

Şekil 4.25’te kodlama standartları arasında kod içi dokümantasyonun çok yaygın olmadığı gözlenmekle birlikte her boyutta ve türde organizasyon, kaçınılmaz olarak kodlamada bazı standartlar uygulamaktadır.



Şekil 4.25 : Kodlama standartları

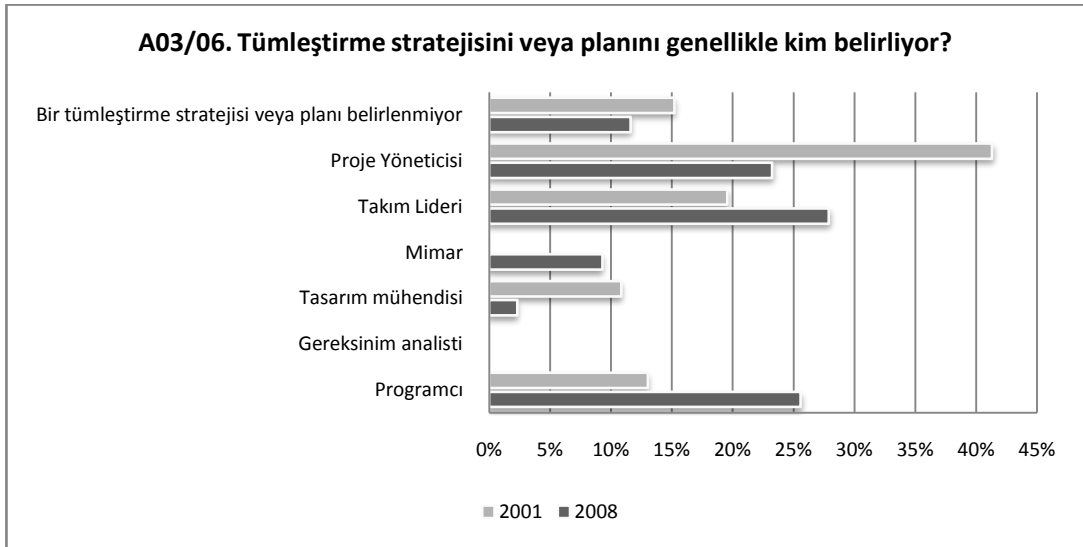
Şekil 4.26’da detayı verilen kodlama kriterleri içinde “Başka kriterler” seçeneğindeki artış dikkat çekicidir. Tanımlı kodlama kriterleri bulunmadığı anlamını da ifade ediyor olabileceği dikkate alınmalıdır.



Şekil 4.26 : Kodlama kriterleri

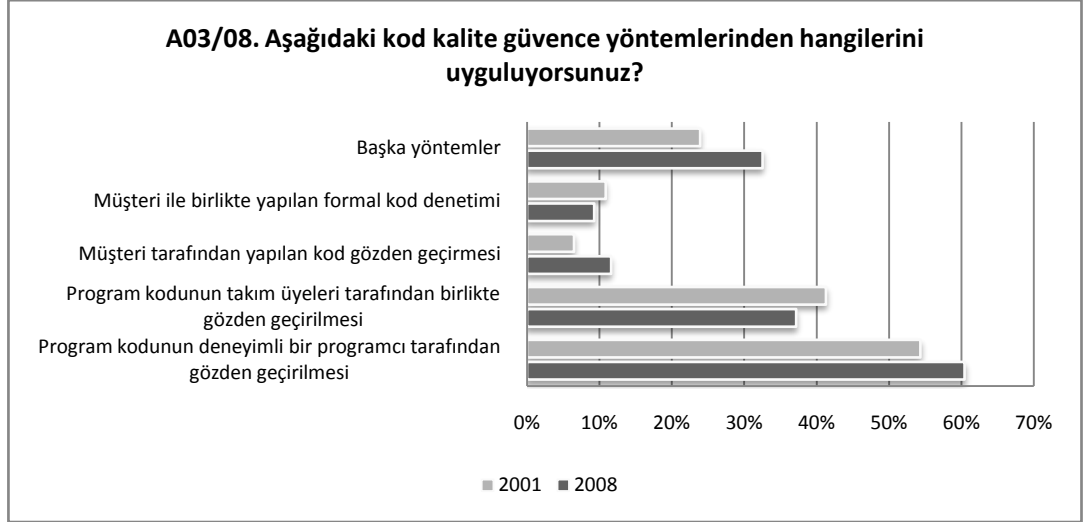
2001 yılı verilerine göre tümleştirme stratejisi veya planını belirleyen kişi olarak karşımıza Proje Yöneticisi çıkarken 2008 yılında bu rol Proje Yöneticisinden Programcı, Mimar ve Takım Liderine kaymaktadır (Şekil 4.27).

2001 yılı beyanları içinde ağırlıklı olarak gözlemlenen kodun tümleştirilmesinin “uzman” ellere teslim edilme yaklaşımı, 2008 yılında yerini daha önce uygulaması görülmeyen mimar gibi farklı rollere bırakmıştır. 2008 yılında tanımlı görevlerin dışındaki rollerin yerine oturmaya başladığı görülmektedir.



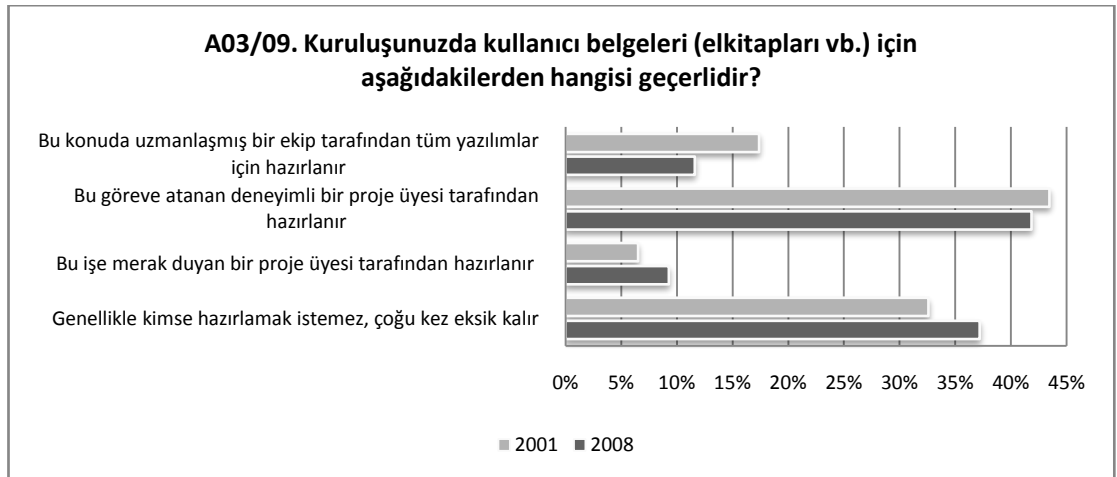
Şekil 4.27 : Tümleştirme stratejisi

Şekil 4.28’de kalite güvence yöntemlerinin kullanımına yönelik olarak; “Müşteri ile kod gözden geçirme” uygulamasının kullanılıyor olması dikkat çeken sonuçlardan biridir. Kodun gözden geçirilmesi çok yaygın olmasa da önemsenen bir uygulama olarak göze çarpmaktadır. Bu soru için başka yöntemler yanıtındaki artış, kod için kalite güvence yöntemi uygulamayan katılımcıların artışı olarak ta yorumlanabilir.



Şekil 4.28 : Kod kalite güvence yöntemleri

Geliştirmenin bir parçası olarak kullanıcı dokümantasyonuna gereken önemin verildiğini söylemek mümkün değildir (Şekil 4.29). Araştırmaya katılan firmaların yüzde 37’si kullanıcı belgelerinin hazırlanmadığını beyan etmektedir. Kullanıcı belgeleri hazırlanmasını sistematik olarak ele alan firma oranı yüzde 12 olarak gerçekleşmiştir.

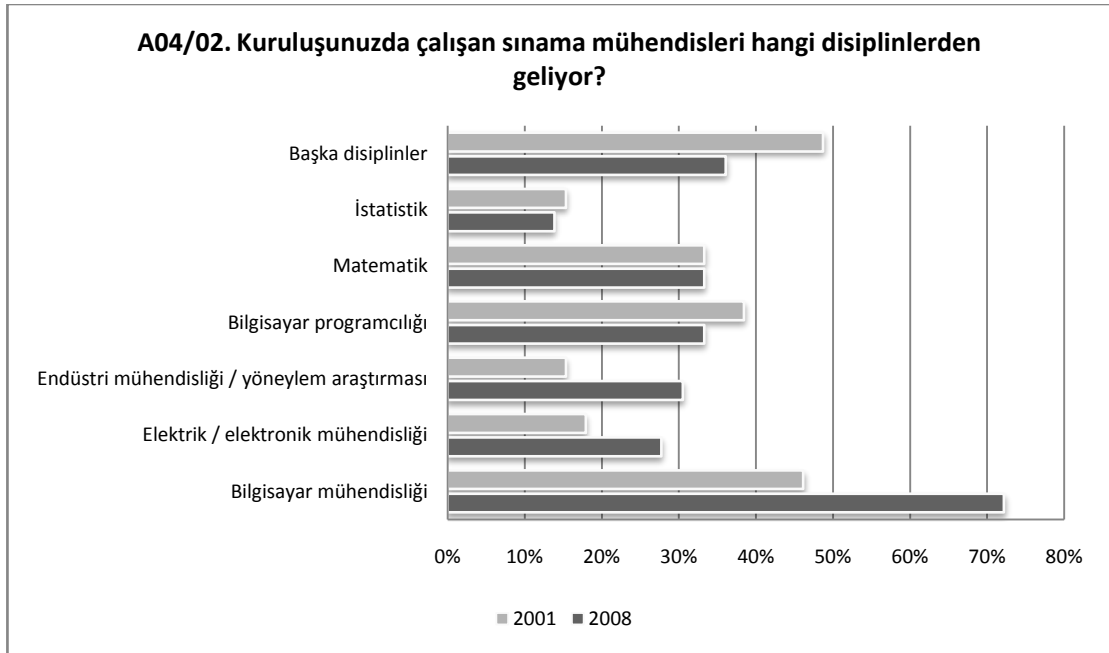


Şekil 4.29 : Kullanıcı belgeleri

4.3.4 Yazılım Sınama

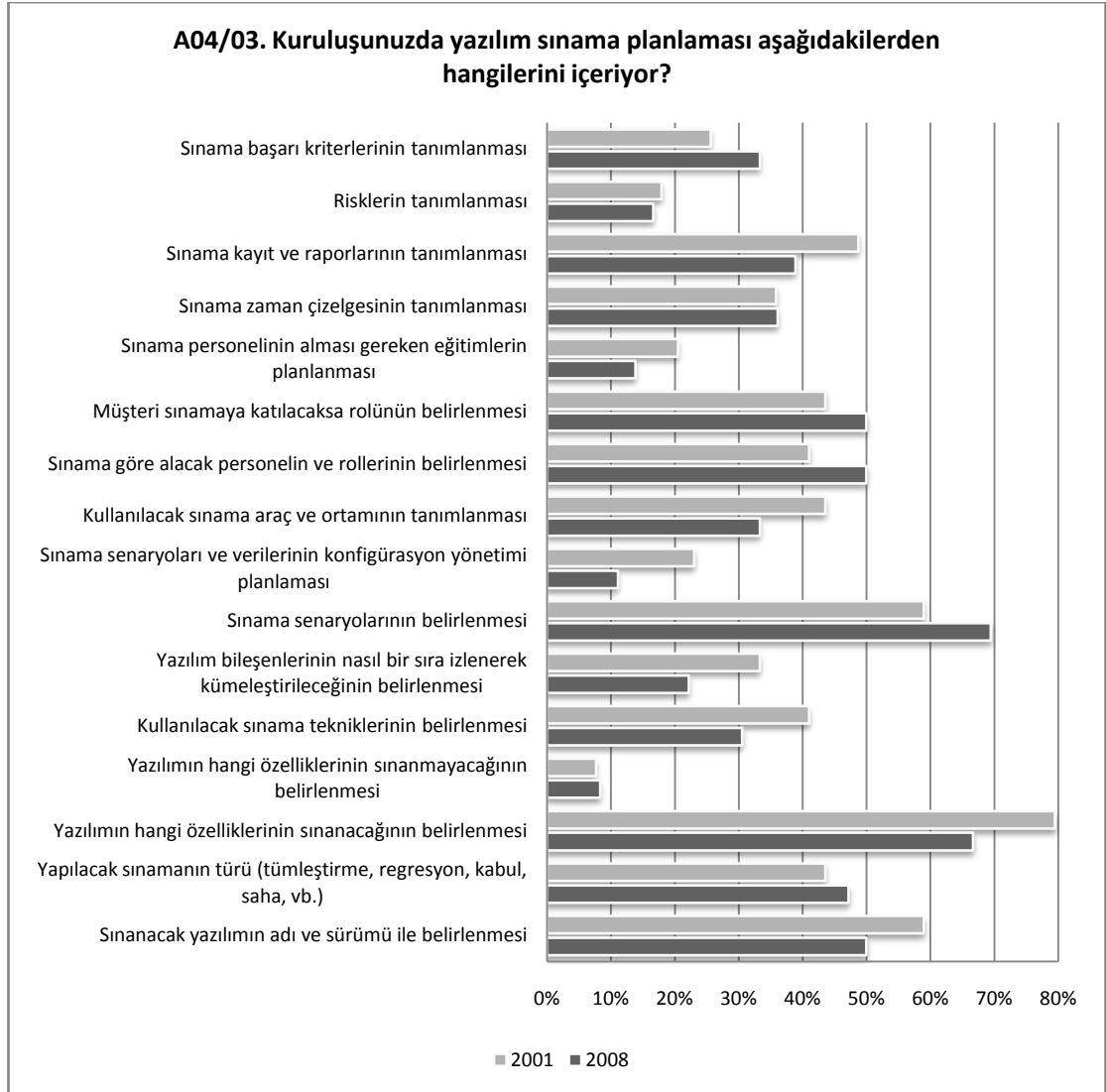
Yazılım sınama, gerçekleştirilen ürünün yazılım gereksinimlerini karşıladığının doğrulanmasıdır.

Şekil 4.30'da detayı görüldüğü şekilde, 2001 – 2008 yılları arasında; Sınama Mühendisleri içinde bilgisayar mühendislerinin oranı önemli ölçüde (yüzde 46 → yüzde 72) artmıştır. Süreçte başka disiplinlerden gelen katılımcı azalırken, endüstri mühendisleri yüzde 15, elektrik ve elektronik mühendisleri yüzde 10 artış göstermiştir. Bununla birlikte başka disiplinlerden gelen sınama mühendisi oranının azımsanmayacak düzeyde olması, yazılımın kara kutu (yazılımın kod, işleyiş ve mantığını bilmeksizin belirli girdiler için beklenen çıktılar ürettiğinin testidir) testlerini yapanların ağırlıklı olarak ilgili iş alanına mensup kişiler olmasından kaynaklıdır.



Şekil 4.30 : Yazılım sınama mühendisleri

Detayı aşağıda yer alan Şekil 4.31'de görüldüğü gibi 2008 yılı verileri içinde senaryoya dayalı test yaklaşımı öne çıkmış, bununla birlikte sınama sürecinde planlama ve rollerin önemi artmıştır.



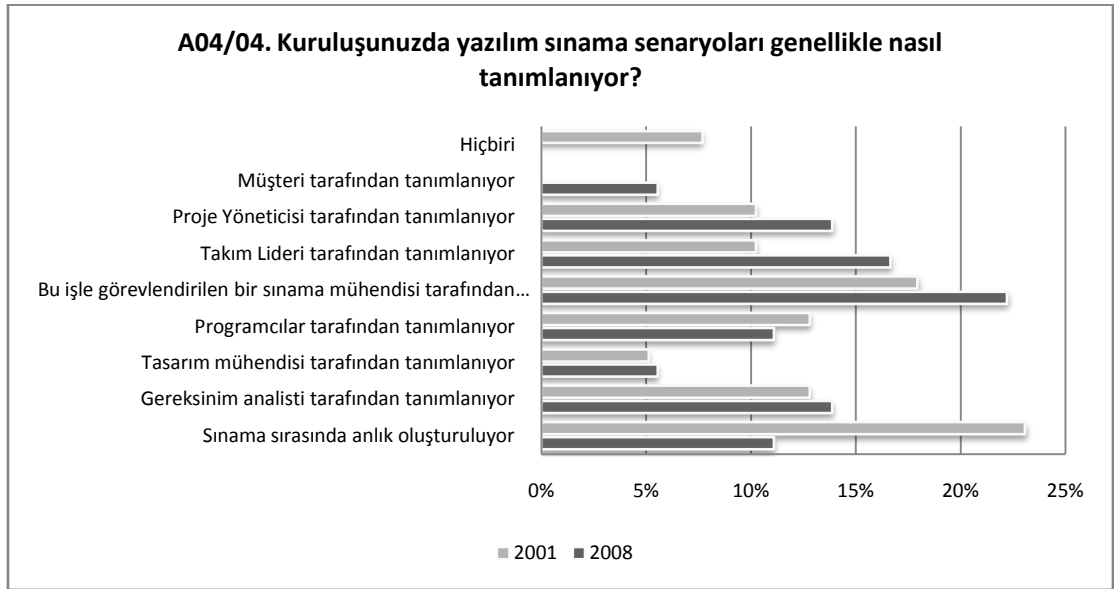
Şekil 4.31 : Yazılım sına planlaması

Şekil 4.32’de görüldüğü şekilde, 2001 yılı verileri içinde sına sürecinin diğere süreçlerle olan zorunlu bağlantısı diğere sonuçlarda kabul edilmekle beraber, bu sorularla aslında dikkate alınmadığı ortaya çıkmaktadır.

2008 yılında bu durum tümü ile değışmiş ve plansız/anlık oluşturulan senaryolar yerine bu iş ile görevlendirilen Sına Mühendisleri tarafından oluşturulan senaryolar ön plana çıkmıştır. Senaryo tanımlama yaklaşımının hangi rol tarafından üstlenildiğinden bağımsız olarak önemli ölçüde kabul gördüğü/arttığı görülmektedir. Oran yüzde 31’den

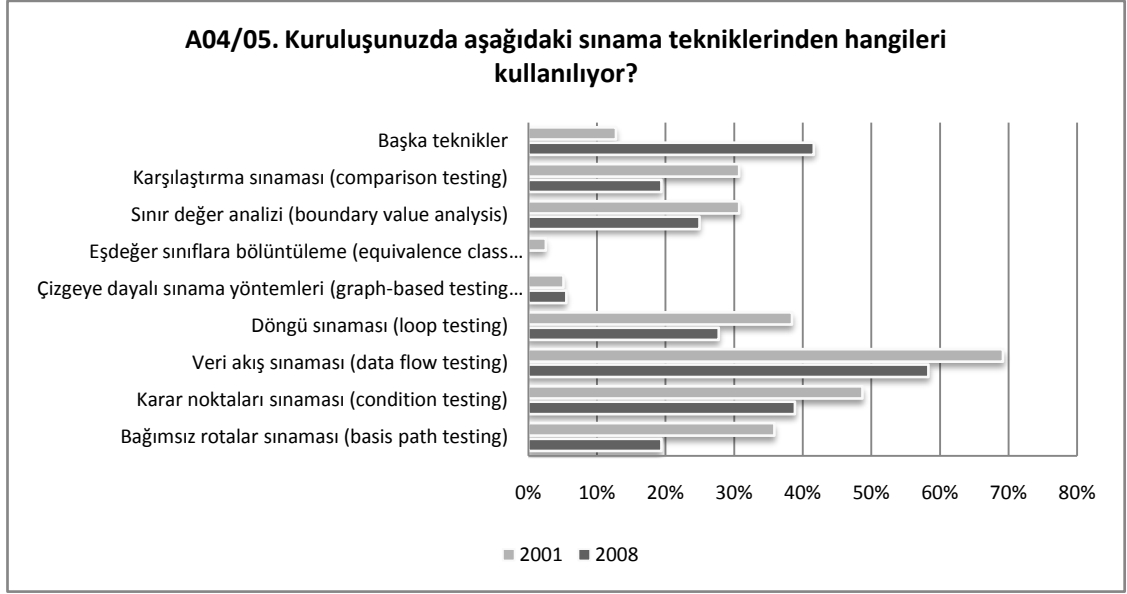
yüzde 8'e gerilemiştir (Sınama sırasında anlık oluşturuluyor ve hiçbiri seçenekleri toplamı).

Sınama senaryolarının yüzde 14 oranında Proje Yöneticisi, yüzde 17 oranında Takım Lideri tarafından tanımlanıyor olması yine proje kaynakları ile ilgilidir. Belli roller kaynak varsa paylaşmakta, olmadığı durumda en yetkin kaynaklar tarafından üstlenilmektedir.



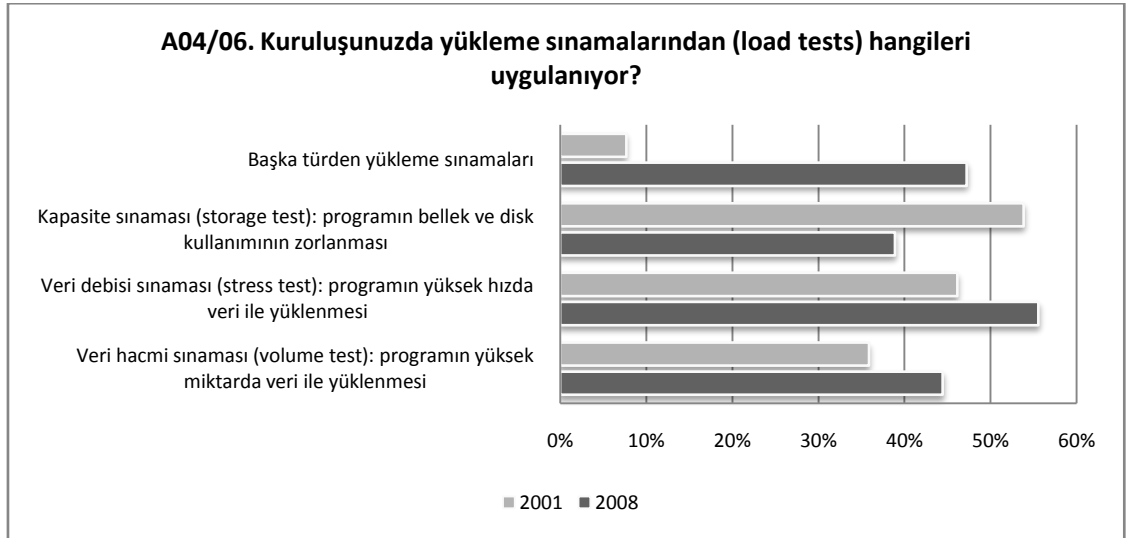
Şekil 4.32 : Sınama senaryoları

Şekil 4.33'te görüldüğü gibi, 2008 yılı verileri içinde, tanımlı sınama teknikleri dışında başka teknikler seçeneği, önemli oranda tercih edilmiştir. Bir sonraki araştırma için tanımlı teknikler gözden geçirilmeli ve gerekiyorsa başka teknikler seçeneğinin detayları katılımcıdan alınmalıdır.



Şekil 4.33 : Sına teknikleri

Benzer şekilde Şekil 4.34: Yükleme sınamaları'nda da başka tür sınamalar seçeneğini yanıtlayanlarda önemli oranda artış görülmektedir.



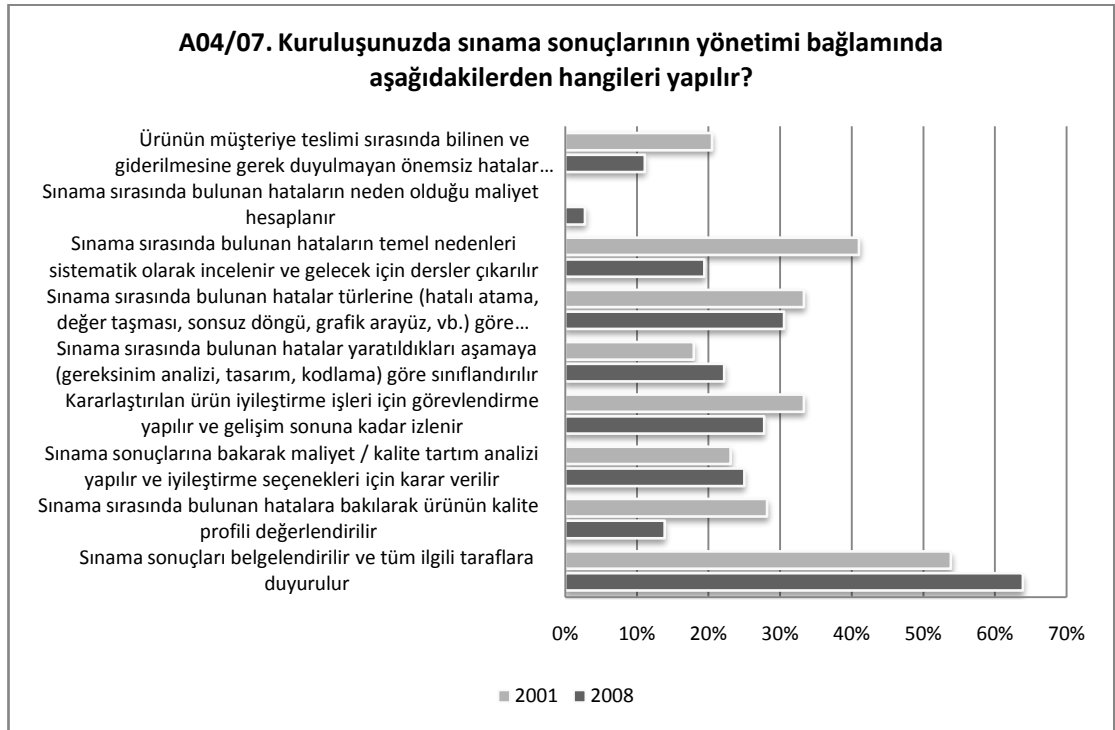
Şekil 4.34 : Yükleme sınamaları

Sına süreci yeniden işleme (rework) maliyetinin en açık hesaplanacağı süreç olmakla birlikte Şekil 4.35 ile özetlenen sonuçlar bu hesabın çok önemsenmediğini göstermektedir.

Sınama sürecinin sadece yazılım ürünlerinin sınaması olmayıp aslında tüm mühendislik süreçlerinin sınaması olduğu yaklaşımının yeterince kabul görmediği gözlenmektedir.

Temel süreç gerekleri göz önüne alındığında, sınama sonuçlarına gereken önemin verildiği söylenebilir. Sınama süreci hala “Sınama = Maliyet” yaklaşımı ile ele alınmaktadır.

Araştırma sonuçları içinde 2008 yılı verileri değerlendirildiği zaman süreç olgunluğunun yazılım kadrosu ile paralellik arzettiği görülmektedir. Yazılım organizasyonu ölçek olarak büyüdükçe süreç olgunluk düzeyi de artış göstermektedir.

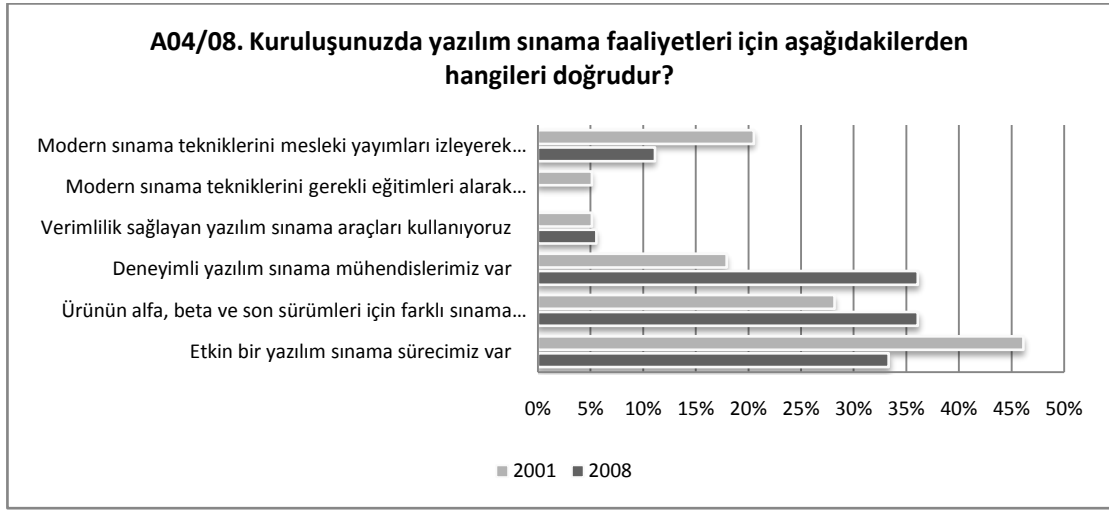


Şekil 4.35 : Sınama sonuçlarının yönetimi

Şekil 4.36 ile özetlenen sonuçlar, sınamada araç kullanımının yaygın bir uygulama olmadığını göstermektedir.

Kuruluşlar, sınama konusunda eğitim almaksızın süreci gerçekleştirmektedir. “Sınama = Maliyet” görüşü burada da ortaya çıkmakta ve araç ve eğitim için kaynak ayrılması ile sürece verilen önem düzeyi ortaya çıkmaktadır.

2008 yılında deneyimli yazılım sına mühendisi kullanımında önemli oranda artış (yüzde 18 → yüzde 36) olmakla birlikte “etkin bir yazılım sına sürecimiz var” diyenlerin oranı yüzde 10’un üzerinde azalmıştır. Bu durum yine yazılım organizasyonlarında iyi pratiklerin artışı ve 2008 yılında yanıtların/beyanların 2001 yılına göre daha tutarlı olduğunun, toplamda ele alınan süreçlerin tümü için bilinç ve farkındalığın arttığıının göstergesidir.

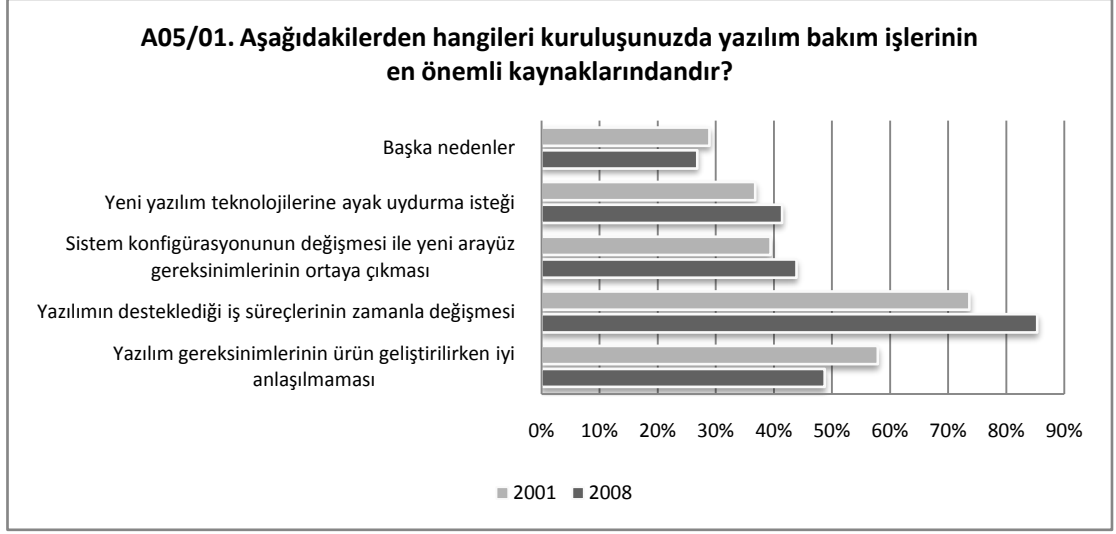


Şekil 4.36 : Yazılım sına faaliyetleri

4.3.5 Yazılım Bakımı

Yazılım bakımı, teslim edilen bir yazılım ürününde hata giderme, performans iyileştirme ve değişen ortama uyarlama amacıyla değişiklikler yapılmasıdır.

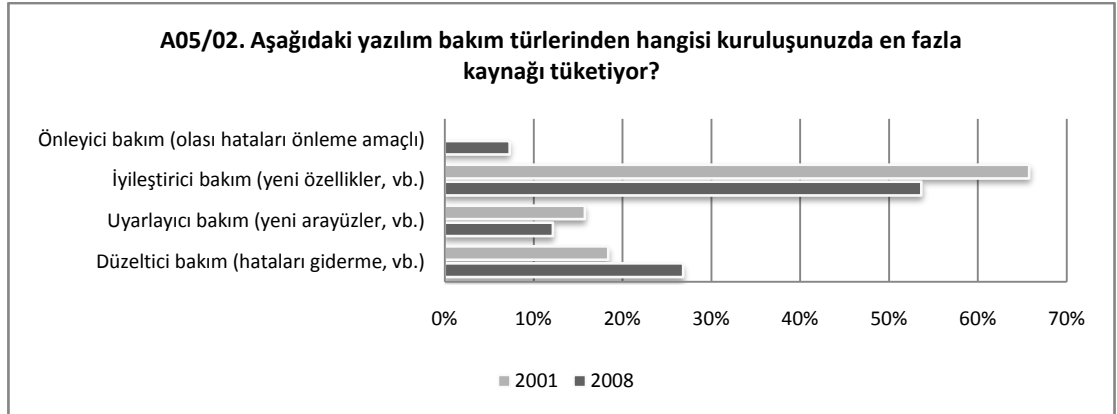
Yazılım bakım işlerinin kaynağı değerlendirilirken, bakım ve gereksinim süreçleri arasındaki doğal ilişki Şekil 4.37’de sonuçlara yansımaktadır. Bakım sürecindeki işlerin sebeplerine bakıldığı zaman 2001 – 2008 yılları arasında sonuçlar önemli ölçüde benzerlik arz etmektedir.



Şekil 4.37 : Yazılım bakım işlerinin kaynağı

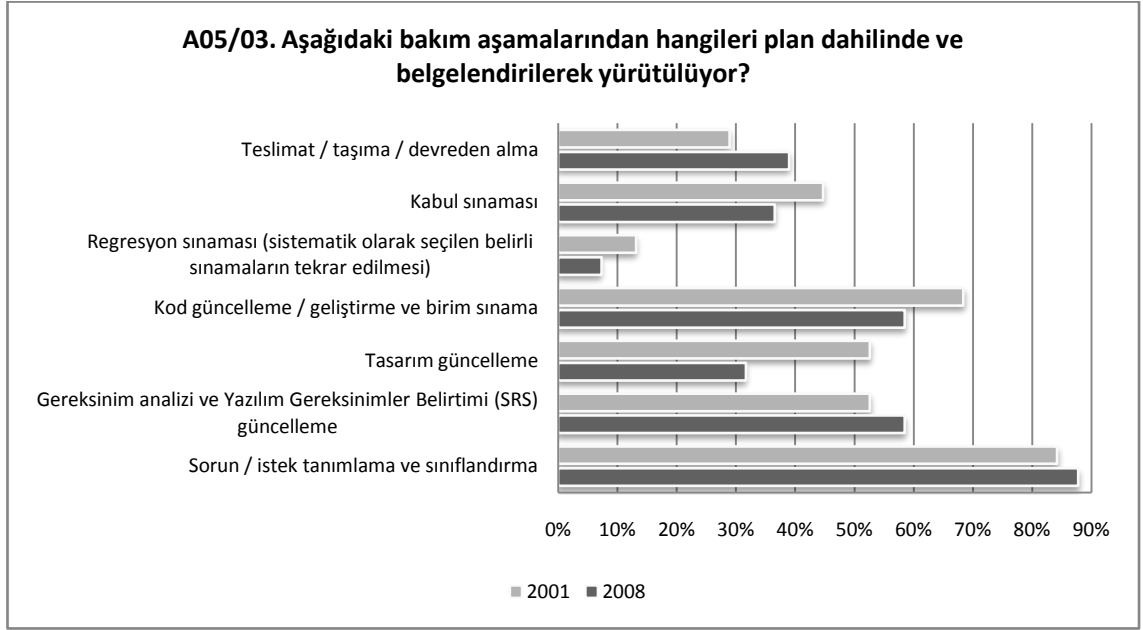
Detayı Şekil 4.38’de yer alan yazılım bakım türleri sorgulandığında, en büyük yüzdeyi (2001 - yüzde 65 → 2008 - yüzde 54) iyileştirici bakım (yeni özelliklerin eklenmesi) almaktadır.

Beyanlar arasında “önleyici bakım” yapan firma oranı çok düşük düzeydedir (ağırlıklı olarak iç kullanım için yazılım geliştiren firmalar). Önleyici bakıma bu derece düşük önem veriliyor olması dikkat çekicidir.



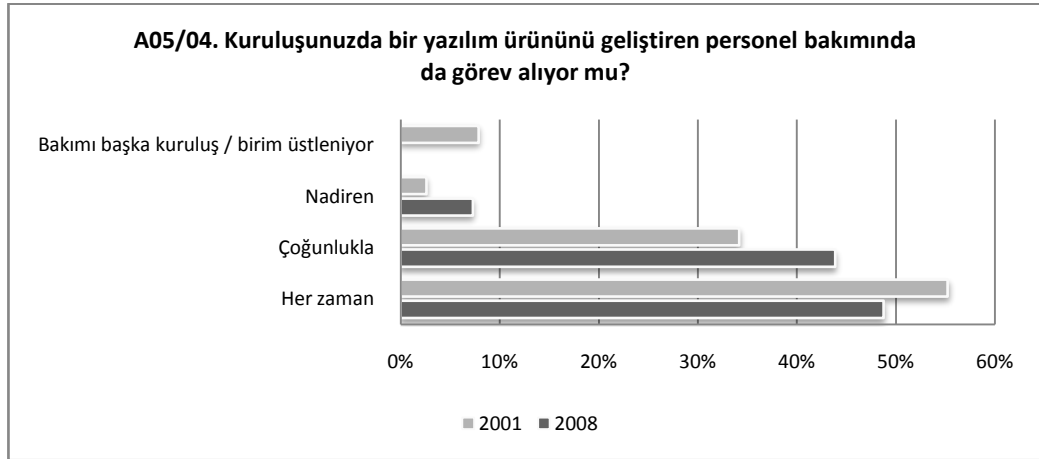
Şekil 4.38 : Yazılım bakım türleri

Şekil 4.39 ile verilen sonuçlar, bakım aşamasında yazılım tasarımının güncelliğinin sağlanmadığını göstermektedir. 2008 yılı yanıtları içinde tasarımı bakım aşamasında güncellediğini beyan eden firmaların oranı yüzde 31’e kadar gerilemiştir.



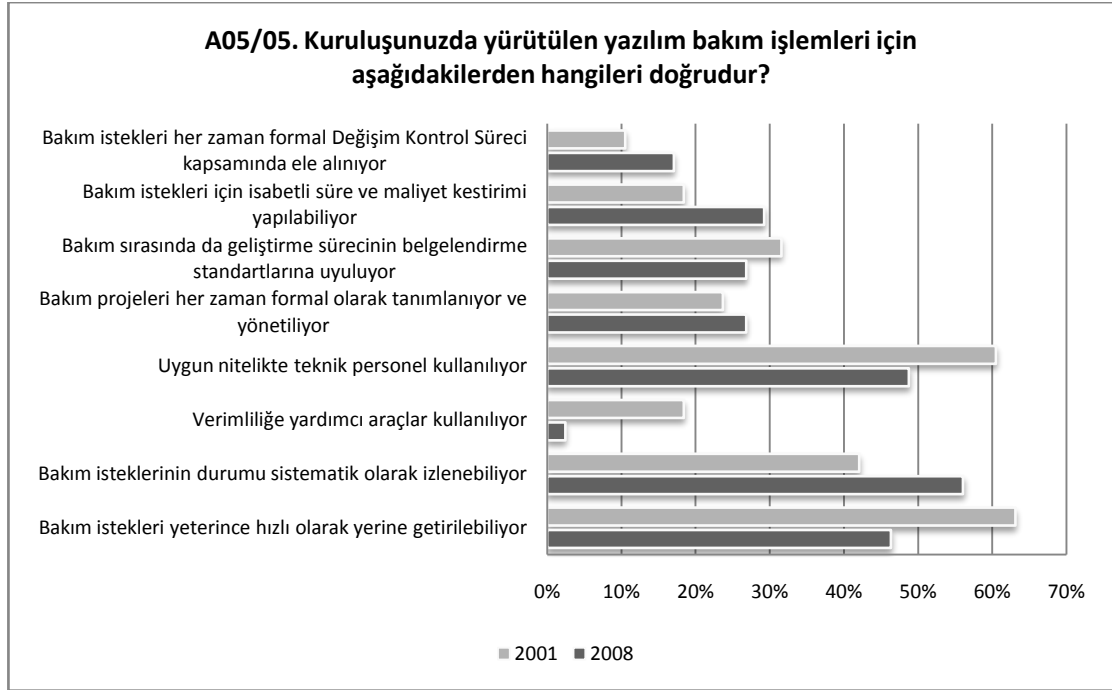
Şekil 4.39 : Yazılım bakım aşamaları

Şekil 4.40’da yer alan sonuçlar, yazılım bakımının geliştirmeyi yapanların dışında başka ekiplere devredilemediğini göstermektedir. Proje ekibinin bakımda görev alması formal/teknolojik yöntemlere ağır basmaktadır.



Şekil 4.40 : Bakımı yapan çalışanlar

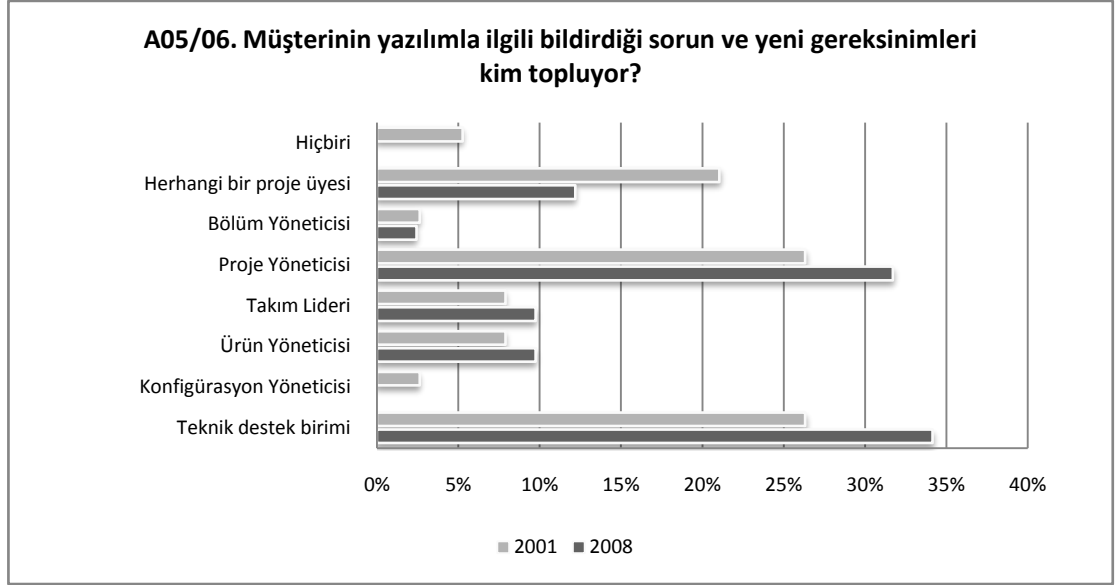
“Şekil 4.41: Yazılım Bakımı”nda yer alan sonuçlara göre bakım maliyetlerinin bilinmesi yeterince önemsenmemektedir. Bakım sürecinde araç kullanımı çok düşük orandadır ve bakım istekleri için genellikle “Değişim Kontrol Süreci” uygulanmamaktadır.



Şekil 4.41 : Yazılım bakımı

Şekil 4.42 ile açıklanan sonuçlara göre, bakım süreci için sorun ve gereksinimler Proje Yöneticisi/Takım Lideri düzeyinde ele alınmaktadır.

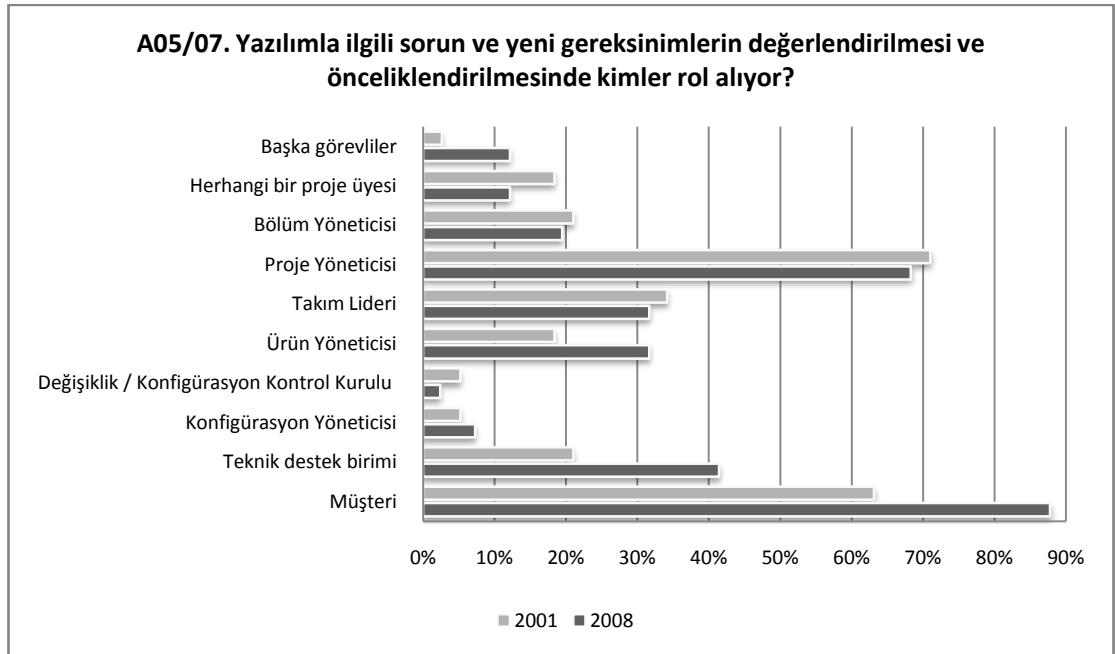
Bu durum, bakıma verilen önem açısından olumlu bir görüntü sunmakla beraber, bakım maliyetleri ve bakımın formal değişim sürecinin bir parçası olması açısından önemli eksikliklere işaret ettiği söylenebilir.



Şekil 4.42 : Yazılım bakımı – sorun ve yeni gereksinimler

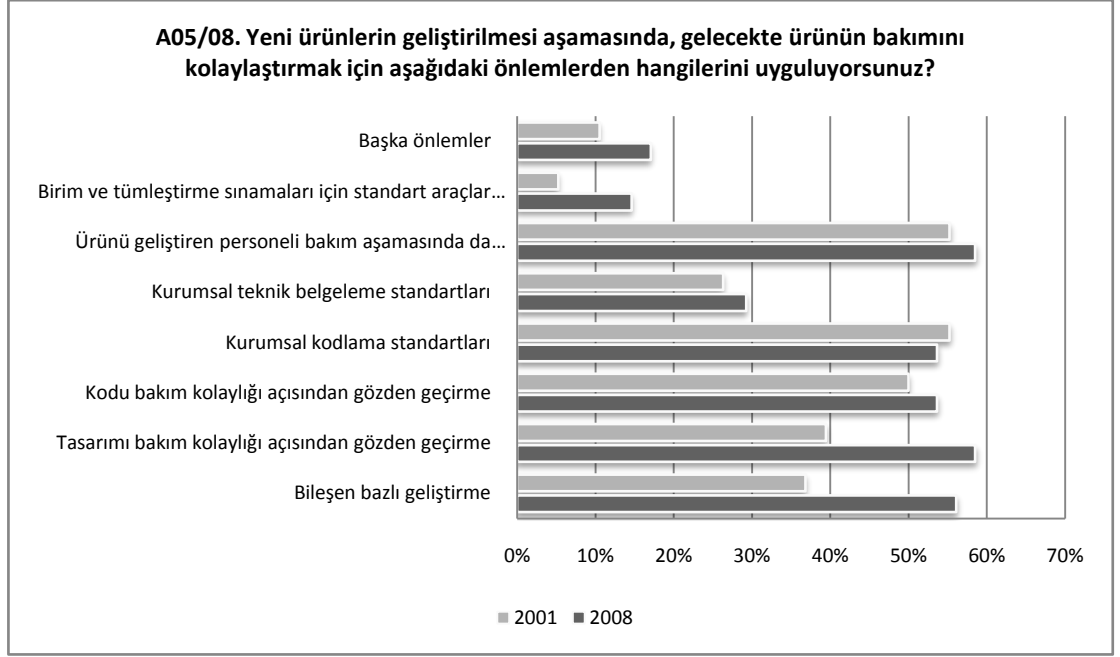
Şekil 4.43'te görüldüğü şekilde, gereksinimlerin değerlendirilmesi ve önceliklendirilmesinde en büyük oranda (yüzde 88) müşteri rol almaktadır.

Bu durum müşteriye verilen önem açısından önemli olmakla birlikte değişiklik veya konfigürasyon kontrol kurullarının süreçte neredeyse hiç yer almaması dikkat çekicidir.



