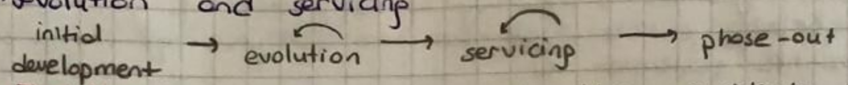
Ch-9 SOFTWARE EVOLUTION

→ Software change is inevitable.

- A key problem for all organizations is implementing and managing change to their existing software systems

* Yeni yazılım yerine varolanı güncellemek !

→ Evolution and servicing



Evolution: Yazılım sisteminin kullanımda olduğu ve sistemde yeni gereksinimler önerildiği ve uygulandığı icin gelisen döngüdeki aşama

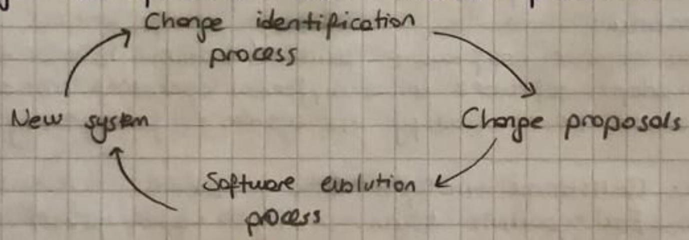
Servicing: Yazılım kullanıslı ama yeni bir islevsellik eklenmedi
↳ tek değisiklik calısır durumda kalması icin gerekli olanlardır

Phase-out: Yazılım hala kullanılabilir ama değisiklik yapılmadı

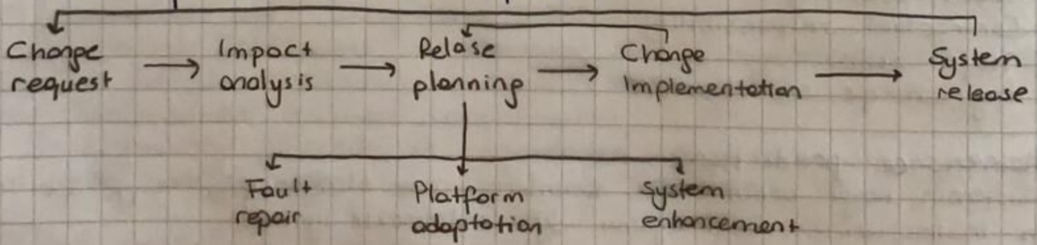
→ Evolution processes

- Software evolution processes depend on
  - The type of software being maintained
  - The development processes used
  - The skills and experience of the people involved.

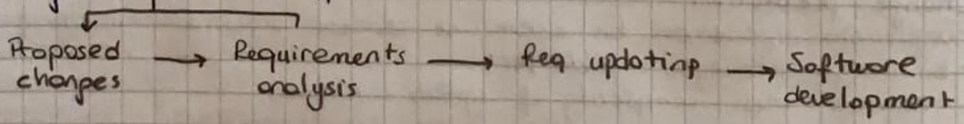
→ Change identification and evolution processes



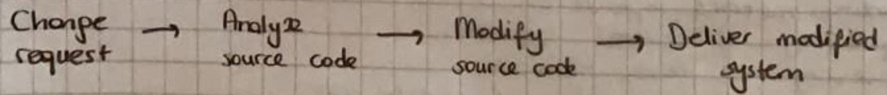
→ The software evolution process



→ Change Implementation



→ The emergency repair process



→ Process metrics → sürdürülebilirliği değerlendirmek için

- Number of requests for corrective maintenance
- Average time required for impact analysis
- Average time taken to implement a change request
- Average time taken [illegible] requests

} Biri artıyorsa sürdürülebilirlikte düşüşe işaret edebilir

→ Agile methods and evolution

- Agile methods are based on incremental development so the transition from development to evolution is a seamless one.

Evolution, sık sık çıkan sis. sürümlerine dayanan geliştirme süreci devamı

→ Program evolution dynamics

- Program evolution dynamics is the study of the processes of system change
- Lehman ve Belady: gelişirken sistemlere bir dizi yasa öne sürdü. Yasalardan çok mantıklı gözlemler var.

