

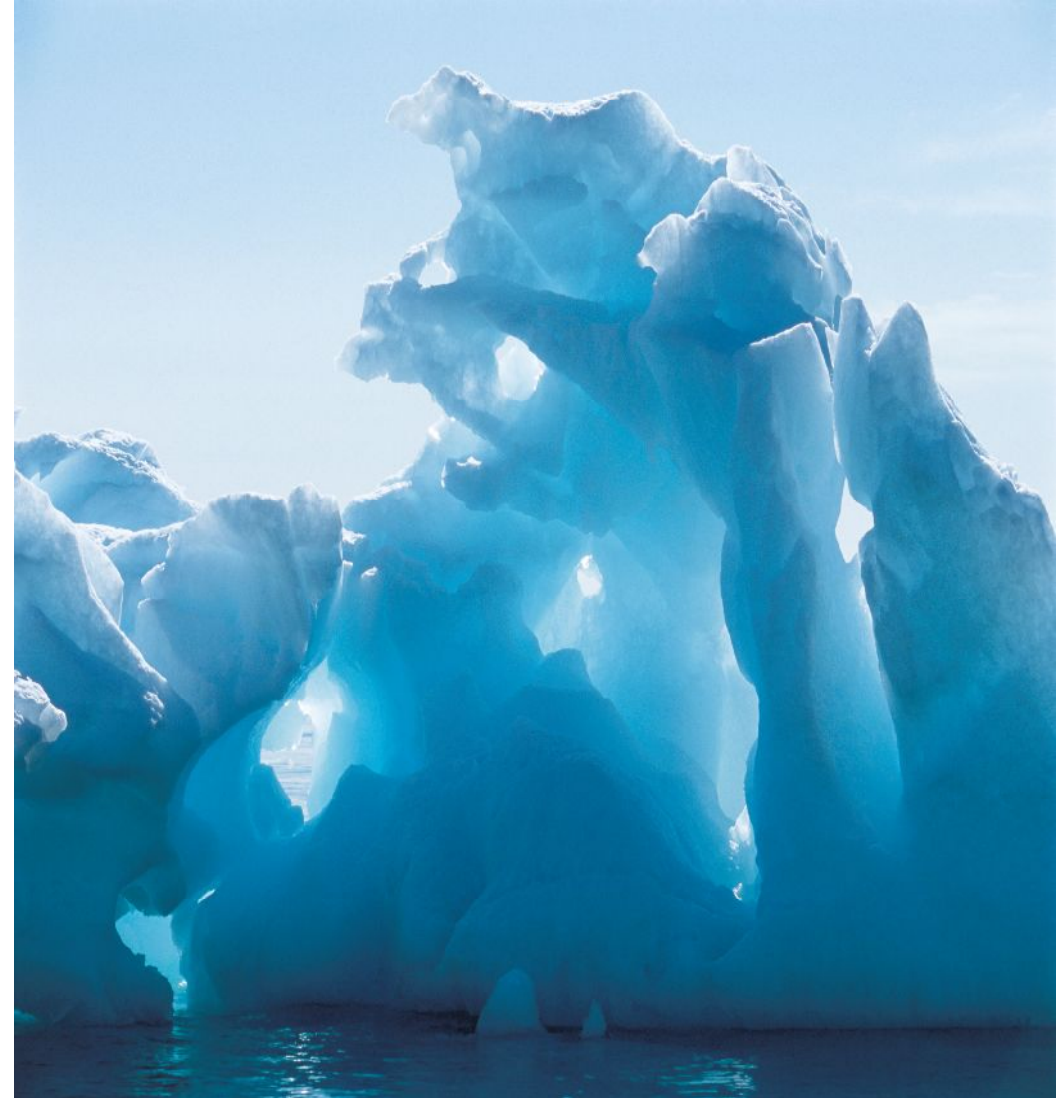
# BTRisk



## YAZILIM GÜVENLİĞİ

*Güvenliği Yazılımla  
Bütünleştirmek*

Fatih Emiral CISSP, CISA, CEH, CIA  
28 Şubat 2009

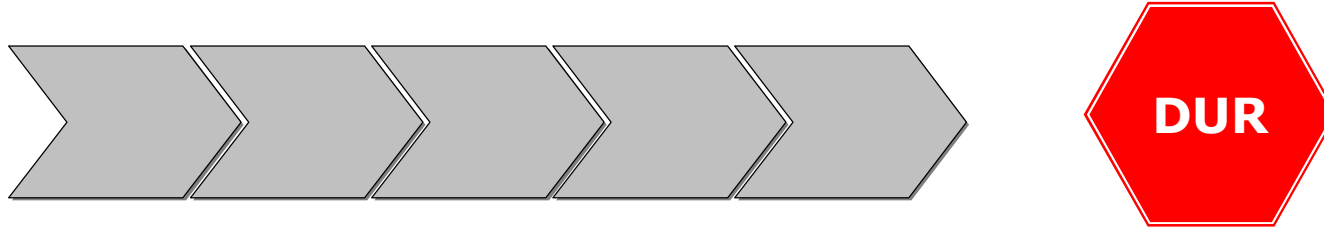


# İçindekiler

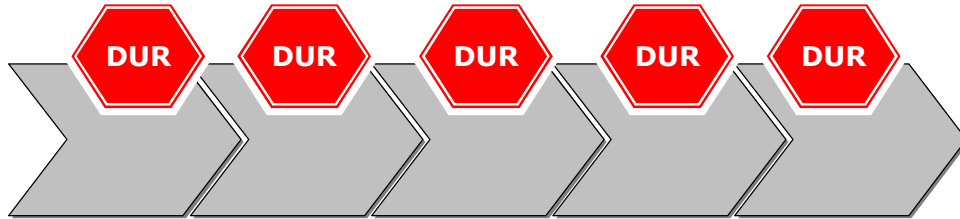
- I. Yazılım Güvenliđi Nedir
- II. Yazılım Güvenliđi Alanında Başarılı Bir Örnek
- III. Yazılım Güvenliđi Alanındaki Çalışmalar
- IV. (Software Security) Touchpoints
- V. CLASP
- VI. SDL
- VII. Yazılım Güvenliđi Metodolojileri ile BT Kontrol Çerçevesi Arasında (Kaba) Analoji
- VIII. Genel İzlenim
- IX. Kimler Güvenliđi Yazılım Sürecine (Başarılı Biçimde) Entegre Edebilir

## I. Yazılım Güvenliđi Nedir

- Fonksiyonel uygulama testinin uygulama kalitesindeki yeri



- Sola itme



- Uygulama güvenliđi için bir gümüş mermi var mı?



## II. Yazılım Güvenliği Alanında Başarılı Bir Örnek

### YIL: 2001 / 2002

- "Gartner Recommends Against Microsoft IIS" (eWeek 2001)
- "IT Bugs Out Over IIS Security" (eWeek 2001)
- "Microsoft's security woes" (CNET 2002)
- "Microsoft's security push lacks oomph" (CNET 2002)



### ... YIL: 2005 / 2006

- "We actually consider Microsoft to be leading the software [industry] now in improvements in their security development life cycle." (CRN 2006)
- "Oltsik gives Microsoft credit for implementing industry-leading security development processes saying, 'Microsoft is ahead of the pack in this area.'" (Enterprise Strategy Group 2006)
- "Overall, security bulletins from Microsoft have decreased in recent years" (eWeek 2005)
- "Microsoft: Software Security Trendsetter?" (eWeek 2005)

Kaynak: The Security Development Lifecycle: SDL: A Process for Developing Demonstrably More Secure Software  
By Michael Howard, Steve Lipner

## II. Yazılım Güvenliği Alanında Başarılı Bir Örnek 2002 Yılında Ne Oldu?

From: Bill Gates  
Sent: Tuesday, January 15, 2002 5:22 PM  
To: Microsoft and Subsidiaries: All FTE  
Subject: Trustworthy computing

...

However, even more important than any of these new capabilities is the fact that it is designed from the ground up to deliver Trustworthy Computing. What I mean by this is that customers will always be able to rely on these systems to be available and to secure their information. **Trustworthy Computing is computing that is as available, reliable and secure as electricity, water services and telephony.**

...