Şekil 4.43 : Yazılım ile ilgili sorun ve gereksinimlerin değerlendirilmesi

Şekil 4.44'te detayı yer alan, bakımı kolaylaştırmak için alınacak önlemler içinde en yüksek oranda görülen yöntem olan, ürünü geliştiren personelin bakımda da görevlendirilmesi bu süreçte kurumsal yaklaşımların yer almama durumunun en önemli göstergesidir.

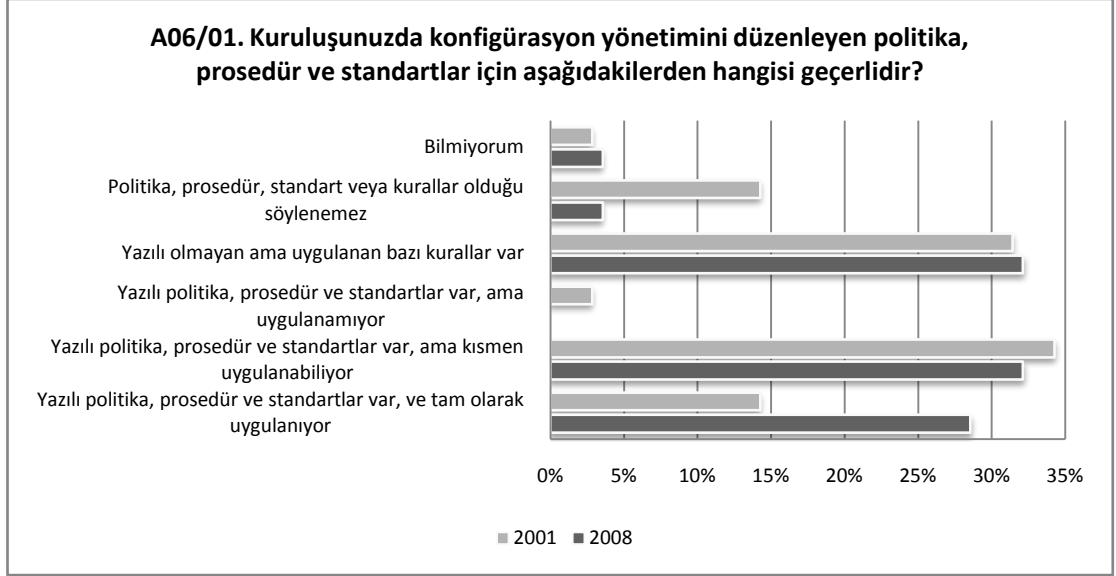


Şekil 4.44 : Yazılım bakımını kolaylaştırmak

4.3.6 Yazılım Konfigürasyon Yönetimi

Yazılım konfigürasyon yönetimi, bir yazılım ürününün işlevsel ve yapısal bütünlüğünü kurmak ve sürdürmek için yürütülen faaliyetlerdir.

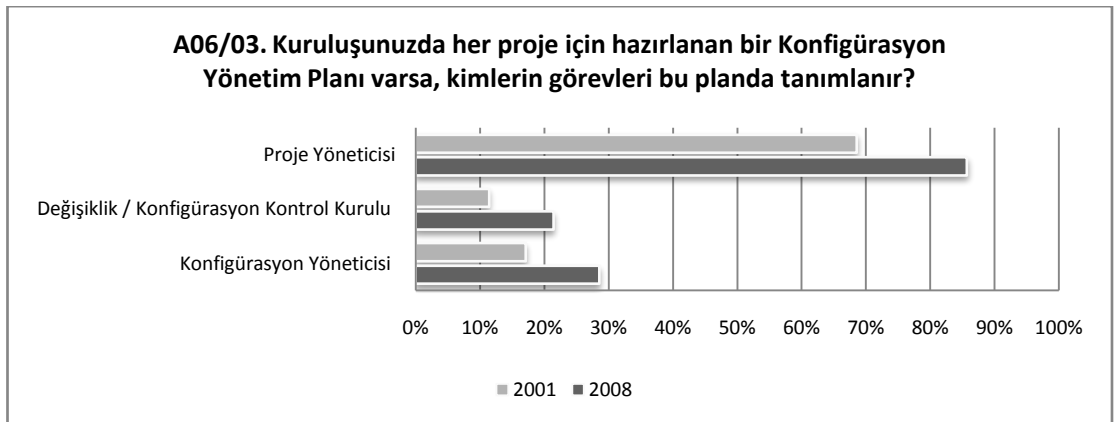
Şekil 4.45: Kuruluşlarda konfigürasyon yönetimini düzenleyen politika prosedür ve standartlar gözönüne alındığında; anket katılımcısı firmalarda yüzde 30 dan büyük oranda yazılı ama kısmen uygulanabilen politika, prosedür ve standartlar görülmektedir. Yine yüzde 30 dan büyük oranda yazılı olmayan ancak uygulanan kurallar olduğu görülmektedir. Yazılı politika, prosedür ve standartları olup bunları tam olarak uygulayabilen firmaların oranı ise 2008 yılında yüzde 14'ten yüzde 28'e yükselmiştir.



Şekil 4.45 : Konfigürasyon yönetimi politika, prosedür ve standartları

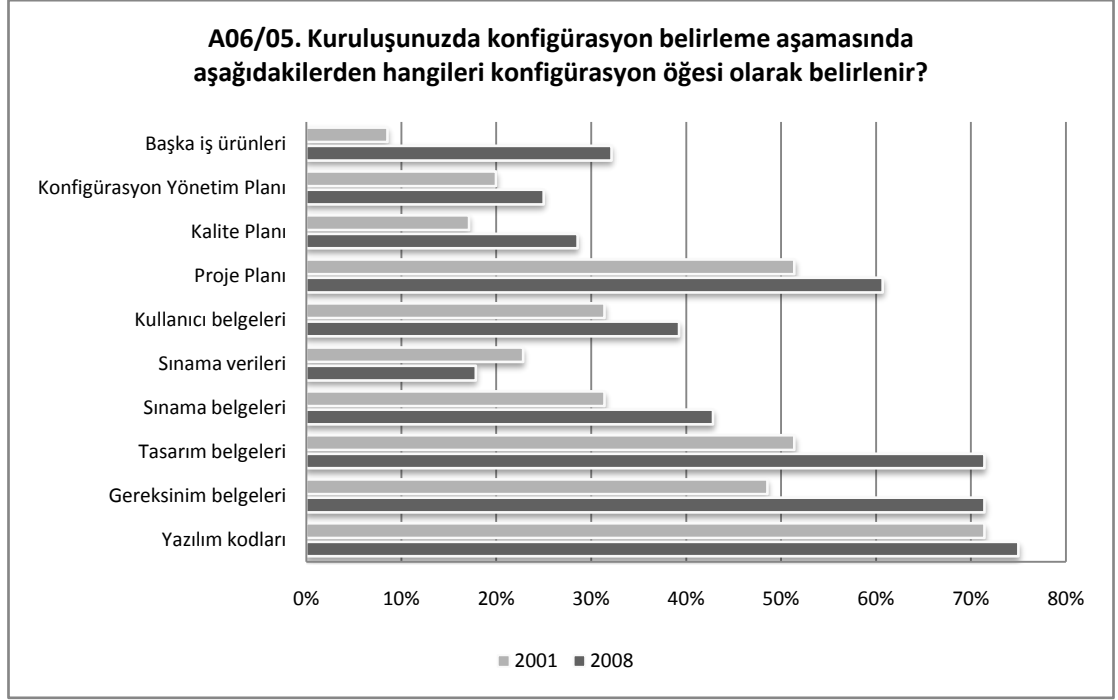
Detayı Şekil 4.46 ile açıklanan, Konfigürasyon Yönetim Planı içinde tanımlı görevler analiz edildiğinde; 2008 yılı verileri içinde en çok (yüzde 86 oranında) “Proje Yöneticisi” rolünün tanımlı olduğu görülmektedir. “Proje Yöneticisi” rolünün algılamadaki farklılığı konfigürasyon yönetimi süreci içinde de dikkat çekmektedir.

Projelere ait konfigürasyon planları içinde yüzde 28 oranında “Konfigürasyon Yöneticisi” ve yüzde 22 oranında “değişiklik/ konfigürasyon yönetim kurulu” yer almaktadır. 2008 yılında 2001 yılına göre, her üç rol de proje planları içinde daha yüksek oranda tanımlanmıştır.



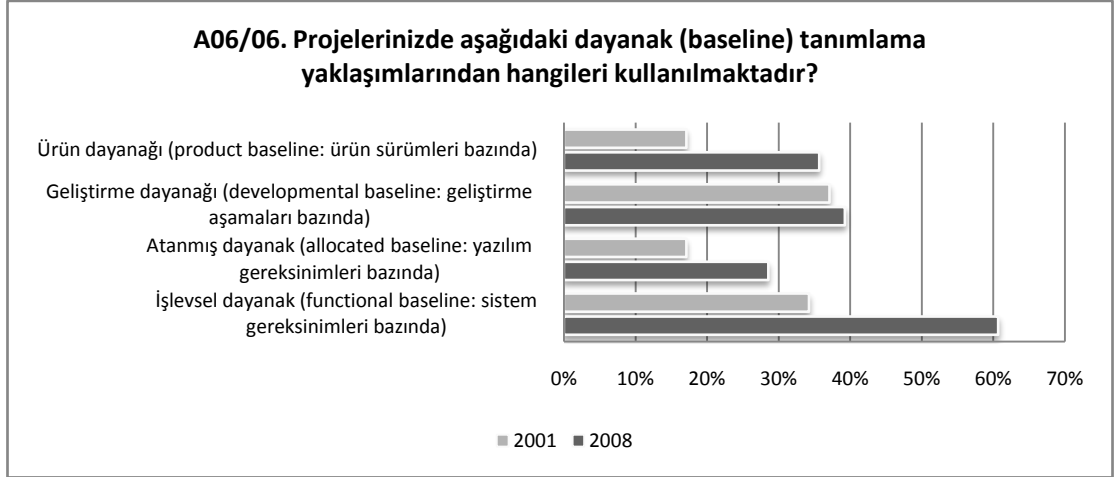
Şekil 4.46 : Konfigürasyon yönetim planı

Konfigürasyon öğeleri içinde (Şekil 4.47) yazılımın kaynak kodlarının yanısıra gereksinim, tasarım, sinama ve kullanıcı belgeleri ile birlikte planlar daha yüksek oranda tanımlanır olmuştur. Ancak burada konfigürasyon öğesi olarak tanımlanan başka iş ürünlerindeki artışı yorumlamak mümkün olmamıştır.



Şekil 4.47 : Konfigürasyon öğeleri

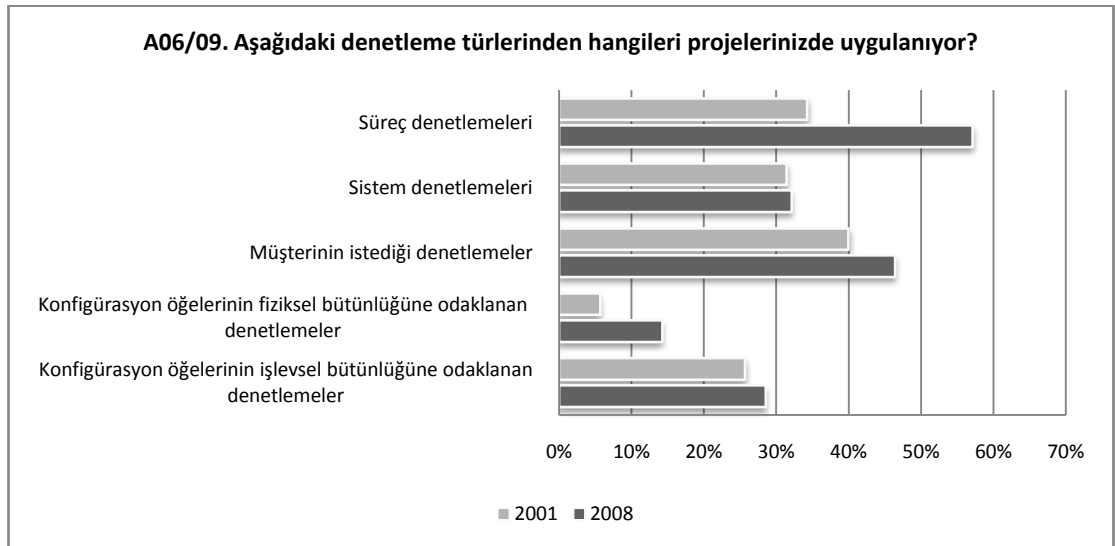
Detayı Şekil 4.48’de yer alan, projelerde dayanak tanımlama yaklaşımları incelendiğinde; 2001 yılı verileri içinde en çok geliştirme aşamaları bazında ve sistem gereksinimleri bazında işlevsel dayanak yaklaşımları görülürken, 2008 yılında sistem gereksinimleri bazında işlevsel dayanak yaklaşımı en çok uygulanır olmuştur.



Şekil 4.48 : Konfigürasyon dayanak tanımlama yaklaşımları

Projelerde konfigürasyon yönetimi itibarı ile uygulanan denetlemeler (Şekil 4.49); 2001 yılında “müşterinin istediği denetlemeler”, “süreç denetlemeleri” ve “sistem denetlemeleri” sırası ile yüzde 31 - yüzde 40 arası oranlarda gerçekleşmektedir.

2008 yılında konfigürasyon denetleme yaklaşımı önemli ölçüde artarken, en çok yapılan denetlemelerin, süreç denetlemeleri (yüzde 57) ve müşterinin istediği denetlemeler (yüzde 46) olduğu görülmektedir.

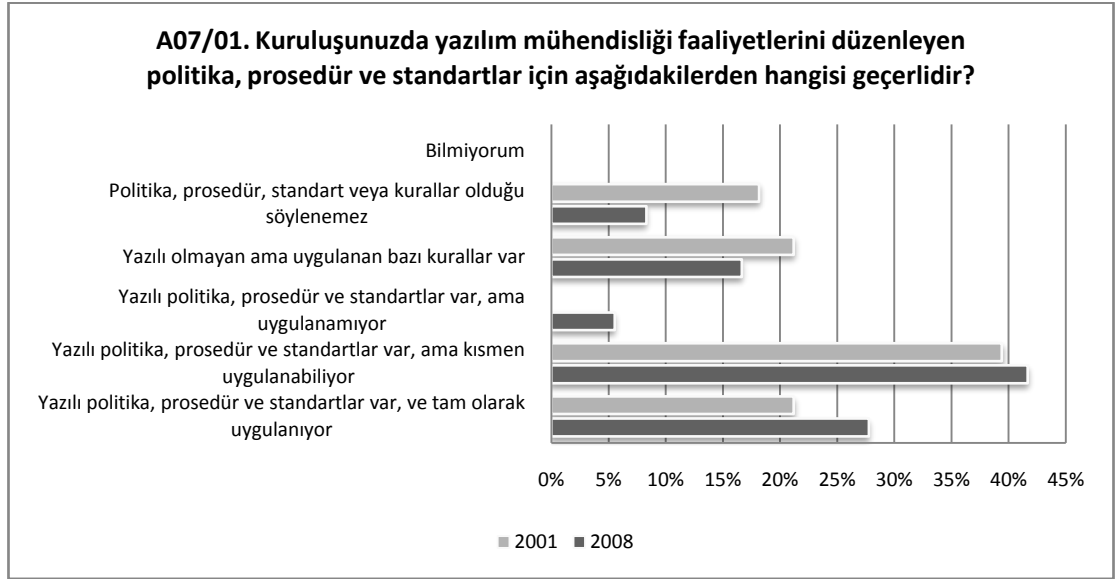


Şekil 4.49 : Konfigürasyon denetleme türleri

4.3.7 Yazılım Mühendisliği Yönetimi

Yazılım mühendisliği yönetimi, yazılım geliştirme ve bakımı bağlamında yürütülen organizasyonel yapılanma, planlama, iletişim, ölçme ve kontrol faaliyetlerini kapsar.

“Şekil 4.50: Yazılım Mühendisliği Politika, Prosedür ve Standartları”nda görüldüğü şekilde, bu sürecin ortalaması SPICE II. düzeyi göstermekte, diğer sorulara verilen yanıtlar da bu durumu desteklemektedir.

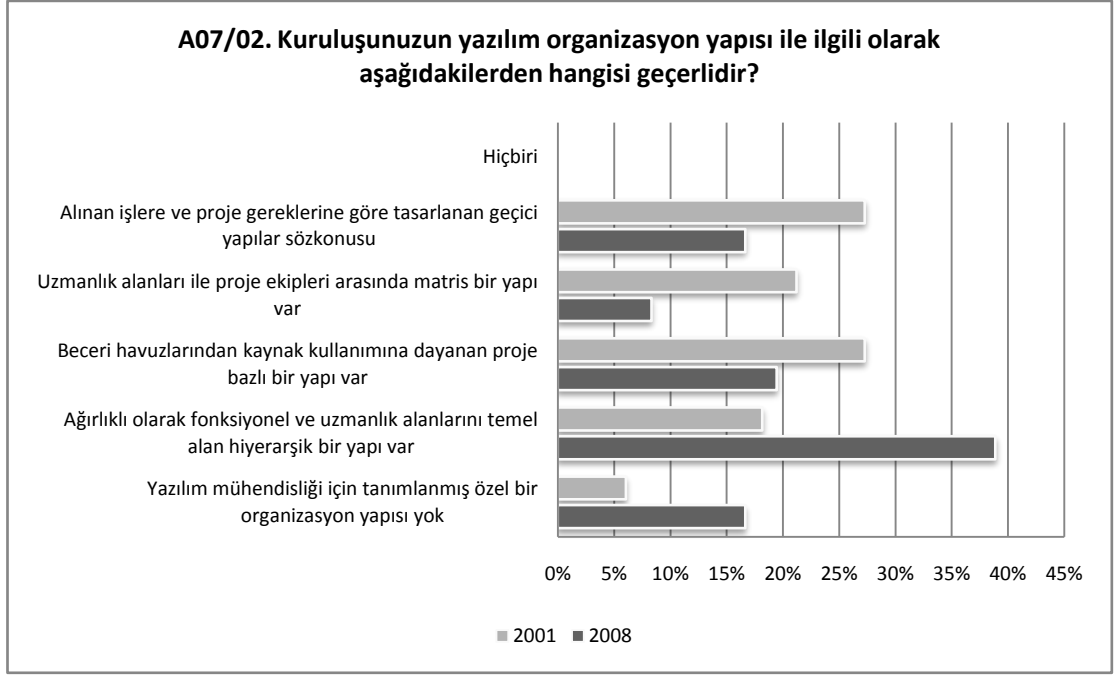


Şekil 4.50 : Yazılım mühendisliği politika, prosedür ve standartları

Şekil 4.51 ile ele alınan sonuçlarda, kuruluşlarda yazılım organizasyon yapılarının, yazılım mühendisliği için tanımlanmış organizasyon yapıları olmadığı görülmektedir.

2001 yılı sonuçları içinde, yüzde 27 oranında “Alınan işler ve proje gereklerine göre tasarlanan geçici yapı” ve aynı oranda “Beceri havuzlarından kaynak kullanımına dayalı proje bazlı yapı” yer almaktadır.

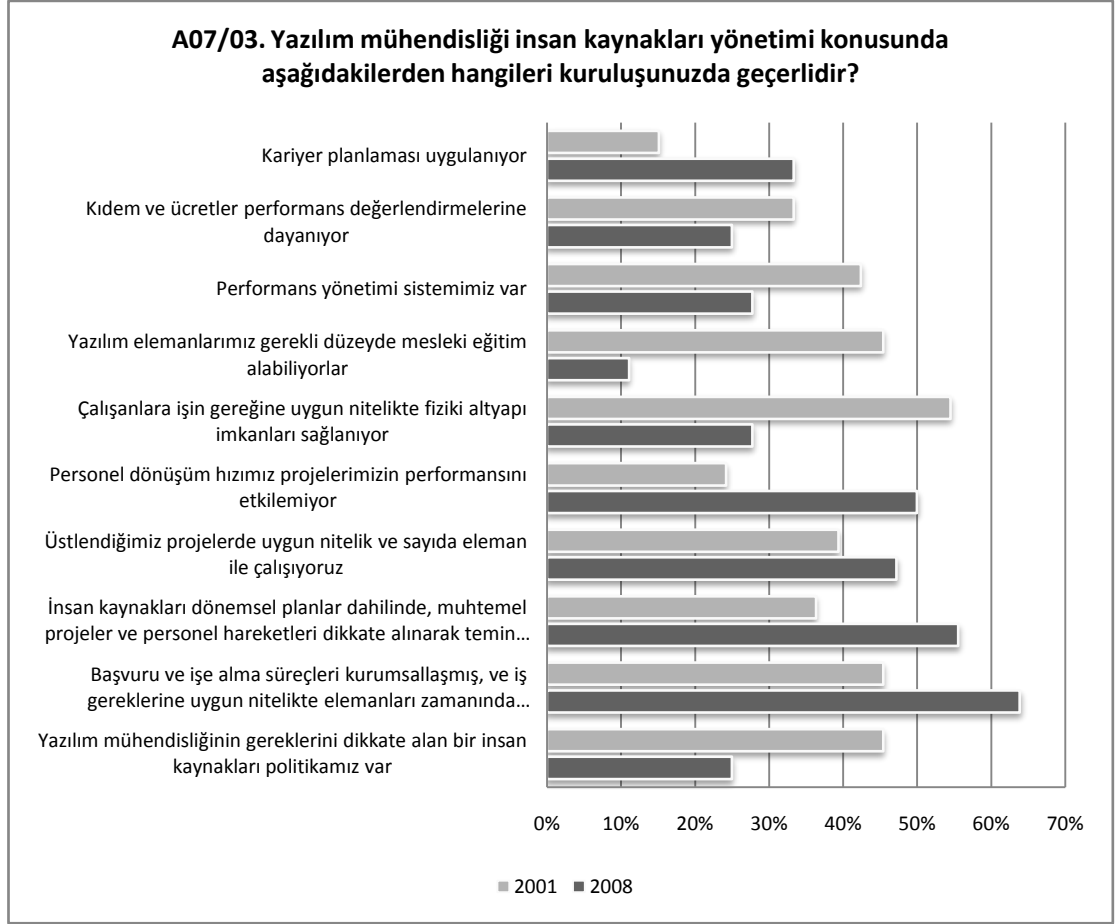
2008 yılı sonuçları içinde ise en çok (yüzde 39) “Fonksiyonel ve uzmanlık alanlarını temel alan hiyerarşik yapı” bulunmaktadır.



Şekil 4.51 : Yazılım organizasyon yapısı

Şekil 4.52 ile açıklanan 2001 yılı yanıtları içinde; yazılım kadrosu 50+ kişi olan firmalarda, çalışanlara sağlanan olanakların genellikle ortalamanın üzerinde olduğu görülmektedir. Kariyer planlama uygulandığını söyleyenlerin oranı sadece yüzde 15, personel dönüşüm hızının projelerin performansını etkilemediğini düşünenlerin oranının yüzde 25, çalışanlara işin gereğine uygun nitelikte fiziki altyapı imkanları sağlayan yüzde 55, yazılım elemanlarına gerekli düzeyde mesleki eğitim sağlayan firmaların oranı ise yüzde 45'i geçmemektedir.

2008 yılı yanıtları içinde ise çalışanlara sağlanan olanakları ifade eden maddelerde oranların düştüğü, kurumsallaşmayı ifade eden maddelerde iyileşme olduğu görülmektedir. Bu konuda süreçteki iyileşmenin sadece kurumlar yararına görünmesi, çalışana verilen eğitim, iş ortamı, performans değerlendirme sistemi ve benzeri olanakların değerlendirmeler içinde oransal olarak azalması dikkat çekici sonuçlar içindedir.

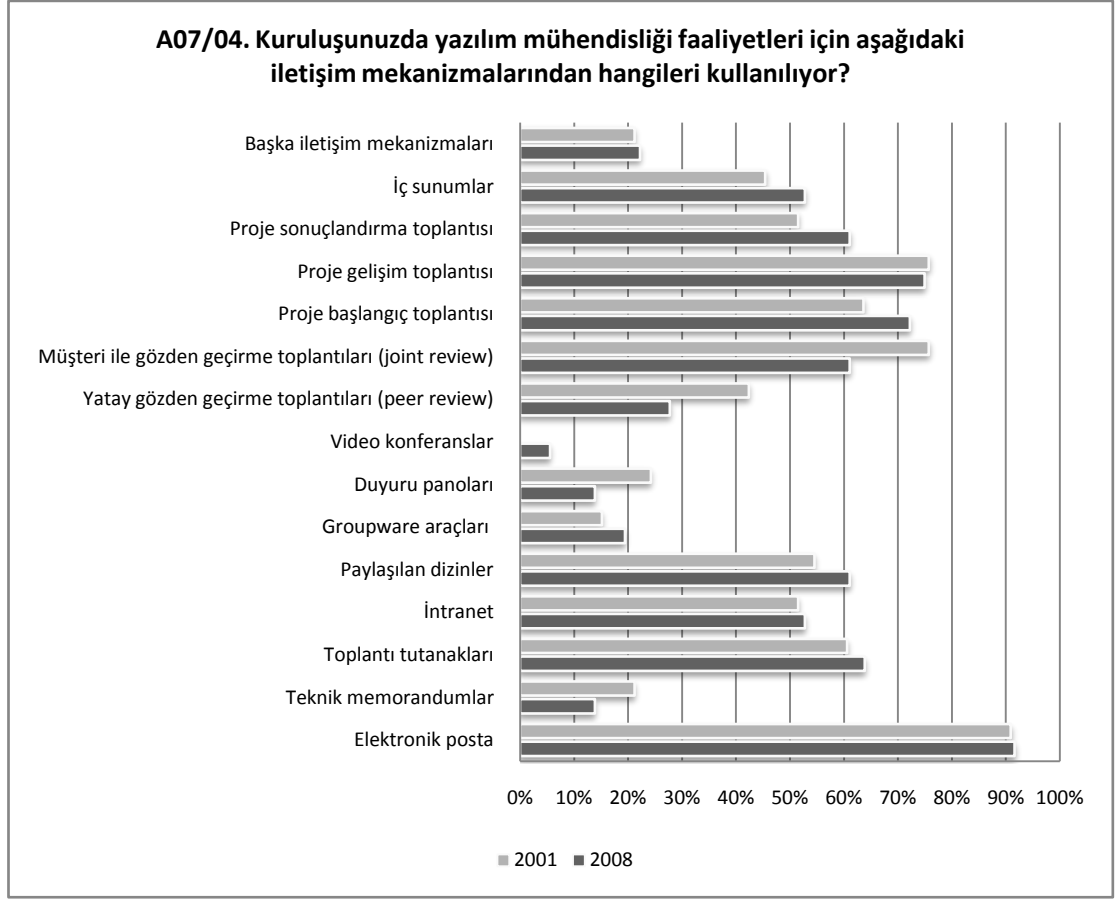


Şekil 4.52 : Yazılım mühendisliği insan kaynakları yönetimi

Projelerde “Video konferanslar” hariç pek çok iletişim aracı kullanılmaktadır (Şekil 4.53). “Groupware” araç kullanımının düşük olması ve yatay gözden geçirme toplantılarının azlığı dikkat çekmektedir.

Sonuçlar içinde kullanımı yüzde 22 oranında yer alan “Başka iletişim mekanizmaları”nın formal olmayan iletişim yöntemlerini ifade etmesi olasıdır.

İletişim mekanizmaları itibarı ile 2001 ve 2008 yılı sonuçları büyük ölçüde paralellik göstermektedir, teknolojik olanakların iletişimde daha çok kullanılması beklenirken gerçekleşen durum beklenmeyen sonuçlar içinde yer almaktadır.

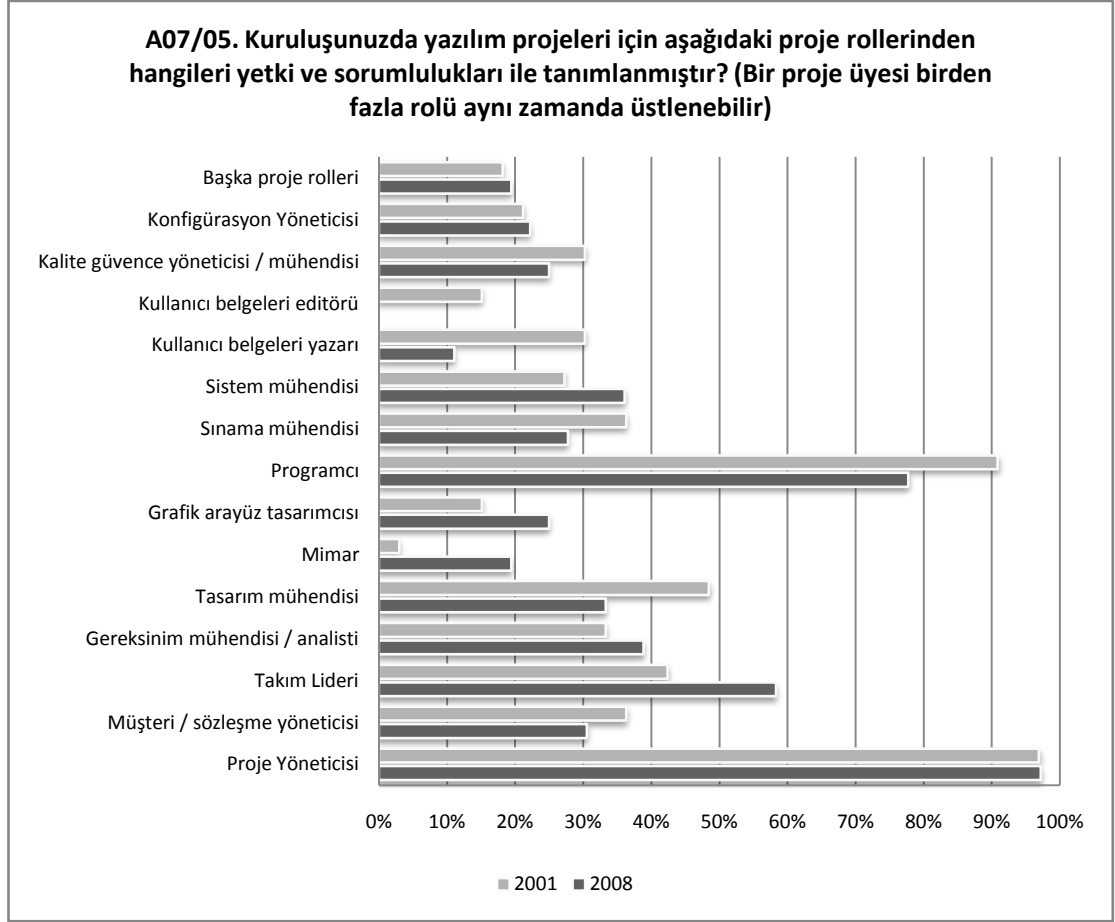


Şekil 4.53 : Yazılım mühendisliği faaliyetleri – İletişim mekanizmaları

Kuruluşlarda yazılım proje rollerinden en çok “Proje Yöneticisi” (yüzde 98) ve “programcı” (yüzde 78) rolleri yetki ve sorumlulukları ile tanımlanmış durumdadır (Şekil 4.54). Diğer mühendislik rolleri ise yüzde 30-50 arasında değişen oranlarda tanımlıdır.

Kalite Güvence Yöneticisi/ Mühendisi 2001 yılında yüzde 30, 2008 yılında yüzde 25 oranında tanımlıdır. Yazılım kadrosu küçük olan firmalarda bu rol bulunmazken, kadro büyüdükçe artan kaynaklar ile birlikte “Kalite Güvence Yöneticisi/ Mühendisi rolü de bulunmaktadır.

Diğer rollerde zaman içinde değişiklikler olsa da 2001 ve 2008 yılı sonuçları oldukça benzer yapıdadır.

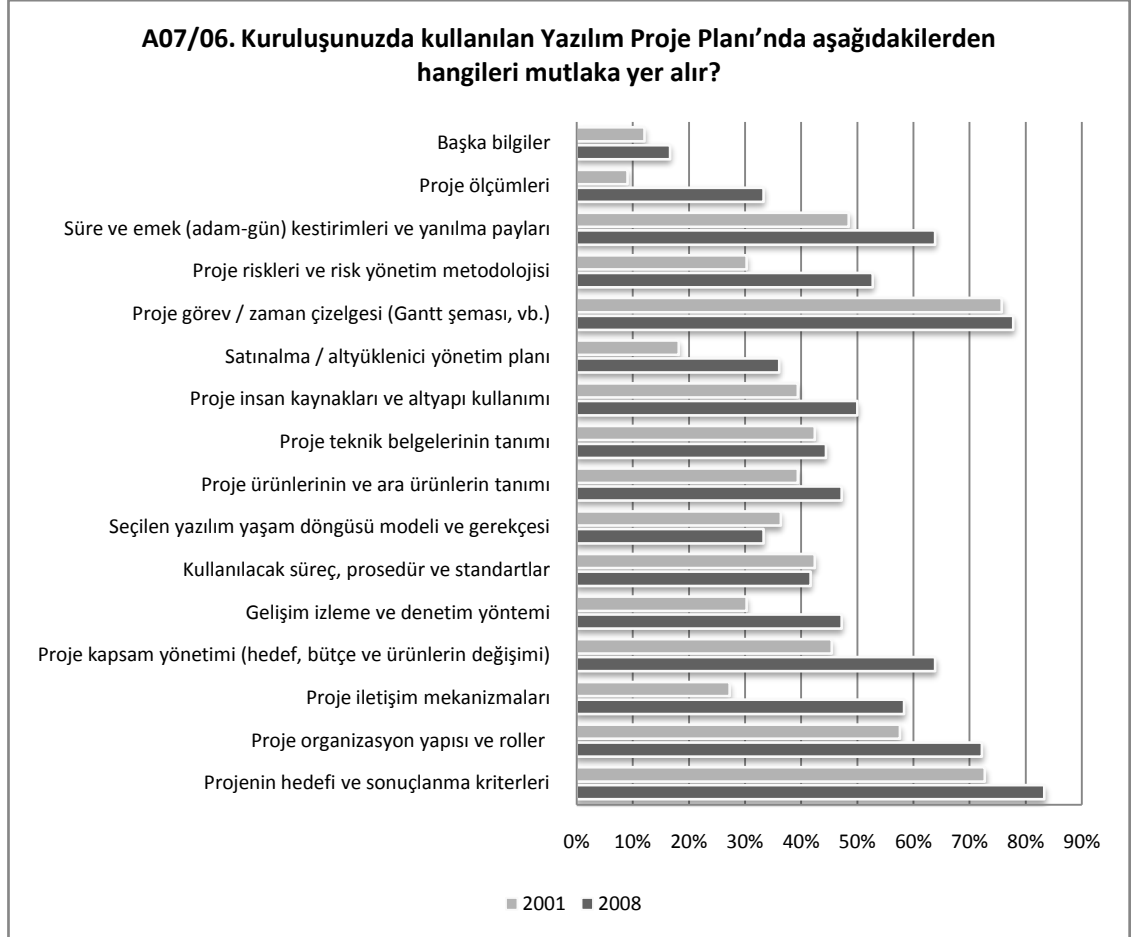


Şekil 4.54 : Yazılım proje rolleri

Şekil 4.55’de yazılım proje planlarının; 2001 yılında yüzde 52’sinde proje süre/emek tahminleri, yüzde 28’inde proje hedefleri, yüzde 25’inde proje görev/zaman çizelgesi yer almamaktadır. yüzde 60 oranda proje ürünleri, yüzde 60 oranda insan kaynağı ve altyapı tanımlanmamaktadır. Proje ölçümleri planın içinde yüzde 8 oranında tanımlıdır.

2008 yılında proje risklerinin planlar için ele alınma oranı yüzde 30’dan yüzde 52’ye artış göstermiştir. Proje süre/emek tahminleri daha çok adreslenmekle birlikte (yüzde 52 → yüzde 64) hala planların yüzde 36’sının içinde süre ve emek tahmini bulunmaması planlama sürecini sorgulanır hale getirmektedir. Projelerde iletişim mekanizmaları (yüzde 58), kapsam (yüzde 63), organizasyon (yüzde 72) ve yapılacak ölçümler (yüzde 34) geçen zaman zarfında planlar içinde daha çok adreslenmektedir.

Proje planlarında yüzde 83 oranında proje hedeflerinin ve sonuçlanma kriterlerinin yer alması, proje planının proje çizelgesinden farklı olarak anlaşılmaya başlandığının göstergesidir. Benzer şekilde proje risklerinin planlama içinde daha yüksek oranda ele alınması risk yönetimi süreci konusunda farkındalığın ve uygulamanın arttığını göstermektedir.



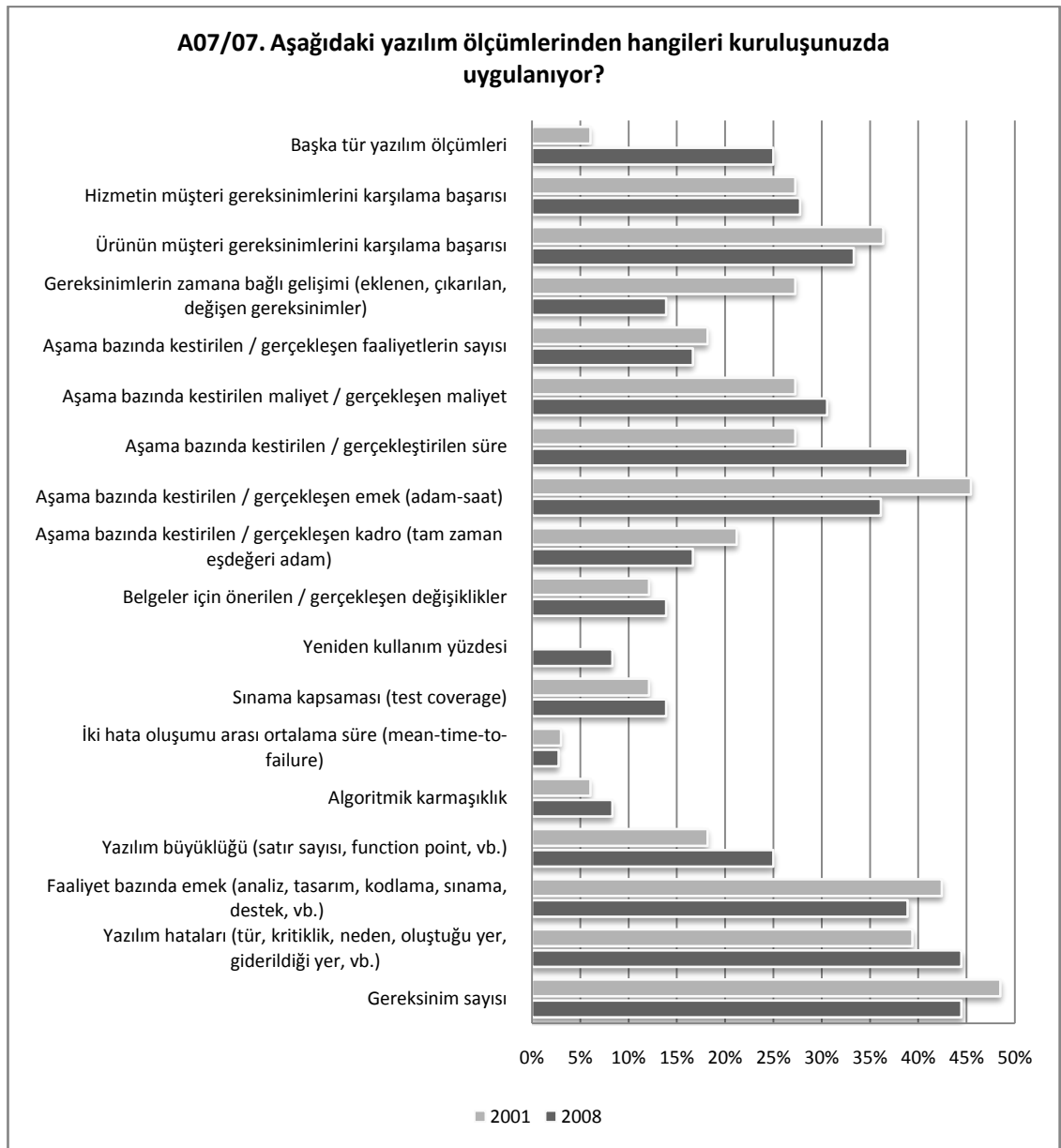
Şekil 4.55 : Yazılım proje planı

Şekil 4.56'da, yazılım ölçümlerinin düşük oranlarda gerçekleştirildiği görülmektedir. Yazılım mühendisliği sonuçları, gereksinimlerin numaralandırılma oranının düşük olduğunu göstermekle birlikte gereksinim sayısı önemli bir yazılım ölçümü (yüzde 48 - yüzde 44) olarak göze çarpmaktadır.

En temel ölçüm kriterlerinden biri olması beklenen “gereksinim değişikliklerinin” ve “müşteri gereksinimlerinin karşılanma oranının” yeterince ölçülenmediği gözlemler

arasındadır. Yazılım üreticileri yeniden kullanım konusunda neredeyse ölçümleme yapmamaktadır. Proje konusunda tahmin edilen ve gerçekleştirilen maliyet, süre, kaynak ve faaliyetlerin ölçümünde ise oranlar oldukça düşüktür.

Sonuçlar içinde, temel ölçümler yeterince yapılmazken seçenekler içinde detayı yer almayan “Başka Ölçümler” in yüzde 25 oranında bulunmasının ölçümleme yapılmadığının bir itirafı olduğu düşünülmektedir. Ölçümleme için hala yeterli oranda veri toplanmamaktadır.



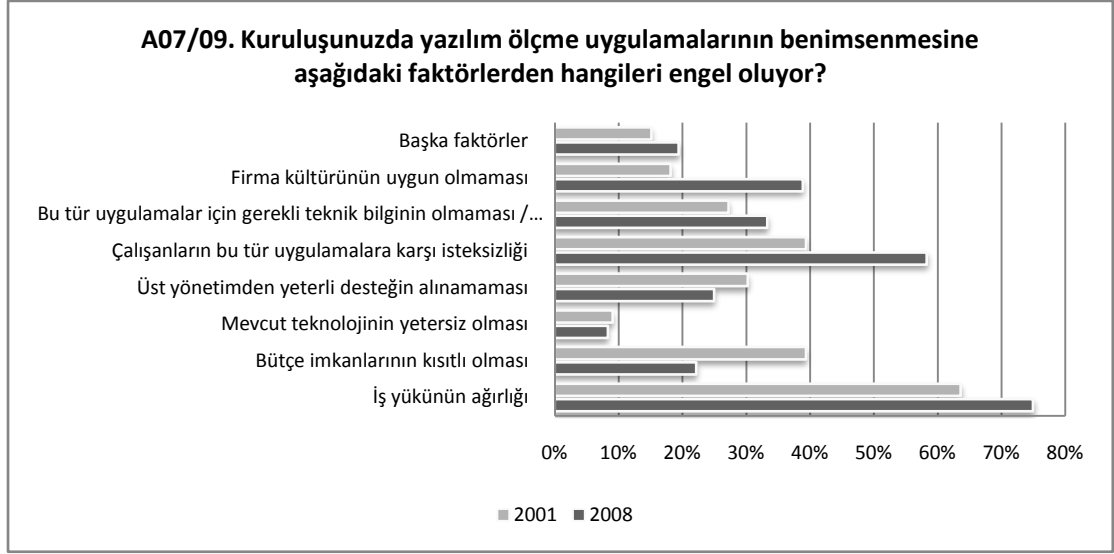
Şekil 4.56 : Yazılım ölçümleri

Şekil 4.57’de yazılım ölçme uygulamalarının benimsenmesi konusunda da sorun olduğu ve soruna bakış açısının çalışanın görevi ve firmanın türüne göre değişkenlik gösterdiği görülmektedir. Bu konuda ortak fikir büyük oranda (2001 - yüzde 64, 2008 yüzde 75) “İş yükünün ağırlığı”nın ölçüm yapmanın benimsenmesine engel olduğu yönündedir. Bütçe imkanlarının kısıtlı olması 2008 yılında arkasına gizlenen bir sebep olmaktan çıkmış, çalışanların ölçmeye isteksizliği ve firma kültürünün uygun olmaması önemli faktörler olarak beyan edilmiştir.

2008 yılı sonuçları rollere göre ayrıştırıldığında;

- a) Yazılım Mühendislerine göre; iş yükünün ağırlığı (yüzde 67), çalışanların isteksizliği, bütçe imkanları ve Üst Yönetim desteği (tümü yüzde 33’er),
- b) Kalite Yöneticilerine göre; işyükünün ağırlığı (yüzde 100) ve çalışanların isteksizliği (yüzde 67),
- c) Diğer Yöneticilere göre; işyükünün ağırlığı (yüzde 73), çalışanların isteksizliği (yüzde 62) ve firma kültürünün uygun olmaması (yüzde 38) faktörleri ölçme konusunda sınırlayıcı olmaktadır.

Ölçümlemenin benimsenmesi konusunda sorunlar devam etmekte, hala iş yükü ve firma kültürü ölçüm yapmaya engel faktörler olarak beyan edilmektedir. Ancak bu faktörlerin dışında ölçümlemenin neden ve nasıl yapılacağı konusunda da hedeflerin bilinmediği görülmektedir.



Şekil 4.57 : Yazılım ölçme uygulamalarının benimsenmesi

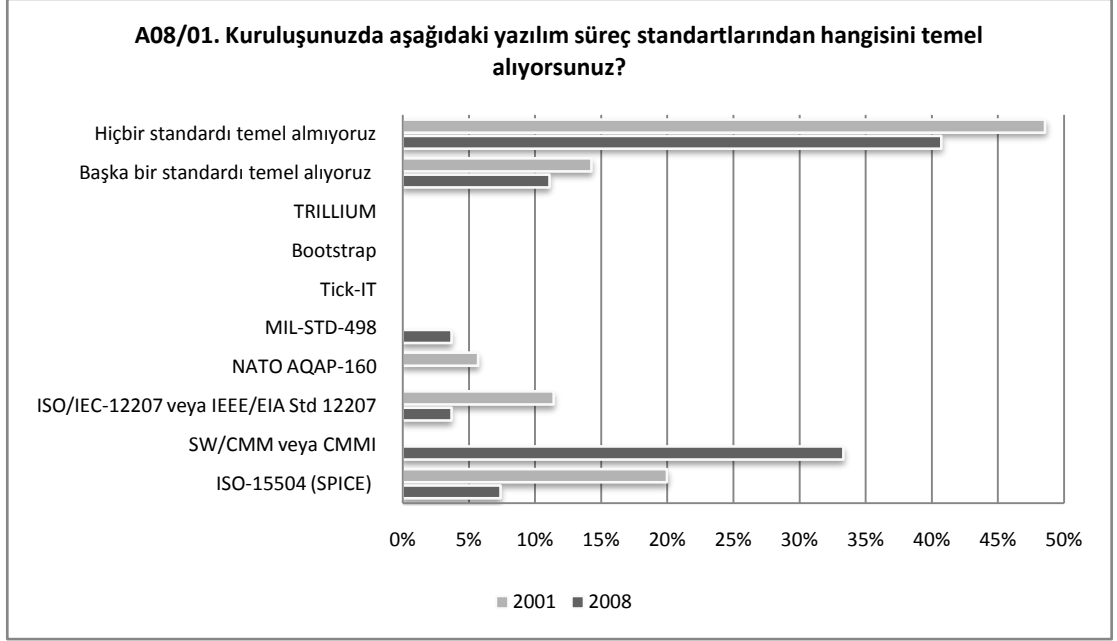
4.3.8 Yazılım Süreç Mühendisliği

Yazılım süreç mühendisliği, yazılım geliştirme ve bakımının kurumlaşması amacıyla yürütülen süreç tanımlama, uygulama, ölçme ve iyileştirme faaliyetlerini kapsar.

Şekil 4.58’de görüldüğü gibi, temel alınan yazılım süreç standartları arasında; 2001 yılında yüzde 20 oranla ISO-15504 (SPICE), 2008 yılında ise yüzde 33 oran ile CMMI öne çıkmaktadır.

“Hiçbir standardı temel almıyoruz” beyanında bulunan firmalar, 2008 yılında yüzde 41 oranındadır. Bu grubun içinde, iç kullanım amaçlı yazılım geliştiren firmalarda hiçbir standardın temel alınmama durumu daha yüksek orandadır.

2008 yılında araştırmaya katılan firmalar arasında, savunma sanayi için geliştirme yapan firma bulunmaması sebebi ile AQAP standartları temel alınmamaktadır. AQAP standartları, askeri alanda üretim gösteren firmalar tarafından, NATO kalite güvence gereklilikleri için temel alınmaktadır (<http://www.nato.int/docu/standard.htm> 2008).