Lehman's laws: degisim sürekli, uzun süreli sistem evrimi çalışmalarından elde edilen bilgileri açık

- Continuing change
- Increasing complexity
- Large program evolution
- Organizational stability
- Conservation of familiarity
- Continuing growth
- Declining quality
- Feedback system

→ Software Maintenance

- Kullanıldıktan sonra bir programın değiştirilmesi

Types of maintenance:

→ Maintenance to repair software faults (Yazılım hatalarını onarmak)
→ Maintenance to adapt software to a different operating environment
→ Maintenance to add to or modify the system's funcionality

→ Maintenance cost: Geliştirme maliyeti daha yüksek
Eski yazılım bakım maliyeti daha yüksek

Cost Factors:

- Team stability
- Contractual responsibility
- Staff skills
- Program age and structure

→ Maintenance prediction:

- Sistemin hangi bölümlerinin sorunlara neden olabileceğini ve yüksek bakım maliyetlerinin bulunduğunu değerlendirmekle ilgilidir.

→ Change prediction

- Değişimin öngörülmesi bir sistem ile çevresi arasındaki ilişkilerin anlaşılmasını gerektir
- Bu ilişkiyi etkileyen faktörler:
  - number and complexity of system interfaces
  - number of inherently volatile system requirements
  - the business processes where the system is used

→ Complexity metrics depends on:

- Complexity of control structures
- Complexity of data structures
- Object, method (procedure) and module size

→ Process metrics → sürdürülebilirligi degerlendirmek icin kullanilabilir

- Number of requests for corrective maintenance
- Average time required for impact analysis
- Average time taken to implement a change request
- Number of outstanding change requests

Biri artiyorsa sürdürülebilirlikte düsüse isaret edebil

→ System re-engineering
- Islevselligini degistirmeden eski bir sis. bir kısmını veya tamamını yeniden yapılandırmak
- Bakımı kolaylastırmak icin

Advantages:
- Reduced risk
- Reduced cost.

→ Re-engineering process activities
- Source code translation
- Reverse engineering
- Program structure improvement
- Program modularisation
- Data reengineering

→ Reengineering cost factors
- The quality of the software to be reengineered.
- The tool support available for reengineering
- The extent of the data conversion which is required.
- The availability of expert staff for reengineering.

→ Preventative maintenance by refactoring
- Degisim yoluyla bozulmayı yavaslatan bir programda iyilestirmeler yapma sürecidir

→ Refactoring and reengineering

Refactoring: Development ve evolution süreci boyunca sürekli bir gelisim sürecidir Bir sis. sürdürmenin maliyetlerini ve zorluklarını artıran yapı ve kod bozulmasından kacınmak amaclanmıstır

Re-engineering: Bir süre sis. saglandıktan sonra re-engineering yapılır ve bakım maliyeti artar.

→ "Bad smells" in program code
- Duplicate code
- Long methods
- Switch (case) statements
- Data clumping
- Speculative generality

→ Legacy system management

- Eski sis. dayanan kuruluslarin bu sistemleri gelistirmek icin bir strateji secmeleri gerekir.
- Secilen strateji: sistem kalitesine ve isletme degerine bagli olmali

→ Legacy system categories

- Low quality, low business value : should be scrapped
- Low quality, high-business value : Uygun sis varsa yeniden yapilandirilmali
- High quality, low-business value : COTS ile yer degistir, hurdala ya da bakim yap.
- High quality, high-business value : Normal sis. bakimini kullanarak devam

→ Business value assessment

- Degerlendirmede farklı bakıs acilarini dikkate alin
  - System end-users
  - Business customers
  - Line managers
  - IT managers
  - Senior managers

→ Issues in business value assesment

- The use of the system
- The business processes that are supported
- System dependability
- The system outputs

→ System quality assessment

- Business process assessment : isin mevcut hedeflerini ne kadar iyi destekliyor
- Environment assessment : Sis. ortami ne kadar etkili
- Application assessment : Uyg. yazilimi sis kalitesi nedir

→ Business process assessment

- Use a viewpoint-oriented approach and seek answers from system stakeholders

→ Factors used in environment assessment

- Supplier stability
- Failure rate
- Age
- Performance
- Support requirements
- Maintenance costs
- Interoperability

→ Factors used in application assessment

- Understandability
- Documentation
- Data
- Performance
- Programming language
- Configuration management
- Test data
- Personnel skills.

