



YMT 312-Yazılım Tasarım Ve Mimarisi

Mimari Tasarım Çözümlemesi

Fırat Üniversitesi Yazılım Mühendisliği Bölümü

Bölüm-10

Bu Haftaki Konular

Mimari Oluşturma Teknikleri..7

AquaLush Kalite Niteliklerine Göre Ayırıştırma15

SAD kalite karakteristikleri.....32

Profiller.....23

Gözden Geçirmeler (Reviews).....34

GENEL BAKIŞ

KISIM 1 – Yazılım Mimarilerinin Oluşturulması, Değerlendirilmesi, İyileştirilmesi ve Seçilmesi

KISIM 2 – Yazılım Mimarilerinin Tamamlanması



KISIM 1

Yazılım Mimarilerinin Oluřturulması, Deęerlendirilmesi, İyileřtirilmesi ve Seęilmesi



Amaçlar

- Mimari oluşturma tekniklerini araştırmak ve göstermek
- Sanal aygıt (virtual device) fikrini tanıtmak
- Mimari iyileştirme tekniklerini araştırmak
- Mimarilerin değerlendirilmesi için senaryo ve profillerin kullanımını göstermek
- Mimarilerin değerlendirilmesi için prototiplerin kullanımını göstermek



İçerik

- Mimari oluşturma teknikleri
 - İşlevsel ayrıştırma (Functional decomposition)
 - Kalite nitelikleri bazlı ayrıştırma (Quality-attribute-based decomposition)
- Sanal aygıtlar (virtual device) ve aygıt arabirim modülleri
- Mimari iyileştirme teknikleri
- Mimari alternatiflerinin değerlendirilmesi
 - Senaryolar, profiller, fayda ağaçları
 - Prototipler
- Mimari alternatiflerin seçilmesi



Mimari Oluřturma Teknikleri

- **İřlevsel Bileřenleri Belirleyin**—İřlevsel ve veri gereksinimlerini bütnleřik biřimde gerřekleřtirilmesinden sorumlu bileřenler oluřturun.
- **Bileřenleri Kalite Niteliklerine Gre Belirleyin**—İřlevsel olmayan gereksinimleri karřılayacak bileřenler oluřturun, ardından kalan iřlevsel ve veri gereksinimi bořluklarını dolduracak bileřenler ekleyin.