**I've spent the past few months** working with Craig Mundie's group and others across the company to define what achieving Trustworthy Computing will entail, and to focus our efforts on building trust into every one of our products and services.

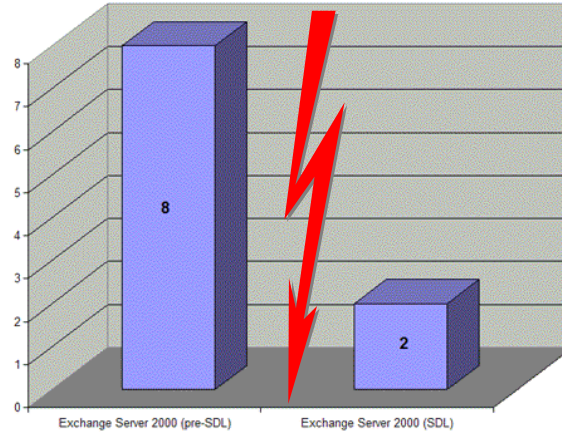
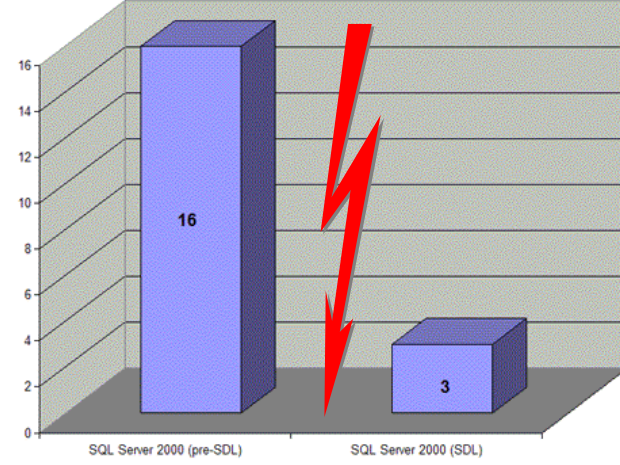
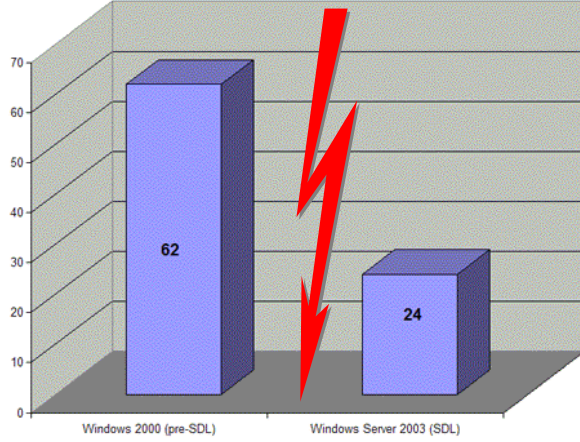
...

So now, when we face a choice between adding features and resolving security issues, **we need to choose security.** Our products should emphasize security right out of the box, and we must constantly refine and improve that security as threats evolve.

...

Kaynak: Bu e-posta internet üzerinde pek çok kaynaktan elde edilebilir.