→ System measurement

- You may collect quantitative data to make an assessment of the quality of the app. system
  - The number of system change requests
  - The number of different user interfaces used by the system
  - The volume of data used by the system

Ch11 Security and Dependability

⇒ Dependability properties

→ System dependability
- Bir sistemin güvenilirliği, kullanıcının bu sisteme olan güvenirliliğini yansıtır
- Dependability covers the related systems attributes of reliability, availability and security. These are all interdependent

→ Importance of dependability
- Güvensiz sis kullanıcılar tarafından reddedilebilir.
- Arıza ekonomik kayıplara veya fiziksel hasara yol açarsa, arıza maliyeti çok yüks

→ Causes of failure
- Hardware failure:
- Software failure:
- Operational failure:

→ Principal dependability properties

Dependability

*Availability	*Reliability	*Safety	*Security
-Sis. çalışır durumda olması	-Sis. beklenen hizmeti doğru sunabilmesi	-Sis insanlara ya da çevreye zarar vermesinin ihtimali	-Sis. saldırılara karşı koyabilmesi

→ Other dependability properties
- * Repairability: Bir arıza durumunda sistemin nasıl tamir edilebileceğini yansıtır
- * Maintainability: Sistemin yeni gereksinimlere ne ölçüde uyarlanabileceğini yansıtır.
- * Survivability: Bir sistemin hizmetlerini kullanıcılarına kasıtlı veya kazara saldırma karşısında sunmaya devam edebilmesi
- * Error tolerance: Kullanıcı giriş hatalarının önlenip ne ölçüde tolere edilebileceğini yansıtır.

→ Dependability attribute dependencies
- Sistem güvenilmez olabilir, çünkü verileri bir saldırıdan dolayı bozulmuş
- Virüs bulaşmışsa sistem güvenliğinden emin olamazsın
- Bir sisteme yapılan hizmet reddi saldırılarının kullanılmaz duruma getirilmesi amaçlanmıştır.

→ Dependability achievement
- Avoid the introduction of accidental errors when developing the system
- Design protection mechanisms that guard against external attacks.
- Configure the system correctly for its operating environment.
- Include recovery mechanisms to help restore normal system service after a failure

→ Dependability cost
- Daha güvenilir olması için daha pahalı teknik kullanmak

and reliability

bir şartnameye uygunluk ve hizmet sunumu ile ilgili

=) Availability and reliability

Reliability: Belirli bir ortamda belirli bir amaç için belirli bir süre sarfında failure-free sistemin çalışabilme olasılığı

Availability: Bir sis belirli bir zamanda istenen hizmetleri sunması

*Sistem arızaları hızlı bir şekilde tamir edilebildiği ve verilere zarar vermediği sürece, bozu sis. arızaları problem olmayabilir.

→Perceptions of reliability

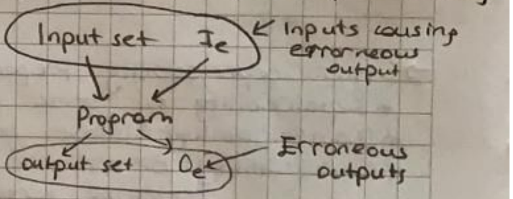
Resmi güvenilirlik tanımı, kullanıcının bir sistemin güvenilirliği algısını her zaman yansıtmaz.

→Reliability and specifications

→Availability perception (Sistemin hizmet sunmaya hazır old. süre) iki faktörü dikkate almaz
- Hizmet kesintisinden etkilenen kullanıcı sayısı
- Kesintinin uzunluğu

→Reliability terminology
- Human error or mistake
- System fault
- System error
- System failure

→ A system as an input/output mapping

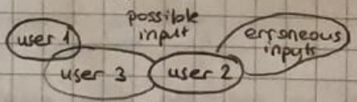


→Faults and failures

mutlaka sistemle sonuçlanması gerekmez

System errors

→Software usage patterns



→Reliability in use

Program hataları nadiren yapılan bölümlerde olabilir Bu yüzden kod kullanıcılar tarafından asla karşılaşılmayabilir. Bunların çıkarılması algılanan güvenilirliği etkilemez

→Reliability achievement
- Fault avoidance: En aza indirgeyen geliştirme tekniği kullanılır
- Fault detection and removal:
- Fault tolerance

→Safety, sistemi yansıtan bir sistemin özelliğidir
- Primary safety-critical systems: Başarısızlığı neden olabilecek gömülü yazılım sistemleri ilişkili donanım başarısız ve doğrudan insanları tehdit etmek
- Secondary safety-critical systems: Arızası diğer arızalara neden olan sistemler

→ Safety and reliability

Reliability → belirli bir sartnameye uygunluk ve hizmet sunumu ile ilgili
Safety → sistemin spesifikasyonuna uygun olup olmadıgına bakılmaksızın hasara neden olamayacagından emin olmakla ilgili