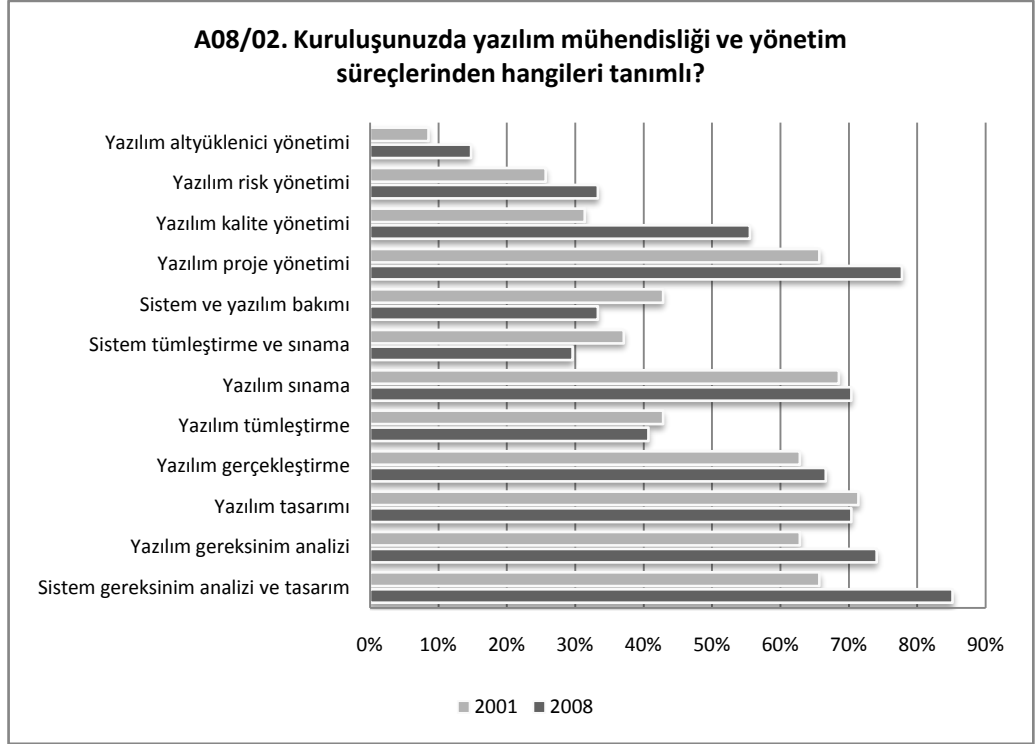


Şekil 4.58 : Yazılım süreç standartları

Şekil 4.59’da yazılım mühendisliği ve yazılım yönetimi süreçlerinin tanımlı olma durumu incelendiğinde, yazılım mühendisliği süreçlerinin daha yüksek oranlarda tanımlı olduğu görülmektedir.

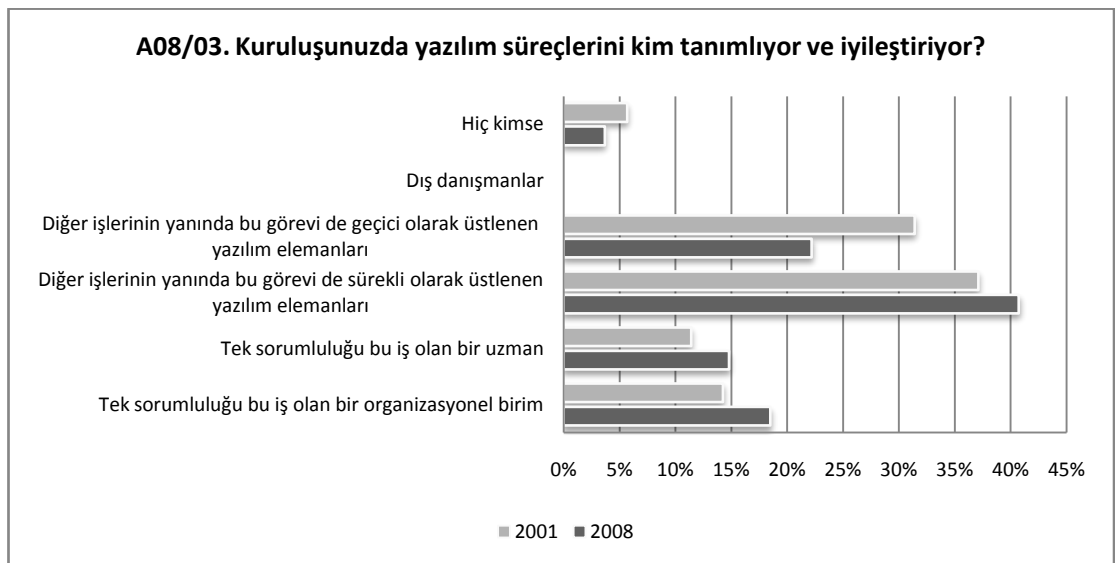
Yazılım tasarımı sürecinin yüksek oranda tanımlı olduğu beyan edilmekle birlikte bu durum yazılım mühendisliği verileri ile tutarlı değildir. Yazılım altyüklenici yönetimi süreci çok düşük oranda (2001 - yüzde 8 → 2008 - yüzde 15) tanımlıdır. Yazılım risk yönetimi ve kalite yönetimi süreçleri yine çok düşük oranda tanımlı görülmektedir.

Yazılım kalite yönetimi sürecinin tanımlanma oranındaki artışın metodoloji kullanımı ile ilgili olduğu ve bu konuda artan uygulama ve farkındalığın bir diğer göstergesi olduğu düşünülmektedir.



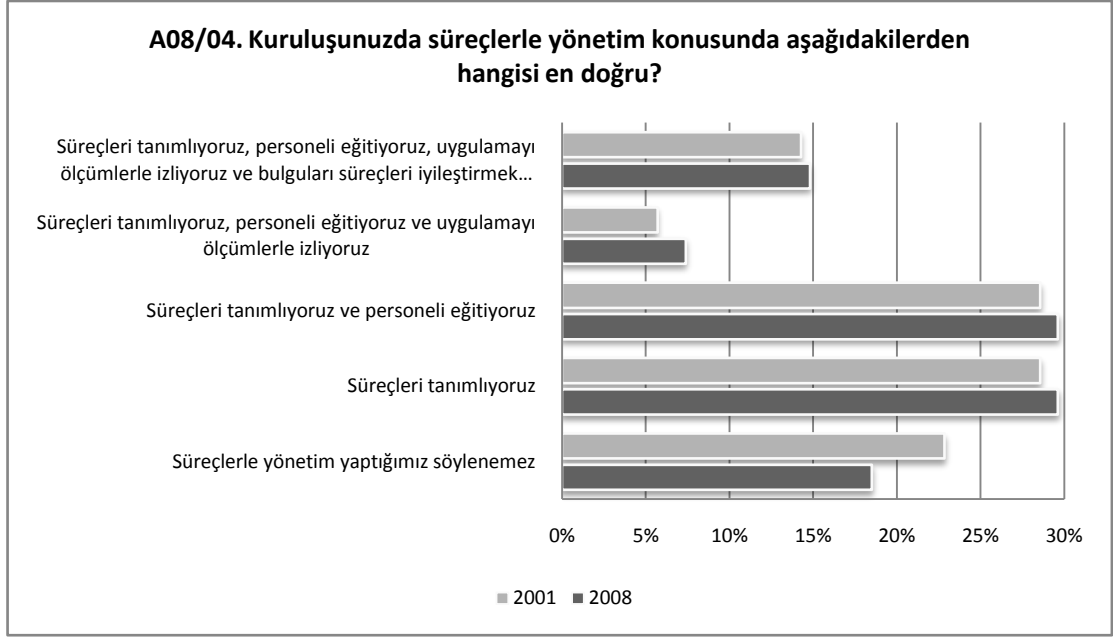
Şekil 4.59 : Yazılım mühendisliği ve yönetim süreçleri

Şekil 4.60 ile açıklanan veriler içinde, yazılım süreçlerinin tanımlanma ve iyileştirilmesi için ayrılan kaynak oranının arttığı görülmektedir. Ayrılan kaynak oranındaki artış, bu konuya odaklanmanın da göstergesidir. Ancak kurumlar dış kaynaklardan yararlanmak yerine süreç tanımlama ve iyileştirme çalışmalarını iç kaynakları görevlendirerek gerçekleştirmeyi tercih etmektedir.



Şekil 4.60 : Yazılım süreçlerinin tanımlanma ve iyileştirilmesi

Şekil 4.61 ile detayı açıklandığı şekilde süreçlerle yönetim konusunda oranların az da olsa iyi yönde artış gösterdiği görülmektedir. Süreç yaklaşımı biraz daha oturmuş olmakla beraber hala ölçümleme ve iyileştirme konusunda zaafiyetler bulunmaktadır.

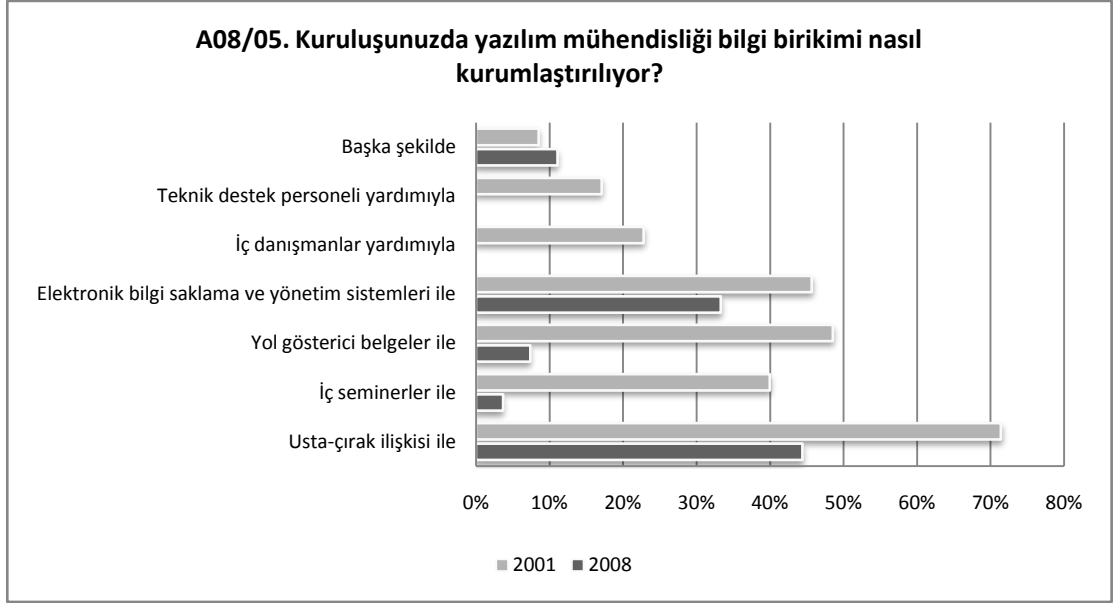


Şekil 4.61 : Süreçlerle yönetim

Şekil 4.62, yazılım mühendisliği bilgi birikiminin kurumsallaşma yöntemi olarak, en çok “usta çırak ilişkisi” ile bilgi birikimi transferinin gerçekleştirildiğini göstermektedir.

2001 yılı verileri içinde usta çırak ilişkisi yöntemi en çok “Yazılım Yüklenicilerinde” görülmektedir. Usta çırak ilişkisi yöntemi ve SPICE 3+ düzey konusundaki beyanlar arasında da büyük bir çelişki olduğu görülmektedir. Bu konudaki beyanları gözden geçirince “sözlü uygulamadan yazılı uygulamaya geçilememiş” olduğunu görmek mümkündür.

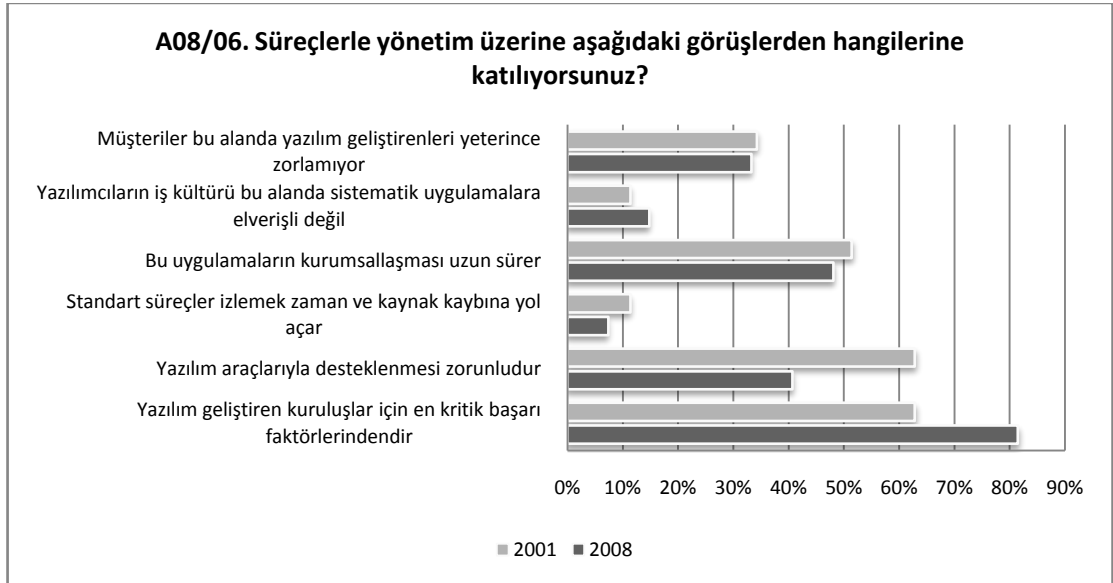
2008 yılı verilerinde bilgi birikiminin kurumsallaştırılması konusunda usta-çırak ilişkisinin yerini önemli ölçüde elektronik bilgi saklama ve yönetim sistemleri almıştır.



Şekil 4.62 : Yazılım mühendisliği bilgi birikimi

Şekil 4.63'te detayı görüldüğü gibi, süreçlerle yönetim konusunda hala yüzde 32 oranında müşterinin süreç yönetimine zorlaması beklenirken diğer tarafta katılımcıların yüzde 81'i bu konuyu en kritik başarı faktörü olarak tanımlamaktadır.

Süreçlerle yönetimin, yazılım geliştiren kuruluşlar için en önemli başarı faktörlerinden olduğunu düşünenlerin oranında yüzde 20'ye yakın oranda artış bulunmaktadır.

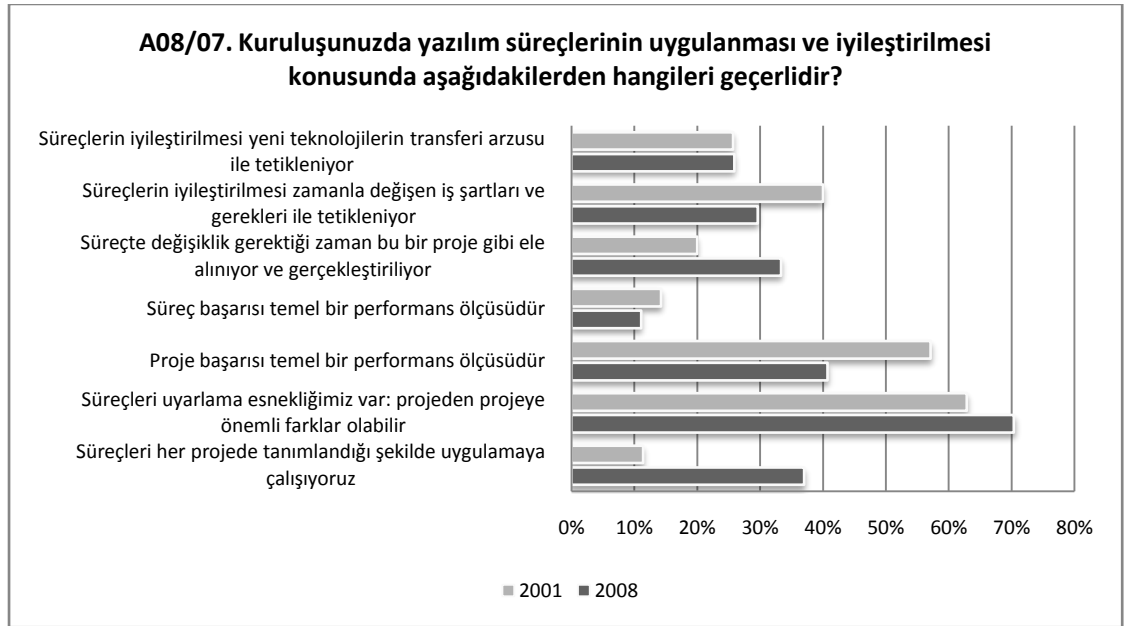


Şekil 4.63 : Süreçlerle yönetim

Şekil 4.64'te yer alan sonuçlar, firmalarda süreç yaklaşımı bulunmadığını göstermektedir.

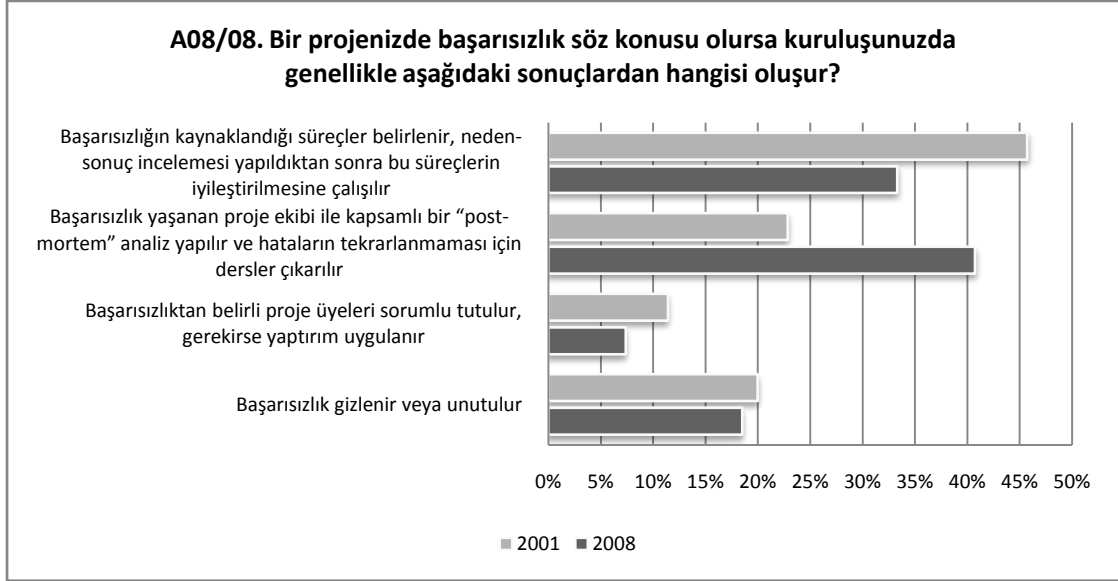
2001 yılında, proje başarısını performans göstergesi olarak değerlendirenler yüzde 57 oranındayken, süreç başarısını performans ölçüsü olarak algılayanların oranı yüzde 14 olarak görülmektedir. 2008 yılında oranlar değişmekle beraber performans yüzde 11 oranda süreç, yüzde 41 oranda proje başarısına bağlı olarak algılanmaktadır.

Proje başarısı ile süreç başarısı arasındaki farklı algılama kurumsallaşma düzeyinin düşük olduğunun göstergesidir. Proje başarısı, temel bir performans ölçüsü olarak beyan edilmekle birlikte, bu konuda ölçüm yapılmaması, proje başarısının nasıl tanımlandığı sorusunu da akla getirmektedir.



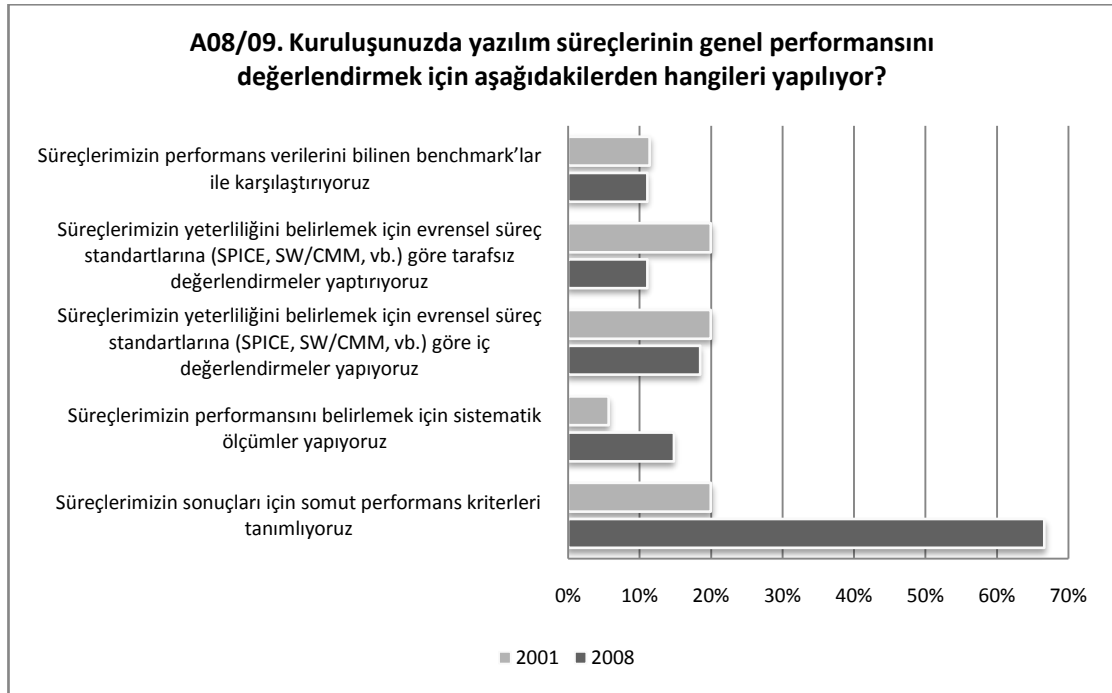
Şekil 4.64 : Yazılım süreçlerinin uygulanması ve iyileştirilmesi

Şekil 4.65 ile açıklanan yanıtlarda, projelerde başarısızlık nedenlerinin analiz edilerek ilgili süreçlerin iyileştirilmesi yerine “hatalardan ders çıkarma” yaklaşımı daha yaygındır. Bu durum projelerde başarısızlıkların kaynağının süreç mi yoksa çalışanlar mı olduğu sorusunu da akla getirmektedir.



Şekil 4.65 : Projede başarısızlık durumu

Şekil 4.66'da, 2001 yılında yüzde 20 olan süreçler için performans kriterlerinin tanımlanma oranı, 2008 yılında yüzde 67'ye çıkmıştır. Bu durum süreç olgunluk düzeyinin SPICE III. düzeyde olduğunu ifade etmekle birlikte bu kriterler doğrultusunda ölçümleme ve süreç değerlendirme yaklaşımları çok sınırlı olarak görülmektedir.



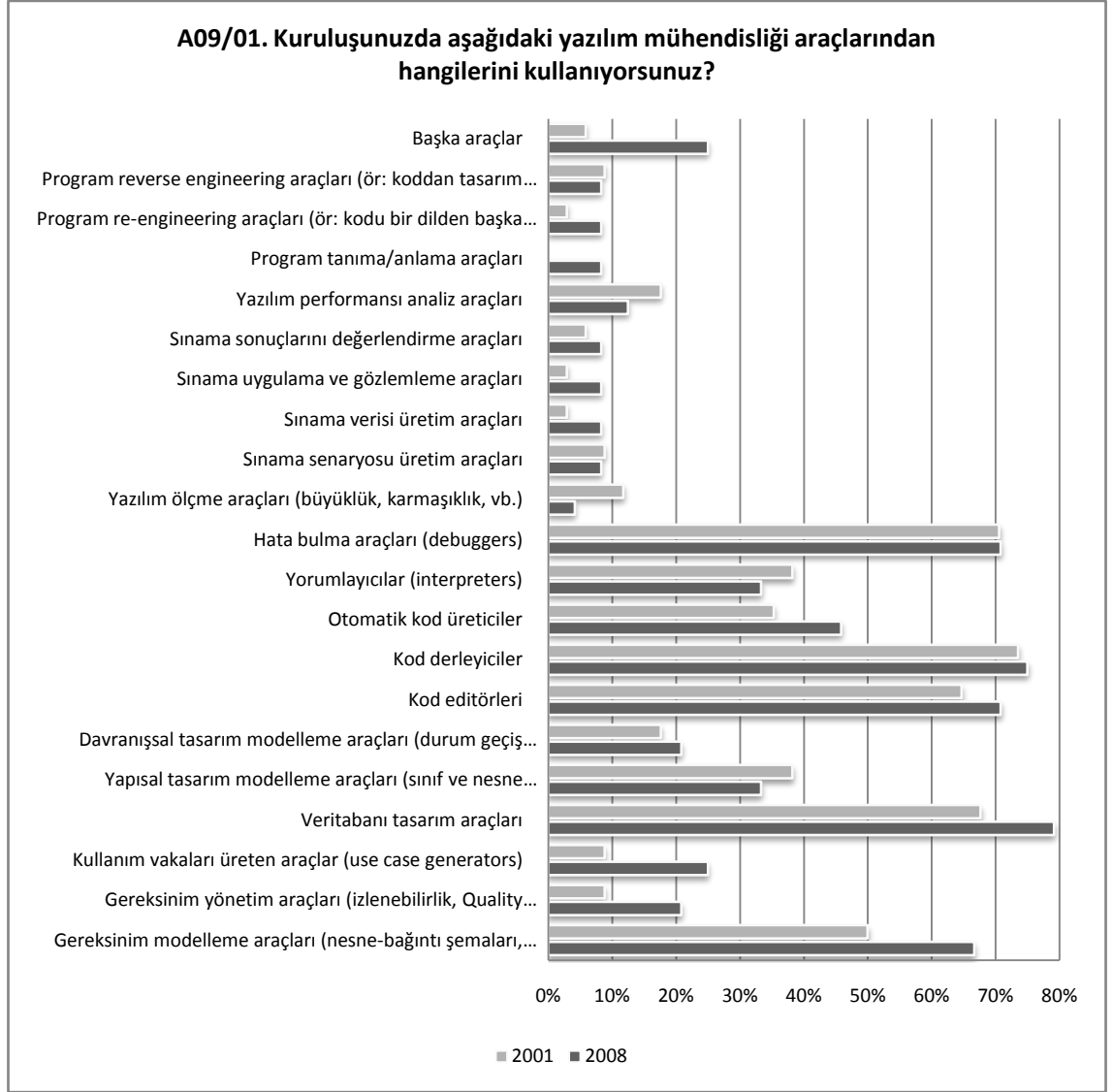
Şekil 4.66 : Yazılım süreçlerinin genel performansı

4.3.9 Yazılım Mühendisliđi Araç ve Yöntemleri

Yazılım mühendisliđi araçları, yazılım geliştirme ve bakımında kullanılan yazılımlardır. Yazılım mühendisliđi yöntemleri, yazılım yaşam döngüsünde kullanılan mühendislik yaklaşımlarını ve teknikleri kapsar.

Şekil 4.67 ile özetlenen sonuçlar içinde, ölçme ve sınama ile ilgili araç kullanımının çok az olduđu ve oranı az da olsa gereksinim yönetimi araçları kullanımının iki katına çıktığı görülmektedir.

Bunun dışında kod editörleri, kod derleyiciler, veritabanı tasarım araçları, hata bulma araçları gibi yazılım geliştirme sürecinde kullanımı zorunluluđu yüksek olan araçların yoğun kullanıldığı görülmektedir.

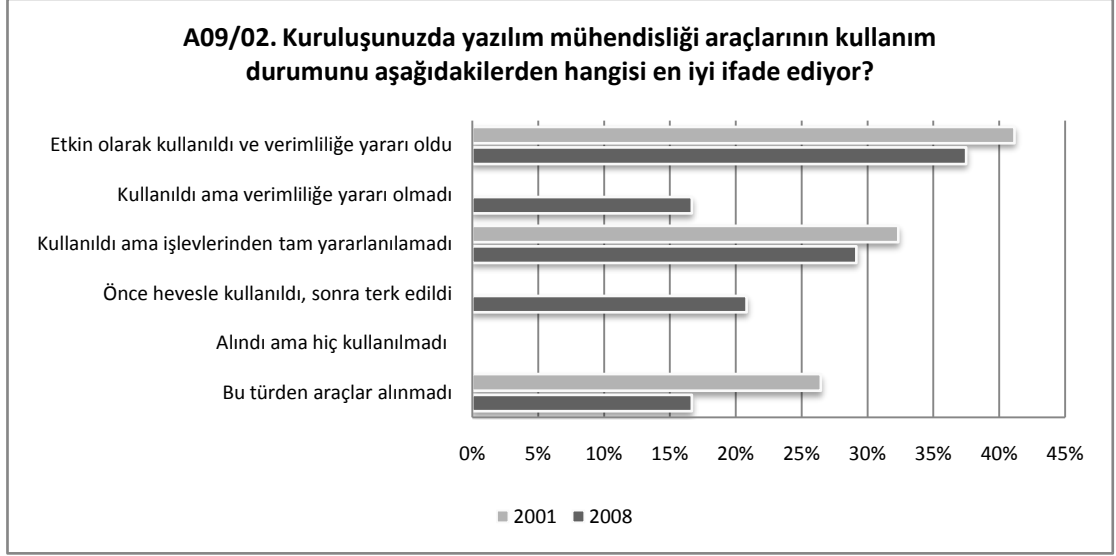


Şekil 4.67 : Kullanılan yazılım mühendisliği araçları

Şekil 4.68’de, araç kullanımında ve araçların verimliliğe katkısında önemli oranda sorun olduğu görülmektedir.

Kullanılan yazılım mühendisliği araçları en çok kalite yöneticileri tarafından verimli bulunmaktadır, yazılım mühendisleri ve yöneticiler farklı görüşlere sahiptir.

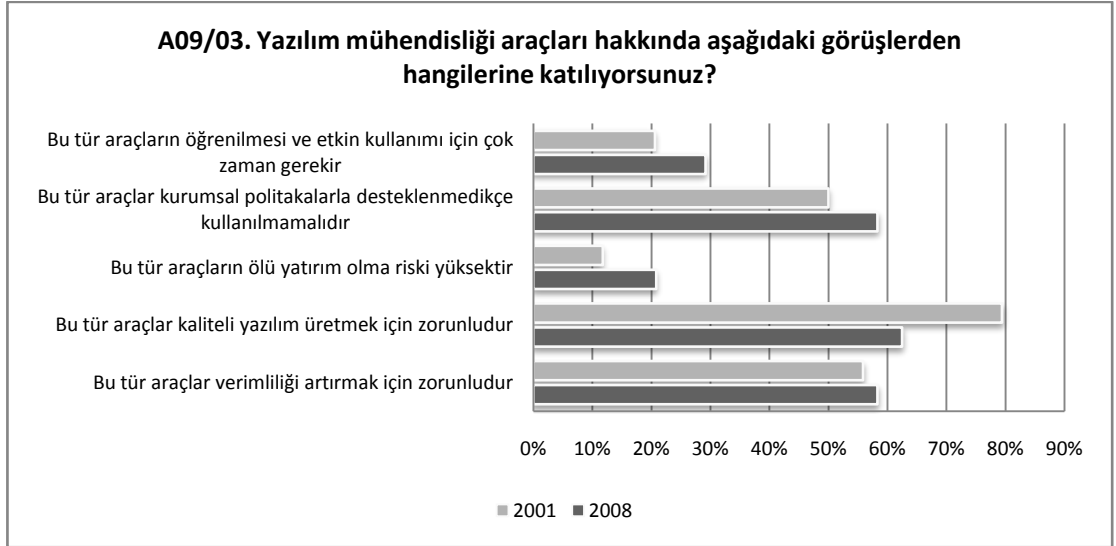
Bu konudaki değerlendirmelerin genel sonucunu “yönetimin zayıf ve araç kullanımı çok düşük olduğu şeklinde özetlemek mümkündür.



Şekil 4.68 : Yazılım mühendisliği araçlarının kullanım durumu

Şekil 4.69, araç kullanımının gerekliliği ve verimliliğe katkısının sorgulanmaya devam ettiğini göstermektedir.

Yazılım mühendisliği araçlarının kullanımının yüzde 20 oranda ölü yatırım olma riskinin yüksek olduğu düşünülmektedir.

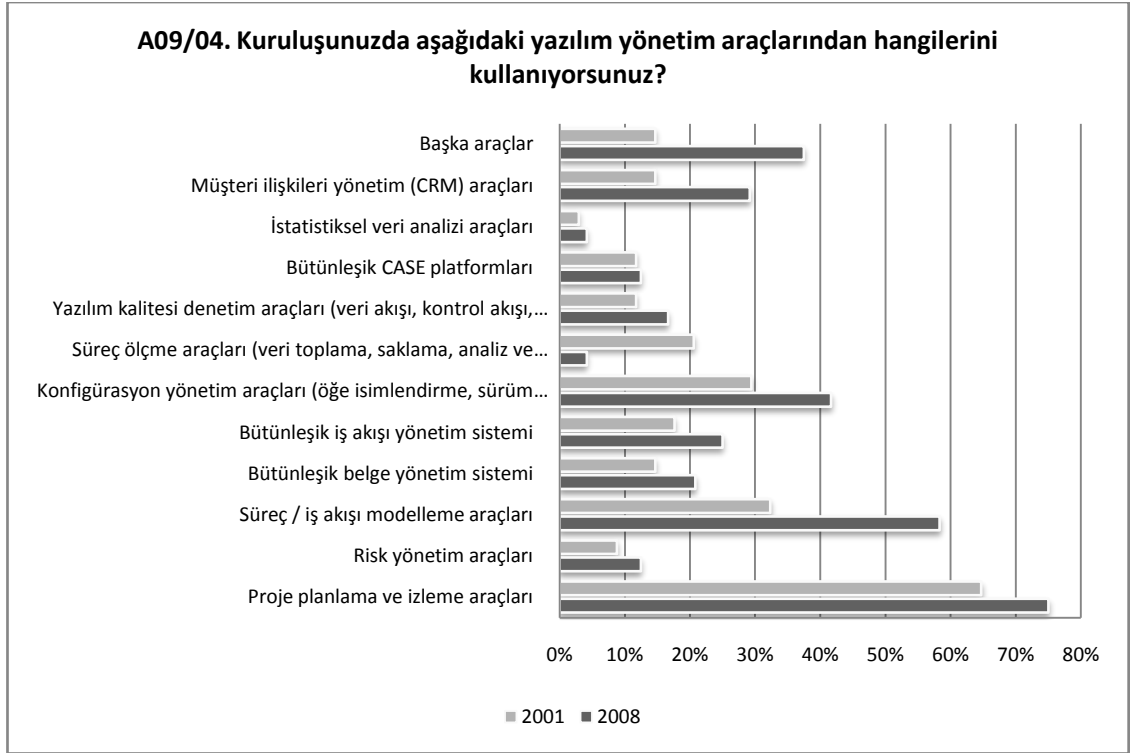


Şekil 4.69 : Yazılım mühendisliği araçları hakkında görüşler

Yazılım yönetimi araçlarının kullanım durumu değerlendirildiğinde (Şekil 4.70), yönetimin yeterince araç desteği ile yapılmadığı ortaya çıkmaktadır.

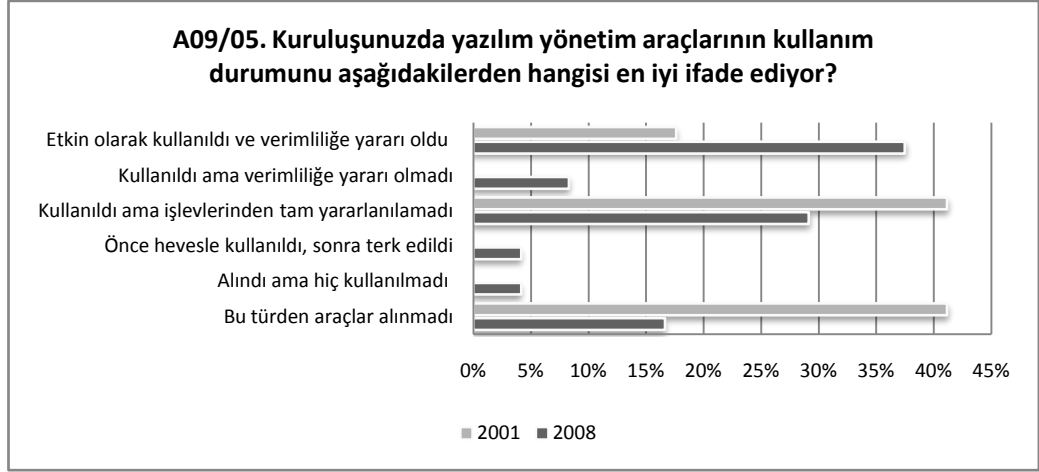
Bu konudaki beyanlara dayanarak, yazılım projelerinde ölçümleme yapılmadığını ve yönetim zaafı olduğunu söylemek mümkündür.

Proje planlama ve izleme araçlarının kullanımının dahi yüzde 75 oranını (2008) geçmediği, iş akış yazılımları kullanımı önceki değerlendirmeye göre çok artmakla birlikte yüzde 60'ı geçmediği görülmektedir. Belge yönetim sistemi bu kadar az kullanılırken, planlama/izleme araçları kullanılmazken proje yönetiminin nasıl başarılı olacağı konusu değerlendirilmelidir.



Şekil 4.70 : Kullanılan yazılım yönetim araçları

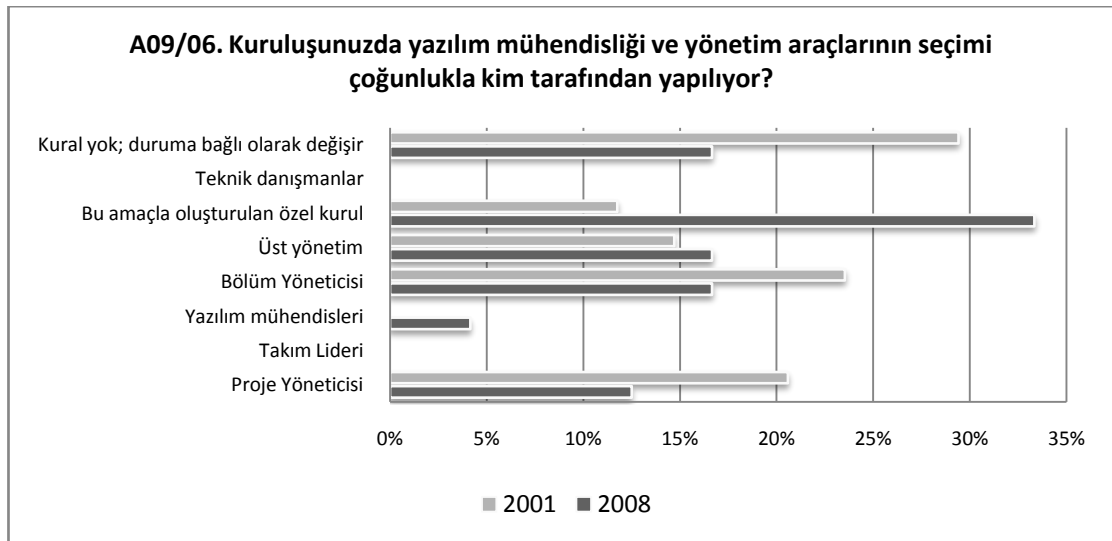
Şekil 4.71 incelendiği zaman, yazılım yönetimi araçlarının geçmişe göre daha çok kullanıldığı ve verimliliğe katkısının önceki değerlendirmeye göre önemli ölçüde arttığı görülmektedir.



Şekil 4.71 : Yazılım yönetim araçlarının kullanım durumu

2001 yılı verileri içinde, yazılım mühendisliği araçlarının seçimi konusunda (Şekil 4.72); araçları kullanacak mühendislerin seçimde etkili olmadığı, teknik danışmanların araç seçimine katılımının olmadığı, kararların bölüm yöneticisi, Proje Yöneticisi ve Üst Yönetim tarafından verildiği görülmektedir. Yanıtlar arasında kurul kararı çok az (yüzde 12) orandadır, yüzde 30'a yaklaşan oranda kural olmadığı yanıtı bulunmaktadır. Kural olmaması bu konuda da kurumsallaşma olmadığını göstermektedir.

2008 yılında ise bu konudaki negatif görüntü değişmiş, araç seçiminde kuralsızlık ve araç seçiminde yönetici kararı önemli ölçüde azalmış ve kurumsallık düzeyi artarak araç seçimi en çok (yüzde 33) kurul kararı ile yapılır olmuştur.

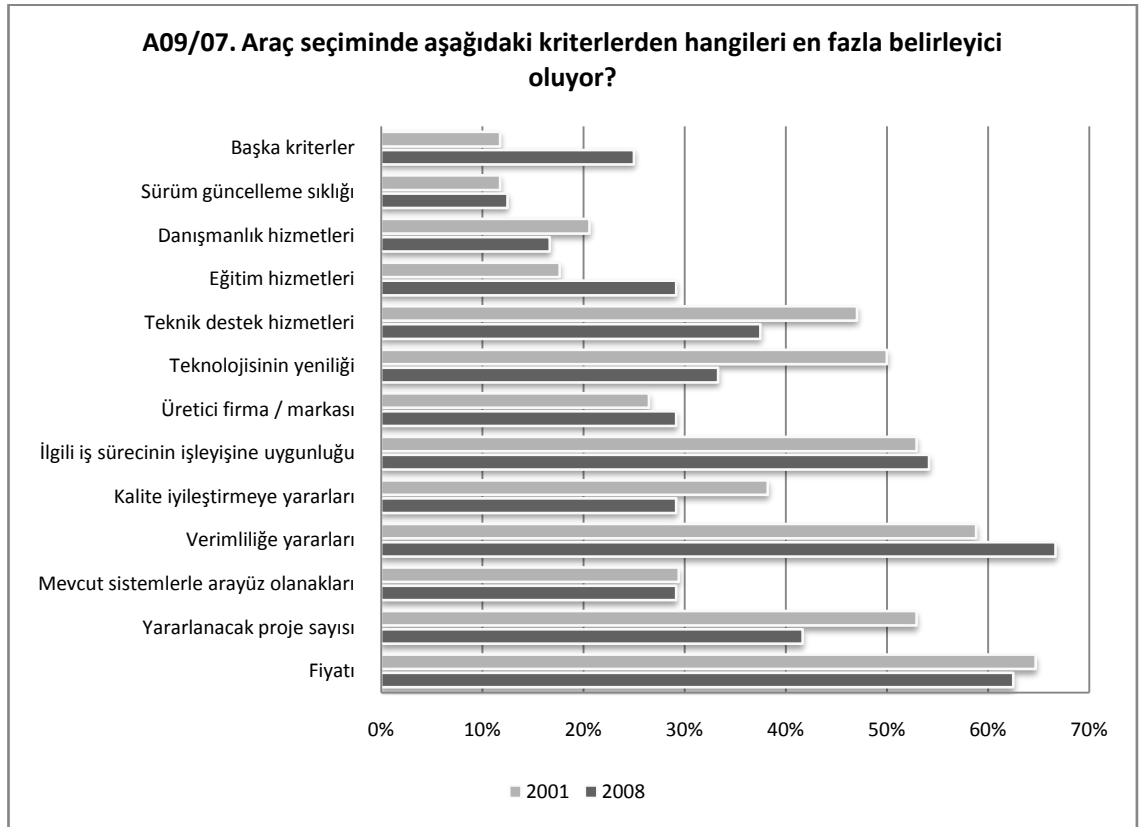


Şekil 4.72 : Yazılım mühendisliği ve yönetim araçlarının seçimi

Detayı Şekil 4.73 ile verilen araç seçim kriterleri arasında, 2001 yılı verileri içinde fiyat en önemli kriter olarak görülmektedir. Aracın verimliliğe yararı, ilgili iş sürecinin işleyişine uygunluğu, yararlanacak proje sayısı ve teknolojisinin yeniliği fiyatı izleyen kriterlerdir. Seçim kriterleri arasında eğitim, danışmanlık, marka ve mevcut sistemler ile arayüzler çok düşük oranlarda belirleyici olmaktadır.

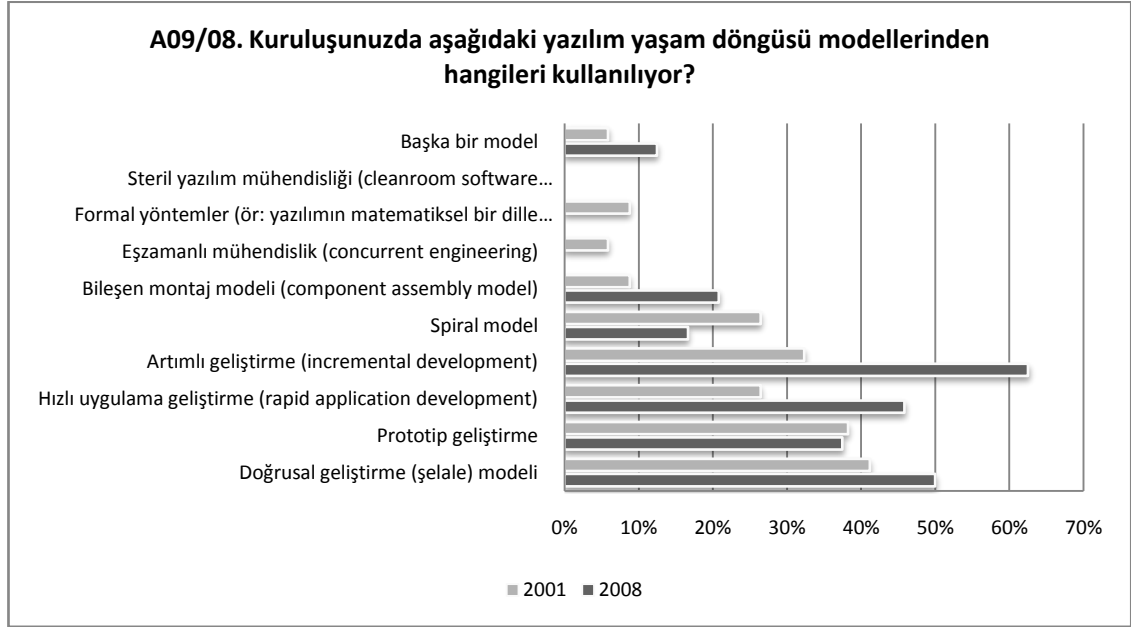
Firmanın yazılım çalışanı sayısı 100 kişiden fazlaysa araç seçiminde fiyat önceliğini kaybetmektedir.

2008 yılında ise fiyat yerine aracın verimliliğe yararı en önemli seçim kriteri olarak yerini almıştır.



Şekil 4.73 : Araç seçim kriterleri

Şekil 4.74 ile açıklanan yazılım yaşam döngüsü modelleri içinde, artımlı geliştirme 2008 yılında önemli ölçüde artarak, doğrusal geliştirmenin önüne geçmiştir.

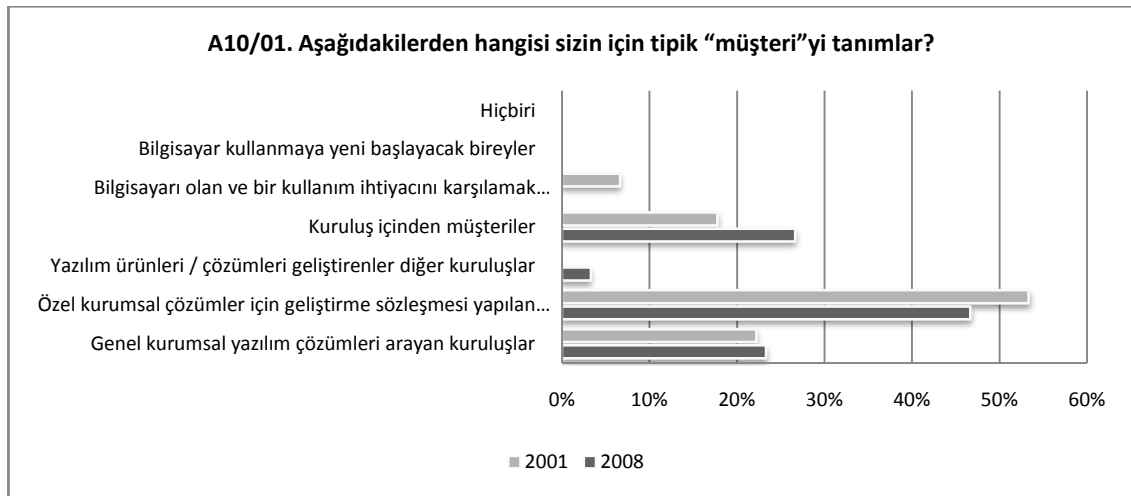


Şekil 4.74 : Yazılım yaşam döngüsü modelleri

4.3.10 Yazılım Kalitesi

Yazılım kalitesi, yazılım ürününün müşteri tarafından belirtilen veya beklenen kalite niteliklerini, ve bu niteliklerin güvenceye alınması için yürütülen faaliyetleri konu alır.