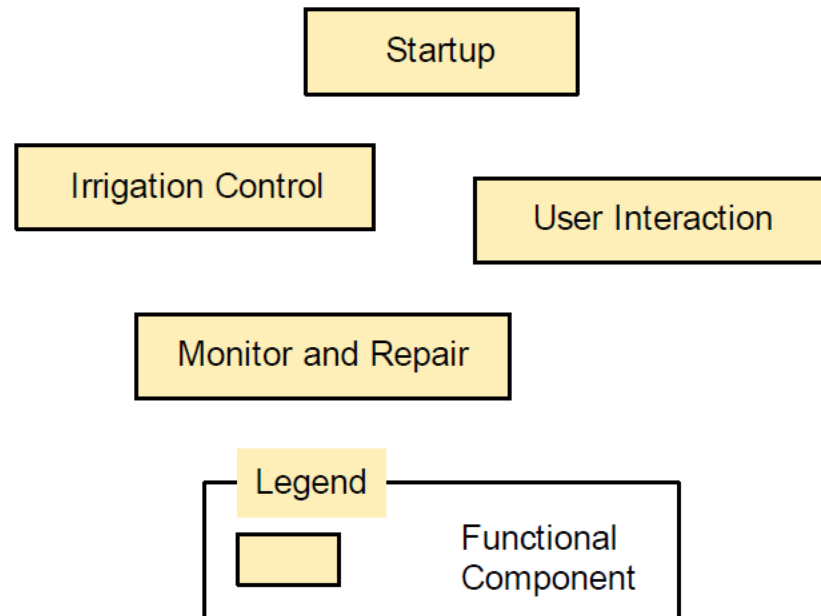


Mimari Oluřturma Teknikleri

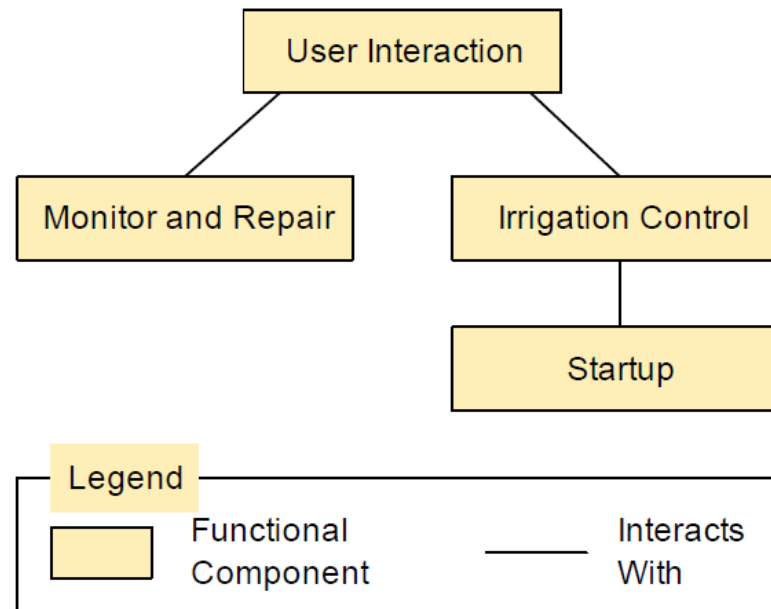
- **Varolan Bir Mimaride Deęişiklik Yapın**— Benzer bir programın mimarisini deęiřtirerek kullanın.
- **Bir Mimari Stilini Ayrıntılılandırın**— Bir mimari stili (architectural style), program veya sistemi oluřturan bileřen tiplerinin ve bunların etkileřimlerinin bir paradigmasıdır. Bir mimari stili ayrıntılılandırarak bir mimari oluřturun.
- **Kavramsal Bir Modeli Dönüřtürün**— Bir problemin kavramsal modelinde deęişiklik yaparak çözüme yönelik bir model oluřturun.



