Nesneye Yönelik Yazılım Mühendisliği (376)

Dr. Öğr. Üyesi Ahmet Arif AYDIN

Yazılım Mühendisliğine Giriş

- Software (Yazılım) nedir ?
  - Bilgisayar **programları**, **prosedürleri** ve bir bilgisayar sisteminin işletilmesine ilişkin **dokümantasyon** ve **veriler** (*Computer **programs**, **procedures**, and possibly associated **documentation** and **data** pertaining to the operation of a computer system*)

- Bir yazılım **varlığı** (entity), **veri kümeleri** (data sets), algoritmalar, fonksiyonlar ve aralarındaki ilişkilerden oluşur.

Yazılım Mühendisliğine Giriş

- **Software Engineering**

  - Yazılımın **geliştirilmesi** (development), **işletilmesi** (operation) ve **bakımı** (maintanence) aşamalarının sistematik, disiplinli ve ölçülebilir bir yaklaşımın uygulanması ve bu alanlardaki çalışmalar yazılım mühendisliği olarak bilinir.

  - (The application of a **systematic**, **disciplined**, **quantifiable** approach to the ***development***, ***operation***, and ***maintenance*** of software)

IEEE: Institute of Electrical and Electronics Engineers, 1963

"IEEE Standard Glossary of Software Engineering Terminology," Office, vol. 121990, no. 1, p. 1, 1990.

No Silver Bullet !

- Frederic P. Brooks

  - ***No Silver Bullet: Essence and Accident of Software Engineering , 1987***

No Silver Bullet !

- Yazılım problemleri
  - hatalı yönetim *(wrong software management)*
  - kaçırılan teslim tarihi *(missed schedules & deadlines)*
  - bütçe ve ödeme problemleri *(payment & budget problems)*
  - hatalı - kusurlu ürünler *(flawed products)*

- Bahsedilen problemler software projelerini canavarlara dönüştürebilir!
- Bu canavarı ortadan kaldırmak için bir sihirli değnek yok !
- (**NO SİLVER BULLET**) !

Frederic P. Brooks,No Silver Bullet: Essence and Accident of Software Engineering , 1987

No Silver Bullet !

- Yazılım problemlerini ortadan kaldırmak için yapılması gerekenler :
  - Replacing the demon and humors theories by the **germ theory**
  - **Stepwise progress** (aşamalı olarak ilerleme)
  - **Persistent** (*ısrarcı*) & **unremitting** (*aralıksız*) effort

Frederic P. Brooks,No Silver Bullet: Essence and Accident of Software Engineering , 1987

No Silver Bullet !

- **Yazılım teknolojileri problemlerini çözmek için tek bir bir sihirli değnek yok çünkü yazılımın verimlilik, güvenilirlik ve basitlik hedeflerine tek bir geliştirme, yönetim veya teknik ile ulaşılamaz** (***There is no silver bullet for software technologies*** since software *productivity, reliability and simplicity* goals can't be achieved by a single development, management or technique)

- **Donanım ve elektronik alanındaki gelişmeler sihirli bir değnek gibi düşünülebilir**.
  - Fast computer hardware development provided ***a kind of silver bullet*** by electronics, transistors and large scale integration.

- **Yazılım geliştirmenin zorlukları problemin tanımlanması, tasarım ve test**
  - *Challenges of building software are specification, design, and testing of this conceptual construct.*

Frederic P. Brooks,No Silver Bullet: Essence and Accident of Software Engineering , 1987

Yazılım teknolojilerinin zorlukları

- Fred Brooks yazılım teknolojilerinin zorluklarını iki ana kategori altında tanımlamaktadır. (*Major difficulties of software technology can be divided in Essence and Accidents*).
  - Öz ile alakalı (**Essence**) olan ve görmezden gelinemeyen zorluklar (***Essence*** is *irreducible challenges that inherited from the nature of software)*
    - complexity
    - conformity
    - changeability
    - invisibility

- ***Accidents*** *are product related and domain specific problems*.
  - Ürün ve alan ile alakalı zorluklar (Accidents)

Frederic P. Brooks,No Silver Bullet: Essence and Accident of Software Engineering , 1987

Essence (Öz ile Alakalı) Zorluklar (Complexity)

1. *Karmaşıklık (Complexity)*

- ***projenin boyutuna bağlı olarak liner olmayan bir biçimde artar*** *(increases in nonlinear fashion when their size scale-up)*
- ***birbirine benzeyen iki parça bulunmamaktadır*** *(there is no alike two parts: variety of modules and classes)*
- ***diğer alanlarda bu özellik görmezden gelinebilir*** *(In math and physics complex phenomena's are modeled by simplified models with "ignoring complexities", however, complexity can't be ignored in software since it is an "essential feature")*

Frederic P. Brooks,No Silver Bullet: Essence and Accident of Software Engineering , 1987

Essence (Öz ile Alakalı) Zorluklar: Conformity

2. *Uyum (Conformity)*

- **Yazılım dış dünyadan izole edilmiş bir biçimde geliştirilemez. Geliştirilen yazılım dış dünyada bulunan kısıtlamalarla uyumlu olmak zorundadır. (***A software entity can't be produced in isolation, it must compatible (conform) to real-world constraints)*

- **Kısıtlamalar mevcut olan donanım önceden var olan donanım, üçüncü taraf bileşenleri, devlet düzenlemeleri ve eski veri formatları (***such as pre-existing hardware, third party components, government regulations, and legacy data formats)*

Frederic P. Brooks,No Silver Bullet: Essence and Accident of Software Engineering , 1987

Essence (Öz ile Alakalı) Zorluklar: Changeability

3. *Changeability(değişebilirlik)*

- ***Yazılım sürekli değişen bir ortam içerisinde geliştirilir.*** (*Software entities are built on "constantly changing environments" applications, users, laws, hardware*)

- **Yazılım değişim isteklerine karşı esnek olmalıdır** (*Software entity must be resilient for change requests which can be user request for new features or compatibility need for new hardware*)

- **Yazılım fikir tabanlı bir yapı olduğu için değişimler araba, bina gibi yapılara göre daha kolaydır.** (*Since software is pure thought-stuff, therefore it can be changed more easily than buildings, car, etc)*

Frederic P. Brooks,No Silver Bullet: Essence and Accident of Software Engineering , 1987

Essence (Öz ile Alakalı) Zorluklar : invisibility

4. **invisibility**

- **Yazılım somut bir varlık değildir (***Software is invisible (un-visualizable) because the reality of software is not embedded in space. Silicon chips have diagrams, computers have connectivity diagrams, and however, there is not a solid diagram for software***)**
- **Yazılımı somut olarak modellenmeye çalışılmaktadır** *(we can't see a tangible, we create abstractions, and create simple models)*
- **Veri ve kontrol akışı bağımlılık kalıpları kolaylıkla tek bir şema yardımıyla temsil edilememektedir** (*Data flow, Control flow, dependency patterns are not easily represented in one representation)*
- **Yazılım yapısısı tasarlarken ortak çalışma gereklidir** *(collaboration is necessary for designing structure of a software system)*

Frederic P. Brooks,No Silver Bullet: Essence and Accident of Software Engineering , 1987

No Silver Bullet !

- **Accidents**
  - **Özel bir alan ve proje ile alakalı olarak ortaya çıkabilen problemler** *(product related , domain specific problems)*

1. **High-level languages**
2. **Time Sharing**
3. **Unified programming environments**

Frederic P. Brooks,No Silver Bullet: Essence and Accident of Software Engineering , 1987