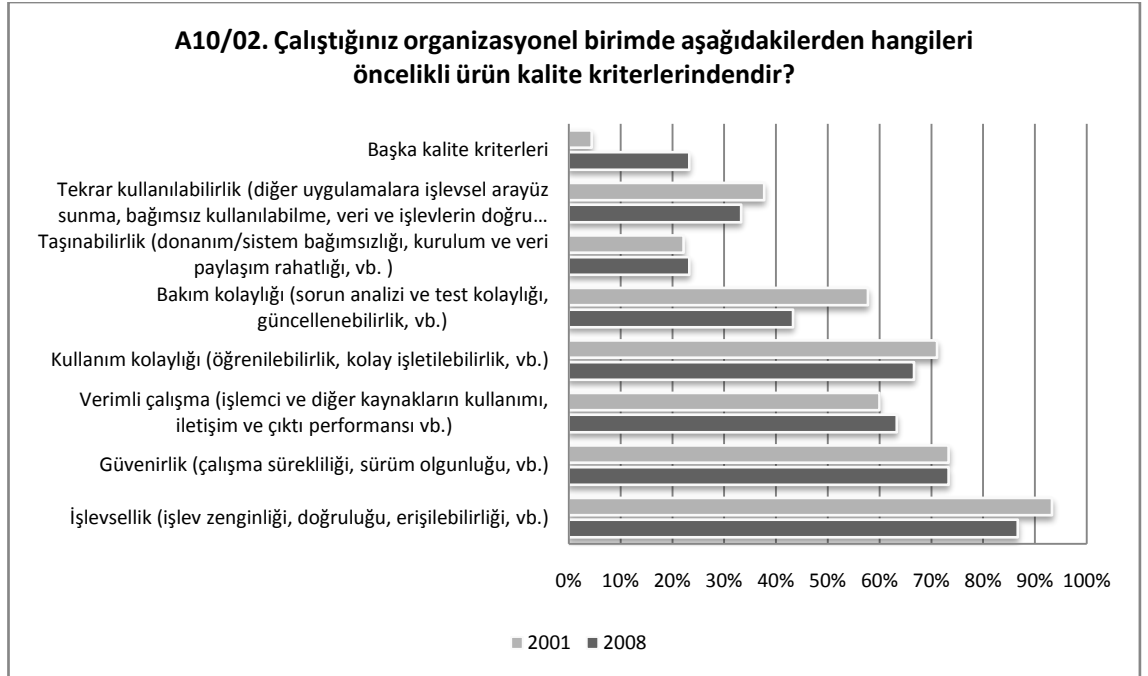
Şekil 4.75’de görüldüğü şekilde, değerlendirmeye katılanlar için müşteriyi en çok (yüzde 47) proje bazlı geliştirme yapılan firmalar oluşturmakta, bunu kurum içi müşteriler ve genel yazılım çözümü arayan kuruluşlar izlemektedir.



Şekil 4.75 : Müşteri tanımı

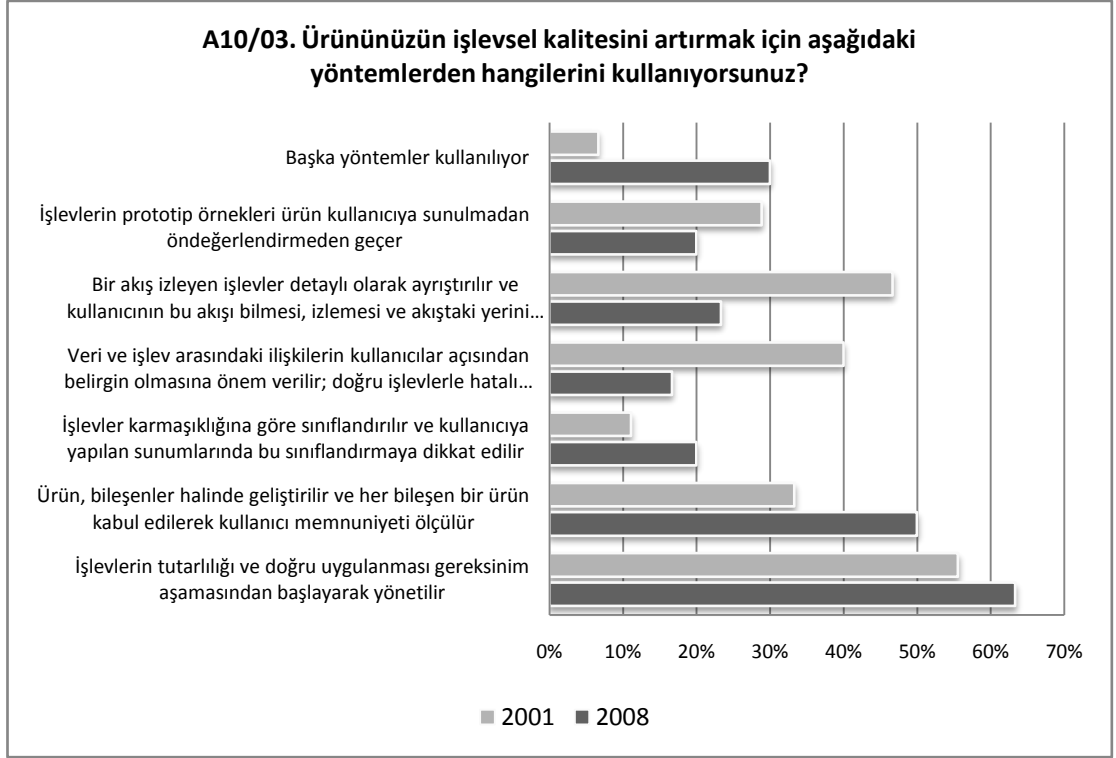
Detayı Şekil 4.76 ile verilen ürün kalite kriterleri arasında; taşınabilirlik, kullanım kolaylığı, bakım kolaylığı, verimli çalışma ve özellikle tekrar kullanılabilirlik yeterince önemsenmemektedir.

Sonuçlar kuruluş türüne göre ayrıştırıldığında, bakım kolaylığı ve tekrar kullanılabilirlik özelliklerinin yazılım yüklenicileri için daha önemli olduğu görülmektedir. Yazılım mühendisliği sonuçları içinde ise paket yazılım üreticileri bakım kolaylığı ve tekrar kullanılabilirlik açısından daha önde görünmektedir.



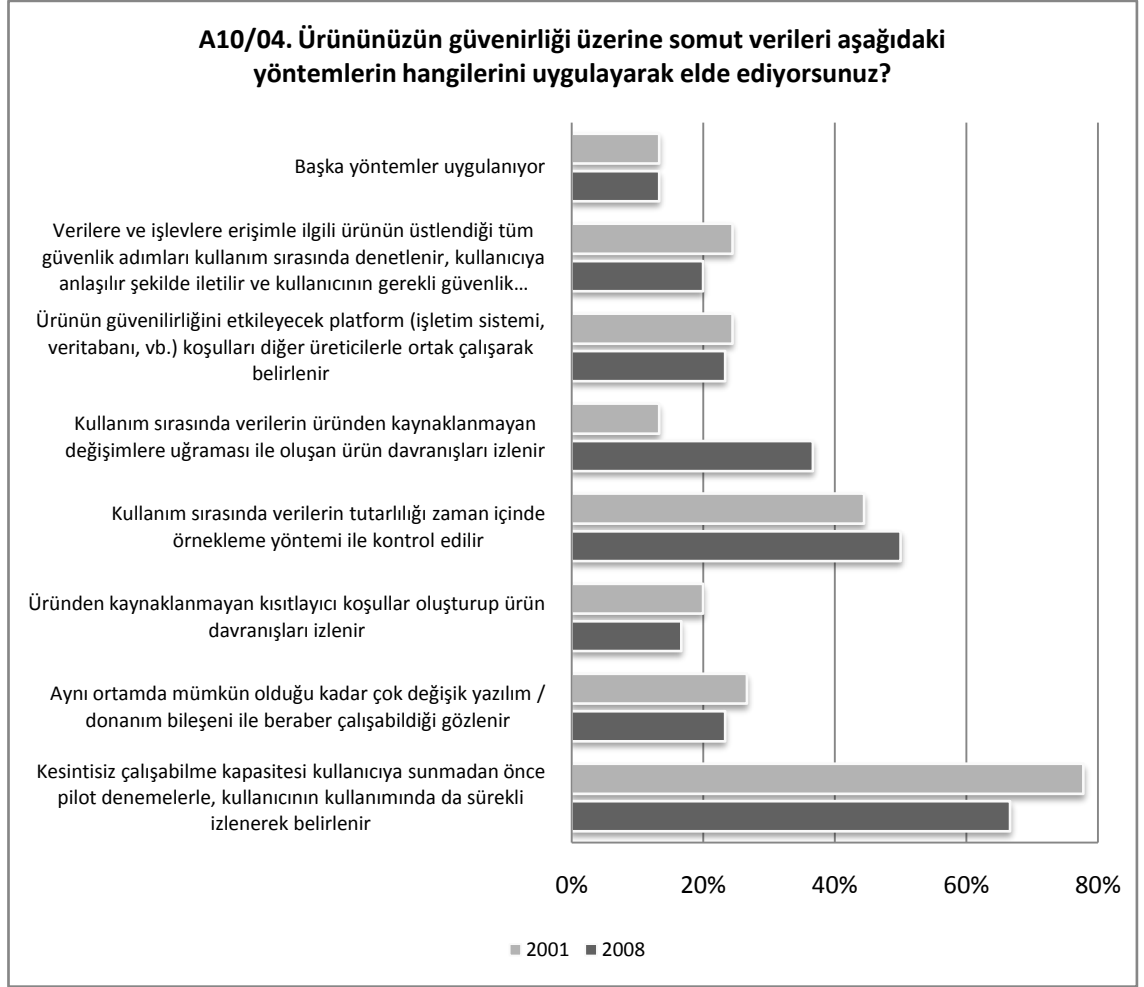
Şekil 4.76 : Ürün kalite kriterleri

Şekil 4.77 ile açıklanan sonuçlar içinde 2001 ve 2008 yılı verileri içinde tutarlı olarak işlevlerin tutarlılığının yönetilmesi ilk sırada yer almaktadır.



Şekil 4.77 : Ürünün işlevsel kalitesi

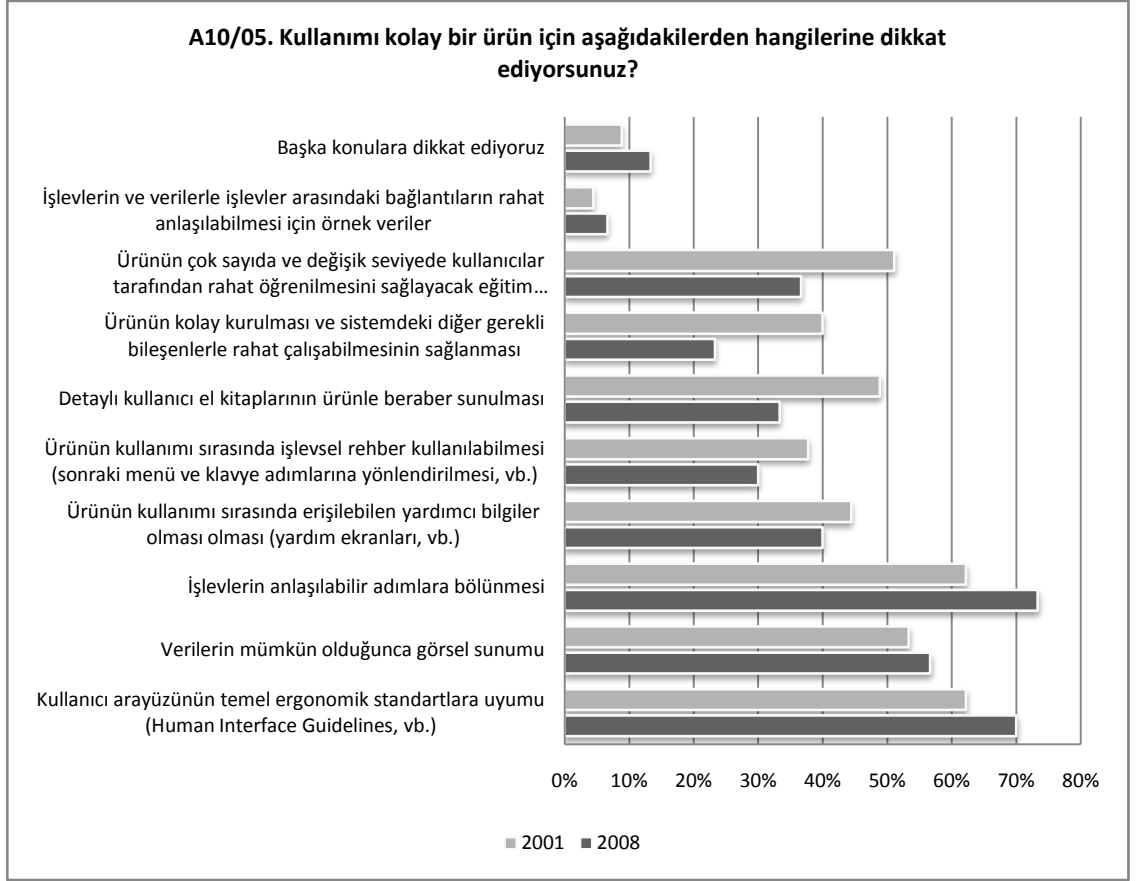
Şekil 4.78’de görüldüğü gibi, ürünün güvenilirliği en çok pilot denemelerle belirlenmekte, diğer güvenilirlik belirleme yöntemleri daha düşük oranlarda uygulanmaktadır.



Şekil 4.78 : Ürünün güvenilirliği

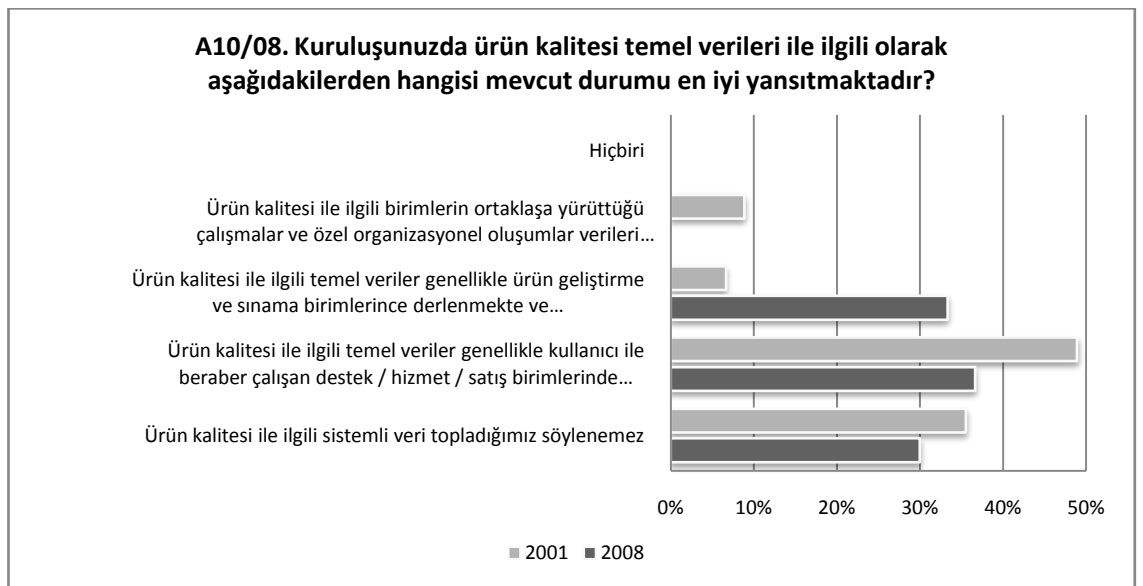
Şekil 4.79’da görüldüğü gibi, yazılımın kullanım kolaylığı için en çok; işlevlerin anlaşılır adımlara bölünmesi, kullanıcı arayüzünün ergonomisi ve verilerin görsel sunumu tercih edilen yöntemler içinde yer almaktadır.

Kullanım kolaylığı paket yazılım üreticileri için daha yüksek bir öneme sahiptir. Yazılım yüklenicileri için kullanıcı eğitimi ve el kitapları daha düşük oranda (ortalamanın altında) önem taşımaktadır.



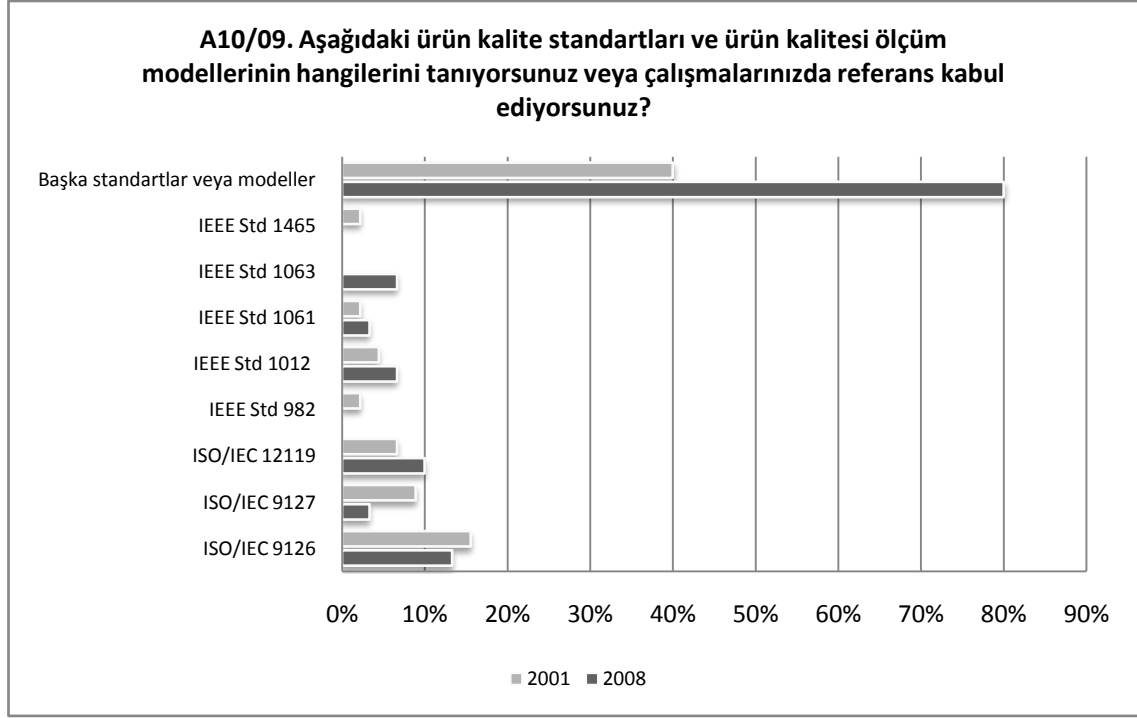
Şekil 4.79 : Yazılımın kullanım kolaylığı

Şekil 4.80 içinde yer alan yanıtlar değerlendirildiğinde, katılımcıların yüzde 30 oranında veri toplamadığını beyan etmesi dikkat çekmektedir.



Şekil 4.80 : Ürün kalitesi

Şekil 4.81 ile gösterilen ürün kalite standartları ve ürün kalitesi ölçüm modelleri içinde, beyanlar ürün kalitesi standartları ve ürün kalitesi ölçüm modelleri ile ilgili olarak bir tanıma sorunu olduğunu açıkça göstermektedir. Yazılım yönetimi süreçleri ile ilgili değerlendirmeler sırasında bilgi eksikliği çok sık ortaya çıkmaktadır.



Şekil 4.81 : Ürün kalite standartları ve ürün kalitesi ölçüm modelleri

4.4 KURULUŞ TÜRÜNE GÖRE ANALİZLER

Araştırma sonuçları üç grupta toplanarak analiz edilmiştir:

- Paket yazılım üreticisi
- Proje bazlı yazılım yüklenicisi
- İç kullanım için yazılım geliştiren veya bakımını yapan kuruluş.

2001 yılında 1.grup 14, 2.grup 30, 3.grup ise 24 kişiden oluşurken 2008 yılı verileri içinde bu sayılar üç paket yazılım üreticisi, 43 proje bazlı yazılım yüklenicisi ve 28 iç kullanım için yazılım geliştiren veya bakımını yapan kuruluştan oluşmaktadır. Paket yazılım üreticilerinden sadece üç firmanın soruları yanıtlaması ve bu firmalardan ikisinin çok küçük ölçekli olması nedeni ile paket yazılım üreticilerinin yanıtları geneli

ifade etmemektedir. Bu nedenle bu başlık altında yanıtlar sadece referans olarak verilmiş, demografik özellikler dışındaki konuların yorumu yapılmamıştır.

Soru kategorilerine göre farklılıklar Şekil analizlerin sonuçlarına göre aşağıdaki şekilde yorumlanmıştır.

4.4.1 Genel Göstergeler

Çalışan Sayısı: 2008 yılında araştırmaya katılan paket yazılım üreticisi firmaların tümünde çalışan sayısı 1-99 kişi arasındadır. İç kullanım için yazılım geliştiren firmalarda çalışan sayısı en yüksek oranda 500'den fazladır. (yüzde 43). Proje bazlı yazılım geliştiren firmalarda ise çalışan sayısı, en yüksek oranda 1-99 kişi arasındadır (yüzde 51). 2001 yılı verilerinde çalışan sayısı itibarı ile daha homojen bir dağılım olduğu söylenebilir.

Yazılım Geliştirme ve Bakım Kadrosu: Paket yazılım üreticisi firmalarda yazılım geliştirme kadrosu büyük oranda 1-9 kişi arasındadır (yüzde 67). İç kullanım için yazılım geliştiren firmalarda (yüzde 47) ve proje bazlı yazılım geliştiren firmalarda da (yüzde 40) yazılım geliştirme kadrosu büyük oranda 1-9 kişi arasındadır. 2001 yılı verilerinde yazılım geliştirme ve bakım kadrosu itibarı ile daha homojen bir dağılım olduğu söylenebilir.

Hitap Edilen Sektör: Tüm kategorilerde hizmet (Bilişim hizmetleri; Danışmanlık; Elektrik-su-gaz dağıtım; Eğitim; Eğlence; Gayrimenkul pazarlama; Haberleşme; Hukuk hizmetleri; İnşaat taahhüt; Lojistik-dağıtım-toptancılık; Mühendislik-ArGe hizmetleri; Perakendecilik; Reklamcılık; Sağlık; Turizm-ağırlama; Ulaştırma; Yazılı-görsel basın; Diğer) ve endüstri (Beyaz eşya; Denizcilik; Elektronik-bilgisayar-haberleşme; Enerji; Gıda; İlaç; İnşaat malzemeleri; Konut; Madencilik; Makine-otomotiv; Mobilya; Müzik-video; Petrokimya; Tarım-hayvancılık; Tekstil; Uzay-havacılık; Yayımcılık; Diğer) en yüksek yüzdeleri oluşturmaktadır.

Araştırmayı Yanıtlayanlar: Proje bazlı yazılım geliştiren firmalarda ankete katılanların büyük bir kısmı Üst Yönetici (yüzde 34), Yazılım Proje Yöneticiliği (yüzde 26) ve BT Yöneticiliği (yüzde 26) görev alanlarından gelmektedir.

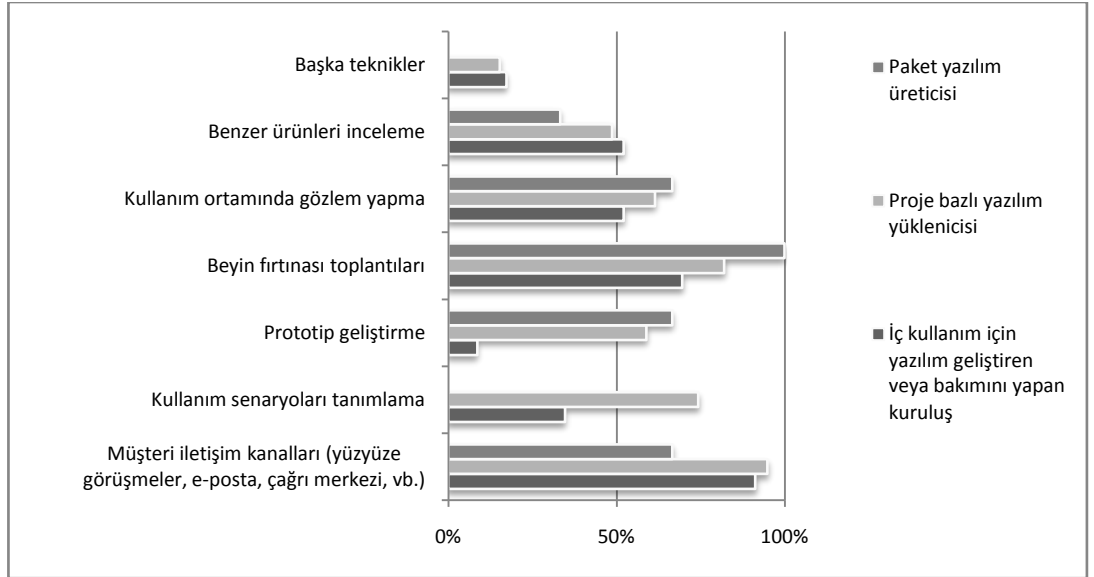
İç kullanım amaçlı yazılım geliştiren firmalarda ise yazılım mühendisleri (yüzde 33), Proje Yöneticileri (yüzde 23) ve BT Yöneticileri (yüzde 20) ankete katılmışlardır.

Görev Süresi: Kuruluş türünden bağımsız olarak, araştırmaya katılanların büyük çoğunluğu firmalarında uzun süredir çalışan kişilerdir. (≥10 yıl)

4.4.2 Yazılım Gereksinimleri

Gereksinimlerin belirlenmesinde, benzer ürünleri inceleme yöntemi en çok iç kullanım için yazılım geliştiren grupta görülmektedir. Bunun sebebinin müşteri tarafından ifade edilemeyen/ima edilen gereksinimlerin de ürün üzerinde görülerek gereksinimlerin eksiksiz olarak tanımlanması olduğu düşünülmektedir.

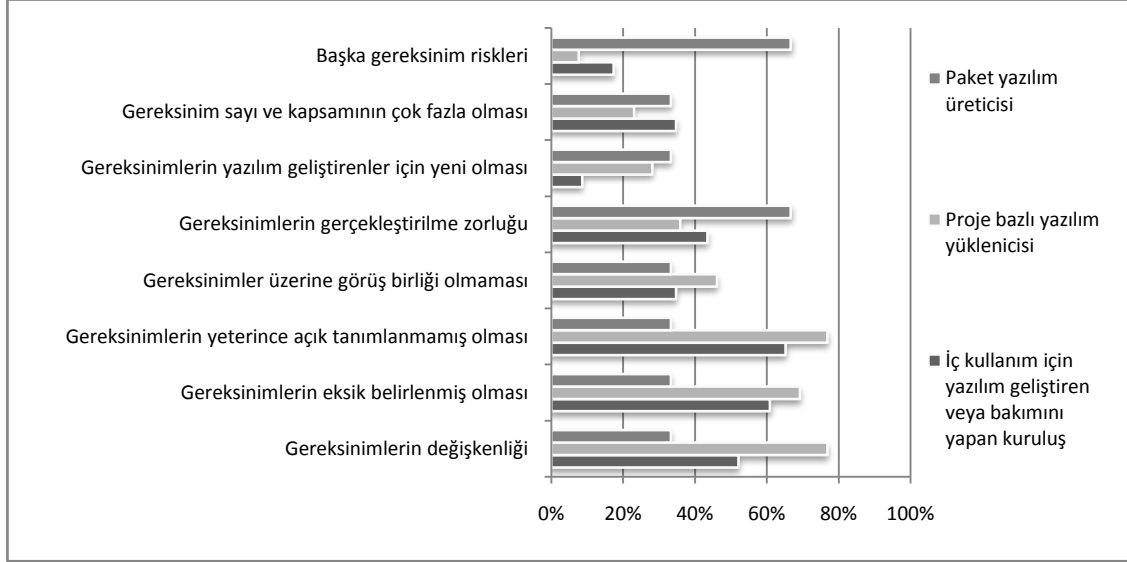
Paket yazılım üreticileri, müşteri ile iletişim yerine beyin fırtınası toplantılarını gereksinimlerin belirlenmesinde daha yoğun olarak tercih etmektedir (Şekil 4.82).



Şekil 4.82 : Gereksinim belirleme teknikleri

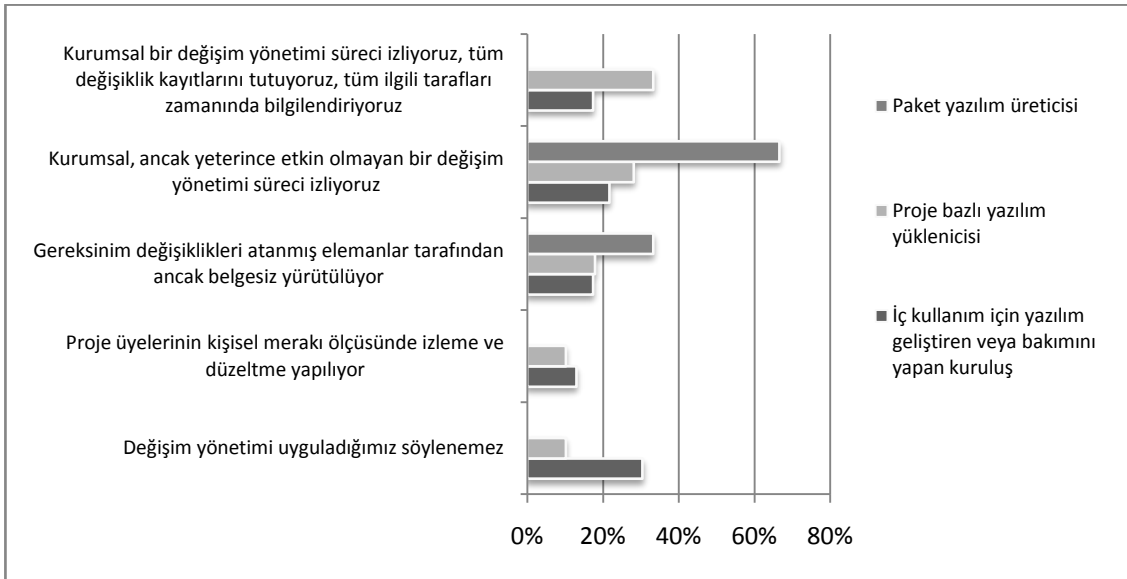
Gereksinim Riskleri: İç kullanım için yazılım geliştiren firmalar ve proje bazlı yazılım geliştiren firmalarda en önemli gereksinim riskleri (Şekil 4.83); gereksinimlerin eksik belirlenmesi, değişkenliği ve yeterince açık tanımlanmamasıdır. Paket yazılım geliştiren

firmalarda ise gereksinimlerin gerçekleştirme zorluğu en önemli risk alanı olarak ortaya çıkmaktadır.



Şekil 4.83 : Gereksinim riskleri ve risk azaltma yöntemleri

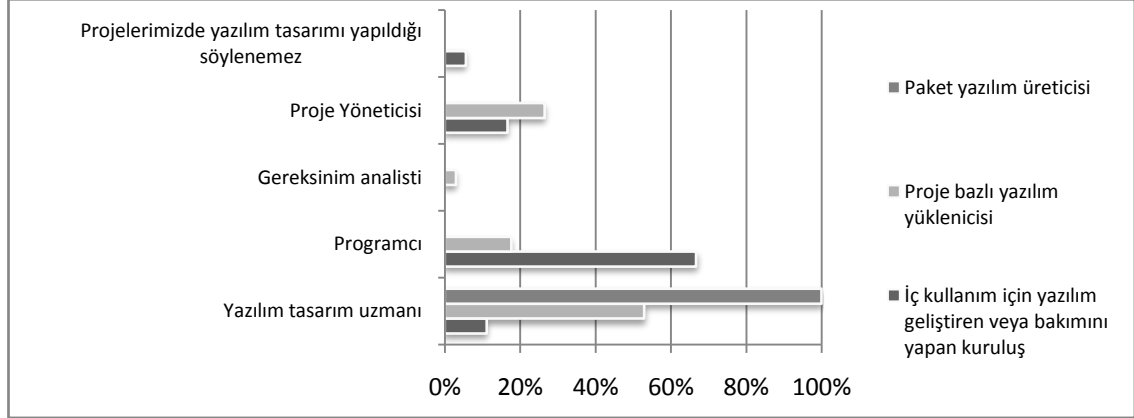
Değişim Yönetimi: Şekil 4.84’te görüldüğü gibi iç kullanım için yazılım geliştiren firmalarda yüzde 30, proje bazlı yazılım geliştiren firmalarda yüzde 10 oranında gereksinim değişikliklerinin yönetiminin yapılmadığı beyan edilmiştir.



Şekil 4.84 : Gereksinim değişim yönetimi

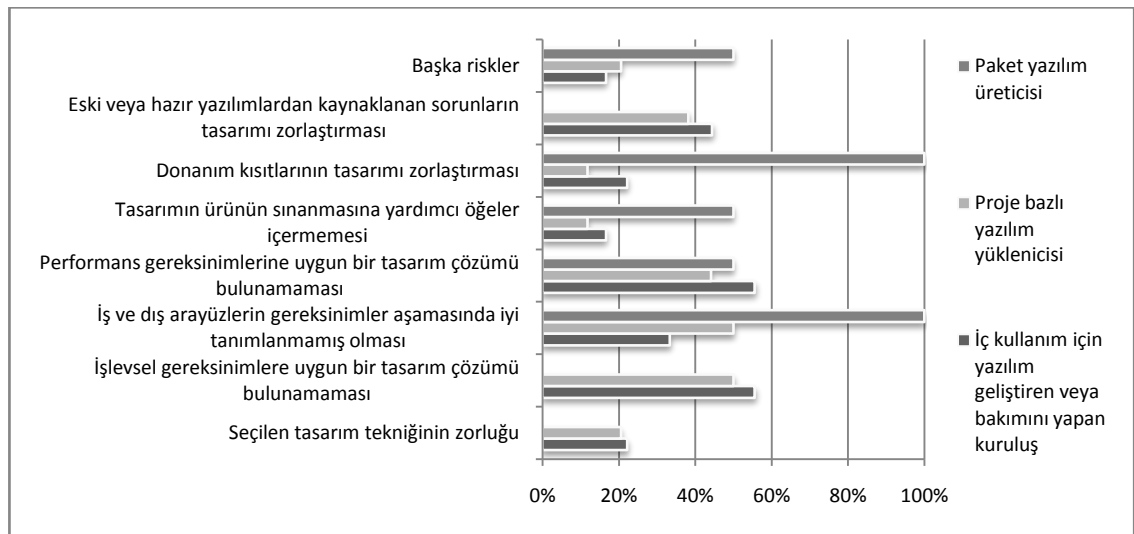
4.4.3 Yazılım Tasarımı

Yazılım Tasarım Sorumlusu: Proje bazlı yazılım geliştiren firmalarda yazılım tasarım uzmanı, Proje Yöneticisi ve programcı, iç kullanım amaçlı yazılım geliştiren firmalarda ise programcı, Proje Yöneticisi ve yazılım tasarım uzmanı tarafından tasarım yapılmaktadır (Şekil 4.85).



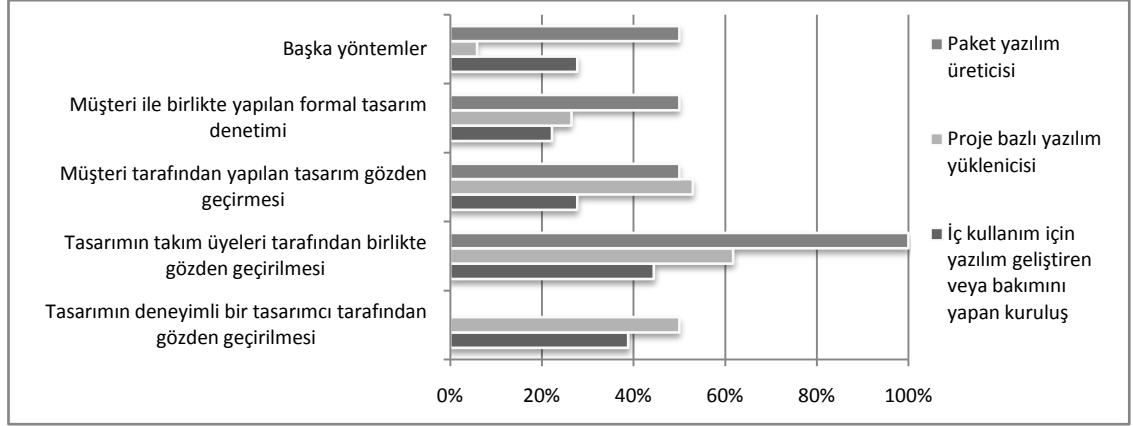
Şekil 4.85 : Projelerde yazılım tasarımı

Tasarım Riskleri: İç kullanım amaçlı ve proje bazlı yazılım geliştiren firmalarda benzer tasarım risklerinin değerlendirildiği Şekil 4.86'da görülmektedir. Performans gereksinimlerine ve işlevsel gereksinimlere uygun bir tasarım çözümü bulunamaması ve eski ya da hazır yazılımlardan kaynaklanan sorunların tasarımı zorlaştırması tasarım riskleri arasında en yüksek değerde yer almaktadır.



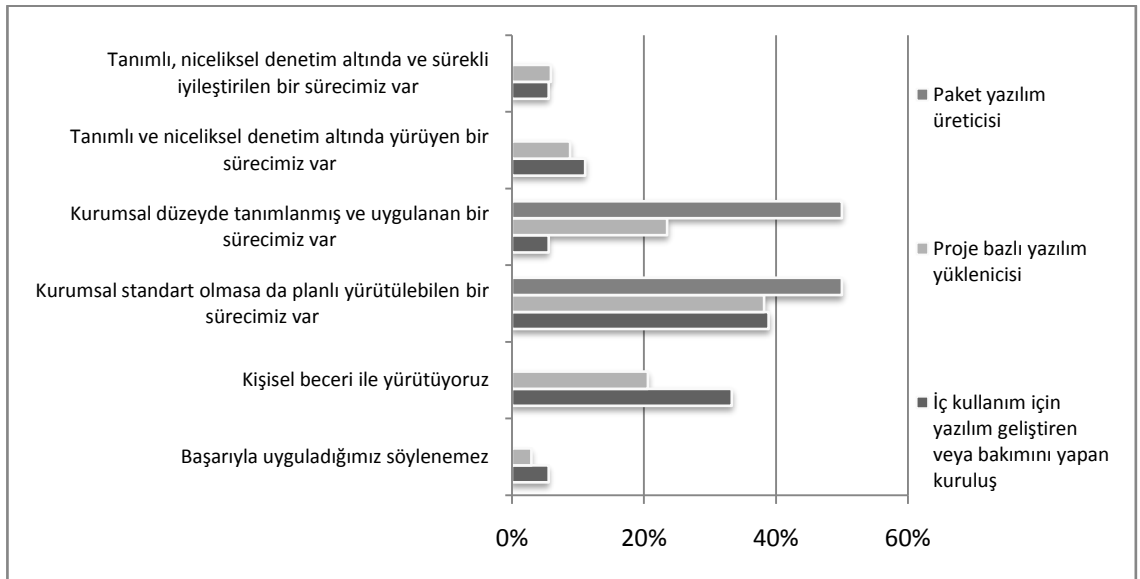
Şekil 4.86 : Tasarım riskleri ve risk azaltma yöntemleri

Tasarım Kalite Güvence: Şekil 4.87’de, iç kullanım amaçlı yazılım geliştiren firmalarda çeşitli yöntemler arasında homojen bir dağılım geçerlidir. Proje bazlı yazılım geliştiren firmalarda ise tasarımın takım üyeleri veya deneyimli bir tasarımcı tarafından gözden geçirilmesi en sık uygulanan kalite güvence yöntemi olarak öne çıkmaktadır.



Şekil 4.87 : Tasarım kalite güvence yöntemleri

Yazılım Tasarımı: Şekil 4.88’de yazılım tasarımı konusunda, proje bazlı ve iç kullanım amaçlı yazılım geliştiren firmalarda “kurumsal standart olmasa da planlı yürütülebilir bir süreç” bulunmaktadır.



Şekil 4.88 : Yazılım tasarımı süreç profili

4.4.4 Yazılım Gerçekleştirme

Programlama Yönetimi: İç kullanım amaçlı yazılım üreticilerinde, diğer kuruluşlara kıyasla yapısal programlama ve nesne yönelimli programlama yöntemleri daha yüksek oranda kullanılmaktadır. Proje bazlı yazılım üreten firmalarda nesne yönelimli programlama ve yeniden kullanılabilir bileşenler ağırlıklı olarak tercih edilmektedir.

Kodlama Standartları: İç kullanım için yazılım üreten firmalarda, değişkenleri ve yazılım birimlerini isimlendirme ve yazılım birimine ilişkin bilgi (amacı, işlevi, son değiştiren, vb.) ağırlıklı olarak uyulan kodlama standartlarıdır. Paket yazılım üreticilerinde diğer gruplara kıyasla daha yüksek oranlarda, kod formatı ve değişkenleri ve yazılım birimlerini isimlendirme kodlama standartları olarak kullanılmaktadır.

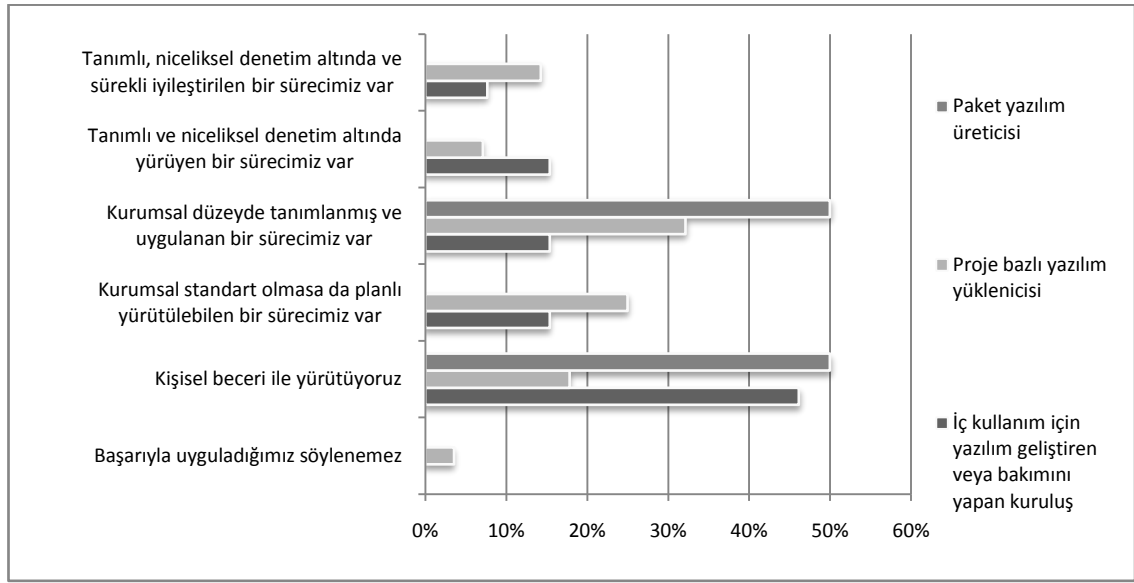
Tümleştirme Stratejisi: Kuruluş türünden bağımsız olarak, tümleştirme stratejisi en çok tasarım aşamasında belirlenmektedir. Proje bazlı yazılım geliştiren firmalarda tümleştirme stratejisi ağırlıklı olarak Proje Yöneticisi ve Takım Lideri tarafından belirlenirken iç kullanım için yazılım geliştiren yapılarda daha çok programcılar tarafından belirlenmektedir. İç kullanım için yazılım geliştiren firmaların yüzde 15'i bir tümleştirme stratejisi veya planı kullanmamaktadır. Bu sonuçlar, iç kullanım için yazılım geliştiren firmalarda tümleştirme aşamasından kaynaklı risklerin, diğer firmalara göre daha az olduğunun/algılandığının göstergesidir.

Kod Kalite Güvencesi: Kuruluş türünden bağımsız olarak, kod kalite güvencesi kapsamında kodun deneyimli bir programcı veya takım üyeleri tarafından gözden geçirilmesi en yüksek yüzdelerde yer almaktadır. Müşteri ile birlikte yapılan kod denetim ve gözden geçirmeleri çok yaygın bir yaklaşım olarak görülmemekle birlikte iç kullanım için yazılım geliştiren firmalarda toplam yüzde 16 oranında uygulandığının beyan edilmesi ilgi çekicidir.

Kullanıcı Belgeleri: Proje bazlı veya iç kullanım amaçlı yazılım geliştiren firmalarda büyük oranda, kullanıcı belgeleri atanmış deneyimli proje üyesi tarafından hazırlanır ya da kimse hazırlamak istemez seçenekleri işaretlenmiştir. İç kullanım için yazılım geliştiren firmaların yüzde 38'inde proje bazlı yazılım geliştiren firmaların ise yüzde

39’unda kullanıcıya yönelik belgelerin hazırlanmıyor olması önemli bir eksikliğe işaret etmektedir. 2001 yılı verileri de kullanıcı belgeleri konusunda aynı paralelde gelişmiştir.

Yazılım Geliştirme: Yazılım geliştirme konusunda, proje bazlı yazılım geliştiren farklı kuruluşlarda farklı düzeylerde uygulamalar gerçekleştirilmektedir (Şekil 4.89). İç kullanım amaçlı yazılım geliştiren firmalarda ise büyük oranda “kişisel beceri ile yürütülen” bir süreç bulunmaktadır.



Şekil 4.89 : Yazılım geliştirme süreç profili

4.4.5 Yazılım Sınama

Sınama Mühendisleri: 2001 yılı verileri içinde sınama mühendisleri; proje bazlı yazılım geliştiren firmalarda bilgisayar mühendisliği ve programcılığından, iç kullanım için yazılım geliştiren firmalarda en yüksek oranda bilgisayar mühendisliği ve matematik alanlarından, paket yazılım üreticilerinde ise diğer disiplinlerden gelmektedir.

2008 yılında ise sınama mühendisleri, proje bazlı yazılım geliştiren firmalarda bilgisayar mühendisliği ve programcılığından, iç kullanım için yazılım geliştiren firmalarda en yüksek oranda bilgisayar mühendisliği ve endüstri mühendisliği alanlarından gelmektedir. 2001 yılından 2008 yılına bilgisayar mühendislerinin bu

süreçte yer alma oranında önemli ölçüde (yüzde 45'ten yüzde 72'ye) artış olduğu görülmüştür.

Sınama Senaryoları: 2001 yılı verileri içinde; proje bazlı yazılım geliştiren firmalarda sınama senaryoları farklı uzmanlık alanları tarafından hazırlanmaktadır. İç kullanım için yazılım geliştiren firmalarda sınama senaryoları en yüksek oranda sınama sırasında anlık olarak oluşturulmaktadır. Paket yazılım üreticilerinde ise en yüksek oranda bu işle görevlendirilen sınama mühendisi tarafından hazırlanmakta ya da sınama sırasında anlık olarak oluşturulmaktadır.

2008 yılı verileri içinde ise Tablo 4.7'de oranları verilen sınama senaryoları; proje bazlı yazılım geliştiren firmalarda, ağırlık Gereksinim Analistlerinde olmak üzere farklı uzmanlık alanları tarafından hazırlanmaktadır. İç kullanım için yazılım geliştiren firmalarda senaryolar, sınama sırasında anlık oluşturulabildiği gibi Sınama Mühendisleri ve Takım Lideri ağırlıklı olmak üzere diğer görevler tarafından da paylaşılabilir.

Geçen süre zarfında Sınama Mühendisleri ve Gereksinim Analistlerinin bu süreçte daha yüksek oranda yer alması sürecin başarısı açısından önemli bir göstergedir.

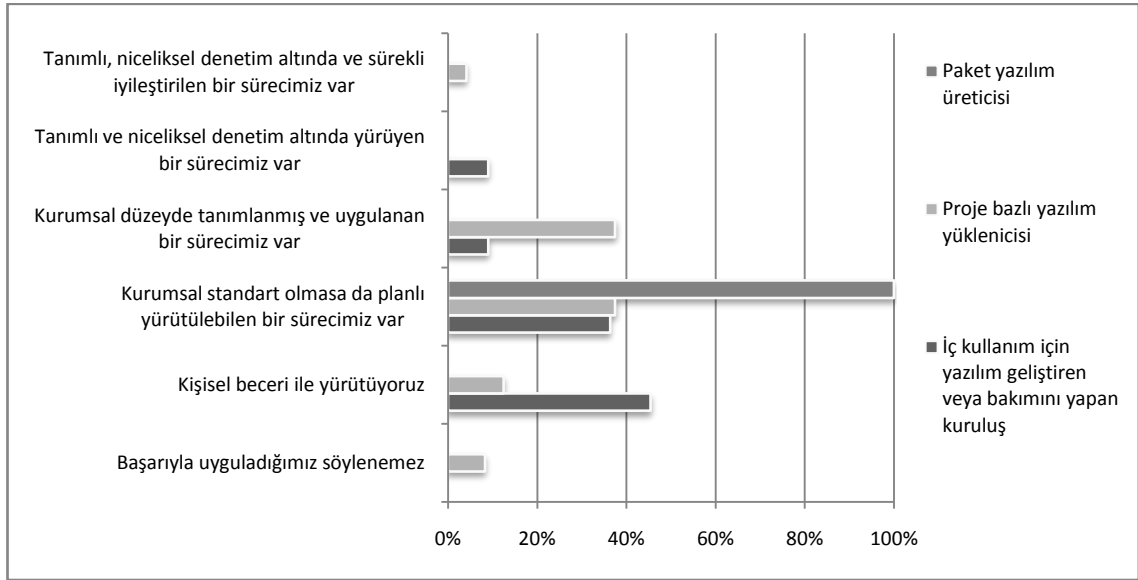
Tablo 4.7 : Kuruluşunuzda yazılım sınama senaryoları nasıl tanımlanıyor?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Sınama sırasında anlık oluşturuluyor	0%	17%	0%
Gereksinim analisti tarafından tanımlanıyor	27%	8%	0%
Tasarım mühendisi tarafından tanımlanıyor	0%	8%	0%
Programcılar tarafından tanımlanıyor	18%	8%	0%
Bu işle görevlendirilen bir sınama mühendisi tarafından tanımlanıyor	18%	25%	0%
Takım Lideri tarafından tanımlanıyor	0%	21%	100%
Proje Yöneticisi tarafından tanımlanıyor	18%	13%	0%
Müşteri tarafından tanımlanıyor	18%	0%	0%
Hiçbiri	0%	0%	0%

Yazılım Sınama: 2001 yılında yazılım sınama konusunda, proje bazlı ve iç kullanım amaçlı yazılım geliştiren firmalarda kurumsal standart olmasa da planlı süreç, paket

yazılım üreticilerinde ise kurumsal düzeyde tanımlanmış ve uygulanan süreç bulunmaktadır.