## II. Yazılım Güvenliđi Alanında Başarılı Bir Örnek Sonuçlar

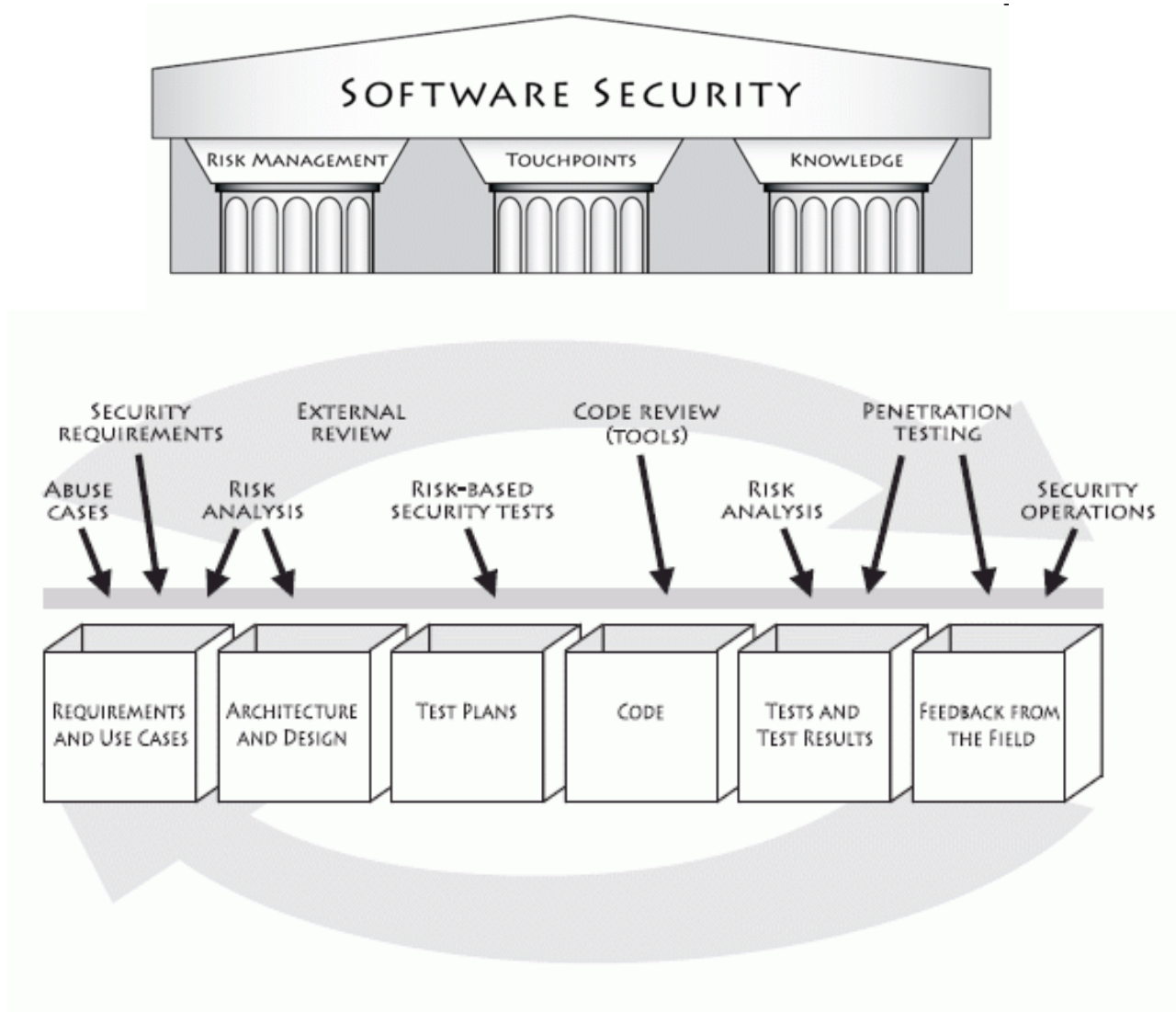


Kaynak: <http://msdn.microsoft.com/en-us/library/ms995349.aspx> (Kritik ve önemli açıklık duyuru sayıları)

