

The background features abstract, overlapping geometric shapes in various shades of blue, ranging from light sky blue to deep navy blue. The shapes are primarily triangles and polygons, creating a dynamic, modern aesthetic. The text is centered in a clean, sans-serif font.

ISO/IEC/IEEE Standardı 29148

- 2011 yılında, üç uluslararası standardizasyon kuruluşu, Uluslararası Standartlar Organizasyonu (ISO), Uluslararası Elektroteknik Komisyonu (IEC) ve Elektrik ve Elektronik Mühendisleri Enstitüsü (IEEE) birlikte ISO/IEC/IEEE Standardı 29148'i yayınladı.

- ▶ Bu belge;
- ▶ Yaşam döngüsü boyunca sistemler ve yazılım ürünleri için mühendislik faaliyetlerinde uygulanan gerekli süreçleri belirtir.
- ▶ ISO/IEC/IEEE 15288 ve ISO/IEC/IEEE 12207'de açıklanan gereksinimler ve gereksinimlerle ilgili süreçlerin uygulanması için kılavuzluk sağlar.
- ▶ Gereksinim süreçlerinin uygulanması yoluyla üretilen gerekli bilgi öğelerini belirtir.
- ▶ Gerekli bilgi öğelerinin gerekli içeriğini belirtir.

- ▶ 29148 yönergelerine uymanın çeşitli faydaları vardır. İlk olarak, gereksinimler belgesinin organizasyonu için basit bir çerçeve sağlarlar. En önemlisi, ISO/IEC/IEEE 29148, SRS'nin işlevsel ve işlevsel olmayan gereksinimlerini organize etmek için rehberlik sağlar.

İşlevsel Olmayan Gereksinimlerin Temsil Edilmesine İlişkin Öneriler

- ▶ IEEE 29148, aşağıdakileri içeren çeşitli işlevsel olmayan gereksinimleri açıklar:
 - ▶ ■ Harici arayüzler
 - ▶ ■ Kullanılabilirlik
 - ▶ ■ Performans
 - ▶ ■ Mantıksal veri tabanı konuları
 - ▶ ■ Tasarım kısıtlamaları
 - ▶ ■ Standartlara uygunluk
 - ▶ ■ Yazılım sistem özniteliği gereksinimleri

- ▶ Harici arayüz gereksinimleri, aşağıdakiler dahil olmak üzere çeşitli şekillerde organize edilebilir:
 - ▶ ■ Öğenin adı
 - ▶ ■ Amacın tanımı
 - ▶ ■ Girdi kaynağı veya çıktı hedefi
 - ▶ ■ Geçerli aralık, doğruluk ve/veya tolerans
 - ▶ ■ Zamanlama
 - ▶ ■ Diğer girdi/çıktılarla ilişkiler
 - ▶ ■ Ekran biçimleri/organizasyonu
 - ▶ ■ Pencere biçimleri/organizasyonu
 - ▶ ■ Veri biçimleri
 - ▶ ■ Komut biçimleri

- ▶ Kullanılabilirlik gereksinimleri, kullanıcı ihtiyaçlarını karşılayan gereksinimleri sağlar. Müşteri memnuniyeti açısından, kullanılabilirlik genellikle en önemli gereksinim sınıfıdır.
- ▶ Bu tür gereksinimlerin yeni sistemlerde veya yeni özelliklerle ilgili olarak keşfedilmesi zordur. Prototipleme, bu tür gereksinimleri keşfetmede yardımcı olabilir.
- ▶ Performans gereksinimleri, yazılıma veya bir bütün olarak yazılımla insan etkileşimine ilişkin statik ve dinamik gereksinimlerdir.
- ▶ Tipik performans gereksinimleri, desteklenecek eşzamanlı kullanıcı sayısını, işlem ve görevlerin sayısını ve içinde işlenecek veri miktarını içerebilir.

- ▶ Mantıksal veri tabanı gereksinimleri, aşağıdakiler gibi çeşitli işlevler tarafından kullanılan bilgi türleridir:
 - ▶ ■ Kullanım sıklığı
 - ▶ ■ Erişim yetenekleri
 - ▶ ■ Veri varlıkları ve ilişkileri
 - ▶ ■ Bütünlük kısıtlamaları
 - ▶ ■ Veri saklama gereksinimleri
- ▶ Tasarım kısıtlaması gereksinimleri, standartlara uygunluk ve donanım sınırlamaları ile ilgilidir. Kısa bir süre sonra gereksinim ifadelerinde kısıtlamaların nasıl organize edileceğini göreceğiz.

- ▶ Standartlara uygunluk, geçerli yasalardan, standartlardan veya düzenlemelerden türetilen gereksinimleri belirtir. Ortaya çıkarma sürecinin başlarında bu belgelerin doğru ve eksiksiz tanımlanması çok önemlidir.
- ▶ Son olarak, yazılım veya sistem öznitelikleri genellikle güvenilirlik, kullanılabilirlik, güvenlik, bakım kolaylığı, taşınabilirlik ve diğer birçok özelliği içerebilir..

İşlevsel Gereksinimlerin Temsil Edilmesine İlişkin Öneriler

- ▶ İşlevsel gereksinimler, sistemin ne yapacağını ortaya koyan ve yapısal olan gereksinimlerdir.
- ▶ İşlevsel gereksinimler, duruma göre açıklamayı veya diğer genel açıklama biçimlerini kullanabilir.

- ▶ İşlevsel gereksinimler şu şekilde organize edilebilir:
- ▶ ■ Sistem modu (ör. navigasyon, savaş, teşhis)
- ▶ ■ Kullanıcı sınıfı (ör. kullanıcı, süpervizör, teşhis)
- ▶ ■ Nesne (sınıfları/nesneleri, öznitelikleri, işlevleri/yöntemleri ve mesajları tanımlayarak)
- ▶ ■ Özellik (sistemin kullanıcıya ne sağladığını açıklar)
- ▶ ■ Uyarı (ör. sensör 1, sensör 2, aktüatör 1, ...)
- ▶ ■ İşlevsel hiyerarşi (örneğin, görevlerin yukarıdan aşağıya ayrıştırılması)

- Bu tekniklerin bir kombinasyonu tek bir SRS belgesinde kullanılabilir.