2008 yılında ise proje bazlı yazılım geliştiren firmaların “kurumsal düzeyde tanımlanmış ve uygulanan süreç” ve “kurumsal standart olmasa da planlı süreç” arasında yoğunlaştığı Şekil 4.90’da görülmektedir. İç kullanım amaçlı yazılım geliştiren firmalarda ise “kurumsal standart olmasa da planlı süreç” ve en çok “kişisel beceri ile yürütülen süreç” yer almaktadır.



Şekil 4.90 : Yazılım sınaama süreç profili

Bu durum, proje bazlı yazılım geliştiren firmalar süreç olgunluğu açısından bir üst düzeye doğru ilerlerken, iç kullanım amaçlı yazılım geliştiren firmaların bu süreçte ilerleme göstermediğine işaret etmektedir.

4.4.6 Yazılım Bakımı

Bakım Kaynakları: Hem proje bazlı hem de iç kullanım için yazılım geliştiren firmalar için 2008 yılında da 2001 yılında olduğu gibi Tablo 4.8’de oranları verilen, bakım işlerinin en önemli kaynağı/nedeni, iş süreçlerinin değişmesi ve ürün geliştirilirken gereksinimlerin iyi anlaşılabilmesidir. Sistem konfigürasyonunun değişmesi ile yeni arayüz gereksinimlerinin ortaya çıkması ve yeni yazılım teknolojilerine ayak uydurma

isteği sonraki sebepler olarak sıralanmaktadır. Özellikle iç kullanım için yazılım geliştiren firmalarda, yüzde 64 oranında gereksinimlerin iyi anlaşılammış olması dikkat çekici bir durumdur. Bu tür iç bölümlerin firmanın iş süreçlerine daha hakim olacağı düşüncesi ile yaklaşıldığında bu sorunun sebebi ancak yazılım süreçlerine bir metodoloji dahilinde sistematik yaklaşılması olabilir.

Tablo 4.8 : Hangileri yazılım bakım işlerinin en önemli kaynaklarındandır?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Yazılım gereksinimlerinin ürün geliştirilirken iyi anlaşılmaması	64%	44%	0%
Yazılımın desteklediği iş süreçlerinin zamanla değişmesi	86%	88%	0%
Sistem konfigürasyonunun değişmesi ile yeni arayüz gereksinimlerinin ortaya çıkması	43%	44%	100%
Yeni yazılım teknolojilerine ayak uydurma isteği	36%	44%	0%
Başka nedenler	14%	36%	0%

Bakım Türleri: 2008 yılında da 2001 yılında olduğu gibi iyileştirici bakım (yeni özellikler, vs) firmalarda en çok kaynağı tüketmektedir. Tablo 4.9’da oranları verilen önleyici bakım yaklaşımı ise sadece iç kullanım için yazılım geliştiren firmalarda görülmektedir.

Tablo 4.9 : Yazılım bakım türlerinden hangisi en fazla kaynağı tüketiyor?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Düzeltilici bakım (hataları giderme, vb.)	14%	36%	0%
Uyarlayıcı bakım (yeni arayüzler, vb.)	7%	8%	100%
İyileştirici bakım (yeni özellikler, vb.)	57%	56%	0%
Önleyici bakım (olası hataları önleme amaçlı)	21%	0%	0%

Bakım Personeli: 2008 yılında da 2001 yılında olduğu gibi yazılım geliştirilmesinde görev alan personel, Tablo 4.10’da verilen oranlarda yazılımın bakımında da görev almaktadır.

Tablo 4.10 : Yazılım ürününü geliştiren personel bakımında görev alıyor mu?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Her zaman	71%	40%	0%
Çoğunlukla	21%	52%	100%
Nadiren	7%	8%	0%
Bakımı başka kuruluş / birim üstleniyor	0%	0%	0%

Gereksinim Toplama: Tablo 4.11’de proje bazlı yazılım geliştiren firmalarda, müşteri bakım istekleri, 2001 yılında en yüksek oranda Proje Yöneticisi tarafından alınırken, 2008 yılında bu görevin ağırlığının teknik destek birimine kaydığı görülmektedir.

2001 yılında iç kullanım için yazılım geliştiren firmalarda çoğunlukla herhangi bir proje üyesi tarafından gerçekleştirilen gereksinim toplama görevinin ise Proje Yöneticisine kaydığı görülmektedir. Bu durum bu konuda da proje bazlı firmaların iç kullanım için yazılım geliştiren firmalardan daha yüksek bir olgunluk düzeyine/sistematik yaklaşımlara sahip olduğunun bir başka göstergesidir.

Tablo 4.11 : Yazılımla ilgili sorun ve yeni gereksinimleri kim topluyor?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Teknik destek birimi	21%	44%	0%
Konfigürasyon Yöneticisi	0%	0%	0%
Ürün Yöneticisi	14%	8%	0%
Takım Lideri	14%	4%	100%
Proje Yöneticisi	29%	32%	0%
Bölüm Yöneticisi	7%	0%	0%
Herhangi bir proje üyesi	14%	12%	0%
Hiçbiri	0%	0%	0%

Gereksinim Değerlendirme: 2001 yılı sonuçlarına göre müşteri bakım isteklerinin değerlendirmesinde en yüksek oranda; proje bazlı ve iç kullanım için yazılım geliştiren firmalarda Proje Yöneticisi ve müşteri, paket yazılım üreticilerinde ise Teknik Destek Birimi ve Proje Yöneticisi etkin olmaktadır. 2008 yılında ise oransal dağılımı Tablo 4.12’de görüldüğü şekilde, bu görevi birinci öncelikli olarak müşterinin aldığı, sonrasında Proje Yöneticisinin yüksek oranda söz sahibi olduğu ve beyanlara göre geçen süre zarfında önemli bir iyileşme olduğu görülmektedir.

Tablo 4.12 : Sorun/gereksinim değerlendirme/önceliklendirmesinde rol alanlar

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Müşteri	86%	88%	100%
Teknik destek birimi	29%	48%	100%
Konfigürasyon Yöneticisi	0%	12%	0%
Değişiklik / Konfigürasyon Kontrol Kurulu	0%	4%	0%
Ürün Yöneticisi	36%	24%	100%
Takım Lideri	36%	28%	100%
Proje Yöneticisi	71%	68%	0%
Bölüm Yöneticisi	43%	8%	0%
Herhangi bir proje üyesi	7%	16%	0%
Başka görevliler	21%	8%	0%

Bakım Kolaylığı: 2001 yılı sonuçlarına göre; gelecekte ürün bakım isteklerinin kolaylaştırılması için; proje bazlı yazılım geliştiren ve paket yazılım üreten firmalarda çoğunlukla kurumsal kodlama standartları kullanılmakta, iç kullanım amaçlı yazılım geliştiren firmalarda ise ürünü geliştiren personeli bakım aşamasında da görevlendirme yaklaşımı yer almaktadır.

Detayı Tablo 4.13'te görülen 2008 yılı verilerine göre; en yüksek oran iç kullanım için yazılım geliştirenlerde (yüzde 71) olmak üzere tasarımı bakım kolaylığı açısından gözden geçirme yaklaşımı öne çıkmakta, sonrasında kodu bakım kolaylığı açısından gözden geçirme yaklaşımı yer almaktadır.

Bu göstergeler de metodoloji kullanımını ön plana çıkarmakta ve iyi uygulamaların zaman içinde arttığını göstermektedir.

Tablo 4.13 : Bakımı kolaylaştırmak için hangi önlemler uygulanıyor?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Bileşen bazlı geliştirme	36%	68%	0%
Tasarımı bakım kolaylığı açısından gözden geçirme	71%	56%	0%
Kodu bakım kolaylığı açısından gözden geçirme	57%	56%	0%
Kurumsal kodlama standartları	50%	56%	100%
Kurumsal teknik belgeleme standartları	21%	36%	0%
Ürünü geliştiren personeli bakım aşamasında da görevlendirme	50%	60%	100%
Birim ve tümleştirme sınamaları için standart araçlar kullanma	7%	20%	0%
Başka önlemler	14%	20%	0%

Bakım Ölçümleri: 2001 yılı verileri içinde; paket yazılım üreticilerinde diğer firmalara göre daha yüksek oranlarda bakım ölçümleri yapılmaktadır. 2008 yılı verileri içinde paket yazılım firma sayısının sadece üç tane olması sebebi ile bu konuda kıyaslama yapmak doğru olmayacaktır.

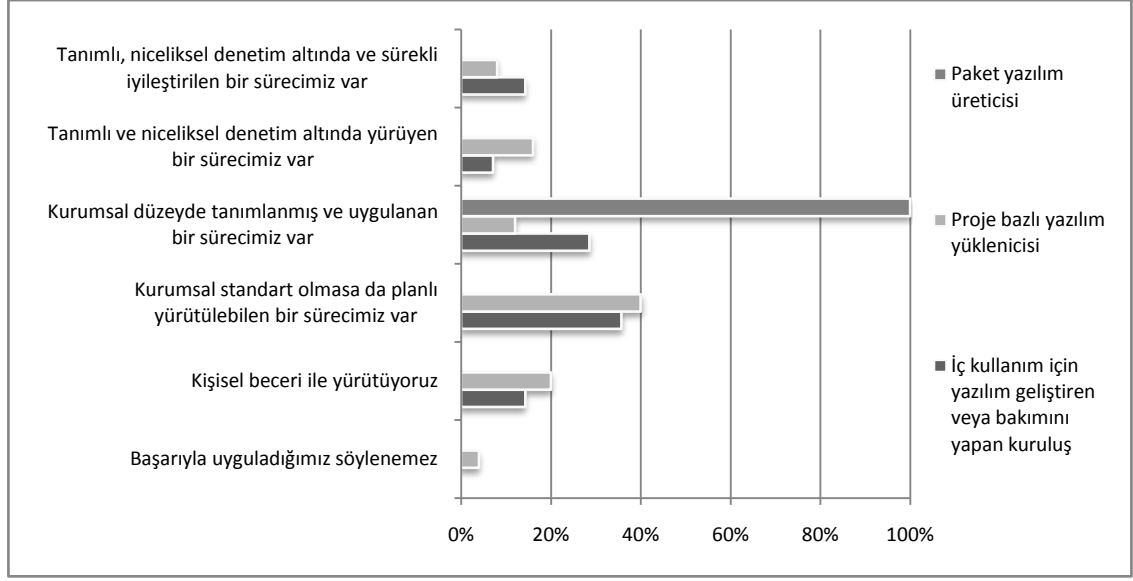
2008 yılında, Tablo 4.14'te görüldüğü gibi, iç kullanım için yazılım geliştiren firmalarda işin doğasına uygun olarak daha yüksek oranda bakım süreç ölçümü yapılmaktadır. Sonuçlandırılmamış sorun bildirimlerinin sayısı ve sorun bildirimlerinin ortalama sonuçlandırılma süresi en çok yapılan ölçümlerdir. Proje bazlı yazılım geliştiren firmalarda ise sonuçlandırılmamış sorun bildirimlerinin sayısı ve sorun bildirimlerinin dönemsel yoğunluğu daha çok ölçümlenmektedir.

Tablo 4.14 : Bakım aşamasında ölçümlerden hangilerini kullanıyorsunuz?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Sorun bildirimlerinin dönemsel yoğunluğu	43%	48%	0%
Sonuçlandırılmamış sorun bildirimlerinin sayısı	86%	68%	0%
Sorun bildirimlerinin ortalama sonuçlandırılma süresi	57%	40%	100%
Sorunları ortalama sonuçlandırma maliyeti	21%	32%	100%
Sorun bildirimlerinin modül bazında dağılımı	50%	40%	0%
Sorun bildirimlerinin müşteri bazında dağılımı	43%	44%	0%
Sorun bildirimlerinin hata türü bazında dağılımı	43%	44%	0%
Başka ölçümler	29%	16%	0%

Yazılım Bakımı: Yazılım bakımı konusunda (Şekil 4.91), 2001 yılında iç kullanım amaçlı yazılım geliştiren firmalarda en yüksek oranda “kurumsal standart olmasa da planlı süreç”, paket yazılım üreticilerinde ise “kurumsal düzeyde tanımlanmış ve uygulanan süreç” ile “tanımlı ve niceliksel denetim altında yürüyen süreç” bulunmaktadır.

2008 yılında ise bu konuda soruları yanıtlayan ve iç kullanım için yazılım geliştiren 28 firmanın beyanları doğrultusunda, süreç olgunluk düzeyinin yükseldiği ve bu firmalarda daha iyi uygulamaların yapıldığı görülmektedir. Proje bazlı yazılım geliştiren firmaların beyanları daha çok SPICE II. düzey ve üzerine işaret etmektedir.



Şekil 4.91 : Yazılım bakımı süreç profili

4.4.7 Yazılım Konfigürasyon Yönetimi

Politika, Prosedür, Standart: 2001 yılında tüm kuruluş türlerinde yazılı politika, prosedür ve standartlar bulunmakla beraber kısmen uygulanabilmektedir. Proje bazlı yazılım geliştiren firmalarda ise daha yüksek oranda “yazılı olmayan, ama uygulanan kurallar” bulunmaktadır.

2008 yılında iç kullanım için yazılım geliştiren firmalarda daha çok yazılı olmayan ama uygulanan bazı kurallar bulunmakta (Tablo 4.15), proje bazlı yazılım geliştiren firmalarda yazılı politika, prosedür ve standartlar bulunmakta ve tam olarak uygulandığı beyan edilmektedir.

Bu süreçte de proje bazlı yazılım geliştiren firmaların daha yüksek olgunluk düzeyine sahip olduğu ve iç kullanım için yazılım geliştiren firmalara ait 2001 yılı beyanlarının, gerçek uygulamalarına göre daha iyimser olduğu görülmektedir.

Tablo 4.15 : Konfigürasyon yönetimi politika, prosedür ve standartları

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Yazılı politika, prosedür ve standartlar var ve tam olarak uygulanıyor	13%	35%	50%
Yazılı politika, prosedür ve standartlar var ama kısmen uygulanabiliyor	38%	29%	0%
Yazılı politika, prosedür ve standartlar var, ama uygulanamıyor	0%	0%	0%
Yazılı olmayan ama uygulanan bazı kurallar var	50%	24%	50%
Politika, prosedür, standart veya kurallar olduğu söylenemez	0%	6%	0%
Bilmiyorum	0%	6%	0%

Konfigürasyon Yönetim Planı: 2001 yılı verilerine göre konfigürasyon yönetim planı içinde, en çok Proje Yöneticisinin görevi tanımlanmaktadır. Proje bazlı ve iç kullanım için yazılım geliştiren bazı firmalarda Konfigürasyon Yöneticisi ile Konfigürasyon Kontrol Kurulunun da görevleri tanımlanmaktadır.

2008 yılında iç kullanım için yazılım geliştiren firmalarda, konfigürasyon planı içinde sadece Proje Yöneticisi, proje bazlı yazılım geliştiren firmalarda ise en çok Proje Yöneticisi (yüzde 76), sonrasında Konfigürasyon Yöneticisi (yüzde 41) ve Değişiklik/Konfigürasyon Kontrol Kurulu (yüzde 24) rolleri tanımlanmaktadır.

Konfigürasyon Denetlemeleri: 2001 yılı verileri içinde projelerde konfigürasyon yönetimi itibarı ile gerçekleştirilen denetlemeler kuruluş türüne göre ayrıştırılarak incelendiğinde; iç kullanım için yazılım üreten firmalarda, müşterinin istediği denetlemeler; paket yazılım üreticilerinde, sistem ve konfigürasyon öğelerinin işlevsel bütünlüğüne odaklanan denetlemeler; yazılım yüklenicilerinde, süreç, sistem ve müşterinin istediği denetlemeler gerçekleştirilmektedir.

2008 yılında da (Tablo 4.16); iç kullanım için yazılım geliştiren firmalarda müşterinin istediği denetlemeler ve süreç denetlemeleri, paket yazılım üreticilerinde ağırlıklı olarak süreç denetlemeleri ve sonrasında diğer denetlemeler görülmektedir.

Yapılan denetlemeler içinde en küçük yüzdeyi konfigürasyon öğelerinin fiziksel bütünlüğüne odaklanan denetlemeler almaktadır.

Tablo 4.16 : Denetleme türlerinden hangileri projelerinizde uygulanıyor?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Konfigürasyon öğelerinin işlevsel bütünlüğüne odaklanan denetlemeler	13%	35%	0%
Konfigürasyon öğelerinin fiziksel bütünlüğüne odaklanan denetlemeler	0%	12%	50%
Müşterinin istediği denetlemeler	63%	35%	50%
Sistem denetlemeleri	13%	35%	50%
Süreç denetlemeleri	50%	65%	0%

Konfigürasyon Dayanağı Yaklaşımları: 2001 yılı verileri içinde (Tablo 4.17), projelerde dayanak tanımlama yaklaşımları kuruluş türüne göre incelendiğinde; iç kullanım için yazılım üretenlerde en çok sistem gereksinimleri bazında işlevsel dayanak, yazılım yüklenicisi firmalarda, geliştirme dayanağı ve paket yazılım üreticisi firmalarda geliştirme dayanağı, ürün dayanağı ve atanmış dayanakları aynı oranda gözlenmektedir.

2008 yılı verileri içinde (Tablo 4.17) en ağırlıklı yöntem olarak işlevsel dayanak, sonrasında iç kullanım için yazılım üreten firmalarda ürün dayanağı yaklaşımı, proje bazlı yazılım üreten firmalarda ise geliştirme dayanağı yaklaşımları görülmektedir.

Tablo 4.17 : Dayanak tanımlama yaklaşımlarından hangileri kullanılıyor?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
İşlevsel dayanak (functional baseline: sistem gereksinimleri bazında)	63%	65%	0%
Atanmış dayanak (allocated baseline: yazılım gereksinimleri bazında)	13%	29%	50%
Geliştirme dayanağı (developmental baseline:geliştirme aşamaları bazında)	13%	47%	50%
Ürün dayanağı (product baseline: ürün sürümleri bazında)	63%	24%	50%

Konfigürasyon Belirleme: 2001 yılı verilerine göre, konfigürasyon belirleme çalışmaları, proje bazlı ve iç kullanım amaçlı yazılım geliştiren firmalarda paket yazılım üreticilerine kıyasla daha fazla alanda ve oranda yapılmaktadır. Diğer kuruluşlarda paket yazılım üreticilerine göre daha fazla konfigürasyon öğesi tanımlanmaktadır.

2008 yılı verilerine göre de (Tablo 4.18) en çok proje bazlı yazılım üreten firmalarda (yüzde 88) olmak üzere konfigürasyon öğeleri belirlenmektedir. Bunun dışındaki süreç

gereklere iç kullanım için yazılım geliştiren firmalarda düşük oranda, proje bazlı yazılım geliştiren firmalarda ise yüzde 29-35 aralığında uygulanmaktadır.

Tablo 4.18 : Projelerin konfigürasyon belirleme aşamasında hangileri yapılır?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Konfigürasyon öğeleri belirlenir	63%	88%	50%
Konfigürasyon öğeleri arasındaki ilişkiler tanımlanır	0%	29%	50%
Konfigürasyon öğelerinin isimlendirme yöntemi tanımlanır	13%	35%	100%
Konfigürasyon öğelerinin sürüm numaralandırma yöntemi tanımlanır	13%	29%	100%
Konfigürasyon öğeleri veritabanında saklanır	0%	29%	50%

Konfigürasyon Değişiklik Kontrolü: 2001 yılı verileri içinde paket yazılım geliştiren firmalarda daha yüksek oranda, “değişikliğin diğer konfigürasyon öğelerine etkileri” kişisel düzeyde ele alınmaktadır.

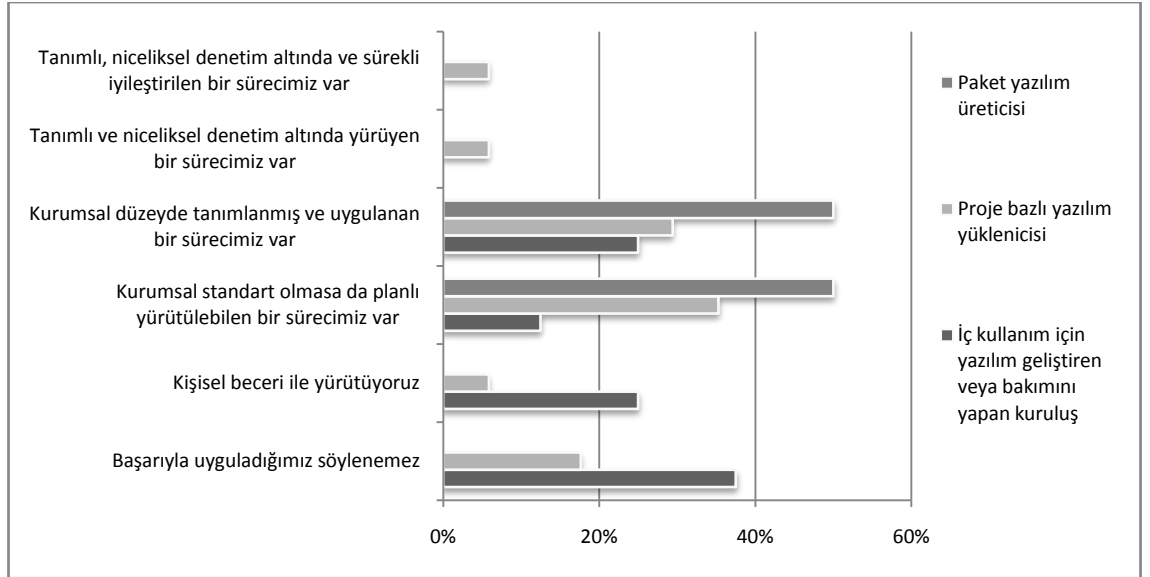
2008 yılında (Tablo 4.19) konfigürasyon değişiklik kontrolü konusundaki iyi uygulamaların, özellikle proje bazlı geliştirme yapan firmalarda arttığı görülmektedir. Kişisel inisiyatifin yerini teknik gözden geçirmeler ve kurumsal düzeyde belgelenmiş prosedürler almıştır. İç kullanım için yazılım geliştiren firmalarda hala kişisel inisiyatif ile değişiklik kontrolü ön plandadır. Bu durum bu süreçte de proje bazlı yazılım üreten firmaların daha iyi uygulamalara sahip olduğunun göstergesidir.

Tablo 4.19 : Konfigürasyon değişiklik kontrolü aşamasında neler uygulanıyor?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Değişiklik kontrolü kişisel inisiyatif ile yapılıyor	63%	41%	50%
Kurumsal düzeyde belgelenmiş bir prosedür izleniyor	13%	35%	0%
Teknik gözden geçirmelerle ürün kontrolü sağlanıyor	13%	47%	0%
Değişiklik istekleri sadece proje yönetimi tarafından değerlendirilip onaylanıyor	25%	24%	50%
Değişiklik istekleri bir Değişiklik Kontrol Kurulu’nda değerlendiriliyor	0%	24%	0%
Değişiklik istekleri bir araç yardımıyla değerlendiriliyor	25%	18%	0%
Değişiklik isteğinden etkilenen taraflara gerekli bildirim yapılıyor	38%	12%	50%
Önerilen değişikliğin diğer konfigürasyon öğelerine etkileri kişisel düzeyde ele alınıyor	25%	0%	0%
Önerilen değişikliğin diğer konfigürasyon öğelerine etkileri kurumsal bir yöntem..	0%	18%	0%

Konfigürasyon Yönetimi: 2001 yılı sonuçları içinde, proje bazlı ve iç kullanım amaçlı yazılım geliştiren firmalarda farklı kuruluşlarda farklı düzeylerde uygulamalar gerçekleştirilmektedir. Paket yazılım üreticilerinde ise büyük oranda “kurumsal standart olmasa da planlı yürütülebilir süreç” bulunmaktadır.

2008 yılı sonuçları gözönüne alındığında, geçen süre zarfında konfigürasyon yönetimi süreci daha iyi anlaşılmiş ve özellikle proje bazlı yazılım üreten firmalarda daha iyi pratiklerle gerçekleştirilmektedir (Şekil 4.92).



Şekil 4.92 : Yazılım konfigürasyon yönetimi süreç profili

4.4.8 Yazılım Mühendisliği Yönetimi

Politika, Prosedür, Standart: 2001 yılında firmalarda yazılı politika, prosedür ve standartlar bulunmakla birlikte kısmen uygulanmaktadır.

2008 yılı verilerinde (Tablo 4.20), iç kullanım için yazılım geliştiren firmalarda yüzde 50 oranında yazılı politika, prosedür ve standartlar bulunmakla birlikte kısmen uygulanabilmektedir. Daha az oranda (yüzde 17) firmada yazılı politika, prosedür ve standartlar, tam olarak uygulanmaktadır.

Proje bazlı çalışan yazılım firmalarında ağırlık tam ve kısmen uygulanabilen yazılı politika, prosedür ve standartlar arasında dağılmıştır. Bu süreçte de proje bazlı çalışan firmaların beyanları, iç kullanım için yazılım geliştiren firmalara göre daha iyi uygulamalara işaret etmektedir.

Tablo 4.20 : Yazılım mühendisliği politika, prosedür ve standartları

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Yazılı politika, prosedür ve standartlar var ve tam olarak uygulanıyor	17%	38%	0%
Yazılı politika, prosedür ve standartlar var ama kısmen uygulanabiliyor	50%	33%	50%
Yazılı politika, prosedür ve standartlar var ama uygulanamıyor	8%	5%	0%
Yazılı olmayan ama uygulanan bazı kurallar var	17%	14%	50%
Politika, prosedür, standart veya kurallar olduğu söylenemez	8%	10%	0%
Bilmiyorum	0%	0%	0%

Organizasyon Yapısı: Kuruluşlarda yazılım organizasyon yapıları firma türüne göre ayrıştırıldığında (Tablo 4.21), proje bazlı yazılım geliştiren firmalarda 2001 ve 2008 yıllarına ait sonuçlar benzer yapıdadır.

İç kullanım için yazılım geliştiren firmalarda 2001 yılında en yüksek oranda uzmanlık alanları ile proje ekipleri arasında matris bir yapı bulunurken, 2008 yılında matris yapı uygulaması azalmış, yerini hiyerarşik yapılar almıştır.

Tablo 4.21 : Yazılım organizasyon yapısı ile ilgili olarak hangisi geçerlidir?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Yazılım mühendisliği için tanımlanmış özel bir organizasyon yapısı yok	25%	14%	0%
Ağırlıklı olarak fonksiyonel ve uzmanlık alanlarını temel alan hiyerarşik yapı	67%	29%	0%
Beceri havuzlarından kaynak kullanımına dayanan proje bazlı bir yapı var	0%	24%	100%
Uzmanlık alanları ile proje ekipleri arasında matris bir yapı var	8%	5%	0%
Alınan işlere ve proje gereklerine göre tasarlanan geçici yapılar	0%	29%	0%
Hiçbiri	0%	0%	0%

Yazılım Proje Planı: 2001 yılı verilerine göre, paket yazılım üreticisi firmalarda proje planları daha çok detay içermektedir. Proje ölçümleri, sadece iç kullanım için yazılım üreten firmaların proje planlarında yer almaktadır. Yazılım yüklenicisi firmalarda

planlama daha önemli olduğu halde, proje planının kapsam olarak daha zayıf olduğu görülmektedir. Altyüklenici yönetim planı, en çok iç kullanım için yazılım üreten firmalarda kullanılmaktadır.

Tablo 4.22 : Kuruluşunuzda yazılım Proje Planı'nda hangileri mutlaka yer alır?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Projenin hedefi ve sonuçlanma kriterleri	75%	90%	50%
Proje organizasyon yapısı ve roller	50%	81%	100%
Proje iletişim mekanizmaları	50%	62%	50%
Proje kapsam yönetimi (hedef, bütçe ve ürünlerin değişimi)	50%	71%	50%
Gelişim izleme ve denetim yöntemi	33%	48%	100%
Kullanılacak süreç, prosedür ve standartlar	25%	52%	0%
Seçilen yazılım yaşam döngüsü modeli ve gerekçesi	25%	38%	0%
Proje ürünlerinin ve ara ürünlerin tanımı	25%	57%	50%
Proje teknik belgelerinin tanımı	42%	43%	50%
Proje insan kaynakları ve altyapı kullanımı	42%	57%	50%
Satınalma / altyüklenici yönetim planı	25%	48%	0%
Proje görev / zaman çizelgesi (Gantt şeması, vb.)	75%	81%	50%
Proje riskleri ve risk yönetim metodolojisi	42%	57%	50%
Süre ve emek (adam-gün) kestirimleri ve yanılma payları	58%	71%	50%
Proje ölçümleri	25%	38%	50%
Başka bilgiler	25%	14%	0%

2008 yılı verileri içinde (Tablo 4.22); proje bazlı yazılım yüklenicisi firmalar tarafından hazırlanan proje planlarının, daha kapsamlı olduğu görülmektedir.

İç kullanım için yazılım geliştiren firmaların hazırladığı proje planlarında; yüzde 75 oranda proje ölçümleri, yüzde 58 oranda proje risk yönetimi, yüzde 50 oranında proje kapsamı, yüzde 50 oranda proje organizasyonu, yüzde 50 oranda proje iletişimi, yüzde 32 oranda süre ve emek tahminleri bulunmamaktadır.

Proje bazlı yazılım geliştiren firmaların hazırladığı proje planlarında; yüzde 62 oranda proje ölçümleri, yüzde 53 oranında proje risk yönetimi, yüzde 29 oranında proje kapsamı, yüzde 38 oranında proje iletişimi, yüzde 29 oranında süre ve emek tahminleri bulunmamaktadır.

Ölçümlemenin Benimsenmesi: 2001 yılı verileri içinde, yazılım ölçme uygulamalarının benimsenmesi konusunda da sorun olduğu ve soruna bakış açısının çalışanın görevi ve firmanın türüne göre değişkenlik gösterdiği görülmektedir.

Sonuçlar firma türüne göre ayrıştırıldığında;

- İç kullanım için yazılım üreten firmalarda; iş yükünün ağırlığı, çalışanların isteksizliği ve teknik bilgi eksikliği,
- Yazılım yüklenicileri için; iş yükünün ağırlığı ve bütçe imkanları,
- Paket yazılım üreticileri için; iş yükünün ağırlığı, bütçe imkanları ve Üst Yönetim desteği ölçme konusunda sınırlayıcı olmaktadır.

2001 yılı verileri içinde de 2008 yılında da yazılım ölçümlerinin benimsenmesine engel olan faktörler içerisinde en yüksek oranı iş yükünün ağırlığı tutmaktadır. Sonrasında çalışanların ölçümleme konusunda isteksizliği ve firma kültürünün uygun olmaması sebepler arasında yer almaktadır.

(Tablo 4.23) İç kullanım için yazılım geliştiren firmalarda, yüzde 42 oranda Üst Yönetim desteği alınmadığı için ölçümleme yapılmaması dikkat çekici sonuçlar içinde yer almaktadır.

Tablo 4.23 : Ölçme uygulamalarının benimsenmesine hangileri engel oluyor?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
İş yükünün ağırlığı	67%	81%	50%
Bütçe imkanlarının kısıtlı olması	8%	29%	0%
Mevcut teknolojinin yetersiz olması	8%	10%	0%
Üst Yönetimden yeterli desteğin alınmaması	42%	19%	0%
Çalışanların bu tür uygulamalara karşı isteksizliği	67%	52%	50%
Bu tür uygulamalar için gerekli teknik bilginin olmaması/edinilme zorluğu	42%	19%	100%
Firma kültürünün uygun olmaması	42%	33%	50%
Başka faktörler	8%	29%	0%

Yazılım Ölçümleri: 2001yılı verileri kuruluş türüne göre incelendiği zaman; yazılım hatalarının en çok paket yazılım üreticileri tarafından ölçüldüğü, yazılım

yüklenicilerinin daha az ölçüm yaptığı ve yazılım büyüklüğünün paket yazılım üreticileri tarafından hiç ölçülmediği dikkat çeken sonuçlar arasında yer almaktadır.

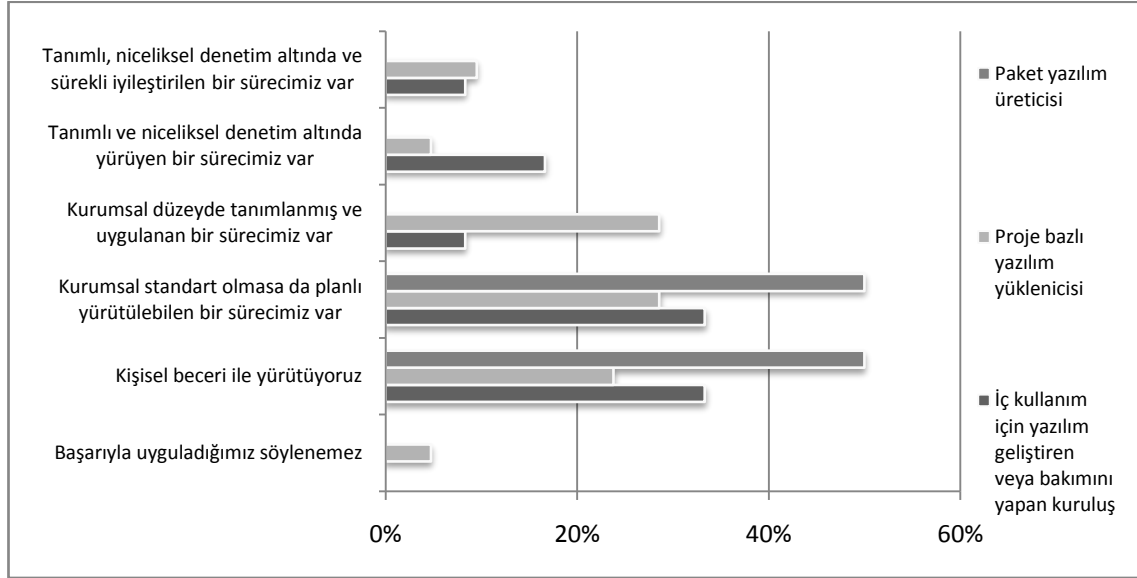
2008 yılında ise (Tablo 4.24), proje bazlı yazılım geliştiren firmalarda daha çok ölçüm yapıldığı görülmektedir. Proje bazlı yazılım geliştiren firmalarda en çok gereksinim sayısı (yüzde 57), yazılım hataları (yüzde 48) ve faaliyet bazında emek ölçümleri (yüzde 43) yapılmaktadır. Proje bazlı firmalarda yüzde 62 oranda projelerde aşama bazında süre ve emek ölçümlerinin yapılmıyor olması proje başarımı açısından dikkat çekicidir.

Tablo 4.24 : Yazılım ölçümlerinden hangileri kuruluşunuzda uygulanıyor?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Gereksinim sayısı	25%	57%	0%
Yazılım hataları (tür, kritiklik, neden, olduğu yer, giderildiği yer, vb.)	33%	48%	50%
Faaliyet bazında emek (analiz, tasarım, kodlama, sınav, destek, vb.)	33%	43%	50%
Yazılım büyüklüğü (satır sayısı, function point, vb.)	25%	19%	100%
Algoritmik karmaşıklık	8%	10%	0%
İki hata oluşumu arası ortalama süre (mean-time-to-failure)	0%	5%	0%
Sınav kapsamı (test coverage)	8%	19%	0%
Yeniden kullanım yüzdesi	0%	10%	0%
Belgeler için önerilen / gerçekleşen değişiklikler	8%	14%	0%
Aşama bazında kestirilen / gerçekleşen kadro (tam zaman eşdeğeri adam)	17%	14%	50%
Aşama bazında kestirilen / gerçekleşen emek (adam-saat)	25%	38%	50%
Aşama bazında kestirilen / gerçekleştirilen süre	33%	38%	50%
Aşama bazında kestirilen maliyet / gerçekleşen maliyet	25%	33%	0%
Aşama bazında kestirilen / gerçekleşen faaliyetlerin sayısı	8%	19%	0%
Gereksinimlerin zamana bağlı gelişimi	8%	19%	0%
Ürünün müşteri gereksinimlerini karşılama başarısı	25%	38%	0%
Hizmetin müşteri gereksinimlerini karşılama başarısı	33%	29%	0%
Başka tür yazılım ölçümleri	25%	24%	50%

Yazılım Mühendisliği: 2001 yılı verilerinde yazılım mühendisliği konusunda, proje bazlı yazılım geliştiren firmalarda farklı düzeylerde uygulamalar gerçekleştirilmektedir. İç kullanım amaçlı yazılım geliştiren firmalarda en yüksek oranda “kurumsal standart olmasa da planlı süreç”, paket yazılım üreticilerinde ise “kurumsal düzeyde tanımlanmış süreç” bulunmaktadır.

2008 yılında (Şekil 4.93) iç kullanım için yazılım geliştiren firmalarda, bu süreçte önemli ölçüde iyileşme olduğu beyan edilmekle birlikte proje bazlı çalışan firmaların daha iyi uygulamalara sahip olduğu görülmektedir.



Şekil 4.93 : Yazılım mühendisliği süreç profili – 2008

4.4.9 Yazılım Süreç Mühendisliği

Süreç Standartları: 2001 yılında iç kullanım amaçlı yazılım geliştiren firmalarda en yüksek oranda, “hiç bir standardı temel almıyoruz” seçeneği işaretlenmiştir. Çeşitli standartlar arasında yaygın olarak SPICE kullanılmaktadır.

2008 yılına gelindiğinde (Tablo 4.25), iç kullanım amaçlı yazılım geliştiren firmaların yüzde 56’sında, proje bazlı yazılım geliştiren firmaların yüzde 36’sında hala hiçbir standart temel alınmamaktadır.

Kullanılan metodolojinin ağırlığı SPICE’den CMMI’a doğru kaymıştır.

Tablo 4.25 : Yazılım süreç standartlarından hangisini temel alıyorsunuz?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
ISO-15504 (SPICE)	11%	7%	0%
SW/CMM veya CMMI	33%	43%	0%
ISO/IEC-12207 veya IEEE/EIA Std 12207	0%	7%	0%
NATO AQAP-160	0%	0%	0%
MIL-STD-498	0%	0%	0%
Tick-IT	0%	0%	0%
Bootstrap	0%	0%	0%
TRILLIUM	0%	0%	0%
Başka bir standardı temel alıyoruz	0%	7%	100%
Hiçbir standardı temel almıyoruz	56%	36%	0%

Süreçlerle Yönetim: 2001 yılı verilerine göre beyanlar; yazılım yüklenicilerinin süreçlerle yönetim yapmadığını, iç kullanım için yazılım üreten firmaların süreçlerini tanımladığını ve paket yazılım üreticisi olan firmaların süreçleri tanımladığını ve süreç eğitimleri verdiğini göstermektedir.

2008 yılı verileri (Tablo 4.26), proje bazlı yazılım üreticilerinin süreçlerle yönetim konusunda da iç kullanım için yazılım üretenlerden daha iyi uygulamalara sahip olduğunu göstermektedir.

Tablo 4.26 : Kuruluşunuzda süreçlerle yönetim konusunda hangisi en doğru?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Süreçlerle yönetim yaptığımız söylenemez	22%	21%	0%
Süreçleri tanımlıyoruz	44%	29%	0%
Süreçleri tanımlıyor ve personeli eğitiyoruz	22%	21%	100%
Süreçleri tanımlıyor, personeli eğitiyor ve uygulamayı ölçümlerle izliyoruz	11%	7%	0%
Süreçleri tanımlıyor, personeli eğitiyor, uygulamayı ölçümlerle izliyor ve iyileştiriyoruz	0%	21%	0%

Yazılım Mühendisliği Bilgi Birikiminin Kurumsallaşması: 2008 yılında da 2001 yılında olduğu gibi proje bazlı ve iç kullanım için yazılım geliştiren firmalarda usta-çırak ilişkisi en yüksek orana sahiptir.

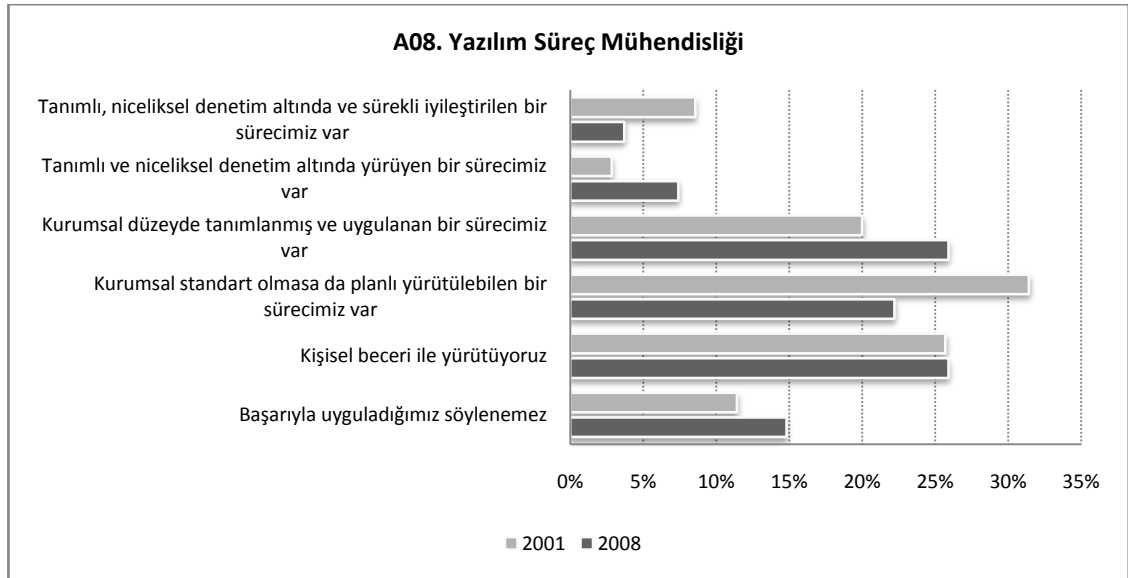
2008 yılında (Tablo 4.27), elektronik bilgi saklama ve yönetim sistemleri, bilgi

aktarımının bir ölçüde de olsa sözlü iletişimden yazılı/kayıtlı ortama taşınmasına destek olmuştur.

Tablo 4.27 : Yazılım mühendisliği bilgi birikimi nasıl kurumlaştırılıyor?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Usta-çırak ilişkisi ile	44%	50%	0%
İç seminerler ile	0%	7%	0%
Yol gösterici belgeler ile	0%	7%	50%
Elektronik bilgi saklama ve yönetim sistemleri ile..	33%	29%	50%
İç danışmanlar yardımıyla	0%	0%	0%
Teknik destek personeli yardımıyla	0%	0%	0%
Başka şekilde	22%	7%	0%

Süreç Tanımlama, İyileştirme: Süreç tanımlama ve iyileştirme konusunda proje bazlı yazılım geliştiren ve iç kullanım amaçlı yazılım geliştiren firmalarda farklı düzeylerde uygulamalar gerçekleştirilmektedir. Şekil 4.94'te yer alan beyanlara göre bu sürecin olgunluk düzeyinde 2001 ve 2008 yılları arasında önemli bir değişiklik bulunmamaktadır.



Şekil 4.94 : Yazılım süreç mühendisliği süreç profili – 2008

4.4.10 Yazılım Mühendisliği Araç ve Yöntemleri

Yazılım Mühendisliği Araçları: Araçların kaliteli yazılım üretmek için zorunlu olduğu görüşü zaman içinde değişmemekle birlikte iç kullanım için yazılım üreten firmalarda yüzde 43, proje bazlı yazılım geliştiren firmalarda yüzde 13 oranında araçların ölü yatırım olma riski olduğu düşünülmektedir (Tablo 4.28).

Tablo 4.28 : Yazılım mühendisliği araçları için hangi görüşlere katılıyorsunuz?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Bu tür araçlar verimliliği artırmak için zorunludur	57%	63%	0%
Bu tür araçlar kaliteli yazılım üretmek için zorunludur	71%	63%	0%
Bu tür araçların ölü yatırım olma riski yüksektir	43%	13%	0%
Bu tür araçlar kurumsal politikalarla desteklenmedikçe kullanılmamalıdır	71%	50%	100%
Bu tür araçların öğrenilmesi ve etkin kullanımı için çok zaman gerekir	43%	25%	0%

Araç Seçimi: Araç seçiminde kurul kararı olması, kurumsallaşmanın önemli bir göstergesi olmakla birlikte hala iç kullanım için yazılım geliştiren firmalarda yüzde 29, proje bazlı yazılım geliştiren firmalarda yüzde 13 oranında kuralsız seçim görülmektedir (Tablo 4.29).

Tablo 4.29 : Yazılım mühendisliği ve yönetimi araç seçimini kim yapıyor?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Proje Yöneticisi	0%	19%	0%
Takım Lideri	0%	0%	0%
Yazılım mühendisleri	0%	0%	100%
Bölüm Yöneticisi	14%	19%	0%
Üst Yönetim	14%	19%	0%
Bu amaçla oluşturulan özel kurul	43%	31%	0%
Teknik danışmanlar	0%	0%	0%
Kural yok; duruma bağlı olarak değişir	29%	13%	0%

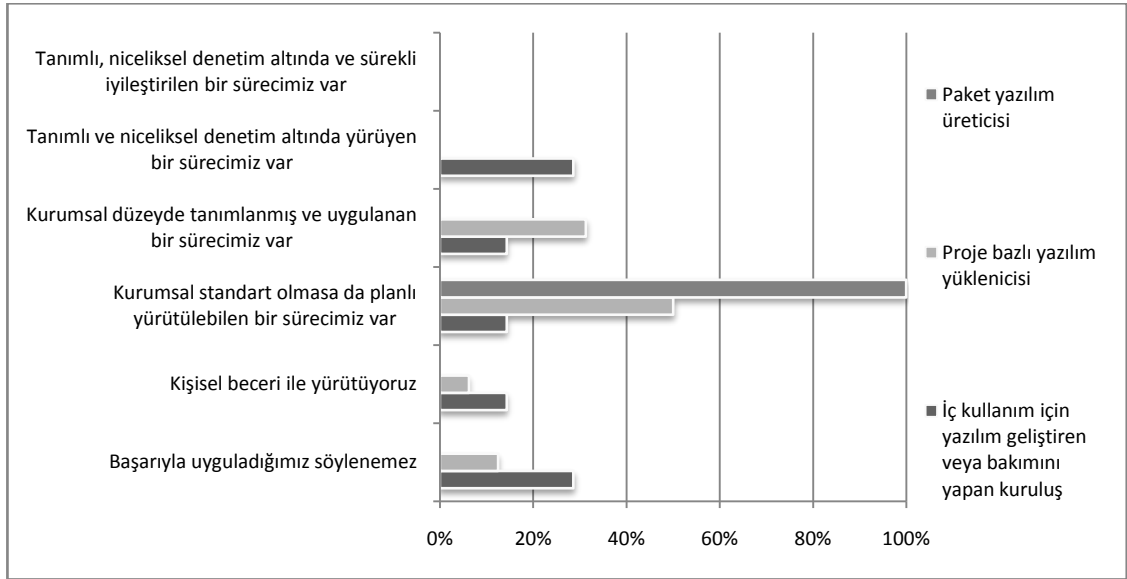
Mühendislik Yöntemleri: 2001 yılı verilerinde, yenilikleri izleme ve şirket politikası, mühendislik yöntemlerinin seçiminde en yüksek oranda etkili olmaktadır.

2008 yılında ise (Tablo 4.30), teknolojik eğilimleri izleme isteği ve ilgili iş sürecinin işleyişine uygunluk seçimde ön plandadır.

Tablo 4.30 : Mühendislik yöntemlerinin seçiminde hangi kriterler belirleyici?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Teknolojik eğilimleri izleme isteği	71%	63%	0%
Müşteri teşviki veya zorlaması	43%	31%	0%
İlgili iş sürecinin işleyişine uygunluk	71%	56%	100%
Rakiplerle farklılık yaratma isteği	29%	31%	0%
Danışman tavsiyesi	43%	25%	100%
Şirket politikası ve uzun vadeli hedeflere uygunluk	43%	56%	0%
Başka kriterler	29%	31%	0%

Araç ve Yöntemler: Araç ve yöntemler konusunda (Şekil 4.95), proje bazlı yazılım geliştiren ve iç kullanım amaçlı yazılım geliştiren firmalarda geçen zaman zarfında hem beyanlarda hem de uygulamalarda iyileşme görülmektedir.



Şekil 4.95 : Yazılım mühendisliği araç ve yöntemleri süreç profili – 2008

4.4.11 Yazılım Kalitesi

Ürün Kalite Kriterleri: 2008 yılı verilerine göre (Tablo 4.31), kullanım kolaylığı ve işlevsellik iç kullanım için yazılım geliştiren ve proje bazlı yazılım geliştiren firmalar için en yaygın ürün kalite kriterleri olarak tanımlanmıştır.

