YAZILIM GELİŞTİRME SÜRECİ

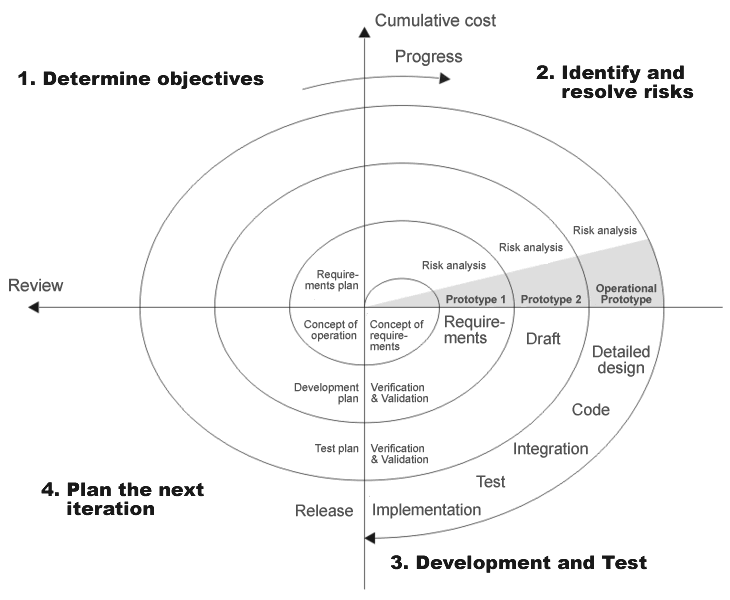
1. Gereksinim Analizi
2. Tasarım
3. Kodlama
4. Test
5. Uygulama

Projemizi **Spiral Yazılım Geliştirme** Modeli kullanarak gerçekleştireceğiz.

Spiral modeli neden kullanıyoruz?

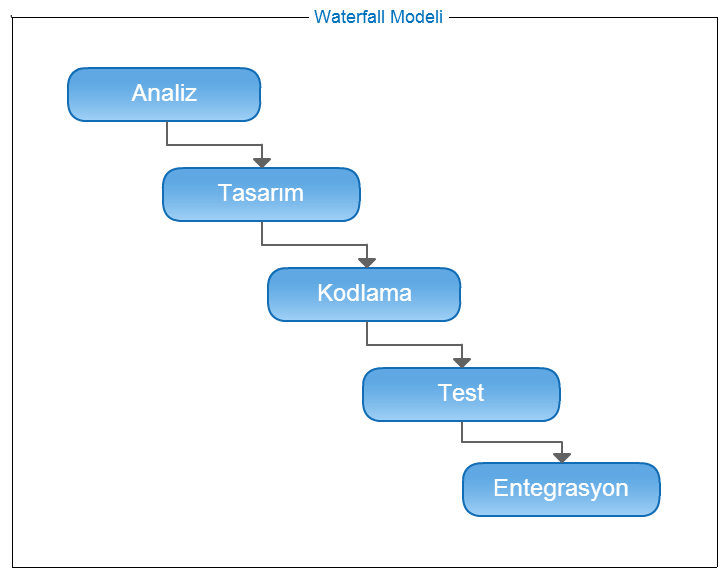
Projemizi gerçekleştirirken en bir yazılım sürecini tamamiyle tamamlamıyoruz. Ufak analiz tasarım ve kodlama yaparak proje bitene kadar ufak prototipler sunuyoruz. Vize ve final kısımlarında projenin işleyişini gösterebilmek adına çıktılara ihtiyacımız olmasından kaynaklı. Yaptığım proje tek bir uygulama olarak çalışmayacak. Ve daha kolaylık sağlayacağını düşündüğümüz için önce website uygulama kısmı, devamında chrome extension kısmı ve devamında bu iki birim arasındaki iletişim kısımları tasarlanıp kodlanacak.

