

BİL 481 - Yazılım Mühendisliği

Ders Tanıtımı

Yrd. Doç. Dr. Çağdaş Evren Gerede

Bilgisayar Mühendisliği, TOBB ETÜ

Table of contents

1. Uyarılar

2. Notlama

3. Konular

4. Kimsin Sen?

Uyarılar

- Pratikte önemli bir ders (iş bulanlarınızın çoğu yazılım mühendisi olarak çalışacak),
- Oldukça zorlanılan bir ders;
- Ders kapsamında Yazılım üretebilmek (ileri seviye kodlama yapabilmek) gerekiyor;
- Programlama, nesnesel tasarım ve veri yapıları derslerinin konularına hakim olmak gerekiyor (Bunlardan CC \geq ile geçtiyseniz, çok zorlanacaksınız);
- Kalırsanız BİL495'i bağladığı için sene kaybı muhtemel;
- Katalog uygulamam muhtemel;
- Geçen dönemden bazı rakamlar:
 - Yalnızca 4 öğrenci dersi geçti,
 - AA alan olmadı;
 - Son ödevi yalnızca 5 kişi teslim etti (tam yapabilen yok);
 - Proje demosuna yalnızca 8 kişi geldi.

Uyarılar (devam)

- Çoğunlukla Java ile programlama yapacağız
- Git kullanacağız
- Github kullanacağız
- Eclipse kullanacağız
- Açık kaynak kodlu projelere göz atacağız
- Büyük kodlar içerisinde yolumuzu bulmayı ve varolan yazılımı bozmadan onu genişletebilmeyi tatbik edeceğiz
- Tanımadığınız insanlarla proje bağlamında çalışmak zorunda kalacaksınız (toplanıp, tartışıp kararlar vereceksiniz; işleri parçalara bölüp parçaları bağımsız olarak tamamlayacak daha sonra parçaları biraraya getiriyor olacaksınız) (Bitirme projesi dersi için de hazırlık olacak)

Ne yapmalısınız?

- "Başarısız olma rehberi" sunumunun anlattıklarının tam tersini yapmalısınız;
- Sabırlı olmayı öğrenmelisiniz;
- Olabildiğince fazla açık kaynak kodlu projelerin kodlarını açıp okumalısınız;
- Analitik düşünme yeteneğinizi geliştirmelisiniz (code basede alakalı değişim noktalarını bulma, code blokları üzerinde neden sonuç ilişkisi kurma, değişiklik etki analizi yapma);
- Yazılım geliştirme, building, testing, debugging araçlarını efektif kullanmayı öğrenmelisiniz;
- Stakoverflow, github, google vb. bilgiye ulaşma yollarını aktif kullanmayı öğrenmelisiniz;
- Yeni teknolojileri hızlı bir şekilde öğrenmeye ve uygulayabilmeye hazır olmalısınız.

Notlama

- %3 Code Repository and Continuous Integration Setup
- %5 Coding Assignment 1
- %7 Coding Assignment 2
- %9 Coding Assignment 3 (Demo)
- %5 Proje Part 1 (Demo)
- %11 Proje Part 2 (Demo)
- %20 Arasınan 1
- %20 Arasınan 2
- %20 Final

Konular

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Software Engineering

Planning (agile, waterfall), Design (patterns, anti-patterns, layers, APIs, separation of concerns), Development (readability, cohesion, coupling, indentation, style guides), Building, Dependency Management (targets, make, ant, maven, packaging), Debugging, Testing (Unit, Integration, End-to-end, regression, whitebox, blackbox, ui, manual), Change List, Code Reviews, Configuration Management (commit, branch, revert, merge, rebase, distributed, central), Static Analysis (Lint), Continuous Integration, Release (Branch cut, staging environment, integration environment, production environment, push, cherrypick), Bug and Feature Tracking (prioritization, assignment, communication thread, linked from commits), Experimentation (A/B testing), Logging, Monitoring, Alerting (paging, oncall rotation), Data Analysis, Code Refactoring, Team work (roles, communication ways), Product design documents, Technical design documents, System architecture, UML, User experience, Project Management Lessons (Mythical Man Month, Second System Syndrome)...

Kimsin Sen?

- Yrd. Doç. Dr. Çağdaş Evren Gerede
- Ofis: Z-77
- E-posta: cegerede@etu.edu.tr
- Eğitim
 - 2002, Lisans, Bilgisayar Müh., Bilkent Üniversitesi (Bölüm Birincisi)
 - 2007, Doktora, Bilgisayar Bilimleri, University of California, Santa Barbara, ABD
- Önceki İş Tecrübesi
 - Android, Google, Mountain View, Kaliforniya, ABD
 - YouTube, Google, San Bruno, Kaliforniya, ABD
 - Citrix Systems, San Francisco, ABD
 - IBM TJ Watson Research Center, Yorktown, New York, ABD
 - University of California, Santa Barbara, Kaliforniya, ABD

Yazılım mühendisliği alanındaki araştırma faaliyetleri yazılım sistemlerinin tasarımı, geliştirilmesi, hatalardan arındırılması, saklanması, versiyonlanması, üretime geçirilmesi, ölçeklenmesi, performans takibi yapılması, bakımı, deneylenmesi, güncellenmesi vb. evrelerinde ortaya çıkan çözmeye değer fakat çözülmemiş problemlerin tespitini ve çözülmesini hedefler. Dolayısıyla buradaki elde edilen sonuçlar ve teknolojiler başarılı yazılım sistemlerinin üretilmesine ve devamlılığına katkıda bulunur.

- Devam eden araştırma projelerim
 - Görme Engelli Yazılım Geliştiriciler İçin Daha Erişilebilir Kod İnceleme ve Denetim Araçları, (Yazılım)
 - Konuşma Yoluyla Etkileşilebilen Mobil Yazılımların Geliştirilmesine Yönelik Bir Yazılım Çerçevesi (Yazılım)
 - Videolardan Kamera Kimliği Belirlenmesi Ve Büyük Video Veri Kümelerinde Kaynak Kamera Tespiti. (Dağıtık Sistem) (Taha Hocayla)
- Araştırma Laboratuvarı: İddialı öğrencileri labıma ve araştırma gurubuma katmaktayım

Bitti