Tablo 4.31 : Çalıştığımız birimde hangileri ürün kalite kriterlerindedir?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
İşlevsellik (işlev zenginliği, doğruluğu, erişilebilirliği, vb.)	88%	89%	0%
Güvenirlilik (çalışma sürekliliği, sürüm olgunluğu, vb.)	75%	74%	0%
Verimli çalışma (işlemci ve diğer kaynak kullanımı, iletişim ve çıktı perf.)	63%	63%	100%
Kullanım kolaylığı (öğrenilebilirlik, kolay işletilebilirlik, vb.)	100%	63%	0%
Bakım kolaylığı (sorun analizi ve test kolaylığı, güncellenebilirlik, vb.)	50%	42%	100%
Taşınabilirlik (donanım/sistem bağımsızlığı, kurulum ve veri paylaşım ..)	25%	21%	100%
Tekrar kullanılabilirlik (diğer uygulamalara işlevsel arayüz sunma, bağımsı..)	25%	42%	0%
Başka kalite kriterleri	25%	26%	0%

Güvenirlilik: 2008 yılında da 2001 yılında olduğu gibi (Tablo 4.32), ürünün güvenirliliği pilot denemelerle ve kullanıcı kullanımında sürekli izlenerek belirlenmektedir.

Tablo 4.32 : Ürün güvenirliliği verilerini hangi yöntemlerle elde ediyorsunuz?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Kesintisiz çalışabilme kapasitesi kullanıcıya sunmadan önce pilot denem..	100%	58%	0%
Aynı ortamda mümkün olduğu kadar çok değişik yazılım / donanım bileş..	25%	21%	0%
Üründen kaynaklanmayan kısıtlayıcı koşullar oluşturup ürün davranışları izlenir	0%	21%	100%
Kullanım sırasında verilerin tutarlılığı zaman içinde örnekleme yöntemi ..	50%	58%	0%
Kullanım sırasında verilerin üründen kaynaklanmayan değişimlere uğra..	38%	26%	100%
Ürünün güvenirliliğini etkileyecek platform (işletim sistemi, veritabanı..)	25%	21%	0%
Verilere ve işlevlere erişimle ilgili ürünün üstlendiği tüm güvenlik adımları	38%	16%	0%
Başka yöntemler uygulanıyor	0%	21%	0%

Ürün kalitesi temel verileri: 2001 yılında, beyanlar ürün kalitesi ile ilgili verilerin yeterince değerlendirilmediğini göstermektedir. Firmalarda yüzde 36 oranında ürün kalitesi ile ilgili veri toplanmadığı sonuçlar arasında yer almaktadır. Paket yazılım üreticileri yüzde 89, Yazılım yüklenicileri yüzde 60, İç kullanım yazılım üreticileri yüzde 43 oranında ürün kalitesi ile ilgili veri toplamaktadır.

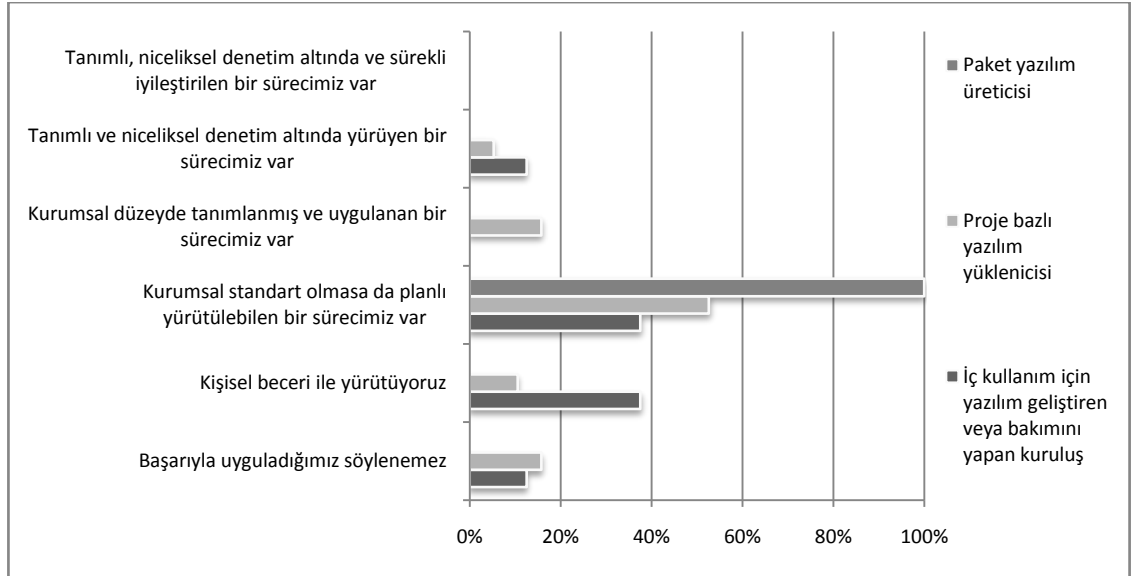
Tablo 4.33, 2008 yılında iç kullanım için yazılım üreten firmaların, ürün kalitesi konusunda daha çok veri topladığını göstermektedir.

Tablo 4.33 : Ürün kalitesi verileri ile ilgili hangisi mevcut durumu yansıtıyor?

SEÇENEK	İç Kullanım	Proje Bazlı	Paket Yazılım
Ürün kalitesi ile ilgili sistemli veri topladığımız söylenemez	25%	37%	0%
Ürün kalitesi ile ilgili temel veriler genellikle kullanıcı ile beraber çalışan	25%	37%	0%
Ürün kalitesi ile ilgili temel veriler genellikle ürün geliştirme ve sına ..	50%	26%	100%
Ürün kalitesi ile ilgili birimlerin ortaklaşa yürüttüğü çalışmalar ve özel or..	0%	0%	0%
Hiçbiri	0%	0%	0%

Aşağıdaki ürün kalite standartları ve ürün kalitesi ölçüm modelleri içinde ISO/IEC 12119 standardı oranı çok yüksek olmasa da hem iç kullanım için yazılım üretenler (yüzde 13) hem de proje bazlı yazılım üreten firmalar (yüzde 11) tarafından referans alınmaktadır.

Yazılım Kalitesi: 2008 yılında yazılım kalitesi konusunda proje bazlı yazılım geliştiren firmalarda, sınırlı da olsa iyileştirme görülmekle birlikte diğer kuruluşlarda değişiklik bulunmamaktadır (Şekil 4.96).



Şekil 4.96 : Yazılım kalitesi süreç profili – 2008

5. SONUÇ VE ÖNERİLER

5.1 GENEL DEĞERLENDİRME

Bu tez çalışması ile, bir anket çalışması vasıtası ile Türkiye’de yazılım mühendisliği ve yazılım yönetimi uygulamalarının gelişimini ve mevcut olgunluk düzeyini değerlendirmek; bu konuda daha önce gerçekleştirilmiş araştırmalar ile kıyaslayarak, bu alanda öncelikleri belirlemek hedeflenmiştir.

2008 yılı araştırma sonuçları da bu doğrultuda değerlendirilerek, 2001 yılında KalDer – Yazılım Toplam Kalite Yönetimi Uzmanlık Grubu tarafından yapılan araştırma sonuçları ile kıyaslanarak üç farklı kesit bazında bulgular araştırılmıştır. “Görev Tanımı”, “Kuruluş Türü” ve “Yazılım Kadrosu Çapı” altında yapılan analizler aşağıdaki başlıklar ile açıklanmıştır. Bu analizler, her bölümün sonunda yer alan ve ait olduğu sürecin olgunluk düzeyini ifade edecek şekilde sayısallaştırılan onuncu sorular ile gerçekleştirilmiştir.

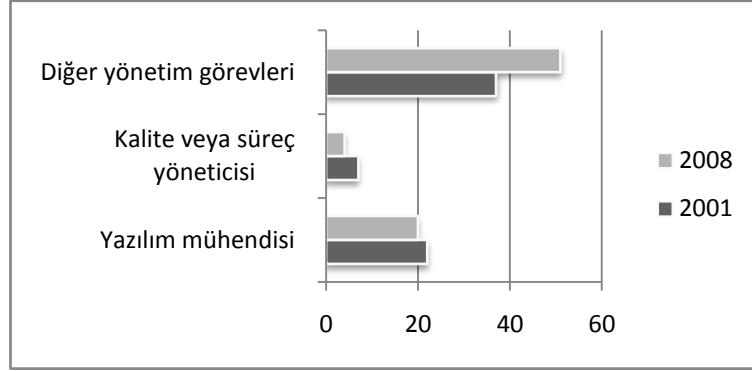
Araştırmanın genel değerlendirmesi yapıldığı zaman 2001 ve 2008 yıllarında genel olgunluk düzeyi açısından önemli bir değişiklik olmadığı ve sonuçların önemli ölçüde paralellik arzettiği görülmektedir.

Ancak her süreç altında yer alan ve o sürece ait pratik/uygulamaları sorgulayan ilk dokuz soru detaylı olarak incelendiği zaman, 2008 yılı sonuçlarının kendi içinde daha tutarlı olduğu görülmektedir. 2008 yılında, süreçler bazında kavramlar, rol ve sorumluluklar daha sağlıklı bir şekilde algılanmakta ve uygulanmaktadır. 2001 yılı verilerinde ise iyi uygulamalar daha az ve beyanlar uygulamadan daha iyi olduğunu ifade edecek şekilde yapılmış durumdadır.

Diğer bir önemli nokta, 2001 yılı araştırmasına katılan ve ortalama olgunluk düzeyini yukarı çıkaran savunma sanayi yazılım tedarikçi firmalarının 2008 yılı araştırmasına katılmamış olmasıdır.

Sonuçlara bu şekilde bakıldığı zaman, beklenen düzey de olmasa da 2008 yılında Türkiye’de yazılım süreçlerinin olgunluk düzeyinin arttığı söylenebilir.

5.1.1 Kesit-1: Görev Tanımı



Şekil 5.1 : Görev tanımı

2008 yılı araştırmasında (Şekil 5.1), yazılım mühendisleri ve kalite/süreç yöneticilerinin katılımı benzer düzeyde kalırken diğer yöneticilerin, özellikle Üst Yöneticilerin katılımında önemli ölçüde artış olmuştur.

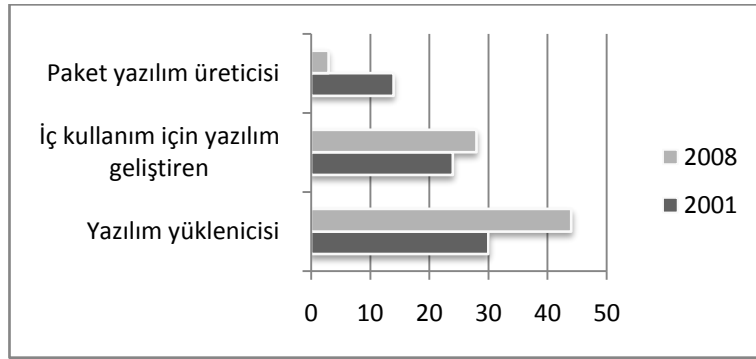
Tablo 5.1 : Göreve göre karşılaştırma

	Tümü		Yazılım Mühendisleri		Kalite Yöneticileri		Diğer Yöneticiler	
	2008	2001	2008	2001	2008	2001	2008	2001
A01. Yazılım Gereksinimleri	2,35	2,55	2,00	2,83	3,25	2,25	2,38	2,46
A02. Yazılım Tasarımı	2,22	2,32	2,20	2,85	2,67	1,75	2,19	2,15
A03. Yazılım Gerçekleştirme	2,49	2,65	2,30	3,00	4,00	2,25	2,45	2,46
A04. Yazılım Sınama	2,08	2,33	1,80	2,69	3,00	2,25	2,04	2,14
A05. Yazılım Bakımı	2,51	2,58	3,00	2,93	3,00	1,80	2,42	2,53
A06. Yazılım Konfigürasyon Yönetimi	1,96	1,74	1,50	1,73	4,00	3,00	1,78	1,57
A07. Yazılım Mühendisliği Yönetimi	2,28	2,18	2,00	2,22	3,00	2,75	2,27	2,05
A08. Yazılım Süreç Mühendisliği	1,96	2,03	3,00	1,50	2,67	2,40	1,63	2,20
A09. Yazılım Mühendisliği Araç ve Yöntem.	2,00	1,71	3,50	1,60	3,00	1,50	1,75	1,80
A10. Yazılım Kalitesi	1,77	1,71	1,00	1,71	2,75	1,67	1,68	1,67
Ortalama	2,16	2,18	2,23	2,31	3,13	2,16	2,06	2,10
Fark	-0,02		-0,08		0,97		-0,04	
N=	75		20		4		51	

Tablo 5.1’de göreve göre segmente edilen süreçler bazında olgunluk düzeylerini sayısal olarak ifade edebilmek için, her sürecin sonundaki düzeyi belirleyen 10. sorularda seçeneklerin (düzeylerin) karşısındaki yüzde oranları ile o seçeneğe ait düzey katsayısı çarpılarak ortaya çıkan sonuç rakamı detayı Tablo 3.2’de açıklandığı şekilde hesaplanarak karşılaştırma için kullanılmıştır.

Beyan edilen olgunluk düzeyleri, katılımcıların görev tanımlarına göre ayrıştırıldığı zaman (Tablo 5.1) sadece Kalite/Süreç Yöneticilerinin beyanlarında iyileşme olduğu görülmektedir. Bu durum, bu rolün daha çok büyük ve kaynakları sınırlı olmayan, kaynak ayıran şirketlerde yer alması, bu tür şirketlerin belli bir metod dahilinde süreçlerini iyileştiriyor olması ile açıklanabilir.

5.1.2 Kesit-2: Kuruluş Türü



Şekil 5.2 : Kuruluş türü

2008 yılı araştırmasında (Şekil 5.2), proje bazlı yazılım geliştiren yazılım yüklenicilerinin katılımı artarken, paket yazılım üreticilerinden sadece üç firma araştırmaya katılmıştır. Kuruluş türüne göre yapılan analizlerde üç katılımcının sektörü temsil etmesi mümkün olmayacağı için paket yazılım üreticileri ayrı bir kategori olarak yorumlanmamıştır.

Tablo 5.2’de kuruluş türüne göre segmente edilen süreçler bazında olgunluk düzeylerini sayısal olarak ifade edebilmek için, her sürecin sonundaki düzeyi belirleyen 10. sorularda seçeneklerin (düzeylerin) karşısındaki yüzde oranları ile o seçeneğe ait düzey

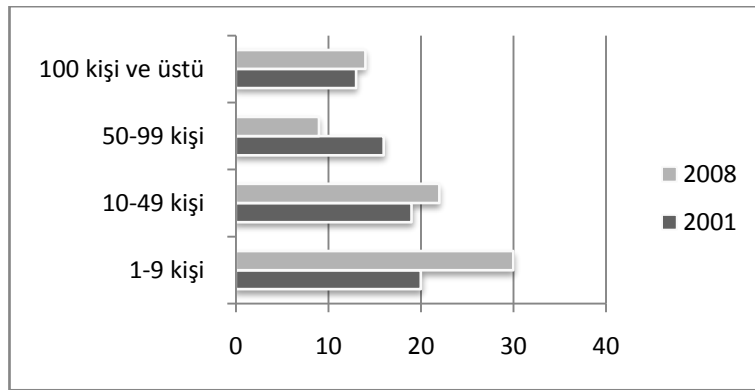
katsayısı çarpılarak ortaya çıkan sonuç rakamı detayı Tablo 3.2’de açıklandığı şekilde hesaplanarak karşılaştırma için kullanılmıştır.

Tablo 5.2 : Kuruluş türüne göre karşılaştırma

	Tümü		İç Kullanım		Proje		Paket	
	2008	2001	2008	2001	2008	2001	2008	2001
A01. Yazılım Gereksinimleri	2,35	2,55	2,09	2,07	2,56	2,81	1,67	2,83
A02. Yazılım Tasarımı	2,22	2,32	2,00	1,60	2,32	2,86	2,50	2,25
A03. Yazılım Gerçekleştirme	2,49	2,65	2,23	2,14	2,64	2,87	2,00	2,89
A04. Yazılım Sınama	2,08	2,33	1,82	2,00	2,21	2,53	2,00	2,43
A05. Yazılım Bakımı	2,51	2,58	2,71	1,54	2,40	3,00	3,00	3,43
A06. Yazılım Konfigürasyon Yönetimi	1,96	1,74	1,25	1,67	2,18	1,89	2,50	1,25
A07. Yazılım Mühendisliği Yönetimi	2,28	2,18	2,33	1,58	2,33	2,53	1,50	2,50
A08. Yazılım Süreç Mühendisliği	1,96	2,03	1,78	1,83	2,00	2,13	2,00	2,13
A09. Yazılım Mühendisliği Araç ve Yöntem.	2,00	1,71	2,00	1,50	2,00	1,57	2,00	2,25
A10. Yazılım Kalitesi	1,77	1,71	1,63	1,75	1,84	1,55	2,00	2,00
Ortalama	2,16	2,18	1,98	1,77	2,25	2,37	2,12	2,40
Fark	-0,02		0,22		-0,13		-0,28	
N=	75		28		44		3	

Sayısal sonuçlar itibarı ile (Tablo 5.2), iç kullanım için yazılım üreten firmaların beyanlarında sınırlı da olsa iyileşme görülmektedir. Bununla birlikte yapılan analiz sonuçları paket yazılım üreticisi firmaların iyi uygulamalar/pratikler açısından daha iyi bir düzeyde olduğunu göstermektedir.

5.1.3 Kesit-3: Yazılım Kadrosu Çapı



Şekil 5.3 : Yazılım kadrosu çapı

Katılımcılar yazılım kadrosu çapına göre segmente edildiğinde (Şekil 5.3), 2001 yılında daha homojen bir dağılım görülürken 2008 yılında katılımcıların ağırlık noktasını küçük yazılım organizasyonları oluşturmuştur.

Tablo 5.3’de yazılım çalışanı sayısına göre segmente edilen süreçler bazında olgunluk düzeylerini sayısal olarak ifade edebilmek için, her sürecin sonundaki düzeyi belirleyen 10. sorularda seçeneklerin (düzeylerin) karşısındaki yüzde oranları ile o seçeneğe ait düzey katsayısı çarpılarak ortaya çıkan sonuç rakamı detayı Tablo 3.2’de açıklandığı şekilde hesaplanarak karşılaştırma için kullanılmıştır.

Tablo 5.3 : Yazılım çalışanı sayısına göre karşılaştırma

	1-9		10-49		50-99		100+	
	2008	2001	2008	2001	2008	2001	2008	2001
A01. Yazılım Gereksinimleri	1,78	2,13	2,24	3,25	2,50	2,63	3,62	2,55
A02. Yazılım Tasarımı	2,18	1,73	1,79	3,30	2,67	2,60	3,00	1,89
A03. Yazılım Gerçekleştirme	2,60	2,00	2,00	3,00	2,20	3,00	3,38	2,78
A04. Yazılım Sınama	1,79	1,33	1,92	3,44	2,25	2,67	3,00	2,22
A05. Yazılım Bakımı	2,38	1,80	2,27	3,00	2,80	3,30	3,14	2,30
A06. Yazılım Konfigürasyon Yön.	1,25	0,91	1,90	2,25	1,75	1,71	3,20	2,33
A07. Yazılım Müh. Yönetimi	1,80	1,11	2,25	2,67	1,50	2,78	3,22	2,33
A08. Yazılım Süreç Mühendisliği	1,63	1,13	1,50	2,00	2,00	2,73	3,00	2,00
A09. Yazılım Müh. Araç ve Yönt.	1,78	1,11	2,00	1,33	1,50	2,00	2,75	2,22
A10. Yazılım Kalitesi	1,22	1,17	2,00	1,64	1,00	1,92	2,60	2,20
Ortalama	1,84	1,44	1,99	2,59	2,02	2,53	3,09	2,28
Fark	0,40		-0,60		-0,52		0,81	
N=	30		22		9		14	

Sonuçlar yazılım organizasyonunun çapına göre segmente edildiği zaman (Tablo 3.2), organizasyon çapı ile süreç olgunluk düzeyleri arasında paralellik olduğu görülmektedir. Paralellik açısından 2008 yılı verileri tutarlılık arz etmektedir.

Bu konuda beklenenin tersine, 1-9 çalışanı olan yazılım firmalarında daha büyük firmalara göre daha fazla iyileşme görülmektedir.

Toplam sonuçlara bakılarak 100'den fazla çalışanı olan yazılım organizasyonlarında hem iyileşme oranı hem de süreç olgunluk düzeyi, diğer segmentlerin üzerinde yer almaktadır.

5.2 GİRDİ ve ÇIKTI BAZINDA DEĞERLENDİRMELER

Sonuçların değerlendirilmesi, girdiler bazında “İnsan, Teknoloji ve Süreç” parametreleri ile tanımlanan “Yetkinlik Üçgeni”, çıktılar bazında “Kalite, Maliyet ve Hız” parametreleri ile tanımlanan “Rekabetçilik Üçgeni” ile yapılmıştır.

5.2.1 GİRDİLER

Süreçler açısından bir değerlendirme yapıldığı zaman “Yazılım Yönetimi” konuları katılımcılar tarafından daha az bilinmektedir. Yazılım yönetimi açısından kurumsallaşma ve süreç olgunluk düzeyleri, yazılım mühendisliği süreçlerine göre daha düşük orandadır. (Yazılım kalitesi, yazılım süreç mühendisliği, yazılım konfigürasyon yönetimi ve yazılım mühendisliği araç ve yöntemleri en az kurumsallaşan süreçler olarak göze çarpmaktadır). Verilere göre yazılım mühendisliği, yazılım yönetiminden daha iyi yapılmaktadır. Bunun sebepleri arasında, gerçek ile algılanan birbirinden farklı olması, organizasyon sorunu, yönetim ve zihniyet sorunu düşünülmektedir.

Katılımcıların “Yazılım Mühendisliği” başlığı altındaki süreçler için süreç yeterliliği beyanı “yönetilen” (SPICE/CMMI Seviye 2) düzeyi işaret etmekle birlikte beyanın aksine genel süreç düzeyi “yapılan” (SPICE/CMMI Seviye 1) düzeyinin üstüne çıkmamaktadır.

Katılımcıların “Yazılım Yönetimi” başlığı altındaki süreçler için de süreç yeterliliği beyanı “yönetilen” (SPICE/CMMI-2) düzeyi işaret etmekle birlikte beyanın aksine genel süreç düzeyi “varolan” (SPICE/CMMI-1) düzeyinin üstüne çıkmamaktadır.

İnsan kaynağı açısından yapılan değerlendirmede, proje ekiplerinde daha çok proje gerekleri ve alınan işe göre tasarlanan geçici yapılar görülmektedir. Mesleki eğitim, fiziksel altyapı ve uygun nitelik ve nicelikte eleman konularında sorunlar bulunmakta, performans ve kariyer yönetimi iyi yapılmamaktadır. Geçen yedi yıl süresince

kurumsallaşma yönünde iyi uygulamalar artmakla birlikte çalışanlara yönelik koşullar (fiziki ortam, eğitim, performans değerlendirme..) önemli ölçüde zorlaşmıştır.

Yazılım mühendisleri ve yöneticilerin beyanları 2008 yılında da 2001 yılında olduğu gibi birbirinden farklıdır. 2008 yılında, yazılım süreç mühendisliği, yazılım mühendisliği araç ve yöntemleri ve konfigürasyon yönetimi konusunda fark çok belirgindir. Yazılım yönetimi konularında, yöneticiler kurumsallaşmanın daha az olduğunu düşünmektedir.

Özellikle küçük boyutlu organizasyonlarda, yazılım geliştirme ve bakım dışındaki süreçlerde zaman ve eleman ayırımında sorunlar beyan edilmektedir. Zaman kullanımının müşteri, Üst Yönetim gibi “dış etmenler” tarafından baskı altında tutulduğu gözlemlenmektedir.

Eleman kaynağının kısıtlı olduğu durumlarda, rol paylaşımına gidilmekle birlikte rol paylaşımının planlı ve stratejik bir uygulama olup olmadığı ancak organizasyon detayında incelenerek bulunabilir.

Altyapı açısından yapılan değerlendirmede, yazılım mühendisliği araçlarının kaliteli yazılım üretmek için zorunlu olduğu düşünülmektedir ancak araç kullanımı çok az ve araçlar oldukça pahalıdır. Araç seçimi ve temini konusunda güçlüklerle işaret edilmektedir.

Mühendislik yöntemleri seçiminde, teknolojik yenilikleri izleme ve şirket politikası etkilidir.

Yeterince ölçüm yapılmamakta, iyileştirme için girdi bulunmamaktadır.

Temel süreçlerde uzmanlaşma ve eğitimin önemi vurgulanmakla birlikte bu konuda girişimleri olduğunu belirten katılımcı sayısı dikkat çekecek düzeyde azdır.

İletişim, deęişim yönetimi, sorun izleme gibi destekleyen süreçlerin yürütülmesi için organizasyonların gereken temel altyapıya sahip oldukları beyan edilmektedir.

5.2.2 ÇIKTILAR

Yazılım mühendisliği süreçleri kalite/verimlilik açısından değerlendirildięi zaman, evrensel süreç standart ve modellerinin bizim yazılım sektörümüzde de referans alınması gereklilięi öne çıkmaktadır. Uygun altyapılarla ve araçlarla çalışma gereęi bulunmaktadır. Eğitim eksikliklerinin giderilmesi gereklidir.

Yazılım mühendisliği süreçleri, maliyet/kaynak kullanımı açısından değerlendirildięi zaman, organizasyonlardaki uzmanlık alanlarının çeşitlendirilmesi gereęi bulunmaktadır. Mühendisler için rol esaslı kariyer geliştirme modellerinin uygulanması yarar sağlayacaktır.

Yazılım yönetimi süreçleri kalite/verimlilik açısından değerlendirildięi zaman, kullanım kolaylığı, bakım kolaylığı, tekrar kullanılabilirlik gibi bazı önemli kalite niteliklerine odaklanmanın yetersiz olduęu görülmektedir.

Yazılım yönetimi süreçleri maliyet/kaynak kullanımı açısından değerlendirildięinde, sistematik maliyet kestirimi ve kontrol bulunmadığı, kaynak ve altyapı kullanım planlaması yetersiz olduęu, sistematik süre kestirimi ve kontrollerin bulunmadığı gözlemler arasındadır.

5.3 ÖNERİLER

Araştırma sonuçları ve yaşanan deneyimler; yazılım süreçlerinde standartlaşma ve evrensel metodların kullanımı gereğini, net olarak ortaya koymaktadır. En yüksek önceliği bilgi teknolojisi firmalarının konuya gereken önemi vermesi ve gerekli kaynağı sağlayarak projeleri tanımlı metod/standartlar ile dünya standartlarında yönetmesi gerekliliği oluşturmaktadır. Alım yapan firmaların/tedarik makamının bilgi ve bilinç düzeylerinin yükseltilmesi ve firma seçiminde, doğru tercihleri yapabilme konumuna gelmesi bu konuda itici bir güç olacaktır.

Yazılım üretiminde kalite bir yaşam tarzı olarak ele alınırsa, her safhada karşılaşılan olaylara refleks bir yaklaşımla yanıt verilirse elde edilecek sonuç, projenin/üretimin kalitesini en üste çıkaracaktır. Bu bir refleks haline getirildiği andan itibaren, yazılım üretim süreçlerinde kaliteyi sağlayacak aşamaya ulaşılmış demektir.

Dünyada ve Türkiye`de kalite standartlarının öneminin artması sonucu firmalardan büyük projelerde teklif verebilmeleri için CMMI, AQAP 160, vb. kalite belgeleri istenir olmuştur. Bu belgeler sadece bazı büyük ihalelere girmek için değil, aynı zamanda şirketin verimliliği ve müşteri memnuniyeti açısından da büyük önem taşımaktadır. Bugün birçok firma, yazılım geliştirme süreçlerini CMMI çerçevesinde değerlendirme ve derecelendirme ihtiyacı duymaktadır.

Savunma sanayinin gerekleri/ihtiyacı üzerine ortaya koyulan CMM modeli 20 yıla yaklaşan süre içinde SPICE modelinin getirdiği anlayışla da birleşerek, verimliliği hedefleyen tüm yazılım organizasyonları için önemli bir rehber oluşturmuş ve yaygınlık kazanmıştır. Coğrafi yaygınlık açısından yapılan değerlendirmeler de artık CMMI`ın arkasındaki itici gücün zorunluluktan çok verimlilik olduğunu açıklamaktadır.

Küreselleşen dünyaya tümleşik olma sürecindeki ülkemizin çözmesi gereken bir konu, uluslararası sistemlerin öğrenilerek uygulanmasıdır. Ancak bu şekilde, uluslararası pazarlarda rekabet şansının artması olasıdır.

EKLER

EK 1 - Sorular

A01. Yazılım Gereksinimleri

Yazılım gereksinimleri, bir yazılım ürününün amaçlarına erişmesi için istenen özellikleridir.

1. **Gereksinim belirleme** tekniklerinden *hangilerini* kullanıyorsunuz?
 - a. Müşteri iletişim kanalları (yüzyüze görüşmeler, e-posta, çağrı merkezi, vb.)
 - b. Kullanım senaryoları tanımlama
 - c. Prototip geliştirme
 - d. Beyin fırtınası toplantıları
 - e. Kullanım ortamında gözlem yapma
 - f. Benzer ürünleri inceleme
 - g. Başka teknikler
2. **Gereksinim tanımlama** aşamasında aşağıdaki tekniklerden *hangilerini* kullanıyorsunuz?
 - a. Numaralandırma
 - b. Müşterinin verdiği öneme göre önceliklendirme
 - c. Teknik zorluk veya karmaşıklık derecesine göre sınıflandırma
 - d. Gerçekleştirme maliyetine göre sınıflandırma
 - e. Değişkenlik derecesine göre sınıflandırma
 - f. Türe göre sınıflandırma
 - g. İlgili olduğu iş sürecini / süreçlerini kaydetme
 - h. Gereksinimin kaynağı olan kişiyi kaydetme
 - i. Gereksinimin tanımlandığı tarihi kaydetme
 - j. Gereksinimler arasındaki bağımlılıkları kaydetme
 - k. Gereksinimin revizyonlarını saklama
 - l. Gereksinimleri yazılım ürününün planlanan sürümlerine atama
 - m. Başka teknikler
3. Aşağıdakilerden gereksinim türlerinden *hangilerini gereksinim sınıflandırmasında* kullanıyorsunuz?
 - a. İşlevsel özellikler
 - b. Veri özellikleri
 - c. İşlem hızı / tepki süresi / işlem kapasitesi
 - d. Hesaplama doğruluğu (computational accuracy)
 - e. Hataların ele alınışı (error handling)
 - f. Kullanım kolaylığı (usability)
 - g. Öğrenme kolaylığı (ease of use)
 - h. Kullanılabilirlik oranı (availability)
 - i. Kesintisiz kullanım / güvenilirlik (reliability)
 - j. Erişim güvenliği (access security)
 - k. Veri güvenliği (data security)
 - l. Sağlık / yaşam güvenliği (safety)
 - m. Kullanıcı arayüzleri (look and feel)
 - n. Diğer yazılımlar ile arayüzler
 - o. İşletim sistemi veya diğer sistemler ile arayüzler
 - p. Bakım kolaylığı (maintainability)
 - q. Platformlar arası taşınabilirlik (portability)
 - r. Kullanıcı için belgeleme
 - s. Teknik belgeleme
 - t. Veri yedekleme ve kurtarma özellikleri
 - u. Geliştirme dili, aracı, tekniği veya metodolojisi ile ilgili kısıtlar
 - v. Teknik standartlara veya spesifikasyonlara uygunluk
 - w. Kültürel standartlara (dil, para birimi vb.) uygunluk
 - x. İş süreci modellerine ve uygulama çerçevelerine (application frameworks) uygunluk
 - y. Yasalara ve mevzuata uygunluk
 - z. Başka gereksinim türleri
4. **Gereksinim risklerinden** *hangilerini* izliyor ve risk azaltma yöntemleri uyguluyorsunuz?
 - a. Gereksinimlerin değişkenliği
 - b. Gereksinimlerin eksik belirlenmiş olması
 - c. Gereksinimlerin yeterince açık tanımlanmamış olması
 - d. Gereksinimler üzerine görüş birliği olmaması

- e. Gereksinimlerin gerçekleştirilme zorluğu
 - f. Gereksinimlerin yazılım geliştirenler için yeni olması
 - g. Gereksinim sayı ve kapsamının çok fazla olması
 - h. Başka riskler
5. **Gereksinim analizi** aşamasında aşağıdaki tekniklerden *hangilerini* kullanıyorsunuz?
- a. Kullanım vakaları (use cases)
 - b. Veri akış şemaları (data flow diagrams)
 - c. Varlık-bağıntı şemaları (entity-relationship diagrams)
 - d. Nesne şemaları (object diagrams)
 - e. Faaliyet şemaları (activity diagrams)
 - f. Etkileşim şemaları (interaction diagrams)
 - g. Durum haritası şemaları (statechart diagrams)
 - h. Durum geçiş şemaları (state transition diagrams)
 - i. Formal belirtim dilleri (formal specification languages)
 - j. Veri sözlüğü
 - k. Başka teknikler
6. Kuruluşunuzda **yazılım gereksinimlerinin tanımlanması** için aşağıdakilerden *hangisi* geçerli?
- a. Gereksinimleri tanımladığımız söylenemez
 - b. Kişisel notlar şeklinde tanımlıyoruz
 - c. Kişisel düzeyde ancak paylaşılabilen belgeler yazıyoruz
 - d. Sadece elektronik ortamda geliştirdiğimiz analiz modelleri üzerinde tanımlıyoruz
 - e. Kurumsal bir Yazılım Gereksinimleri Belirtimi (SRS) şablonu kullanıyoruz
 - f. Kurumsal bir Yazılım Gereksinimleri Belirtimi (SRS) şablonu kullanıyoruz ve belgenin sürekli güncel kalmasını sağlıyoruz
 - g. Hiçbiri
7. Kuruluşunuzda **gereksinimlerin onaylanması** için aşağıdakilerden *hangileri* geçerli?
- a. İş süreci uzmanları ile gereksinim gözden geçirmeleri yapıyoruz
 - b. Müşteri ile formal gereksinim gözden geçirmeleri yapıyoruz
 - c. Müşteri ile prototipler üzerinde çalışıyoruz
 - d. Formal model doğrulama teknikleri uyguluyoruz
 - e. Kabul sınamaları uyguluyoruz
 - f. Başka gereksinim onaylama yöntemleri uyguluyoruz
8. Kuruluşunuzda **gereksinim değişim yönetimi** için aşağıdakilerden *hangisi* geçerli?
- a. Değişim yönetimi uyguladığımız söylenemez
 - b. Proje üyelerinin kişisel merakı ölçüsünde izleme ve düzeltme yapıyor
 - c. Gereksinim değişiklikleri atanmış elemanlar tarafından ancak belgesiz yürütülüyor
 - d. Kurumsal, ancak yeterince etkin olmayan bir değişim yönetimi süreci izliyoruz
 - e. Kurumsal bir değişim yönetimi süreci izliyoruz, tüm değişiklik kayıtlarını tutuyoruz, tüm ilgili tarafları zamanında bilgilendiriyoruz
 - f. Hiçbiri
9. **Gereksinimlerin yönetimi** üzerine aşağıdaki görüşlerden *hangilerine* katılıyorsunuz?
- a. Yazılım geliştiren kuruluşlar için en kritik başarı faktörlerindedir
 - b. Yazılım araçlarıyla desteklenmesi zorunludur
 - c. Bu alanda standart süreçler izlemek zaman ve kaynak kaybına yol açar
 - d. Bu uygulamaların kurumsallaşması uzun sürer
 - e. Yazılımcıların iş kültürü bu alanda sistematik uygulamalara elverişli değil
 - f. Müşteriler bu alanda yazılım geliştirenleri yeterince zorlamıyor
10. Kuruluşunuzda **gereksinimlerin yönetimi** ile ilgili genel bir değerlendirme yapmak gerekirse, aşağıdakilerden *hangisi* mevcut durumu en iyi yansıtır?
- a. Başarıyla uyguladığımız söylenemez
 - b. Kişisel beceri ile yürütüyoruz
 - c. Kurumsal standart olmasa da planlı yürütülebilen bir sürecimiz var
 - d. Kurumsal düzeyde tanımlanmış ve uygulanan bir sürecimiz var
 - e. Tanımlı ve niceliksel denetim altında yürüyen bir sürecimiz var
 - f. Tanımlı, niceliksel denetim altına ve sürekli iyileştirilen bir sürecimiz var

A02. Yazılım Tasarımı

Yazılım tasarımı, yazılım gereksinimlerini karşılayan bir ürünün gerçekleştirilmesine temel oluşturan yapıyı ve arayüzleri gerekli ve yeterli ayrıntı düzeyinde tanımlayan mühendislik çözümdür.

1. Projelerinizde **yazılım tasarımı** genellikle *kim* tarafından yapılır?
 - a. Yazılım tasarım uzmanı
 - b. Programcı
 - c. Gereksinim analisti
 - d. Proje Yöneticisi
 - e. Projelerimize yazılım tasarımı yapıldığı söylenemez
2. **Yazılım tasarım strateji ve yöntemlerinden** *hangilerini* kullanıyorsunuz?
 - a. Böl ve yönet (divide-and-conquer)
 - b. Tümevarım (bottom-up)
 - c. Tümdengelim (top-down)
 - d. Aşamalı ayrıntılandırma (stepwise refinement)
 - e. Veri soyutlama (data abstraction)
 - f. Bilgi saklama (information hiding)
 - g. Buluşsal yöntemler (heuristic methods)
 - h. Örüntüler ve örüntü dilleri (patterns and pattern languages)
 - i. Deneme yanılma yaklaşımı (iterative approach)
 - j. Artımlı yaklaşım (incremental approach)
 - k. İşleve yönelik tasarım (function-oriented design)
 - l. Nesne yönelimli tasarım (object-oriented design)
 - m. Veri yapısı odaklı tasarım (data structure centered design)
 - n. Unified Modeling Language (UML)
 - o. Başka yöntemler
3. Aşağıdaki **yapısal tasarım (statik gösterim) tekniklerinden** *hangilerini* kullanıyorsunuz?
 - a. Mimari tanımlama dilleri (architecture description languages)
 - b. Sınıf ve nesne şemaları (class and object diagrams)
 - c. Bileşen şemaları (component diagrams)
 - d. İntikal şemaları (deployment diagrams)
 - e. Varlık-bağıntı şemaları (entity-relationship diagrams)
 - f. Arayüz tanımlama dilleri (interface description languages)
 - g. Jackson yapı şemaları (Jackson structure charts)
 - h. Yapı şemaları
 - i. Başka teknikler
4. Aşağıdaki **davranışsal tasarım (dinamik gösterim) tekniklerinden** *hangilerini* kullanıyorsunuz?
 - a. Faaliyet şemaları (activity diagrams)
 - b. İşbirliği şemaları (collaboration diagrams)
 - c. Veri akış şemaları (data flow diagrams)
 - d. Karar tabloları ve şemaları (decision tables and diagrams)
 - e. Akış şemaları (flowcharts)
 - f. Formal belirtim dilleri (formal specification languages)
 - g. Sözde kod (pseudo code)
 - h. Program tasarım dilleri (program design languages)
 - i. İş sıralama şemaları (sequence diagrams)
 - j. Etkileşim şemaları (interaction diagrams)
 - k. Durum geçiş şemaları (state transition diagrams)
 - l. Durum haritası şemaları (statechart diagrams)
 - m. Başka teknikler
5. **Tasarım risklerinden** *hangilerini* izliyor ve risk azaltma yöntemleri uyguluyorsunuz?
 - a. Seçilen tasarım tekniğinin zorluğu
 - b. İşlevsel gereksinimlere uygun bir tasarım çözümü bulunamaması
 - c. İş ve dış arayüzlerin gereksinimler aşamasında iyi tanımlanmamış olması
 - d. Performans gereksinimlerine uygun bir tasarım çözümü bulunamaması
 - e. Tasarımın ürünün sınımanmasına yardımcı öğeler içermemesi
 - f. Donanım kısıtlarının tasarımı zorlaştırması
 - g. Eski veya hazır yazılımlardan kaynaklanan sorunların tasarımı zorlaştırması
 - h. Başka riskler
6. **Yazılım altyapısında** aşağıdaki teknolojilerden *hangilerini* kullanıyorsunuz?
 - a. Java, J2EE, Enterprise JavaBeans
 - b. Microsoft Windows, VB, DCOM, .NET
 - c. OMG standartları, CORBA, IIOP, XMI
 - d. İnternet standartları: HTML, CGI, XML, SOAP, ebXML
 - e. İlişkisel veritabanı sistemleri

- f. Başka teknolojiler
7. Yazılım tasarımını geliştirirken öncelikle odaklandığınız **tasarım nitelikleri** hangileri?
- Gereksinimler ile izlenebilirliği olması
 - Kod ile izlenebilirliği olması
 - Programlama dilinden mümkün olduğunca bağımsız olması
 - İşlevselliği yüksek bir ürünü temsil etmesi
 - Güvenirliği yüksek bir ürünü temsil etmesi
 - Ürünün kullanımı sırasında oluşabilecek hataların ilgili iş sürecini aksatmayacak şekilde yönetilmesini (error handling, recovery) sağlaması
 - Performansı yüksek bir ürünü temsil etmesi
 - Güvenlik özellikleri yüksek bir ürünü temsil etmesi
 - Kullanması ve öğrenmesi kolay bir ürünü temsil etmesi
 - Tekrar kullanılabilir olması; bir ürün ailesine temel oluşturabilmesi
 - Kolay anlaşılması
 - Kolay değiştirilebilmesi
 - Kurumsal tasarım standartlarına uygun olması
 - Diğer tasarımlarla kolay birleştirilebilmesi
 - Başka nitelikler
8. Aşağıdaki **tasarım kalite güvence** yöntemlerinden *hangilerini* uyguluyorsunuz?
- Tasarımın deneyimli bir tasarımcı tarafından gözden geçirilmesi
 - Tasarımın takım üyeleri tarafından birlikte gözden geçirilmesi
 - Müşteri tarafından yapılan tasarım gözden geçirmesi
 - Müşteri ile birlikte yapılan formal tasarım denetimi
 - Başka yöntemler
9. Kuruluşunuzda **formal tasarım tekniklerinin benimsenmesine** aşağıdaki faktörlerden *hangileri* engel oluyor?
- Proje sürelerinin yeterli olmaması
 - Tasarım araçlarına yatırım yapılamaması
 - Gerekli eğitimlerin alınamaması
 - Üst Yönetimden yeterli desteğin alınamaması
 - Yazılımcıların isteksizliği
 - Başka faktörler
10. Kuruluşunuzda **yazılım tasarımı** ile ilgili genel bir değerlendirme yapmak gerekirse, aşağıdakilerden *hangisi* mevcut durumu en iyi yansıtıyor?
- Başarıyla uyguladığımız söylenemez
 - Kişisel beceri ile yürütüyoruz
 - Kurumsal standart olmasa da planlı yürütülebilen bir sürecimiz var
 - Kurumsal düzeyde tanımlanmış ve uygulanan bir sürecimiz var
 - Tanımlı ve niceliksel denetim altında yürüten bir sürecimiz var
 - Tanımlı, niceliksel denetim altına ve sürekli iyileştirilen bir sürecimiz var

A03. Yazılım Gerçekleştirme

Yazılım gerçekleştirme, tasarımın öngördüğü yazılım birimlerinin üretilmesini ve sınanmasını, aşamalı olarak tümleştirilmesini ve müşteriye teslim edilecek ürün belgelerinin yazılmasını kapsar.

1. Aşağıdaki **programlama yöntemlerinden** *hangilerini* kullanıyorsunuz?
- Yapısal programlama
 - Nesne yönelimli programlama
 - Otomatik kod üretimi
 - Yeniden kullanılabilir bileşenler
 - Başka yöntemler
2. Yazılımlarınızda *ne tür ortak standart modüller* kullanılıyor?
- Veri tanımları
 - Hata karşılama ve bildirim
 - Veritabanı erişimi
 - İşlem kayıtlarını tutma (operation logging)
 - Güvenlik kontrolü ve yetkilendirme
 - İşlem modülünü çağırma
 - Mesaj alımı ve gönderimi
 - Şekil kullanıcı arayüzü

- i. Başka modüller
3. Aşağıdaki **kodlama standartlarından hangilerini** kurumlaştırdınız?
 - a. Değişkenleri ve yazılım birimlerini isimlendirme
 - b. Yazılım birimine ilişkin bilgi (amacı, işlevi, son değiştiren, vb.)
 - c. Kod formatı
 - d. Satırsarı açıklama formatı
 - e. Satırsarı açıklama içeriği
4. Programcıları **hangi kodlama kriterlerine** uymaya zorluyorsunuz?
 - a. Yeniden kullanılabilirlik
 - b. Basitlik ve kısalık
 - c. Yapısalılık (GO TO kullanmama, tek giriş ve çıkış noktası olması vb.)
 - d. Başka kriterler
5. Programcılar **hangi birim sınamaları** mutlaka yapıyor?
 - a. Tüm komutlar
 - b. Tüm dallanmalar
 - c. Tüm koşullar
 - d. Tüm erişim yolları
 - e. Tüm sınır değerler
 - f. Tüm özel değerler
6. **Tümleştirme stratejisini veya planını** genellikle *kim* belirliyor?
 - a. Programcı
 - b. Gereksinim analisti
 - c. Tasarım mühendisi
 - d. Mimar
 - e. Takım Lideri
 - f. Proje Yöneticisi
 - g. Bir tümleştirme stratejisi veya planı belirlenmiyor
7. **Tümleştirme stratejisi veya planı** genellikle geliştirmenin *hangi* aşamasında belirleniyor?
 - a. Proje planlama
 - b. Gereksinim analizi
 - c. Tasarım
 - d. Kodlama
 - e. Tümleştirme
 - f. Bir tümleştirme stratejisi veya planı kullanılmıyor
8. Aşağıdaki **kod kalite güvence** yöntemlerinden *hangilerini* uyguluyorsunuz?
 - a. Program kodunun deneyimli bir programcı tarafından gözden geçirilmesi
 - b. Program kodunun takım üyeleri tarafından birlikte gözden geçirilmesi
 - c. Müşteri tarafından yapılan kod gözden geçirmesi
 - d. Müşteri ile birlikte yapılan formal kod denetimi
 - e. Başka yöntemler
9. Kuruluşunuzda **kullanıcı belgeleri** (elkitapları vb.) için aşağıdakilerden *hangisi* geçerlidir?
 - a. Genellikle kimse hazırlamak istemez, çoğu kez eksik kalır
 - b. Bu işe merak duyan bir proje üyesi tarafından hazırlanır
 - c. Bu göreve atanan deneyimli bir proje üyesi tarafından hazırlanır
 - d. Bu konuda uzmanlaşmış bir ekip tarafından tüm yazılımlar için hazırlanır
10. Kuruluşunuzda **yazılım gerçekleştirme** ile ilgili genel bir değerlendirme yapmak gerekirse, aşağıdakilerden *hangisi* mevcut durumu en iyi yansıtıyor?
 - a. Başarıyla uyguladığımız söylenemez
 - b. Kişisel beceri ile yürütüyoruz
 - c. Kurumsal standart olmasa da planlı yürütülebilen bir sürecimiz var
 - d. Kurumsal düzeyde tanımlanmış ve uygulanan bir sürecimiz var
 - e. Tanımlı ve niceliksel denetim altında yürüten bir sürecimiz var
 - f. Tanımlı, niceliksel denetim altına ve sürekli iyileştirilen bir sürecimiz var

A04. Yazılım Sınama

Yazılım sınama, gerçekleştirilen ürünün yazılım gereksinimlerini karşıladığının doğrulanmasıdır.