### III. Yazılım Güvenliği Alanındaki Çalışmalar

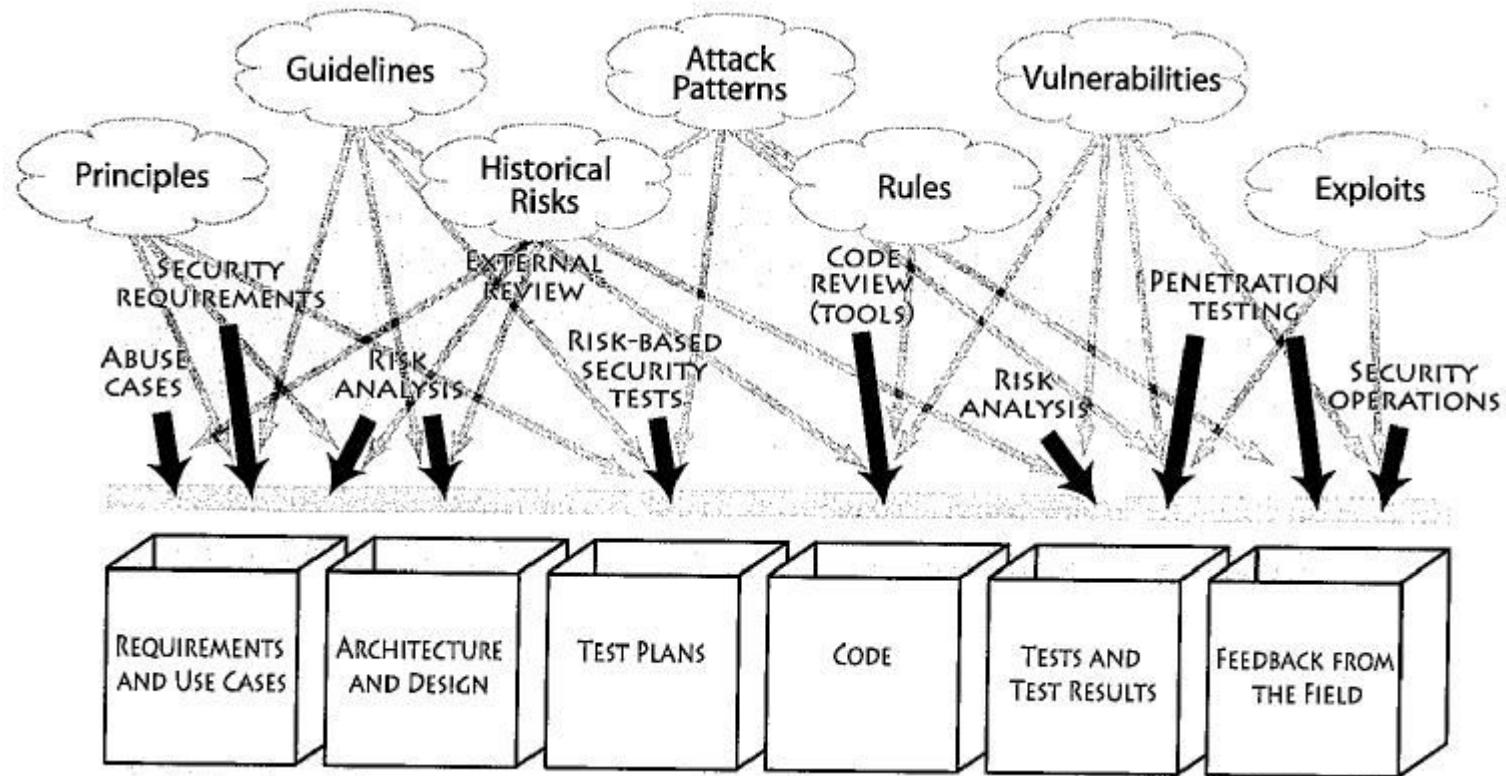
- CLASP (Comprehensive, Lightweight Application Security Process)
  - Secure Software tarafından geliştirildi ve OWASP projeleri arasına katıldı (**2006**).
  - **2007** yılında ciddi oranda değişikliğe uğradı.
  - Secure Software Fortify Software tarafından 2007 yılında satın alındı.
- SDL (Secure Development Lifecycle)
  - Microsoft Security Technology Unit tarafından geliştirilmiştir - The Security Development Lifecycle: SDL: A Process for Developing Demonstrably More Secure Software, Michael Howard, Steve Lipner, **2006**
- Touch Points
  - Gary McGraw tarafından geliştirildi – Software Security, Building Security In, Gary McGraw, **2006** - Not: Kitabın önsözünü Dan Geer yazmıştır.
  - Gary McGraw'ın Cigital isimli bir firması ve Silver Bullet Security Podcast'i bulunmaktadır.
  - McGraw aynı zamanda Fortify Software şirketinde Teknik Yönlendirme Komitesi üyesidir.