→ Unsafe reliable systems
*Sistemde uyuyan hatalar olabilir
- Specification errors
  yanlışsa sistem belirtildigi gibi davranabilir.
- Hardware failures generating spurious inputs
  sartname tahmin etmek zor.
- Context-sensitive commands i.e. issuing the right command at the wrong time

→ Safety terminology
- Accident
- Hazard
- Damage
- Hazard severity
- Hazard probability
- Risk

→ Safety achievement
- Hazard avoidance
- Hazard detection and removal
- Damage limitation

→ Normal accidents
Karmasık sistemlerdeki kazaların nadiren tek bir nedeni vardır, cünkü bu sistemler tek bir arıza noktasına dayanıklı

→ Security: Bir sis güvenligi sistemin kendisini yanlıslıkla veya kasıtlı bir dıs saldırıdan koruma yetenegini yansıtan sis özelligi.

→ Security terminology
- Asset
- Exposure
- Vulnerability
- Attack
- Threats
- Control

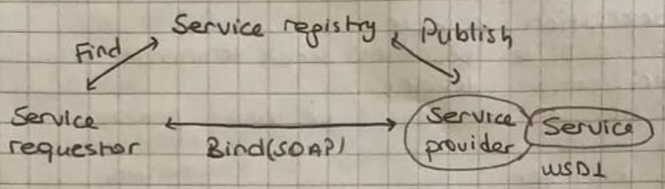
→ Damage from insecurity
- Denial of service
- Corruption of programs or data
- Disclosure of confidential inf.

→ Security assurance
- Vulnerability avoidance
- Attack detection and elimination
- Exposue limitation and recovery

Ch-19 Service-oriented Architecture

→ Web service: daha genel servis anlayısı örnegi; (Bir tarafın digerine sundugu eylem)

→ Service-Oriented architecture: Bilesenlerin tek basına servis oldugu dagıtık sistemler gelistirme aracı



→ Benefits of SOA
- Services can be provided locally or outsourced to external providers
- Services are language-independent
- Investment in legacy system can be preserved
- Inter-organisational computing is facilitated through simplified inf. excha

→ Key Standards
- SOAP → Servis iletisimini destekleyen mesajlasma standardı
- WSDL (Web Service Definition Language) → servis arayüzü ve baglantı tanımlarınasına için ve
- WS-BPEL → Servis kompozisyonunu tanımlamak için kullanılan is akısı dilleri için bir standart

→ Web service standards

XML technologies (XML, XSD ...)
Support (WS-Security, WS-Addressing ...)
Process (WS-BPEL)
Service definition (UDDI, WSDL)
Messaging (SOAP)
Transport (HTTP, HTTPS, SMTP ...)

→ RESTful web service
- REST bir sunucudan müsteriye kaynakların temsilini aktarmaya dayanan mimari
- büyük web hizmetlerinden daha düsük bir ek yük icerir

→ Araç içi bilgi sistemi (konum için GPS)

Advantage of SOA for this app

Sistemin ne zaman programlanıp dagıtılacagına, hangi servis saglayıcının kullanılması gerektigine karar vermek gerekli degil
- Cevrimici hizmet kullanımı nedeniyle sınırlar içinde hareket eder

→ Service-oriented software engineering
- Service engineering. The development of dependable, reusable services
- Software development with services

→ Services as reusable components

Service: A loosely-coupled, reusable software component that encapsulates discrete functionality which may be distributed and programmatically accessed
→ A critical distinction between a service and a component as defined in CBSE is that services are independent

→ Web service description language
Servis arayüzü WSDL'de belirtilen servis tanımında tanımlanmıştır.
- The WSDL specification defines
  - Servisin hangi işlemleri desteklediği ve mesaj formatı
  - Hizmete nasıl erişilir
  - Servisin bulunduğu yer.

→ Organization of WSDL specification
- Intro → XML namespace declarations
- Abstract Interface → Type, Interface, Message declarations
- Concrete implementation → Binding, endpoint declaration

→ WSDL specification component
What → hizmet hangi işlemleri destekler
How → Soyut arayüzü somut protokoller ile eşleştirin
Where → web hizmeti uyg. konumunu açıklar

→ Service Engineering
- Hizmet odaklı uygulamalarda yeniden kullanım için hizmet geliştirme süreci

→ The service engineering process
1) Service candidate identification → Service requirements
2) Service design → Service interface specification
3) Service implementation and deployment → Validated and deployed service

Service candidate identification: uygulanabilecek olası hizmetleri tanımladığınız ve servis gereksinimlerini tanımladığınız yerde
- Hizmet iş süreçlerini desteklemelidir
- 3 temel hizmet türü
  - Utility service → Farklı iş süreçleri tarafından kullanılan
  - Business service → Belirli bir işletme işleviyle ilişkilendirilen hizmet
  - Coordination service → Sipariş verme gibi birleşik işlemleri destekleyen koordinasyon hizmet.