1. **Yazılım sınama süreciniz** aşağıdaki konulardan *hangilerine* öncelikle odaklanmaktadır?
 - a. Yazılımın müşteri tarafından belirlenen işlevsel gereksinimleri karşılaması
 - b. Kullanımı engelleyecek kritik hatalarının öngörülmesi ve önlenmesi
 - c. Kullanım sırasında oluşabilecek istisnai hataların ürün kullanımını aksatmayacak şekilde yönetilmesi
 - d. Yazılım tasarımının doğruluğu
 - e. Yazılımın bakımının kolay yapılabilmesi

- f. Yazılımın kullanıcı tarafından kolay kullanılabilir olması
 - g. Yazılım birimlerinin birbirleriyle arayüzlerinin sağlam kurulması
 - h. Yazılım birimlerinin performansları
 - i. Tümüleştirilmiş yazılımın performansı
 - j. Yazılımın sistem ile arayüzlerinin sağlam kurulması
 - k. Yazılımın yer aldığı sistemin performansı
2. Kuruluşunuzda çalışan **Sinama Mühendisleri** hangi disiplinlerden geliyor?
 - a. Bilgisayar mühendisliği
 - b. Elektrik / elektronik mühendisliği
 - c. Endüstri mühendisliği / yöneylem araştırması
 - d. Bilgisayar programcılığı
 - e. Matematik
 - f. İstatistik
 - g. Başka disiplinler
 3. Kuruluşunuzda **yazılım sinama planlaması** aşağıdakilerden *hangilerini* içeriyor?
 - a. Sınanacak yazılımın adı ve sürümü ile belirlenmesi
 - b. Yapılacak sınamanın türü (tümüleştirme, regresyon, kabul, saha, vb.)
 - c. Yazılımın hangi özelliklerinin sınanacağını belirlenmesi
 - d. Yazılımın hangi özelliklerinin sınanmayacağını belirlenmesi
 - e. Kullanılacak sinama tekniklerinin belirlenmesi
 - f. Yazılım bileşenlerinin nasıl bir sıra izlenerek kümeleştirileceğinin belirlenmesi
 - g. Sinama senaryolarının belirlenmesi
 - h. Sinama senaryoları ve verilerinin konfigürasyon yönetimi planlaması
 - i. Kullanılacak sinama araç ve ortamının tanımlanması
 - j. Sinama göre alacak personelin ve rollerinin belirlenmesi
 - k. Müşteri sinamaya katılacaksa rolünün belirlenmesi
 - l. Sinama personelinin alması gereken eğitimlerin planlanması
 - m. Sinama zaman çizelgesinin tanımlanması
 - n. Sinama kayıt ve raporlarının tanımlanması
 - o. Risklerin tanımlanması
 - p. Sinama başarı kriterlerinin tanımlanması
 4. Kuruluşunuzda **yazılım sinama senaryoları** genellikle *nasıl* tanımlanıyor?
 - a. Sinama sırasında anlık oluşturuluyor
 - b. Gereksinim analisti tarafından tanımlanıyor
 - c. Tasarım mühendisi tarafından tanımlanıyor
 - d. Programcılar tarafından tanımlanıyor
 - e. Bu işle görevlendirilen bir sinama mühendisi tarafından tanımlanıyor
 - f. Takım Lideri tarafından tanımlanıyor
 - g. Proje Yöneticisi tarafından tanımlanıyor
 - h. Müşteri tarafından tanımlanıyor
 - i. Hiçbiri
 5. Kuruluşunuzda aşağıdaki **sinama tekniklerinden** *hangileri* kullanılıyor?
 - a. Bağımsız rotalar sinaması (basis path testing)
 - b. Karar noktaları sinaması (condition testing)
 - c. Veri akış sinaması (data flow testing)
 - d. Döngü sinaması (loop testing)
 - e. Çizgeye dayalı sinama yöntemleri (graph-based testing methods)
 - f. Eşdeğer sınıflara bölüntüleme (equivalence class partitioning)
 - g. Sınır değer analizi (boundary value analysis)
 - h. Karşılaştırma sinaması (comparison testing)
 - i. Başka teknikler
 6. Kuruluşunuzda **yükleme sinamalarından** (load tests) *hangileri* uygulanıyor?
 - a. Veri hacmi sinaması (volume test): programın yüksek miktarda veri ile yüklenmesi
 - b. Veri debisi sinaması (stress test): programın yüksek hızda veri ile yüklenmesi
 - c. Kapasite sinaması (storage test): programın bellek ve disk kullanımının zorlanması
 - d. Başka türden yükleme sinamaları
 7. Kuruluşunuzda **sinama sonuçlarının yönetimi** bağlamında aşağıdakilerden *hangileri* yapılır?
 - a. Sinama sonuçları belgelendirilir ve tüm ilgili taraflara duyurulur
 - b. Sinama sırasında bulunan hatalara bakılarak ürünün kalite profili değerlendirilir
 - c. Sinama sonuçlarına bakarak maliyet / kalite tartım analizi yapılır ve iyileştirme seçenekleri için karar verilir
 - d. Kararlaştırılan ürün iyileştirme işleri için görevlendirme yapılır ve gelişim sonuna kadar izlenir

- e. Sınama sırasında bulunan hatalar yaratıldıkları aşamaya (gereksinim analizi, tasarım, kodlama) göre sınıflandırılır
 - f. Sınama sırasında bulunan hatalar türlerine (hatalı atama, değer taşması, sonsuz döngü, Şekil arayüz, vb.) göre sınıflandırılır
 - g. Sınama sırasında bulunan hataların temel nedenleri sistematik olarak incelenir ve gelecek için dersler çıkarılır
 - h. Sınama sırasında bulunan hataların neden olduğu maliyet hesaplanır
 - i. Ürünün müşteriye teslimi sırasında bilinen ve giderilmesine gerek duyulmayan önemsiz hatalar belgelendirilir
8. Kuruluşunuzda **yazılım sınama faaliyetleri** için aşağıdakilerden *hangileri* doğrudur?
- a. Etkin bir yazılım sınama sürecimiz var
 - b. Ürünün alfa, beta ve son sürümleri için farklı sınama yaklaşımları kullanıyoruz
 - c. Deneyimli yazılım sınama mühendislerimiz var
 - d. Verimlilik sağlayan yazılım sınama araçları kullanıyoruz
 - e. Modern sınama tekniklerini gerekli eğitimleri alarak öğreniyoruz
 - f. Modern sınama tekniklerini mesleki yayımları izleyerek öğreniyoruz
9. Kuruluşunuzda **yazılım sınama belgelerinden** hangileri kullanılıyor?
- a. Sınama planı (test plan)
 - b. Sınama tasarım belirtimi (test design specification)
 - c. Sınama senaryosu belirtimi (test case specification)
 - d. Sınama prosedürü belirtimi (test procedure specification)
 - e. Sınama ögesi teslimat raporu (test item transmittal report)
 - f. Sınama kayıtları (test log)
 - g. Sınama olay raporu (test incident report)
 - h. Sınama özet raporu (test summary report)
 - i. Kabul sınaması tutanağı (acceptance test record)
10. Kuruluşunuzda **yazılım sınama** ile ilgili genel bir değerlendirme yapmak gerekirse, aşağıdakilerden *hangisi* mevcut durumu en iyi yansıtıyor?
- a. Başarıyla uyguladığımız söylenemez
 - b. Kişisel beceri ile yürütüyoruz
 - c. Kurumsal standart olmasa da planlı yürütülebilen bir sürecimiz var
 - d. Kurumsal düzeyde tanımlanmış ve uygulanan bir sürecimiz var
 - e. Tanımlı ve niceliksel denetim altında yürüten bir sürecimiz var
 - f. Tanımlı, niceliksel denetim altına ve sürekli iyileştirilen bir sürecimiz var

A05. Yazılım Bakımı

Yazılım bakımı, teslim edilen bir yazılım ürününde hata giderme, performans iyileştirme ve değişen ortama uyarlama amacıyla değişiklikler yapılmasıdır.

1. Aşağıdakilerden *hangileri* kuruluşunuzda **yazılım bakım işlerinin** en önemli kaynaklarındandır?
- a. Yazılım gereksinimlerinin ürün geliştirilirken iyi anlaşılması
 - b. Yazılımın desteklediği iş süreçlerinin zamanla değişmesi
 - c. Sistem konfigürasyonunun değişmesi ile yeni arayüz gereksinimlerinin ortaya çıkması
 - d. Yeni yazılım teknolojilerine ayak uydurma isteği
 - e. Başka nedenler
2. Aşağıdaki **yazılım bakım türlerinden** *hangisi* kuruluşunuzda en fazla kaynağı tüketiyor?
- a. Düzeltici bakım (hataları giderme, vb.)
 - b. Uyarlayıcı bakım (yeni arayüzler, vb.)
 - c. İyileştirici bakım (yeni özellikler, vb.)
 - d. Önleyici bakım (olası hataları önleme amaçlı)
3. Aşağıdaki **bakım aşamalarından** *hangileri* plan dahilinde ve belgelendirilerek yürütülüyor?
- a. Sorun / istek tanımlama ve sınıflandırma
 - b. Gereksinim analizi ve Yazılım Gereksinimler Belirtimi (SRS) güncelleme
 - c. Tasarım güncelleme
 - d. Kod güncelleme / geliştirme ve birim sınama
 - e. Regresyon sınaması (sistematik olarak seçilen belirli sınamaların tekrar edilmesi)
 - f. Kabul sınaması
 - g. Teslimat / taşıma / devreden alma
4. Kuruluşunuzda bir yazılım ürünü **geliştiren personel** bakımında da görev alıyor mu?
- a. Her zaman
 - b. Çoğunlukla

- c. Nadiren
 - d. Bakımı başka kuruluş / birim üstleniyor
5. Kuruluşunuzda yürütülen **yazılım bakım işlemleri** için aşağıdakilerden *hangileri* doğrudur?
- a. Bakım istekleri yeterince hızlı olarak yerine getirilebiliyor
 - b. Bakım isteklerinin durumu sistematik olarak izlenebiliyor
 - c. Verimliliğe yardımcı araçlar kullanılıyor
 - d. Uygun nitelikte teknik personel kullanılıyor
 - e. Bakım projeleri her zaman formal olarak tanımlanıyor ve yönetiliyor
 - f. Bakım sırasında da geliştirme sürecinin belgelendirme standartlarına uyuluyor
 - g. Bakım istekleri için isabetli süre ve maliyet kestirimi yapılabiliyor
 - h. Bakım istekleri her zaman formal Değişim Kontrol Süreci kapsamında ele alınıyor
6. Müşterinin yazılımla ilgili bildirdiği sorun ve yeni gereksinimleri **kim topluyor**?
- a. Teknik destek birimi
 - b. Konfigürasyon Yöneticisi
 - c. Ürün Yöneticisi
 - d. Takım Lideri
 - e. Proje Yöneticisi
 - f. Bölüm Yöneticisi
 - g. Herhangi bir proje üyesi
 - h. Hiçbiri
7. Yazılımla ilgili sorun ve yeni gereksinimlerin **değerlendirilmesi ve önceliklendirilmesinde kimler** rol alıyor?
- a. Müşteri
 - b. Teknik destek birimi
 - c. Konfigürasyon Yöneticisi
 - d. Değişiklik / Konfigürasyon Kontrol Kurulu
 - e. Ürün Yöneticisi
 - f. Takım Lideri
 - g. Proje Yöneticisi
 - h. Bölüm Yöneticisi
 - i. Herhangi bir proje üyesi
 - j. Başka görevliler
8. Yeni ürünlerin geliştirilmesi aşamasında, gelecekte **ürünün bakımını kolaylaştırmak için** aşağıdaki önlemlerden *hangilerini* uyguluyorsunuz?
- a. Bileşen bazlı geliştirme
 - b. Tasarımı bakım kolaylığı açısından gözden geçirme
 - c. Kodu bakım kolaylığı açısından gözden geçirme
 - d. Kurumsal kodlama standartları
 - e. Kurumsal teknik belgeleme standartları
 - f. Ürünü geliştiren personeli bakım aşamasında da görevlendirme
 - g. Birim ve tümleştirme sınamaları için standart araçlar kullanma
 - h. Başka önlemler
9. Bakım aşamasında aşağıdaki **ölçümlerden hangilerini** kullanıyorsunuz?
- a. Sorun bildirimlerinin dönemsel yoğunluğu
 - b. Sonuçlandırılmamış sorun bildirimlerinin sayısı
 - c. Sorun bildirimlerinin ortalama sonuçlandırılma süresi
 - d. Sorunları ortalama sonuçlandırma maliyeti
 - e. Sorun bildirimlerinin modül bazında dağılımı
 - f. Sorun bildirimlerinin müşteri bazında dağılımı
 - g. Sorun bildirimlerinin hata türü bazında dağılımı
 - h. Başka ölçümler
10. Kuruluşunuzda **yazılım bakımı** ile ilgili genel bir değerlendirme yapmak gerekirse, aşağıdakilerden *hangisi* mevcut durumu en iyi yansıtıyor?
- a. Başarıyla uyguladığımız söylenemez
 - b. Kişisel beceri ile yürütüyoruz
 - c. Kurumsal standart olmasa da planlı yürütülebilen bir sürecimiz var
 - d. Kurumsal düzeyde tanımlanmış ve uygulanan bir sürecimiz var
 - e. Tanımlı ve niceliksel denetim altında yürüten bir sürecimiz var
 - f. Tanımlı, niceliksel denetim altına ve sürekli iyileştirilen bir sürecimiz var

A06. Yazılım Konfigürasyon Yönetimi

Yazılım konfigürasyon yönetimi, bir yazılım ürününün işlevsel ve yapısal bütünlüğünü kurmak ve sürdürmek için yürütülen faaliyetlerdir.

1. Kuruluşunuzda konfigürasyon yönetimini düzenleyen **politika, prosedür ve standartlar** için aşağıdakilerden *hangisi* geçerlidir?
 - a. Yazılı politika, prosedür ve standartlar var, ve tam olarak uygulanıyor
 - b. Yazılı politika, prosedür ve standartlar var, ama kısmen uygulanabiliyor
 - c. Yazılı politika, prosedür ve standartlar var, ama uygulanamıyor
 - d. Yazılı olmayan ama uygulanan bazı kurallar var
 - e. Politika, prosedür, standart veya kurallar olduğu söylenemez
 - f. Bilmiyorum
2. Projelerinizin **planlama aşamasında** konfigürasyon yönetimi için aşağıdakilerden *hangileri* yapılır?
 - a. Konfigürasyon yönetimi organizasyonu ve sorumlulukları belirlenir
 - b. Konfigürasyon yönetimi için gerekli kaynak ve zaman ayrılır
 - c. Kullanılacak araç ve kullanım ilkeleri belirlenir
 - d. Konfigürasyon yönetimi faaliyetleri planlanır
 - e. Altyüklenicinin bu konuda nasıl kontrol edileceği tanımlanır
 - f. Konfigürasyon yönetimi Proje Yöneticisi'nin sorumluluğuna bırakılır
3. Kuruluşunuzda her proje için hazırlanan bir **Konfigürasyon Yönetim Planı** varsa, *kimlerin* görevleri bu planda tanımlanır?
 - a. Konfigürasyon Yöneticisi
 - b. Değişiklik / Konfigürasyon Kontrol Kurulu
 - c. Proje Yöneticisi
4. Projelerinizin **konfigürasyon belirleme aşamasında** aşağıdakilerden *hangileri* yapılır?
 - a. Konfigürasyon öğeleri belirlenir
 - b. Konfigürasyon öğeleri arasındaki ilişkiler tanımlanır
 - c. Konfigürasyon öğelerinin isimlendirme yöntemi tanımlanır
 - d. Konfigürasyon öğelerinin sürüm numaralandırma yöntemi tanımlanır
 - e. Konfigürasyon öğeleri veritabanında saklanır
5. Kuruluşunuzda konfigürasyon belirleme aşamasında aşağıdakilerden *hangileri konfigürasyon ögesi* olarak belirlenir?
 - a. Yazılım kodları
 - b. Gereksinim belgeleri
 - c. Tasarım belgeleri
 - d. Sınama belgeleri
 - e. Sınama verileri
 - f. Kullanıcı belgeleri
 - g. Proje Planı
 - h. Kalite Planı
 - i. Konfigürasyon Yönetim Planı
 - j. Başka iş ürünleri
6. Projelerinizde aşağıdaki **dayanak (baseline) tanımlama** yaklaşımlarından *hangileri* kullanılmaktadır?
 - a. İşlevsel dayanak (functional baseline: sistem gereksinimleri bazında)
 - b. Atanmış dayanak (allocated baseline: yazılım gereksinimleri bazında)
 - c. Geliştirme dayanağı (developmental baseline: geliştirme aşamaları bazında)
 - d. Ürün dayanağı (product baseline: ürün sürümleri bazında)
7. Kuruluşunuzda aşağıdakilerden *hangileri konfigürasyon değişiklik kontrolü aşamasında* uygulanıyor?
 - a. Değişiklik kontrolü kişisel inisiyatif ile yapılıyor
 - b. Kurumsal düzeyde belgelenmiş bir prosedür izleniyor
 - c. Teknik gözden geçirmelerle ürün kontrolü sağlanıyor
 - d. Değişiklik istekleri sadece proje yönetimi tarafından değerlendirilip onaylanıyor
 - e. Değişiklik istekleri bir Değişiklik Kontrol Kurulu'nda değerlendiriliyor
 - f. Değişiklik istekleri bir araç yardımıyla değerlendiriliyor
 - g. Değişiklik isteğinden etkilenen taraflara gerekli bildirim yapılıyor
 - h. Önerilen değişikliğin diğer konfigürasyon öğelerine etkileri kişisel düzeyde ele alınıyor
 - i. Önerilen değişikliğin diğer konfigürasyon öğelerine etkileri kurumsal bir yöntem ile izleniyor
8. Projelerinizde **konfigürasyon yönetimi süreci** boyunca *hangilerinin* kaydı tutuluyor?
 - a. Konfigürasyon öğelerinin isimleri ve sürüm numaraları
 - b. Önerilen konfigürasyon değişiklikleri

- c. Önerilen konfigürasyon değişikliklerinin süre ve maliyet tahminleri
 - d. Önerilen konfigürasyon değişikliklerinin onay ve öncelik durumları
 - e. Onaylanan değişikliklerin yerine getirilme veya getirilememe durumu
 - f. Dönemsel üretilen konfigürasyon durum raporları
 - g. Konfigürasyon denetlemelerinin sonuçları
9. Aşağıdaki **denetleme (audit)** türlerinden *hangileri* projelerinizde uygulanıyor?
- a. Konfigürasyon öğelerinin işlevsel bütünlüğüne odaklanan denetlemeler
 - b. Konfigürasyon öğelerinin fiziksel bütünlüğüne odaklanan denetlemeler
 - c. Müşterinin istediği denetlemeler
 - d. Sistem denetlemeleri
 - e. Süreç denetlemeleri
10. Kuruluşunuzda **konfigürasyon yönetimi** ile ilgili genel bir değerlendirme yapmak gerekirse, aşağıdakilerden *hangisi* mevcut durumu en iyi yansıtıyor?
- a. Başarıyla uyguladığımız söylenemez
 - b. Kişisel beceri ile yürütüyoruz
 - c. Kurumsal standart olmasa da planlı yürütülebilen bir sürecimiz var
 - d. Kurumsal düzeyde tanımlanmış ve uygulanan bir sürecimiz var
 - e. Tanımlı ve niceliksel denetim altında yürüten bir sürecimiz var
 - f. Tanımlı, niceliksel denetim altına ve sürekli iyileştirilen bir sürecimiz var

A07. Yazılım Mühendisliği Yönetimi

Yazılım mühendisliği yönetimi, yazılım geliştirme ve bakımı bağlamında yürütülen organizasyonel yapılanma, planlama, iletişim, ölçme ve kontrol faaliyetlerini kapsar.

1. Kuruluşunuzda yazılım mühendisliği faaliyetlerini düzenleyen **politika, prosedür ve standartlar** için aşağıdakilerden *hangisi* geçerlidir?
- a. Yazılı politika, prosedür ve standartlar var ve tam olarak uygulanıyor
 - b. Yazılı politika, prosedür ve standartlar var ama kısmen uygulanabiliyor
 - c. Yazılı politika, prosedür ve standartlar var ama uygulanamıyor
 - d. Yazılı olmayan ama uygulanan bazı kurallar var
 - e. Politika, prosedür, standart veya kurallar olduğu söylenemez
 - f. Bilmiyorum
2. Kuruluşunuzun **yazılım organizasyon yapısı** ile ilgili olarak aşağıdakilerden *hangisi* geçerlidir?
- a. Yazılım mühendisliği için tanımlanmış özel bir organizasyon yapısı yok
 - b. Ağırlıklı olarak fonksiyonel ve uzmanlık alanlarını temel alan hiyerarşik bir yapı var
 - c. Beceri havuzlarından kaynak kullanımına dayanan proje bazlı bir yapı var
 - d. Uzmanlık alanları ile proje ekipleri arasında matris bir yapı var
 - e. Alınan işlere ve proje gereklerine göre tasarlanan geçici yapılar sözkonusu
 - f. Hiçbiri
3. Yazılım mühendisliği **insan kaynakları yönetimi** konusunda aşağıdakilerden *hangileri* kuruluşunuzda geçerlidir?
- a. Yazılım mühendisliğinin gereklerini dikkate alan bir insan kaynakları politikamız var
 - b. Başvuru ve işe alma süreçleri kurumsallaşmış, ve iş gereklerine uygun nitelikte elemanları zamanında bulmayı başarıyor
 - c. İnsan kaynakları dönemsel planlar dahilinde, muhtemel projeler ve personel hareketleri dikkate alınarak temin ediliyor
 - d. Üstlendiğimiz projelerde uygun nitelik ve sayıda eleman ile çalışıyoruz
 - e. Personel dönüşüm hızımız projelerimizin performansını etkilemiyor
 - f. Çalışanlara işin gereğine uygun nitelikte fiziki altyapı imkanları sağlanıyor
 - g. Yazılım elemanlarımız gerekli düzeyde mesleki eğitim alabiliyorlar
 - h. Performans yönetimi sistemimiz var
 - i. Kıdem ve ücretler performans değerlendirmelerine dayanıyor
 - j. Kariyer planlaması uygulanıyor
4. Kuruluşunuzda yazılım mühendisliği faaliyetleri için aşağıdaki **iletişim mekanizmalarından** *hangileri* kullanılıyor?
- a. Elektronik posta
 - b. Teknik memorandumlar
 - c. Toplantı tutanakları

- d. Intranet
 - e. Paylaşılan dizinler
 - f. Groupware araçları
 - g. Duyuru panoları
 - h. Video konferanslar
 - i. Yatay gözden geçirme toplantıları (peer review)
 - j. Müşteri ile gözden geçirme toplantıları (joint review)
 - k. Proje başlangıç toplantısı
 - l. Proje gelişim toplantısı
 - m. Proje sonuçlandırma toplantısı
 - n. İç sunumlar
 - o. Başka iletişim mekanizmaları
5. Kuruluşunuzda yazılım projeleri için aşağıdaki **proje rollerinden** *hangileri* yetki ve sorumlulukları ile tanımlanmıştır? (Bir proje üyesi birden fazla rolü aynı zamanda üstlenebilir)
- a. Proje Yöneticisi
 - b. Müşteri / sözleşme yöneticisi
 - c. Takım Lideri
 - d. Gereksinim mühendisi / analisti
 - e. Tasarım mühendisi
 - f. Mimar
 - g. Şekil arayüz tasarımcısı
 - h. Programcı
 - i. Test mühendisi
 - j. Sistem mühendisi
 - k. Kullanıcı belgeleri yazarı
 - l. Kullanıcı belgeleri editörü
 - m. Kalite güvence yöneticisi / mühendisi
 - n. Konfigürasyon Yöneticisi
 - o. Başka proje rolleri
6. Kuruluşunuzda kullanılan **Yazılım Proje Planı**'nda aşağıdakilerden *hangileri* mutlaka yer alır?
- a. Projenin hedefi ve sonuçlanma kriterleri
 - b. Proje organizasyon yapısı ve roller
 - c. Proje iletişim mekanizmaları
 - d. Proje kapsam yönetimi (hedef, bütçe ve ürünlerin değişimi)
 - e. Gelişim izleme ve denetim yöntemi
 - f. Kullanılacak süreç, prosedür ve standartlar
 - g. Seçilen yazılım yaşam döngüsü modeli ve gerekçesi
 - h. Proje ürünlerinin ve ara ürünlerin tanımı
 - i. Proje teknik belgelerinin tanımı
 - j. Proje insan kaynakları ve altyapı kullanımı
 - k. Satınalma / altyüklenici yönetim planı
 - l. Proje görev / zaman çizelgesi (Gantt şeması, vb.)
 - m. Proje riskleri ve risk yönetim metodolojisi
 - n. Süre ve emek (adam-gün) kestirimleri, ve yanılma payları
 - o. Proje ölçümleri
 - p. Başka bilgiler
7. Aşağıdaki **yazılım ölçümlerinden** *hangileri* kuruluşunuzda uygulanıyor?
- a. Gereksinim sayısı
 - b. Yazılım hataları (tür, kritiklik, neden, olduğu yer, giderildiği yer, vb.)
 - c. Faaliyet bazında emek (analiz, tasarım, kodlama, sına, destek, vb.)
 - d. Yazılım büyüklüğü (satır sayısı, function point, vb.)
 - e. Algoritmik karmaşıklık
 - f. İki hata oluşumu arası ortalama süre (mean-time-to-failure)
 - g. Sınama kapsamı (test coverage)
 - h. Yeniden kullanım yüzdesi
 - i. Belgeler için önerilen / gerçekleşen değişiklikler
 - j. Aşama bazında kestirilen / gerçekleşen kadro (tam zaman eşdeğeri adam)
 - k. Aşama bazında kestirilen / gerçekleşen emek (adam-saat)
 - l. Aşama bazında kestirilen / gerçekleştirilen süre
 - m. Aşama bazında kestirilen maliyet / gerçekleşen maliyet
 - n. Aşama bazında kestirilen / gerçekleşen faaliyetlerin sayısı
 - o. Gereksinimlerin zamana bağlı gelişimi (eklenen, çıkarılan, değişen gereksinimler)

- p. Ürünün müşteri gereksinimlerini karşılama başarısı
 - q. Hizmetin müşteri gereksinimlerini karşılama başarısı
 - r. Başka tür yazılım ölçümleri
8. Kuruluşunuzda aşağıdaki **yazılım ölçme zorluklarından hangileri** yaşanıyor?
- a. Ölçme amacının, ölçümü yapanlarca tam olarak anlaşılması
 - b. Ölçüm tanımlarının tam olarak anlaşılması
 - c. Veri toplama, inceleme ve raporlama gibi faaliyetlerin önceden planlanmaması
 - d. Veri toplamanın düzenli bir şekilde yapılamaması
 - e. Veri saklama ortamının ve araçlarının tanımlanmaması / yetersizliği
 - f. Ölçüm sonuçlarının inceleme amacı ile etkin bir şekilde kullanılmaması (istatistiksel analiz tekniklerinin kullanılmaması, vb.)
 - g. Ölçüm sonuçlarının iyileştirme amacı ile kullanılmaması (Sonuçların yorumlanmasında güçlükler, sonuçlara kuşku ile yaklaşılması, vb.)
 - h. Ölçme faaliyetlerine katılım ve desteğin yetersizliği
 - i. Başka tür zorluklar yaşanıyor
9. Kuruluşunuzda **yazılım ölçme uygulamalarının benimsenmesine** aşağıdaki faktörlerden **hangileri** engel oluyor?
- a. İş yükünün ağırlığı
 - b. Bütçe imkanlarının kısıtlı olması
 - c. Mevcut teknolojinin yetersiz olması
 - d. Üst Yönetimden yeterli desteğin alınmaması
 - e. Çalışanların bu tür uygulamalara karşı isteksizliği
 - f. Bu tür uygulamalar için gerekli teknik bilginin olmaması / edinilme zorluğu
 - g. Firma kültürünün uygun olmaması
 - h. Başka faktörler
10. Kuruluşunuzda **yazılım projelerinin yönetimi** ile ilgili genel bir değerlendirme yapmak gerekirse, aşağıdakilerden **hangisi** mevcut durumu en iyi yansıtıyor?
- a. Başarıyla uyguladığımız söylenemez
 - b. Kişisel beceri ile yürütüyoruz
 - c. Kurumsal standart olmasa da planlı yürütülebilen bir sürecimiz var
 - d. Kurumsal düzeyde tanımlanmış ve uygulanan bir sürecimiz var
 - e. Tanımlı ve niceliksel denetim altında yürüyen bir sürecimiz var
 - f. Tanımlı, niceliksel denetim altına ve sürekli iyileştirilen bir sürecimiz var

A08. Yazılım Süreç Mühendisliği

Yazılım süreç mühendisliği, yazılım geliştirme ve bakımının kurumlaşması amacıyla yürütülen süreç tanımlama, uygulama, ölçme ve iyileştirme faaliyetlerini kapsar.

1. Kuruluşunuzda aşağıdaki **yazılım süreç standartlarından hangisini** temel alıyorsunuz?
- a. ISO-15504 (SPICE)
 - b. SW/CMM veya CMMI
 - c. ISO/IEC-12207 veya IEEE/EIA Std 12207
 - d. NATO AQAP-160
 - e. MIL-STD-498
 - f. Tick-IT
 - g. Bootstrap
 - h. TRILLIUM
 - i. Başka bir standardı temel alıyoruz
 - j. Hiçbir standardı temel almıyoruz
2. Kuruluşunuzda **yazılım mühendisliği ve yönetim süreçlerinden hangileri** tanımlı?
- a. Sistem gereksinim analizi ve tasarım
 - b. Yazılım gereksinim analizi
 - c. Yazılım tasarımı
 - d. Yazılım gerçekleştirme
 - e. Yazılım tümleştirme
 - f. Yazılım sına
 - g. Sistem tümleştirme ve sına
 - h. Sistem ve yazılım bakımı
 - i. Yazılım proje yönetimi

- j. Yazılım kalite yönetimi
 - k. Yazılım risk yönetimi
 - l. Yazılım altyüklenici yönetimi
3. Kuruluşunuzda **yazılım süreçlerini** *kim* tanımlıyor ve iyileştiriyor?
 - a. Tek sorumluluğu bu iş olan bir organizasyonel birim
 - b. Tek sorumluluğu bu iş olan bir uzman
 - c. Diğer işlerinin yanında bu görevi de sürekli olarak üstlenen yazılım elemanları
 - d. Diğer işlerinin yanında bu görevi de geçici olarak üstlenen yazılım elemanları
 - e. Dış danışmanlar
 - f. Hiç kimse
 4. Kuruluşunuzda **süreçlerle yönetim** konusunda aşağıdakilerden *hangisi* en doğru?
 - a. Süreçlerle yönetim yaptığımız söylenemez
 - b. Süreçleri tanımlıyoruz
 - c. Süreçleri tanımlıyoruz ve personeli eğitiyoruz
 - d. Süreçleri tanımlıyoruz, personeli eğitiyoruz ve uygulamayı ölçümlerle izliyoruz
 - e. Süreçleri tanımlıyoruz, personeli eğitiyoruz, uygulamayı ölçümlerle izliyoruz ve bulguları süreçleri iyileştirmek için kullanıyoruz
 5. Kuruluşunuzda **yazılım mühendisliği bilgi birikimi** nasıl kurumlaştırılıyor?
 - a. Usta-çırak ilişkisi ile
 - b. İç seminerler ile
 - c. Yol gösterici belgeler ile
 - d. Elektronik bilgi saklama ve yönetim sistemleri ile
 - e. İç danışmanlar yardımıyla
 - f. Teknik destek personeli yardımıyla
 - g. Başka şekilde
 6. **Süreçlerle yönetim** üzerine aşağıdaki görüşlerden *hangilerine* katılıyorsunuz?
 - a. Yazılım geliştiren kuruluşlar için en kritik başarı faktörlerindedir
 - b. Yazılım araçlarıyla desteklenmesi zorunludur
 - c. Standart süreçler izlemek zaman ve kaynak kaybına yol açar
 - d. Bu uygulamaların kurumsallaşması uzun sürer
 - e. Yazılımcıların iş kültürü bu alanda sistematik uygulamalara elverişli değil
 - f. Müşteriler bu alanda yazılım geliştirenleri yeterince zorlamıyor
 7. Kuruluşunuzda **yazılım süreçlerinin uygulanması ve iyileştirilmesi** konusunda aşağıdakilerden *hangileri* geçerlidir?
 - a. Süreçleri her projede tanımlandığı şekilde uygulamaya çalışıyoruz
 - b. Süreçleri uyarlama esnekliğimiz var: projeden projeye önemli farklar olabilir
 - c. Proje başarısı temel bir performans ölçüsüdür
 - d. Süreç başarısı temel bir performans ölçüsüdür
 - e. Süreçte değişiklik gerektiği zaman bu bir proje gibi ele alınıyor ve gerçekleştiriliyor
 - f. Süreçlerin iyileştirilmesi zamanla değişen iş şartları ve gerekleri ile tetikleniyor
 - g. Süreçlerin iyileştirilmesi yeni teknolojilerin transferi arzusu ile tetikleniyor
 8. Bir projenizde **başarısızlık söz konusu olursa** kuruluşunuzda genellikle aşağıdaki sonuçlardan *hangisi* oluşur?
 - a. Başarısızlık gizlenir veya unutulur
 - b. Başarısızlıktan belirli proje üyeleri sorumlu tutulur, gerekirse yaptırım uygulanır
 - c. Başarısızlık yaşanan proje ekibi ile kapsamlı bir “post-mortem” analiz yapılır ve hataların tekrarlanmaması için dersler çıkarılır
 - d. Başarısızlığın kaynaklandığı süreçler belirlenir, neden-sonuç incelemesi yapıldıktan sonra bu süreçlerin iyileştirilmesine çalışılır
 9. Kuruluşunuzda **yazılım süreçlerinin genel performansını değerlendirmek** için aşağıdakilerden *hangileri* yapılıyor?
 - a. Süreçlerimizin sonuçları için somut performans kriterleri tanımlıyoruz
 - b. Süreçlerimizin performansını belirlemek için sistematik ölçümler yapıyoruz
 - c. Süreçlerimizin yeterliliğini belirlemek için evrensel süreç standartlarına (SPICE, SW/CMM, vb.) göre iç değerlendirmeler yapıyoruz
 - d. Süreçlerimizin yeterliliğini belirlemek için evrensel süreç standartlarına (SPICE, SW/CMM, vb.) göre tarafsız değerlendirmeler yaptırıyoruz
 - e. Süreçlerimizin performans verilerini bilinen benchmark’lar ile karşılaştırıyoruz
 10. Kuruluşunuzda **süreçlerin tanımlanması ve iyileştirilmesi** ile ilgili genel bir değerlendirme yapmak gerekirse, aşağıdakilerden *hangisi* mevcut durumu en iyi yansıtıyor?
 - a. Süreçlerimizin tanımlanması ve iyileştirilmesi için sistematik ölçümler yapıyoruz
 - b. Süreçlerimizin tanımlanması ve iyileştirilmesi için sistematik ölçümler yapıyoruz ve bulguları süreçleri iyileştirmek için kullanıyoruz
 - c. Süreçlerimizin tanımlanması ve iyileştirilmesi için sistematik ölçümler yapıyoruz ve bulguları süreçleri iyileştirmek için kullanıyoruz ve bulguları süreçleri iyileştirmek için kullanıyoruz
 - d. Süreçlerimizin tanımlanması ve iyileştirilmesi için sistematik ölçümler yapıyoruz ve bulguları süreçleri iyileştirmek için kullanıyoruz ve bulguları süreçleri iyileştirmek için kullanıyoruz

- a. Başarıyla uyguladığımız söylenemez
- b. Kişisel beceri ile yürütüyoruz
- c. Kurumsal standart olmasa da planlı yürütülebilen bir sürecimiz var
- d. Kurumsal düzeyde tanımlanmış ve uygulanan bir sürecimiz var
- e. Tanımlı ve niceliksel denetim altında yürüten bir sürecimiz var
- f. Tanımlı, niceliksel denetim altına ve sürekli iyileştirilen bir sürecimiz var

A09. Yazılım Mühendisliği Araç ve Yöntemleri

Yazılım mühendisliği araçları, yazılım geliştirme ve bakımında kullanılan yazılımlardır. Yazılım mühendisliği yöntemleri, yazılım yaşam döngüsünde kullanılan mühendislik yaklaşımlarını ve teknikleri kapsar.

1. Kuruluşunuzda aşağıdaki **yazılım mühendisliği araçlarından hangilerini** kullanıyorsunuz?
 - a. Gereksinim modelleme araçları (nesne-bağıntı şemaları, veri akış şemaları vb.)
 - b. Gereksinim yönetim araçları (izlenebilirlik, Quality Function Deployment vb.)
 - c. Kullanım vakaları üreten araçlar (use case generators)
 - d. Veritabanı tasarım araçları
 - e. Yapısal tasarım modelleme araçları (sınıf ve nesne şemaları, vb.)
 - f. Davranışsal tasarım modelleme araçları (durum geçiş şemaları, vb.)
 - g. Kod editörleri
 - h. Kod derleyiciler
 - i. Otomatik kod üreticiler
 - j. Yorumlayıcılar (interpreters)
 - k. Hata bulma araçları (debuggers)
 - l. Yazılım ölçme araçları (büyüklük, karmaşıklık, vb.)
 - m. Sınama senaryosu üretim araçları
 - n. Sınama verisi üretim araçları
 - o. Sınama uygulama ve gözleme araçları
 - p. Sınama sonuçlarını değerlendirme araçları
 - q. Yazılım performansı analiz araçları
 - r. Program tanıma/anlama araçları
 - s. Program re-engineering araçları (ör: kodu bir dilden başka bir dile çevirme)
 - t. Program reverse engineering araçları (ör: koddan tasarım üretme)
 - u. Başka araçlar
2. Kuruluşunuzda **yazılım mühendisliği araçlarının kullanım durumunu** aşağıdakilerden *hangisi* en iyi ifade ediyor?
 - a. Bu türden araçlar alınmadı
 - b. Alındı ama hiç kullanılmadı
 - c. Önce hevesle kullanıldı, sonra terk edildi
 - d. Kullanıldı ama işlevlerinden tam yararlanılmadı
 - e. Kullanıldı ama verimliliğe yararı olmadı
 - f. Etkin olarak kullanıldı ve verimliliğe yararı oldu
3. **Yazılım mühendisliği araçları hakkında** aşağıdaki görüşlerden *hangilerine* katılıyorsunuz?
 - a. Bu tür araçlar verimliliği artırmak için zorunludur
 - b. Bu tür araçlar kaliteli yazılım üretmek için zorunludur
 - c. Bu tür araçların ölü yatırım olma riski yüksektir
 - d. Bu tür araçlar kurumsal politikalarla desteklenmedikçe kullanılmamalıdır
 - e. Bu tür araçların öğrenilmesi ve etkin kullanımı için çok zaman gerekir
4. Kuruluşunuzda aşağıdaki **yazılım yönetim araçlarından hangilerini** kullanıyorsunuz?
 - a. Proje planlama ve izleme araçları
 - b. Risk yönetim araçları
 - c. Süreç / iş akışı modelleme araçları
 - d. Bütünleşik belge yönetim sistemi
 - e. Bütünleşik iş akışı yönetim sistemi
 - f. Konfigürasyon yönetim araçları (öge isimlendirme, sürüm yönetimi, vb.)
 - g. Süreç ölçme araçları (veri toplama, saklama, analiz ve raporlama)
 - h. Yazılım kalitesi denetim araçları (veri akışı, kontrol akışı, bağımlılık analizleri, vb.)
 - i. Bütünleşik CASE platformları
 - j. İstatistiksel veri analizi araçları
 - k. Müşteri ilişkileri yönetim (CRM) araçları
 - l. Başka araçlar
5. Kuruluşunuzda **yazılım yönetim araçlarının kullanım durumunu** aşağıdakilerden *hangisi* en iyi ifade ediyor?

- a. Bu türden araçlar alınmadı
 - b. Alındı ama hiç kullanılmadı
 - c. Önce hevesle kullanıldı, sonra terk edildi
 - d. Kullanıldı ama işlevlerinden tam yararlanılmadı
 - e. Kullanıldı ama verimliliğe yararı olmadı
 - f. Etkin olarak kullanıldı ve verimliliğe yararı oldu
6. Kuruluşunuzda **yazılım mühendisliği ve yönetim araçlarının seçimi** çoğunlukla *kim* tarafından yapılıyor?
- a. Proje Yöneticisi
 - b. Takım Lideri
 - c. Yazılım mühendisleri
 - d. Bölüm Yöneticisi
 - e. Üst Yönetim
 - f. Bu amaçla oluşturulan özel kurul
 - g. Teknik danışmanlar
 - h. Kural yok; duruma bağlı olarak değişir
7. **Araç seçiminde** aşağıdaki kriterlerden *hangileri* en fazla belirleyici oluyor?
- a. Fiyatı
 - b. Yararlanacak proje sayısı
 - c. Mevcut sistemlerle arayüz olanakları
 - d. Verimliliğe yararları
 - e. Kalite iyileştirmeye yararları
 - f. İlgili iş sürecinin işleyişine uygunluğu
 - g. Üretici firma / markası
 - h. Teknolojisinin yeniliği
 - i. Teknik destek hizmetleri
 - j. Eğitim hizmetleri
 - k. Danışmanlık hizmetleri
 - l. Sürüm güncelleme sıklığı
 - m. Başka kriterler
8. Kuruluşunuzda aşağıdaki **yazılım yaşam döngüsü modellerinden** *hangileri* kullanılıyor?
- a. Doğrusal geliştirme (şelale) modeli
 - b. Prototip geliştirme
 - c. Hızlı yazılım geliştirme (rapid application development)
 - d. Artımlı geliştirme (incremental development)
 - e. Spiral model
 - f. Bileşen montaj modeli (component assembly model)
 - g. Eşzamanlı mühendislik (concurrent engineering)
 - h. Formal yöntemler (ör: yazılımın matematiksel bir dille tanımlanması)
 - i. Steril yazılım mühendisliği (cleanroom software engineering)
 - j. Başka bir model
9. **Mühendislik yöntemlerinin seçiminde** aşağıdaki kriterlerden *hangileri* belirleyici oluyor?
- a. Teknolojik eğilimleri izleme isteği
 - b. Müşteri teşviki veya zorlaması
 - c. İlgili iş sürecinin işleyişine uygunluk
 - d. Rakiplerle farklılık yaratma isteği
 - e. Danışman tavsiyesi
 - f. Şirket politikası ve uzun vadeli hedeflere uygunluk
 - g. Başka kriterler
10. Kuruluşunuzda **yazılım araç ve yöntemlerinin seçim ve kullanımı** ile ilgili genel bir değerlendirme yapmak gerekirse, aşağıdakilerden *hangisi* mevcut durumu en iyi yansıtıyor?
- a. Başarıyla uyguladığımız söylenemez
 - b. Kişisel beceri ile yürütüyoruz
 - c. Kurumsal standart olmasa da planlı yürütülebilen bir sürecimiz var
 - d. Kurumsal düzeyde tanımlanmış ve uygulanan bir sürecimiz var
 - e. Tanımlı ve niceliksel denetim altında yürüten bir sürecimiz var
 - f. Tanımlı, niceliksel denetim altına ve sürekli iyileştirilen bir sürecimiz var

A10. Yazılım Kalitesi

Yazılım kalitesi, yazılım ürününün müşteri tarafından belirtilen veya beklenen kalite niteliklerini, ve bu niteliklerin güvenceye alınması için yürütülen faaliyetleri konu alır.

1. Aşağıdakilerden *hangisi* sizin için tipik “**müşteri**”yi tanımlar?
 - a. Genel kurumsal yazılım çözümleri arayan kuruluşlar
 - b. Özel kurumsal çözümler için geliştirme sözleşmesi yapılan kuruluşlar
 - c. Yazılım ürünleri / çözümleri geliştirenler diğer kuruluşlar
 - d. Kuruluş içinden müşteriler
 - e. Bilgisayarı olan ve bir kullanım ihtiyacını karşılamak isteyen bireyler
 - f. Bilgisayar kullanmaya yeni başlayacak bireyler
 - g. Hiçbiri
2. Çalıştığınız organizasyonel birimde aşağıdakilerden *hangileri öncelikli ürün kalite kriterlerindendir*?
 - a. İşlevsellik (işlev zenginliği, doğruluğu, erişilebilirliği, vb.)
 - b. Güvenirlilik (çalışma sürekliliği, sürüm olgunluğu, vb.)
 - c. Verimli çalışma (işlemci ve diğer kaynakların kullanımı, iletişim ve çıktı performansı vb.)
 - d. Kullanım kolaylığı (öğrenilebilirlik, kolay işletilebilirlik, vb.)
 - e. Bakım kolaylığı (sorun analizi ve test kolaylığı, güncellenebilirlik, vb.)
 - f. Taşınabilirlik (donanım/sistem bağımsızlığı, kurulum ve veri paylaşım rahatlığı, vb.)
 - g. Tekrar kullanılabilirlik (diğer uygulamalara işlevsel arayüz sunma, bağımsız kullanılabilme, veri ve işlevlerin doğru ayrıştırılması/gruplanması, vb.)
 - h. Başka kalite kriterleri
3. Ürününüzün **işlevsel kalitesini** artırmak için aşağıdaki yöntemlerden *hangilerini* kullanıyorsunuz?
 - a. İşlevlerin tutarlılığı ve doğru uygulanması gereksinim aşamasından başlayarak yönetilir
 - b. Ürün, bileşenler halinde geliştirilir ve her bileşen bir ürün kabul edilerek kullanıcı memnuniyeti ölçülür
 - c. İşlevler karmaşıklığına göre sınıflandırılır ve kullanıcıya yapılan sunumlarında bu sınıflandırmaya dikkat edilir
 - d. Veri ve işlev arasındaki ilişkilerin kullanıcılar açısından belirgin olmasına önem verilir; doğru işlevlerle hatalı verilerin ayrışmasının rahatlıkla yapılabilmesi sağlanır
 - e. Bir akış izleyen işlevler detaylı olarak ayrıştırılır ve kullanıcının bu akışı bilmesi, izlemesi ve akıştaki yerini anlaması sağlanır
 - f. İşlevlerin prototip örnekleri ürün kullanıcıya sunulmadan öndeğerlendirmeden geçer
 - g. Başka yöntemler kullanılıyor
4. Ürününüzün **güvenirliliği** üzerine somut verileri aşağıdaki yöntemlerin *hangilerini* uygulayarak elde ediyorsunuz?
 - a. Kesintisiz çalışabilme kapasitesi kullanıcıya sunmadan önce pilot denemelerle, kullanıcının kullanımında da sürekli izlenerek belirlenir
 - b. Aynı ortamda mümkün olduğu kadar çok değişik yazılım / donanım bileşeni ile beraber çalışabildiği gözlenir
 - c. Üründen kaynaklanmayan kısıtlayıcı koşullar oluşturup ürün davranışları izlenir
 - d. Kullanım sırasında verilerin tutarlılığı zaman içinde örnekleme yöntemi ile kontrol edilir
 - e. Kullanım sırasında verilerin üründen kaynaklanmayan değişimlere uğraması ile oluşan ürün davranışları izlenir
 - f. Ürünün güvenirliliğini etkileyecek platform (işletim sistemi, veritabanı, vb.) koşulları diğer üreticilerle ortak çalışarak belirlenir
 - g. Verilere ve işlevlere erişimle ilgili ürünün üstlendiği tüm güvenlik adımları kullanım sırasında denetlenir, kullanıcıya anlaşılır şekilde iletilir ve kullanıcının gerekli güvenlik tanımlarını yapabildiği izlenir
 - h. Başka yöntemler uygulanıyor
5. **Kullanımı kolay** bir ürün için aşağıdakilerden *hangilerine* dikkat ediyorsunuz?
 - a. Kullanıcı arayüzünün temel ergonomik standartlara uyumu (Human Interface Guidelines, vb.)
 - b. Verilerin mümkün olduğunca görsel sunumu
 - c. İşlevlerin anlaşılabilir adımlara bölünmesi
 - d. Ürünün kullanımı sırasında erişilebilen yardımcı bilgiler olması (yardım ekranları, vb.)
 - e. Ürünün kullanımı sırasında işlevsel rehber kullanılabilmesi (sonraki menü ve klavye adımlarına yönlendirilmesi, vb.)
 - f. Detaylı kullanıcı el kitaplarının ürünle beraber sunulması
 - g. Ürünün kolay kurulması ve sistemdeki diğer gerekli bileşenlerle rahat çalışabilmesinin sağlanması
 - h. Ürünün çok sayıda ve değişik düzeyde kullanıcılar tarafından rahat öğrenilmesini sağlayacak eğitim malzemeleri sağlanması, eğiticilerin eğitimi
 - i. İşlevlerin ve verilerle işlevler arasındaki bağlantıların rahat anlaşılabilmesi için örnek veriler
 - j. Başka konulara dikkat ediyoruz
6. Ürününüzün **bakımı kolaylığı** için aşağıdaki yöntemlerden *hangilerini* uyguluyorsunuz?