## IV. (Software Security) Touchpoints



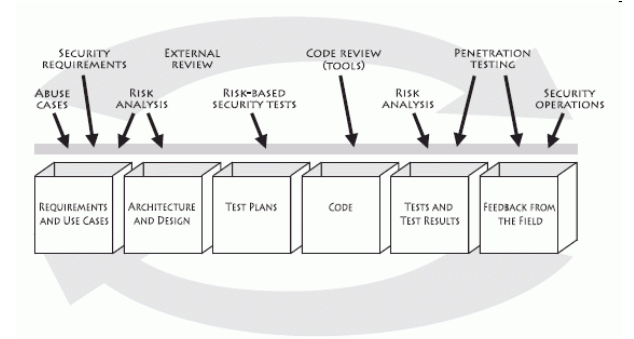
Kaynak: <http://www.swsec.com/resources/touchpoints/>

## IV. (Software Security) Touchpoints



Kaynak: Software Security, Building Security In, Gary McGraw, 2006

## IV. (Software Security) Touchpoints



### • Olumlu Yönler:

- Pedagojik olarak en başarılı anlatıma sahip kavramsal olarak (ilk okunacak kaynak olabilir)
- Yazılım Geliştirme Yaşam Döngüsü (YGYD) bağımsız (en esnek metod)
- 7 Touchpoints içinde tanımlanan aktiviteler için bir etkililik sıralaması öneriyor (ancak bu yaklaşım yazarın kurumsal ilişkileri nedeniyle biraz yanlı olabilir)

### • Olumsuz Yönler:

- Kavramsal anlatım olarak başarılı olmasına rağmen diğer metodlara nazaran kaynak ve netlik açısından geride kalıyor
- Risk Yönetimi'ni yazılım güvenliğini destekleyen 3 kolondan biri olarak tanımlamakla birlikte 7 Touchpoints'in içinde yer alan aktivitelerle bağlantısını net bir biçimde kuramıyor

### • Diğer Konular:

- Yazar yazılım güvenliği ile ilgili aktiviteleri 7 adetle (pratiklik açısından) bilinçli olarak sınırlandığını ve geliştirdiği metodun YGYD'den bağımsız olması için çalıştığını açıkça ifade ediyor.
- Yazarın uzmanlık alanı kitaba da yansımış, dolayısı ile iyi bir kod gözden geçirme aracı seçmek için gerekli ön bilgiyi ve bu araçların geçirdiği evrime ilişkin bilgiyi sağlıyor.

## V. CLASP

- CLASP içeriđi:
  - Kavram Bakışı
  - Rol Tabanlı Bakış
  - Aktivite Deđerlendirme Bakışı
  - Aktivite Uygulama Bakışı
  - Açıklıklar Bakışı
  - Açıklık Kullanma Durumları
  - CLASP Kaynakları
  - Ve diđer dokümanlar (uygulama güvenlik sözlüđü, güvenlik prensipleri ve geliştirme aşamasındaki dokümanlar)

## V. CLASP

- Olumlu Yönler:

- Aktivite olarak en detaylı metod, ayrıca YGYD'nü en geniş anlamda ele alıyor. Örneğin kod imzalama, operasyonel ortamı tanımlama gibi aktiviteler kapsamında.
- Kapsamının genişliği ve genelliği (yani SDL gibi bir organizasyona özel olmaması) nedeniyle iyi bir güvenli YGYD denetimi aracı olabilir. Aktivite Değerlendirme dokümanı aktivitenin yapılması gereken sıklık, yerine getirilmemesi durumunda doğacak risklere ilişkin de bilgi vererek denetim çalışmasını destekliyor.
- Rol odaklı olması uygulamak isteyen kurumlar için organizasyonel rolleri belirleyebilmek için iyi bir özellik
- Güncellemek zor olsa da yazılım açıklık türleri için bir veritabanına ve açıklık kullanım durumlarına sahip
- Yeni uygulamalar ve eski uygulamalar için önerdiği aktivite listeleri mevcut

## V. CLASP

- Olumsuz Yönler:
  - Metod parçaları (kavram, rol, aktivite, açıklık bakış açıları) arasındaki bağlantı yapılmaya çalışılmış olsa da çok net değil. Bu nedenle kavranması zor bir metod.
  - Esnekmiş gibi görünmekle birlikte çok detaylı olması nedeniyle (örneğin rol ve aktivite sayısı olarak) aslında kuruma özelleştirme konusunda daha fazla çalışma ihtiyacı var.
- Diğer Konular:
  - Rol odaklı bir metod, yani aktiviteler ilk olarak rollerle haritalanmış durumda, bu şekilde esnekliği sağlamak hedeflenmiş