**Spiral:**Bu modelin temelinde,**yazılım geliştirme** süreci boyunca risk analizi önemlidir. Bu modelde,her bir klasik **waterfall** modeli, çok sayıda iterasyona bölünür ve her iterasyonda planlamayı ve risk analizini inceler. Bu modelde bir **yazılım** geliştirildiği zaman, her bir iterasyonda bir çıktı elde edilecektir ve bu çıktılar elde edilirken belirli riskler altında gerçekleştirilecektir. Bu modele uyum sağlamanın maliyeti oldukça fazladır.



Öncelikle Yazılım Geliştirme Süreci Belirlenir.

* 1. Örnek: Waterfall Modeli



**Gereksinim Analizi**

**1-) İşlevsel (Functional) Gereksinimler :**

Sistemin ne yapacağını yapısal olarak ortaya koyan gereksinim türüdür. Ayrıca sistemin sunacağı hizmetleri de tanımlar. İşlevsel gereksinim    ‘Ne’ sorusuna cevap verir.

**2-)İşlevsel Olmayan (Nonfunctional) Gereksinimler:**

Sistemin nasıl olacağına karar veren gereksinim türüdür. Arayüzler, güvenlik, kalite gibi soyut nitelikleri belirler. ‘Nasıl’ sorusuna cevap verir.

**Gereksinim Analizi Nedir?**

Sistem gereksinimlerin ve paydaşların ihtiyaçlarının belirlendiği, arayüzlerin, sistem tasarımlarının nasıl olacağına karar verilen ve sistemin, gereksinimlerin özetinin oluşturulduğu kısımdır.

Ayrıca analiz yapmamızın belirli **amaçları** vardır. Peki nedir bu amaçlar ?

* Yazılımın ya da sistemin sınırlarının  belirlenmesi
* Kullanıcı gereksinimlerinin çözüme ulaştırılması
* Sistemin kalitesinin belirlenmesi
* Kalitenin gerçekleştirilmesi için gerekli olan gereksinimin geliştirilmesi
* Paydaşların önceliklerinin belirlenip , görüşülmesi
* Gereksinimlerin kategoriler halinde sınıflandırılması ve alt sınıflara ayrılması

**ÇALIŞMA NASIIL OLACAK?**

1. Kullanıcı websitesine girer.

* **Kayıt olmamışsa:** İlk olarak Login/Register sayfasına yönlenir. Kayıt oluşturur.

Facebook ya da gmaille login yapıcak. Kaydı onaylarsa kullanıcı adı ve şifre oluşturmasını s

* **Kayıtlı ise:** anasayfaya yönlenir.

1. Anasayfa’da

* Kullanıcı bilgilerini girmemişse;
* Kullanıcının düzenli aylık geliri. (Çalıştığı maaşı ) Ve devamında resmi olmayan geliri.
* Harcamaları
  + Ev kira ise kira:
  + Elektrik, su, doğalgaz, internet vb.
  + Günlük minimum harcama / max. Harcama
  + Ekstra harcamalar: yaklaşık miktarı:

Aylık Gelir/Gider tablosu sisteme eklenir.

Bi tane infografik => net = gelir – gider .. bunu html grafiğiyle profil kısmına eklicez.

Bu bilgiler eklenirken asıl hedef olan kredi kartı borçlanması bilgileri test aşaması olduğunda kullanıcıdan istenir. Sebebi anlık kredi kartı borç bilgisinin sisteme girilmesidir. Teste eklenen anlık kredi kartı borcu o ondaki aylık NET ten düşülür. Son gelir gider tablosuna göre test sonucu belirlenir. Testin detayları aşağıdaki “Test” kısmında belirtilecek.

Eğer kullanıcı test kısmında anlık kredi kartı borcularını girerse, kullanıcıya testin sonunda kaydedip kaydetmeyeceği sorulur. Eğer kabul ederse anlık veriler kullanıcı detayları kısmına eklenir. Ve Haftalık olarak website ziyaretinde güncellemesi için bildirim oluşturma yapılır.