- a. Ürünün tüm bileşenlerinin bakım kapsamında yer alacak değişiklikleri sorunsuz içerebilmeleri için tasarımda ekler yapmak, esnek tasarımlar geliştirmek
 - b. Üründe oluşabilecek temel sorunlar için teşhis / tedavi bileşenleri içermek, ürünün işlevsel durumunu (sorunlu / sorunsuz) görsel olarak kullanıcıya sunmasını sağlamak
 - c. Bakım için tüm temel faaliyetleri belgelendirmek, süreçleri tanımlamak
 - d. Bakımdan sorumlu geliştiricileri ve tüm ilgili personeli değişikliklerin kolay uyarlanabilmesi için eğitmek
 - e. Ürünün dönemsel bakım zamanlarında bunu otomatik gerçekleştirmesi, bunun için gerekli adımları başlatmayı sağlayan bileşenler içermesi (yedekleme dahil)
 - f. Ürünün içeriğinde olmayan ancak ürünle beraber kurulumu zorunlu tüm sistem bileşenleri / platformlar (veritabanı, vb.) için bakım rehberi ve uyarıların ürün belgelerinde içerilmesi, ürünün gereken durumlarda anlaşılır mesajlarla kullanıcıyı uyarması
 - g. Ürünün bakım merkezlerinin kullanıcı tarafından bilinmesini ve kolay erişilmelerini belgelemeyle / elektronik iletişimle sağlamak
 - h. Başka yöntemler uyguluyoruz
7. **Ürün performansı** için aşağıdaki ölçümlerden *hangilerini* kullanıyorsunuz?
- a. Sistem kaynaklarının (bellek, işlemci zamanı) kullanımı
 - b. İşlem hızına kullanıcı tarafından önem verilen işlevlerin zaman performansı (çıktılar, bakım bileşenlerinin çalışması, vb.)
 - c. Bellek, zaman gibi kaynaklarla ilgili ölçümlerin yazılım kullanılmayan veya başka bir ürün kullanılan ortamlarla kıyaslanması
 - d. Kullanıcının işlemlere erişim adımlarının ölçümü (sayı ve zaman olarak)
 - e. Kullanıcının verilere ulaşım adımlarının ölçümü (sayı ve zaman olarak)
 - f. Ulaşılan veri depolama kapasitesi
 - g. Ürünle beraber çalışan sistem bileşenlerinin kaynak / zaman kullanımları
 - h. Ürünün ağ iletişimine getirdiği yükün ölçümü
 - i. Başka ölçümler kullanıyoruz
8. Kuruluşunuzda **ürün kalitesi temel verileri** ile ilgili olarak aşağıdakilerden *hangisi* mevcut durumu en iyi yansıtmaktadır?
- a. Ürün kalitesi ile ilgili sistemli veri topladığımız söylenemez
 - b. Ürün kalitesi ile ilgili temel veriler genellikle kullanıcı ile beraber çalışan destek / hizmet / satış birimlerinde toplanmakta ve değerlendirilmektedir
 - c. Ürün kalitesi ile ilgili temel veriler genellikle ürün geliştirme ve sınama birimlerince derlenmekte ve değerlendirilmektedir
 - d. Ürün kalitesi ile ilgili birimlerin ortaklaşa yürüttüğü çalışmalar ve özel organizasyonel oluşumlar verileri derlemekte ve değerlendirmektedir
 - e. Hiçbiri
9. Aşağıdaki **ürün kalite standartları ve ürün kalitesi ölçüm modellerinin** *hangilerini* tanıyorsunuz veya çalışmalarınızda referans kabul ediyorsunuz?
- a. ISO/IEC 9126
 - b. ISO/IEC 9127
 - c. ISO/IEC 12119
 - d. IEEE Std 982
 - e. IEEE Std 1012
 - f. IEEE Std 1061
 - g. IEEE Std 1063
 - h. IEEE Std 1465
 - i. Başka standartlar veya modeller
10. Kuruluşunuzda **ürün kalitesini artıracak faaliyetler** ile ilgili genel bir değerlendirme yapmak gerekirse, aşağıdakilerden *hangisi* mevcut durumu en iyi yansıtıyor?
- a. Başarıyla uyguladığımız söylenemez
 - b. Kişisel beceri ile yürütüyoruz
 - c. Kurumsal standart olmasa da planlı yürütülebilir bir sürecimiz var
 - d. Kurumsal düzeyde tanımlanmış ve uygulanan bir sürecimiz var
 - e. Tanımlı ve niceliksel denetim altında yürüten bir sürecimiz var
 - f. Tanımlı, niceliksel denetim altına ve sürekli iyileştirilen bir sürecimiz var

EK 2 - Katılımcı Profili

Anket katılımcısının profili aşağıdaki bilgilerle tanımlanmıştır:

1. **Çalıştığı kuruluşun ölçeği:** Kuruluşunuzda kaç kişi çalışıyor?
 - a. 1-99
 - b. 100-499
 - c. 500 ve üstü
2. **Yazılım kadrosunun ölçeği:** Kuruluşunuzun yazılım geliştirme/bakım ve yönetimi kadrosu kaç kişiden oluşuyor?
 - a. 1-9
 - b. 10-49
 - c. 50-99
 - d. 100 ve üstü
3. **Çalıştığı kuruluşun türü:** Kuruluşunuz hangi gruba giriyor?
 - a. Paket yazılım üreticisi
 - b. Yazılım yüklenicisi
 - c. İç kullanım için yazılım geliştiren veya bakımını yapan kuruluş
4. **Sektör bilgisi:** Kuruluşunuz paket yazılım üreticisi veya yazılım yüklenicisi ise *ağırlıklı hitap ettiği* ana sektör; iç kullanım için yazılım geliştiren veya bakımını yapan bir kuruluş ise *ait olduğu* ana sektör hangisi?
 - a. Endüstri (Beyaz eşya; Denizcilik; Elektronik-bilgisayar-haberleşme; Enerji; Gıda; İlaç; İnşaat malzemeleri; Konut; Madencilik; Makine-otomotiv; Mobilya; Müzik-video; Petrokimya; Tarım-hayvancılık; Tekstil; Uzay-havacılık; Yayımcılık; Diğer)
 - b. Finans (Bankacılık; Sigortacılık; Diğer)
 - c. Hizmet (Bilişim hizmetleri; Danışmanlık; Elektrik-su-gaz dağıtım; Eğitim; Eğlence; Gayrimenkul pazarlama; Haberleşme; Hukuk hizmetleri; İnşaat taahhüt; Lojistik-dağıtım-toptancılık; Mühendislik-ArGe hizmetleri; Perakendecilik; Reklamcılık; Sağlık; Turizm-ağırlama; Ulaştırma; Yazılı-görsel basın; Diğer)
 - d. Kamu (Adalet; Bayındırlık; Güvenlik; Merkezi-yerel yönetim; Savunma; Sivil toplum kuruluşları, Diğer)
5. **Görev:** Aşağıdakilerden hangisi mevcut görevinizi en iyi tanımlıyor?
 - a. Üst düzey yönetici
 - b. IT yöneticisi
 - c. Diğer bölüm yöneticisi
 - d. Yazılım geliştirme / bakım grubu yöneticisi
 - e. Yazılım Proje Yöneticisi
 - f. Yazılım mühendisi
 - g. Danışman / teknik uzman
 - h. Yazılım kalite veya süreç yöneticisi
 - i. Diğer
6. **E-posta adresi** (Zorunlu değil)

EK 3 - Tek Yanıtlı Sorular

Konu	Soru numarası									
	1	2	3	4	5	6	7	8	9	10
A01						X		X		X
A02	X									X
A03						X	X		X	X
A04				X						X
A05		X		X		X				X
A06	X									X
A07	X	X								X
A08		X			X	X				X
A09	X		X	X				X		X
A10	X							X		X

X tek bir yanıtın işaretlenmesi zorunlu olan soru

EK 4 - Anket Sorularının Güvenilirlik Analizi

Aşağıdaki hesaplamaların tümü Minitab yazılımı kullanılarak yapılmıştır.

2008 Yılı Verileri – 17 Örnek

Birim Analizi 1; 2; 3; 4; 5; 6; 7; 8; 9; 10

Korelasyon Matrisi

	1	2	3	4	5	6	7	8	9
2	0,858								
3	0,859	0,914							
4	0,827	0,745	0,728						
5	0,842	0,756	0,836	0,670					
6	0,712	0,659	0,659	0,816	0,696				
7	0,946	0,852	0,872	0,799	0,851	0,714			
8	0,845	0,686	0,649	0,707	0,714	0,560	0,798		
9	0,786	0,601	0,687	0,612	0,694	0,448	0,697	0,779	
10	0,797	0,802	0,775	0,849	0,854	0,768	0,789	0,639	0,641

Hücre İçerikleri: Pearson Korelasyonu

Birim and Toplam İstatistikleri

Değişken	Toplam		
	Sayı	Ortalama	StSap
A1	17	3,294	1,532
A2	17	3,353	1,057
A3	17	3,647	1,367
A4	17	2,941	1,298
A5	17	3,353	1,320
A6	17	2,882	1,409
A7	17	3,294	1,404
A8	17	3,059	1,435
A9	17	3,118	1,111
A10	17	2,647	1,115
Toplam	17	31,588	11,517

Cronbach's Alpha = 0,9666

Omitted Birim İstatistikleri

Omitted Değişken	Ayar.	Ayar.	Birim-Ayar. Toplam Korel.	Squared Multiple Korel.	Cronbach's Alpha
	Toplam	Toplam			
A1	28,29	10,05	0,9525	0,9582	0,9589
A2	28,24	10,59	0,8689	0,9548	0,9631
A3	27,94	10,30	0,8777	0,9574	0,9618
A4	28,65	10,39	0,8512	0,9451	0,9628
A5	28,24	10,35	0,8721	0,9581	0,9621
A6	28,71	10,42	0,7509	0,7512	0,9669
A7	28,29	10,20	0,9324	0,9233	0,9597
A8	28,53	10,34	0,7983	0,8791	0,9651
A9	28,47	10,67	0,7437	0,8462	0,9666
A10	28,94	10,53	0,8711	0,9594	0,9627

2001 Yılı Verileri – 27 Örnek

Birim Analizi B1; B2; B3; B4; B5; B6; B7; B8; B9; B10

Korelasyon Matrisi

	B1	B2	B3	B4	B5	B6	B7	B8	B9
B2	0,491								
B3	0,481	0,573							
B4	0,461	0,587	0,538						
B5	0,374	0,727	0,699	0,682					
B6	0,384	0,356	0,547	0,334	0,355				
B7	0,418	0,623	0,514	0,733	0,716	0,551			
B8	0,354	0,645	0,315	0,582	0,467	0,530	0,689		
B9	0,330	0,358	0,409	0,336	0,369	0,436	0,492	0,499	
B10	0,297	0,301	0,395	0,367	0,425	0,499	0,537	0,366	0,563

Hücre İçerikleri: Pearson Korelasyonu

Birim ve Toplam İstatistikleri

Değişken	Toplam		
	Sayı	Ortalama	StSap
B1	27	2,444	1,121
B2	27	2,148	1,064
B3	27	2,481	1,014
B4	27	2,148	1,134
B5	27	2,481	1,397
B6	27	1,778	1,311
B7	27	2,111	1,368
B8	27	2,037	1,506
B9	27	1,667	1,144
B10	27	1,593	0,797
Toplam	27	20,889	8,741

Cronbach's Alpha = 0,9009

Omitted Birim İstatistikleri

Omitted Değişken	Ayar. Toplam Ortalama	Ayar. Toplam StSap	Squared		Cronbach's Alpha
			Birim-Ayar. Toplam Korel.	Multiple Korel.	
B1	18,444	8,097	0,5283	0,3793	0,8986
B2	18,741	7,940	0,7244	0,7072	0,8873
B3	18,407	8,025	0,6744	0,6952	0,8905
B4	18,741	7,896	0,7135	0,6642	0,8874
B5	18,407	7,672	0,7274	0,7677	0,8861
B6	19,111	7,900	0,5931	0,6031	0,8954
B7	18,778	7,582	0,8219	0,7608	0,8788
B8	18,852	7,645	0,6821	0,7094	0,8904
B9	19,222	8,050	0,5599	0,4617	0,8968
B10	19,296	8,269	0,5617	0,4632	0,8976

EK 5 - Anket Verileri – Ortalama Değerler

Aşağıdaki hesaplamaların tümü Minitab yazılımı kullanılarak yapılmıştır.

Betimleyici İstatistikler: A01; A02; A03; A04; A05; A06; A07; A08; A09; A10

Değişken	N	Ortalama	SE Ort	StSap	Minimum	Maksimum
A01	66	2,348	0,171	1,387	0,000	5,000
A02	55	2,218	0,159	1,182	0,000	5,000
A03	43	2,488	0,209	1,369	0,000	5,000
A04	36	2,083	0,175	1,052	0,000	5,000
A05	41	2,512	0,198	1,267	0,000	5,000
A06	28	1,964	0,254	1,347	0,000	5,000
A07	36	2,278	0,213	1,279	0,000	5,000
A08	27	1,963	0,259	1,344	0,000	5,000
A09	24	2,000	0,241	1,180	0,000	4,000
A10	30	1,767	0,190	1,040	0,000	4,000

Betimleyici İstatistikler: A01; A02; A03; A04; A05; A06; A07; A08; A09; A10

Değişken	Yazılım kadrosu	N	Ortalama	SE Ort	StSap	Minimum	Maksimum
A01		1	2,0000	*	*	2,0000	2,0000
	1-9	27	1,778	0,209	1,086	0,000	4,000
	10-49	17	2,235	0,304	1,251	0,000	5,000
	100+	13	3,615	0,385	1,387	2,000	5,000
	50-99	8	2,500	0,535	1,512	1,000	5,000
A02		1	2,0000	*	*	2,0000	2,0000
	1-9	22	2,182	0,156	0,733	1,000	4,000
	10-49	19	1,789	0,292	1,273	0,000	5,000
	100+	10	3,000	0,471	1,491	1,000	5,000
	50-99	3	2,667	0,882	1,528	1,000	4,000
A03		0	*	*	*	*	*
	1-9	15	2,600	0,335	1,298	1,000	5,000
	10-49	15	2,000	0,324	1,254	0,000	5,000
	100+	8	3,375	0,420	1,188	2,000	5,000
	50-99	5	2,200	0,800	1,789	1,000	5,000
A04		0	*	*	*	*	*
	1-9	14	1,786	0,239	0,893	0,000	3,000
	10-49	12	1,917	0,193	0,669	1,000	3,000
	100+	6	3,000	0,577	1,414	1,000	5,000
	50-99	4	2,250	0,750	1,500	0,000	3,000
A05		1	2,0000	*	*	2,0000	2,0000
	1-9	13	2,385	0,385	1,387	0,000	5,000
	10-49	15	2,267	0,228	0,884	1,000	5,000
	100+	7	3,143	0,553	1,464	1,000	5,000
	50-99	5	2,800	0,800	1,789	1,000	5,000
A06		1	3,0000	*	*	3,0000	3,0000
	1-9	8	1,250	0,453	1,282	0,000	3,000
	10-49	10	1,900	0,348	1,101	0,000	3,000
	100+	5	3,200	0,583	1,304	2,000	5,000
	50-99	4	1,750	0,750	1,500	0,000	3,000
A07		1	2,0000	*	*	2,0000	2,0000
	1-9	10	1,800	0,359	1,135	0,000	4,000
	10-49	12	2,250	0,329	1,138	1,000	5,000
	100+	9	3,222	0,465	1,394	1,000	5,000
	50-99	4	1,500	0,500	1,000	1,000	3,000
A08		2	2,500	0,500	0,707	2,000	3,000

	1-9	8	1,625	0,532	1,506	0,000	4,000
	10-49	8	1,500	0,378	1,069	0,000	3,000
	100+	5	3,000	0,707	1,581	1,000	5,000
	50-99	4	2,000	0,577	1,155	1,000	3,000
A09		0	*	*	*	*	*
	1-9	9	1,778	0,434	1,302	0,000	4,000
	10-49	9	2,000	0,333	1,000	0,000	3,000
	100+	4	2,750	0,479	0,957	2,000	4,000
	50-99	2	1,50	1,50	2,12	0,00	3,00
A10		2	1,500	0,500	0,707	1,000	2,000
	1-9	9	1,222	0,278	0,833	0,000	2,000
	10-49	12	2,000	0,302	1,044	0,000	4,000
	100+	5	2,600	0,400	0,894	2,000	4,000
	50-99	2	1,00	1,00	1,41	0,00	2,00

Betimleyici İstatistikler: A01; A02; A03; A04; A05; A06; A07; A08; A09; A10

Değişken	Grup	N	Ortalama	SE Ort	StSap	Minimum	Maksimum
A01		1	2,0000	*	*	2,0000	2,0000
	İç Kullanım	23	2,087	0,294	1,411	0,000	5,000
	Paket yazılım	3	1,667	0,333	0,577	1,000	2,000
	Proje Bazlı	39	2,564	0,226	1,410	0,000	5,000
A02		1	2,0000	*	*	2,0000	2,0000
	İç Kullanım	18	2,000	0,302	1,283	0,000	5,000
	Paket yazılım	2	2,500	0,500	0,707	2,000	3,000
	Proje Bazlı	34	2,324	0,201	1,173	0,000	5,000
A03		0	*	*	*	*	*
	İç Kullanım	13	2,231	0,395	1,423	1,000	5,000
	Paket yazılım	2	2,00	1,00	1,41	1,00	3,00
	Proje Bazlı	28	2,643	0,258	1,367	0,000	5,000
A04		0	*	*	*	*	*
	İç Kullanım	11	1,818	0,296	0,982	1,000	4,000
	Paket yazılım	1	2,0000	*	*	2,0000	2,0000
	Proje Bazlı	24	2,208	0,225	1,103	0,000	5,000
A05		1	2,0000	*	*	2,0000	2,0000
	İç Kullanım	14	2,714	0,339	1,267	1,000	5,000
	Paket yazılım	1	3,0000	*	*	3,0000	3,0000
	Proje Bazlı	25	2,400	0,265	1,323	0,000	5,000
A06		1	3,0000	*	*	3,0000	3,0000
	İç Kullanım	8	1,250	0,453	1,282	0,000	3,000
	Paket yazılım	2	2,500	0,500	0,707	2,000	3,000
	Proje Bazlı	17	2,176	0,335	1,380	0,000	5,000
A07		1	2,0000	*	*	2,0000	2,0000
	İç Kullanım	12	2,333	0,396	1,371	1,000	5,000
	Paket yazılım	2	1,500	0,500	0,707	1,000	2,000
	Proje Bazlı	21	2,333	0,287	1,317	0,000	5,000
A08		2	2,500	0,500	0,707	2,000	3,000
	İç Kullanım	9	1,778	0,465	1,394	0,000	4,000
	Paket yazılım	2	2,00	1,00	1,41	1,00	3,00
	Proje Bazlı	14	2,000	0,392	1,468	0,000	5,000
A09		0	*	*	*	*	*
	İç Kullanım	7	2,000	0,655	1,732	0,000	4,000
	Paket yazılım	1	2,0000	*	*	2,0000	2,0000
	Proje Bazlı	16	2,000	0,242	0,966	0,000	3,000
A10		2	1,500	0,500	0,707	1,000	2,000
	İç Kullanım	8	1,625	0,420	1,188	0,000	4,000
	Paket yazılım	1	2,0000	*	*	2,0000	2,0000
	Proje Bazlı	19	1,842	0,245	1,068	0,000	4,000

Betimleyici İstatistikler: A01; A02; A03; A04; A05; A06; A07; A08; A09; A10

Değişken	Sektor	N	Ortalama	SE Ort	StSap	Minimum	Maksimum
A01		1	2,0000	*	*	2,0000	2,0000
	Endüstri	23	2,261	0,283	1,356	0,000	5,000
	Finans	13	3,000	0,358	1,291	1,000	5,000
	Hizmet	23	1,826	0,257	1,230	0,000	5,000
	Kamu	6	3,333	0,667	1,633	1,000	5,000
A02		1	2,0000	*	*	2,0000	2,0000
	Endüstri	20	2,350	0,264	1,182	0,000	5,000
	Finans	11	2,182	0,423	1,401	1,000	5,000
	Hizmet	17	1,941	0,264	1,088	0,000	5,000
	Kamu	6	2,667	0,494	1,211	1,000	4,000
A03		0	*	*	*	*	*
	Endüstri	19	2,421	0,279	1,216	1,000	5,000
	Finans	9	2,667	0,441	1,323	1,000	5,000
	Hizmet	10	1,900	0,482	1,524	0,000	5,000
	Kamu	5	3,600	0,600	1,342	2,000	5,000
A04		0	*	*	*	*	*
	Endüstri	16	2,000	0,204	0,816	0,000	3,000
	Finans	5	2,000	0,632	1,414	1,000	4,000
	Hizmet	11	1,818	0,296	0,982	0,000	3,000
	Kamu	4	3,250	0,629	1,258	2,000	5,000
A05		1	2,0000	*	*	2,0000	2,0000
	Endüstri	15	2,733	0,316	1,223	1,000	5,000
	Finans	7	2,429	0,571	1,512	1,000	5,000
	Hizmet	14	2,071	0,305	1,141	0,000	4,000
	Kamu	4	3,500	0,645	1,291	2,000	5,000
A06		1	3,0000	*	*	3,0000	3,0000
	Endüstri	13	2,000	0,376	1,354	0,000	4,000
	Finans	5	2,200	0,200	0,447	2,000	3,000
	Hizmet	6	0,667	0,333	0,816	0,000	2,000
	Kamu	3	3,667	0,667	1,155	3,000	5,000
A07		1	2,0000	*	*	2,0000	2,0000
	Endüstri	15	2,533	0,307	1,187	1,000	5,000
	Finans	8	2,500	0,500	1,414	1,000	5,000
	Hizmet	9	1,444	0,294	0,882	0,000	3,000
	Kamu	3	3,00	1,15	2,00	1,00	5,00
A08		2	2,500	0,500	0,707	2,000	3,000
	Endüstri	12	2,083	0,313	1,084	0,000	4,000
	Finans	4	2,250	0,750	1,500	1,000	4,000
	Hizmet	6	1,000	0,447	1,095	0,000	3,000
	Kamu	3	2,67	1,45	2,52	0,00	5,00
A09		0	*	*	*	*	*
	Endüstri	11	2,364	0,338	1,120	0,000	4,000
	Finans	5	2,200	0,490	1,095	1,000	4,000
	Hizmet	6	1,000	0,447	1,095	0,000	2,000
	Kamu	2	2,500	0,500	0,707	2,000	3,000
A10		2	1,500	0,500	0,707	1,000	2,000
	Endüstri	11	1,636	0,310	1,027	0,000	3,000
	Finans	5	2,600	0,600	1,342	1,000	4,000
	Hizmet	10	1,400	0,267	0,843	0,000	2,000
	Kamu	2	2,500	0,500	0,707	2,000	3,000

Betimleyici İstatistikler: A01; A02; A03; A04; A05; A06; A07; A08; A09; A10

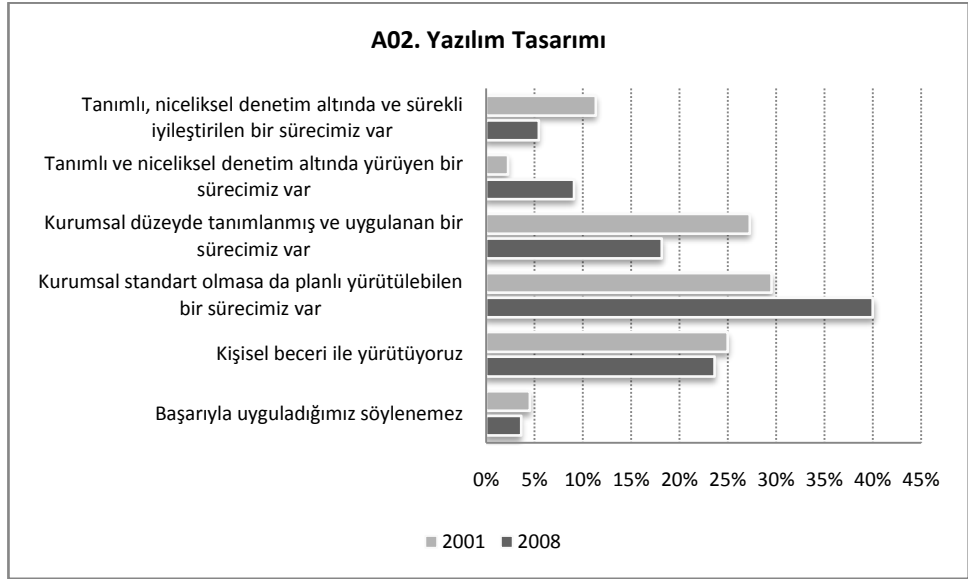
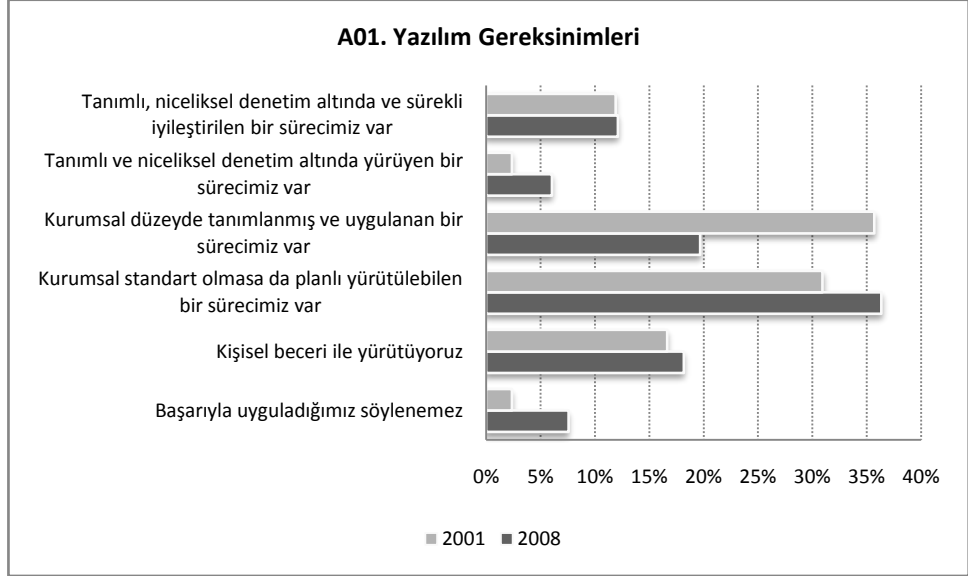
Değişken	Gorev	N	Ortalama	SE Ort	StSap	Minimum	Maksimum
A01		1	2,0000	*	*	2,0000	2,0000
	Diğer yöneticiler	47	2,383	0,201	1,376	0,000	5,000
	Yazılım kalite	4	3,250	0,629	1,258	2,000	5,000
	Yazılım mühendisi	14	2,000	0,392	1,468	0,000	5,000
A02		1	2,0000	*	*	2,0000	2,0000
	Diğer yöneticiler	36	2,194	0,182	1,091	0,000	5,000
	Yazılım kalite	3	2,667	0,882	1,528	1,000	4,000
	Yazılım mühendisi	15	2,200	0,368	1,424	1,000	5,000
A03		0	*	*	*	*	*
	Diğer yöneticiler	31	2,452	0,236	1,312	0,000	5,000
	Yazılım kalite	2	4,00	1,00	1,41	3,00	5,00
	Yazılım mühendisi	10	2,300	0,473	1,494	1,000	5,000
A04		0	*	*	*	*	*
	Diğer yöneticiler	28	2,036	0,174	0,922	0,000	4,000
	Yazılım kalite	3	3,00	1,15	2,00	1,00	5,00
	Yazılım mühendisi	5	1,800	0,490	1,095	0,000	3,000
A05		1	2,0000	*	*	2,0000	2,0000
	Diğer yöneticiler	33	2,424	0,209	1,200	0,000	5,000
	Yazılım kalite	2	3,00	1,00	1,41	2,00	4,00
	Yazılım mühendisi	5	3,000	0,837	1,871	1,000	5,000
A06		1	3,0000	*	*	3,0000	3,0000
	Diğer yöneticiler	23	1,783	0,251	1,204	0,000	4,000
	Yazılım kalite	2	4,00	1,00	1,41	3,00	5,00
	Yazılım mühendisi	2	1,50	1,50	2,12	0,00	3,00
A07		1	2,0000	*	*	2,0000	2,0000
	Diğer yöneticiler	26	2,269	0,245	1,251	0,000	5,000
	Yazılım kalite	3	3,00	1,15	2,00	1,00	5,00
	Yazılım mühendisi	6	2,000	0,516	1,265	1,000	4,000
A08		2	2,500	0,500	0,707	2,000	3,000
	Diğer yöneticiler	19	1,632	0,288	1,257	0,000	4,000
	Yazılım kalite	3	2,67	1,20	2,08	1,00	5,00
	Yazılım mühendisi	3	3,000	0,577	1,000	2,000	4,000
A09		0	*	*	*	*	*
	Diğer yöneticiler	20	1,750	0,250	1,118	0,000	4,000
	Yazılım kalite	2	3,0000	0,000000	0,000000	3,0000	3,0000
	Yazılım mühendisi	2	3,500	0,500	0,707	3,000	4,000
A10		2	1,500	0,500	0,707	1,000	2,000
	Diğer yöneticiler	22	1,682	0,212	0,995	0,000	4,000
	Yazılım kalite	4	2,750	0,479	0,957	2,000	4,000
	Yazılım mühendisi	2	1,00	1,00	1,41	0,00	2,00

Betimleyici İstatistikler: A01; A02; A03; A04; A05; A06; A07; A08; A09; A10

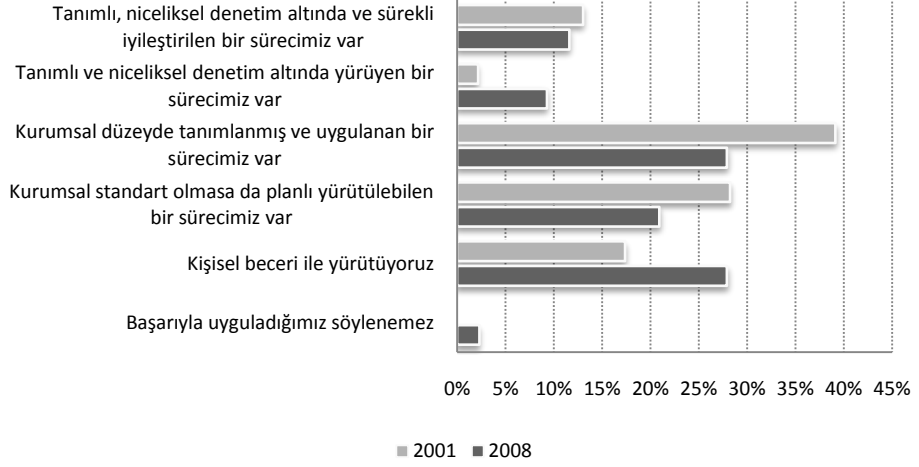
Değişken	Deneyim	N	Ortalama	SE Ort	StSap	Minimum	Maksimum
A01		1	2,0000	*	*	2,0000	2,0000
	0-1 yıl	0	*	*	*	*	*
	10-19 yıl	25	2,480	0,295	1,475	0,000	5,000
	2-4 yıl	10	2,200	0,359	1,135	1,000	5,000
	20+ yıl	10	2,800	0,512	1,619	0,000	5,000
	5-9 yıl	20	2,050	0,294	1,317	0,000	5,000
A02		1	2,0000	*	*	2,0000	2,0000
	0-1 yıl	1	1,0000	*	*	1,0000	1,0000
	10-19 yıl	20	2,050	0,294	1,317	0,000	4,000
	2-4 yıl	8	2,250	0,412	1,165	1,000	5,000
	20+ yıl	10	2,500	0,401	1,269	1,000	5,000
	5-9 yıl	15	2,333	0,270	1,047	1,000	5,000

A03		0	*	*	*	*	*
	0-1 y1l	1	1,0000	*	*	1,0000	1,0000
	10-19 y1l	18	2,444	0,345	1,464	0,000	5,000
	2-4 y1l	4	1,750	0,250	0,500	1,000	2,000
	20+ y1l	8	3,125	0,515	1,458	1,000	5,000
	5-9 y1l	12	2,500	0,379	1,314	1,000	5,000
A04		0	*	*	*	*	*
	0-1 y1l	0	*	*	*	*	*
	10-19 y1l	15	2,267	0,330	1,280	0,000	5,000
	2-4 y1l	2	2,0000	0,000000	0,000000	2,0000	2,0000
	20+ y1l	9	2,000	0,289	0,866	1,000	3,000
	5-9 y1l	10	1,900	0,314	0,994	0,000	3,000
A05		1	2,0000	*	*	2,0000	2,0000
	0-1 y1l	0	*	*	*	*	*
	10-19 y1l	17	2,824	0,324	1,334	1,000	5,000
	2-4 y1l	2	2,0000	0,000000	0,000000	2,0000	2,0000
	20+ y1l	9	2,889	0,389	1,167	1,000	5,000
	5-9 y1l	12	1,917	0,358	1,240	0,000	5,000
A06		1	3,0000	*	*	3,0000	3,0000
	0-1 y1l	0	*	*	*	*	*
	10-19 y1l	14	1,929	0,425	1,592	0,000	5,000
	2-4 y1l	0	*	*	*	*	*
	20+ y1l	5	1,800	0,374	0,837	1,000	3,000
	5-9 y1l	8	2,000	0,463	1,309	0,000	3,000
A07		1	2,0000	*	*	2,0000	2,0000
	0-1 y1l	1	1,0000	*	*	1,0000	1,0000
	10-19 y1l	17	2,765	0,359	1,480	1,000	5,000
	2-4 y1l	3	2,000	0,577	1,000	1,000	3,000
	20+ y1l	5	1,800	0,374	0,837	1,000	3,000
	5-9 y1l	9	1,889	0,351	1,054	0,000	3,000
A08		2	2,500	0,500	0,707	2,000	3,000
	0-1 y1l	0	*	*	*	*	*
	10-19 y1l	12	2,333	0,432	1,497	0,000	5,000
	2-4 y1l	1	2,0000	*	*	2,0000	2,0000
	20+ y1l	5	1,600	0,600	1,342	0,000	3,000
	5-9 y1l	7	1,429	0,481	1,272	0,000	3,000
A09		0	*	*	*	*	*
	0-1 y1l	0	*	*	*	*	*
	10-19 y1l	14	2,214	0,334	1,251	0,000	4,000
	2-4 y1l	0	*	*	*	*	*
	20+ y1l	4	1,500	0,645	1,291	0,000	3,000
	5-9 y1l	6	1,833	0,401	0,983	0,000	3,000
A10		2	1,500	0,500	0,707	1,000	2,000
	0-1 y1l	0	*	*	*	*	*
	10-19 y1l	13	1,769	0,323	1,166	0,000	4,000
	2-4 y1l	1	2,0000	*	*	2,0000	2,0000
	20+ y1l	6	1,833	0,307	0,753	1,000	3,000
	5-9 y1l	8	1,750	0,453	1,282	0,000	4,000

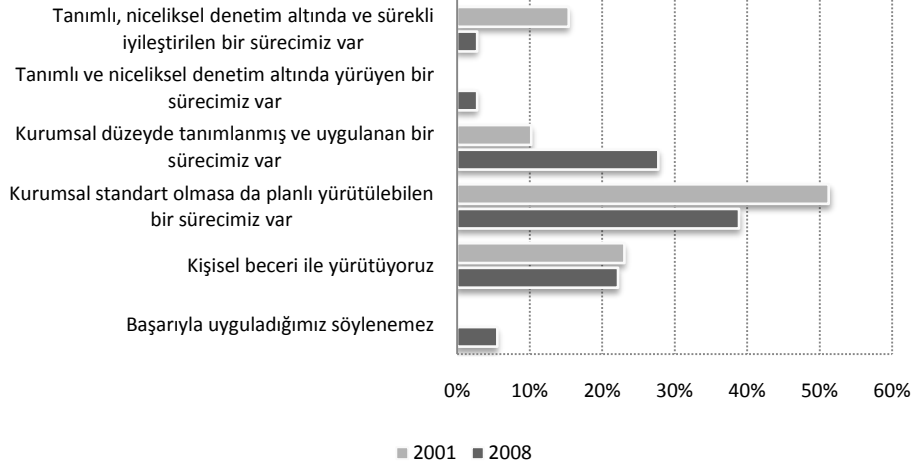
EK 6 - Süreç Özetleri



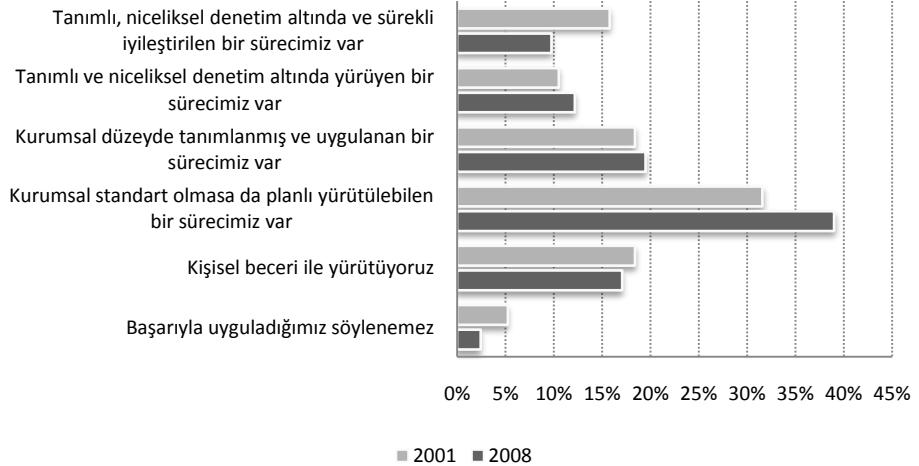
A03. Yazılım Gerçekleştirme



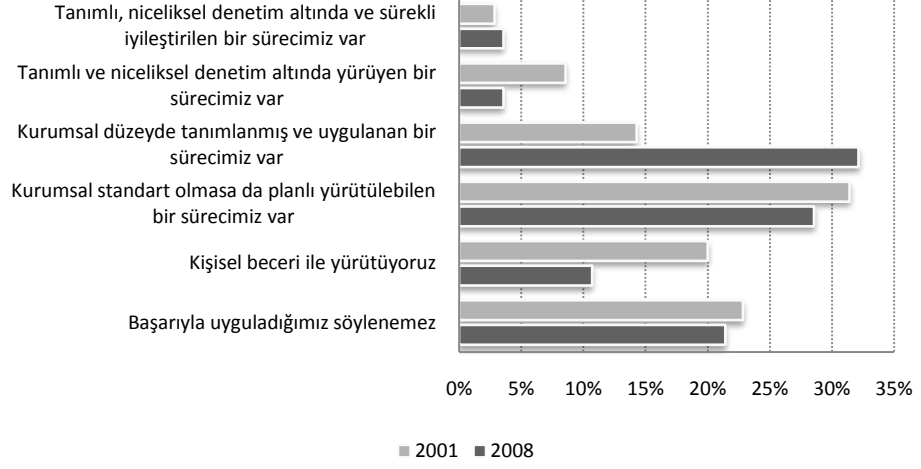
A04. Yazılım Sınama



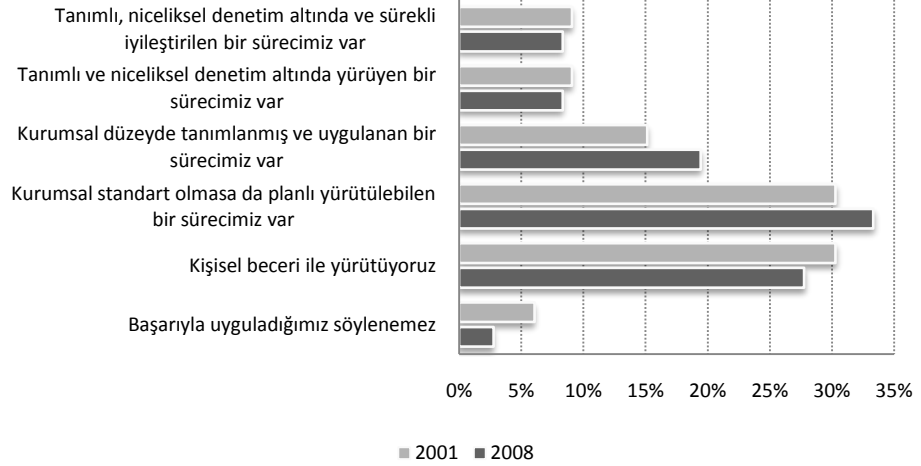
A05. Yazılım Bakımı



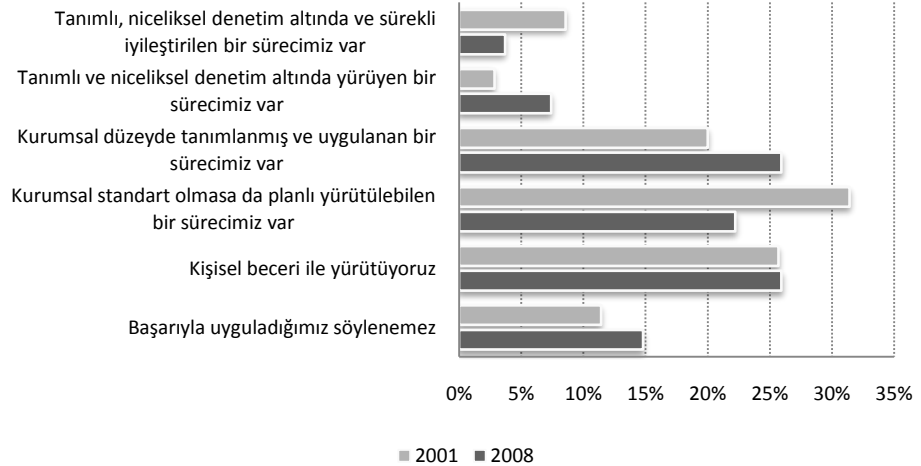
A06. Yazılım Konfigürasyon Yönetimi



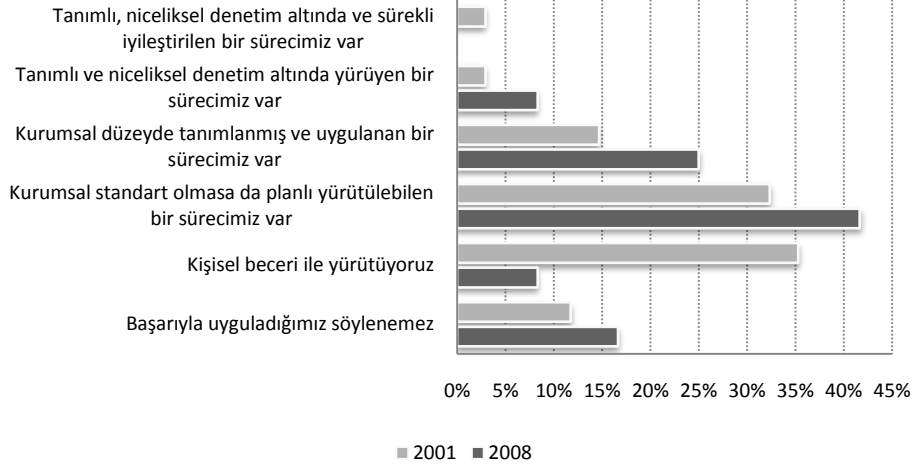
A07. Yazılım Mühendisliği Yönetimi



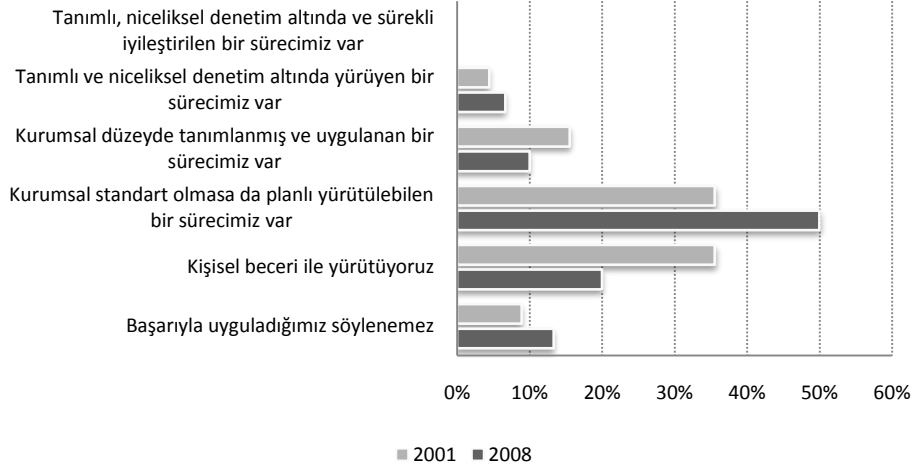
A08. Yazılım Süreç Mühendisliği



A09. Yazılım Mühendisliği Araç ve Yöntemleri



A10. Yazılım Kalitesi



KAYNAKÇA

Kitaplar

Hair, J.F., Anderson, R.E., Tatham, R.L. And Black, W.C., 1998. *Multivariate data analysis*. 5th ed. Prentice Hall, Upper Saddle River, NJ.

Süreli Yayınlar

Charalambos L. Iacovou, Robbie Nakatsu, 2008. Industry A risk profile of offshore-outsourced development projects, *Communications of the ACM*, Volume 51 Issue 6, Publisher: ACM, June.

D. Gefen, M. Zviran, 2006. N. Elman, What Can be Learned from CMMI Failures, *Communications of AIS*, Volume 17, Article 36.

Glass R., 2000. Talk about a software crisis - not!, *Journal of Systems and Software*, 55(1): 1-2, 2000.

Soumitra Dutta, Luk N. Vanwassenhove, Selvan Kulandaiswamy, 1998. Benchmarking European Software Management Practices, *Communications of the ACM*, Volume 41 Issue 6, Publisher: ACM, June.

Wilkie F.G, McFall D. and Mc Caffery F., 2005. An Evaluation of CMMI Process Areas for Small to Medium Sized Software Development Organisations, *Software Process: Improvement and Practice Journal*, Issue 10: 2, June 2005, Pages 189-201 - Wiley Publishers.

IT Week, June 14, 2004. Firms take outsourcing risks.

Diğer Yayınlar

European Software Institute, 1997. Software Best Practices Questionnaire Analysis of Results [online], www2.umassd.edu/swpi/esi/tr-sbpqaor3.pdf . [cited Dec, 1997].

Junyoung Park, Hoyeon Ryu, Ho-Jin Choi, Dong-Kuk Ryu, 2008. A Survey on Software Test Maturity in Korean Defense, India Software Engineering Conference, *Proceedings of the 1st conference on India software engineering conference*, Hyderabad, India, Pages 149-150, ISBN:978-1-59593-917-3.

Raymund Sison, Stanislaw Jarzabek, Ow Siew Hock, Wanchai Rivepiboon, Nguyen Nam Hai, 2006. Software practices in five ASEAN countries: an exploratory study, Pages: 628-631, ISBN:1-59593-375-1.

McCarthy, J.C. The State of Development of the IT Services Global Delivery Model Forrester Research Inc., Cambridge, MA, 2007.

Meriç Aykol, Fuat İnce, 2008. Yazılım Süreç Değerlendirme Modeli olarak CMMI, *Türkiye Bilişim Derneği 2. İstanbul Bilişim Kongresi*, 3-4 Haziran, ss. 2-8.

Monvorath Phongpaibul, Barry Boehm, 2005. Improving quality through software process improvement in Thailand: initial analysis, Pages: 1-6, ISBN:1-59593-122-8.

Peter Kulik, Catherine Weber, 2001. Software Risk Management Practices by KLCI, August, 2001.

Seçkin İkiz, Meriç Aykol, Turgay Aytaç, 2003. Turkey's Software Industry Survey Paper, *3.International SPICE Conference (Variety of International Experience – Holland/Noordwijk)* – March.

Carnegie Mellon University, 2008. CMMI® Version 1.2 and Beyond, A Tutorial by Carnegie Mellon University [online] www.sei.cmu.edu/programs/acquisition-support/presentations/cmmiv2beyond.pdf [cited March, 2008].

Carnegie Mellon University, 2008. Process Maturity © 2008 [online] www.sei.cmu.edu/appraisal-program/profile/profile.html, [cited March, 2008].

Guide to the Software Engineering Body of Knowledge (SWEBOK) [online], www.swebok.org , IEEE and ACM, [cited 2004].

International Standards Organization, 1998. ISO/IEC TR15504:1998(E)

Software Engineering Institute, 2006. CMMI® for Development, Version 1.2 [online] www.sei.cmu.edu/publications/documents/06.reports/06tr008.html, [cited 2006].

ÖZGEÇMİŞ

Adı Soyadı : Meriç Merih Aykol

Sürekli Adresi : 37. Ada İnci-1 D:14 Ataşehir / İstanbul / Türkiye

Doğum Yeri ve Yılı : İstanbul – 1967

Yabancı Dili : İngilizce

İlk Öğretim : İstanbul – Basıncöy İlkokulu – 1978

Orta Öğretim : Gölcük – Barbaros Hayrettin Lisesi – 1984

Lisans : İTÜ – Kontrol ve Bilgisayar Mühendisliği – 1989

Yüksek Lisans : Bahçeşehir Üniversitesi – 2009

Enstitü Adı : Fen Bilimleri Enstitüsü

Program Adı : Bilgi Teknolojileri

Yayımları :

- Kırılıdoğ, M., Aykol, M., Gülseçen, S. (forthcoming) “Interpersonal Communication and Gender in the ICT Profession”, IEEE Technology & Society Magazine (SCI)
- TBD Bilişim Kurultayı – Yazılım Süreç Değerlendirme Modeli olarak CMMI konulu bildiri – Haziran 2008
- IPYD Ulusal Proje Yönetimi Kongresi – BT Proje Yönetimi konulu bildiri – Nisan 2006
- Akademik Bilişim – Dünya’da Yazılım Endüstrisi’nin Gelişimi konulu bildiri – Şubat 2004
- TBD Bilişim Kurultayı – Yazılımda Öncü Ülkeler konulu bildiri – Eylül 2003
- Uluslararası SPICE Konferansı - KalDer Yazılım Sektörü Anketi konulu bildiri (Hollanda/Noordwijk)- Mart 2003
- IPYD Ulusal Proje Yönetimi Kongresi – Uluslararası Yazılım Kalitesi Belgeleri ve Türkiye’deki Durum konulu bildiri – Nisan 2003

- Ulusal Kalite Kongresi Türkiye’de Yazılım Mühendisliği ve Yazılım Yönetimi Uygulamaları konusunda grup adına bildiri – 2002
- Bilişim Zirvesi 2002 - Türkiye’de Yazılım Mühendisliği ve Yazılım Yönetimi Uygulamaları konusunda grup adına bildiri – 2002
- Bilişim Zirvesi 2002 - ISO/IEC TR-15504 SPICE (Yazılım Süreç İyileştirme ve Yeterlilik Belirleme) Metodunun Diğer Bilgi Teknolojisi Hizmet Alanlarına Uyarlanması - KoçSistem Deneyimi konulu bildiri
- Ulusal Kalite Kongresi Yazılım TKY Uygulamaları konulu bildiri - 2001

Çalışma Hayatı :

MERSİS Bilgi Teknolojileri Danışmanlık Ltd. Danışman	2008 –
KOÇSİSTEM A.Ş. Kalite ve Bilgi Sistemleri Grup Yöneticisi	1996 – 2008
YKB – BİLPA A.Ş. Proje Lideri	1990 – 1996
T.EMLAK BANKASI Programcı	1988 – 1990