## VI. SDL

- Stage 0: Education and Awareness
- Stage 1: Project Inception
- Stage 2: Define and Follow Design Best Practices
- Stage 3: Product Risk Assessment
- Stage 4: Risk Analysis
- Stage 5: Creating Security Documents, Tools, and Best Practices for Customers
- Stage 6: Secure Coding Policies
- Stage 7: Secure Testing Policies
- Stage 8: The Security Push
- Stage 9: The Final Security Review
- Stage 10: Security Response Planning
- Stage 11: Product Release
- Stage 12: Security Response Execution

## VI. SDL

- Olumlu Yönler:

- Kendi içinde çok tutarlı ve çok net bir metodoloji
- Microsoft'un dokümantasyonu içinde pratik uyarı ve etkili uygulama yöntemleri mevcut
- Kaynak olarak 3 metod arasında iyi konumda, ayrıca jenerik tehdit ağaç yapıları da mevcut
- Organizasyonel rolleri de net ve detaylı olarak açıklayarak güvenli YGYD açısından organizasyonun önemini net biçimde ifade ediyor

- Olumsuz Yönler:

- Genel ve her kuruma yönelik bir metod değil, doğrudan Microsoft'un ihtiyaçlarına yönelik olarak geliştirilmiş
- Kaynaklar da yukarıda sayılan nedenle genellikle Microsoft ürünlerine yönelik (örneğin gerekli araç ve derleyici opsiyonları Microsoft teknolojilerine özel)

- Diğer Konular:

- Yazılım güvenliği için süreç anlamında dahi bir gümüş mermi olmadığı çok net bir örneği. Microsoft için işe yaradığı kanıtlanmış ayrıca çok iyi dokümante edilmiş bir metodoloji olmakla birlikte her çözümün her kuruma uymayacağı anlaşılabilirliği için iyi bir kaynak.

## VII. Yazılım Güvenliđi Metodolojileri ile BT Kontrol ereveseleri Arasında (Kaba) Analoji

**Yazılım Güvenliđi  
Metodolojisi (veya  
yaklaşımı)**

**BT Kontrol erevesesi**

**Benzerlik Nedeni**

**SDL**

**ITIL**

Herhangi bir yaşam döngüsüne uysun diye deđil belli ve sabit bir şekilde tasarlanmış. Kendi içinde çok tutarlı ancak herkese uyacak biçimde esnek deđil.

**CLASP**

**Cobit**

Uygulama yaşam döngüsünü çok geniş anlamda ele alan ve güvenli uygulama geliştirme kontrolleri için denetim aracı olarak kullanılabilen bir araç.

**Touchpoints**

**ISO27001**

Kontrol uygulamalarında ihtiyaca yönelik olarak esnek.

## VIII. Genel İzlenim

- Organizasyonel yapılanmanın güvenli YGYD'nü destekler nitelikte olması
- Hem uygulama geliştirme ekiplerinin hem de merkezi güvenlik organizasyonunun sürekli eğitim programına tabi tutulması
- Uygulanan güvenli YGYD'nün hem değişen tehditlere göre yenilenebilmesi hem de değişen tehditleri öngören yapıda oluşturulması
- Risk analizi (Microsoft terimiyle tehdit modellemesi)'nin güvenli YGYD'deki önemi, daha sonra gelen güvenlik testleri ve kontrollerin geliştirilmesine katkısı
- Ve tabi yönetimin desteğinin önemi

## IX. Kimler Güvenliđi Yazılım Sürecine (Başarılı Biçimde) Entegre Edebilir

- Yazılımın kurum gelirlerine etkisinin (olumlu ve olumsuz yönde) büyüklüğü
- Yönetimin problemi kavraması ve desteklemesi
- Mevcut olgunluk düzeyi:
  - Mevcut yazılım kalitesi süreçleri
  - Güvenlik yetkinliđi (organizasyonel yapılanma, ekip üyelerinin becerileri)
  - Ağ ve uygulama geliştirme ve operasyon ekipleri arasındaki uyum
  - Teknik ekipler ile güvenlik yönetimi arasındaki uyum

# BTRisk