→ Task and entity-oriented services
- Task-oriented → bazı faaliyetlerle ilişkili olanlardır.
- Entity-oriented → nesnelere benzer

✱ Utility or business services may be entity or task oriented
Coordination services are always task oriented.

→ Service classification
- Task
- Entity

Ch-13

→ Catalog services: Diğer şirketler tarafından onlardan hangi malların sipariş edilebileceğini göstermek için bir tedarikçi tarafından yaratılmıştır.

→ Functional descriptions of catalog service operations

- Make Catalog
- Compare
- Lookup
- Search
- Check Delivery
- MakeVirtualOrder

⇒ Interface design stages

- Logical interface design
- Message design
- WSDL description

⇒ Service implementation and deployment

- Standart bir programlama dili kullanarak programlama hizmetleri
- Input message oluşturarak output beklenen gibi mi diye test edilmeli
- Deployment servis tanıtmayı ve bir web sunucusuna yüklemeyi içerir

→ Legacy system services

- Hizmetlerin önemli bir uygulaması, eski sistemlere gömülü işlevlere erişim sağlamaktır
- Kapsamlı işlevsellik sunar

→ Software development with services

- Mevcut servisler yeni bileşik servisler oluşturmak için oluşturulmakta ve yapılandırılmakta

→ Construction by composition

- Formulate outline workflow
- Discover services
- Select possible services
- Refine workflow
- Create workflow program
- Test completed service or application

Ch-22 Project Management

→Software project management

- Yazılım geliştirme bütçeye bağlı old. belirlenen kısıtlamadan dolayı proje yönetimi gerekir.

→Success criteria

- Deliver the software to the customer at the agreed time.
- Keep overall costs within budget.
- Deliver software that meets the customer's expectations.
- Maintain a happy and well-functioning development team

→Software managment distinctions

- The product is intangible
- Many software projects are one-off projects
- Software processes are variable and organization specific.

→Mangement activities

- Project planning
- Reporting
- Risk management → Risklerin belirlenmesi ve bir proje üzerindeki etkilerini en aza indirgemek için planların oluş ilgili.
- People management
- Proposal writing

→Examples of common project, product and business risk.

	Affects
Staff turnover Management change Hardware unavailability	Project
Requirements change Specification delays Size underestimate	Project and product
CASE tool underperformance	→ Product
Technology change Product competition	Business

→ The risk management process

- Risk identification
- Risk analysis
- Risk planing
- Risk monitoring

→Risk identification

Bir projedeki riskleri belirlemek için ortak risklerin bir kontrol list. kullan

Risk indicator
- Technology risks
- People "
- Organisational "
- Requirements "
- Estimation "

→ Risk analysis : Riskin olasılığını ve ciddiyetini değerlendirin 200
-low, very low , high , moderate, very high

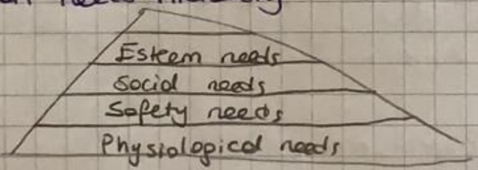
→ Risk planning : Riski yönetmek için strateji geliştir.
-Avoidance strategies
-Minimisation strategies
-Contingency plans
Strategies to help manage risk
-Organizational financial problems
-Recruitment problems
-Staff illness
-Defective components
-Requirements changes
-Organizational restructuring
-Database performance

→ Risk monitoring
-Belirlenen riskleri düzenli olarak değerlendirerek daha az mı yoksa daha da muhtemel mi olacağına karar ver.

→ People mangement factors
- Consistency
- Respect
- Inclusion
- Honesty

→ Motivating people
- Basic needs
- Personal "
- Social "

→ Human needs hierarchy



→ Need satisfaction
-Social
-Esteem
-Self-realization

→ Personality types
- Task -oriented
- Self -oriented
- Interaction -oriented

→ Group communications
- Group size
- Group structure
- Group composition
- The physical work environment