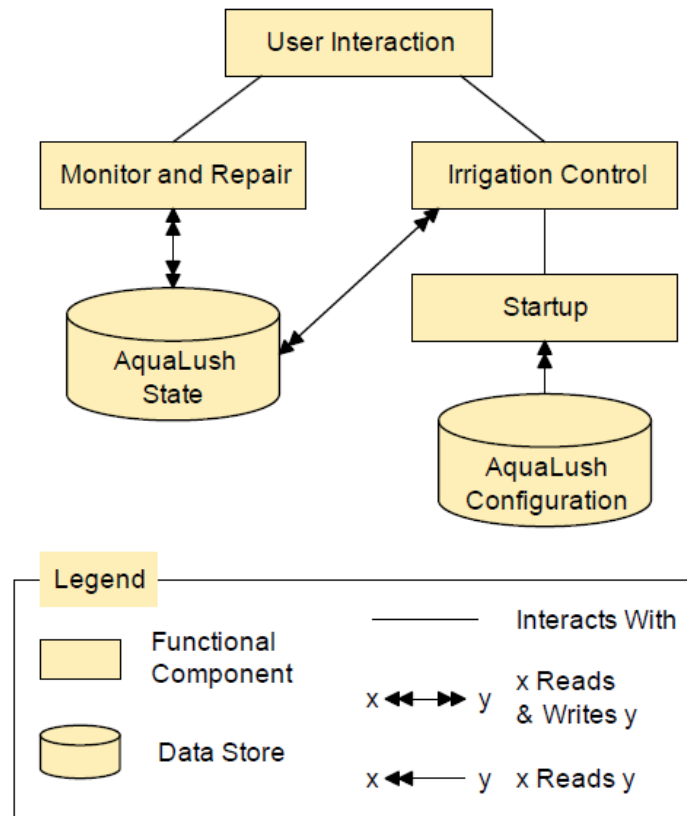
AquaLush İşlevsel Ayırıştırma 1



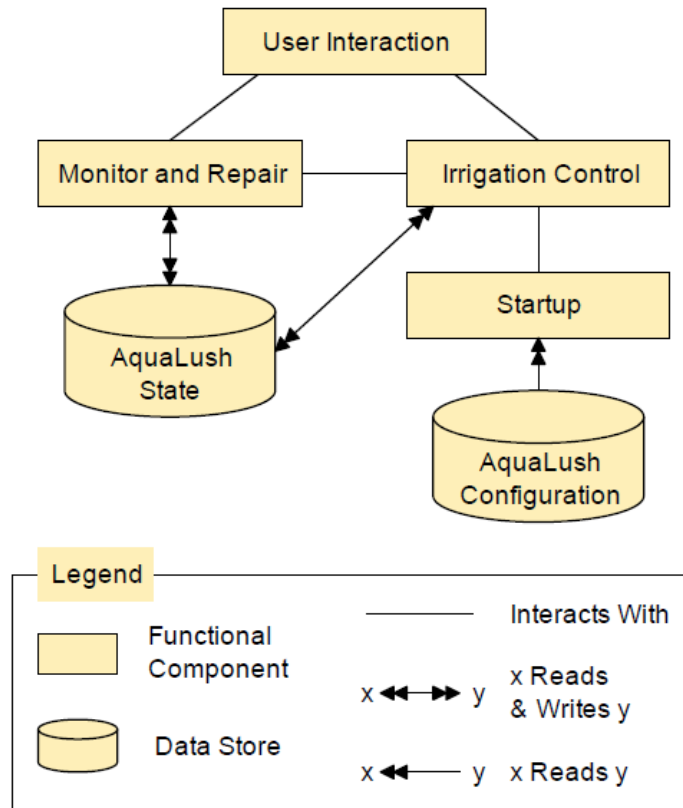
AquaLush İşlevsel Ayırıştırma 2



AquaLush İşlevsel Ayırıştırma 3



AquaLush İşlevsel Ayırıştırma 4



Uyarlanabilirlik Sağlamak (Adaptability)

- Donanım uyarlanabilirliği sağlamanın standart bir yolu sanal aygıtlar içeren bir aygıt arabirim modülü (device interface module) kullanmaktır.

Bir **sanal aygıt** (virtual device), gerçek bir donanım aygıtı veya sisteminin yazılımsal bir **simülasyonu** ya da aygıt/sisteme bir **arabirim**dir.

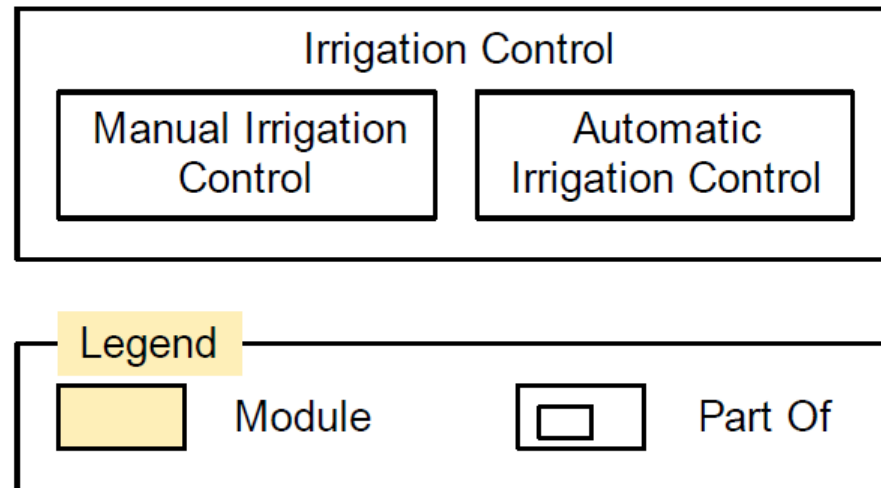
Sanal Aygıt Karakteristikleri

“İdeal” bir cihazı simüle eder

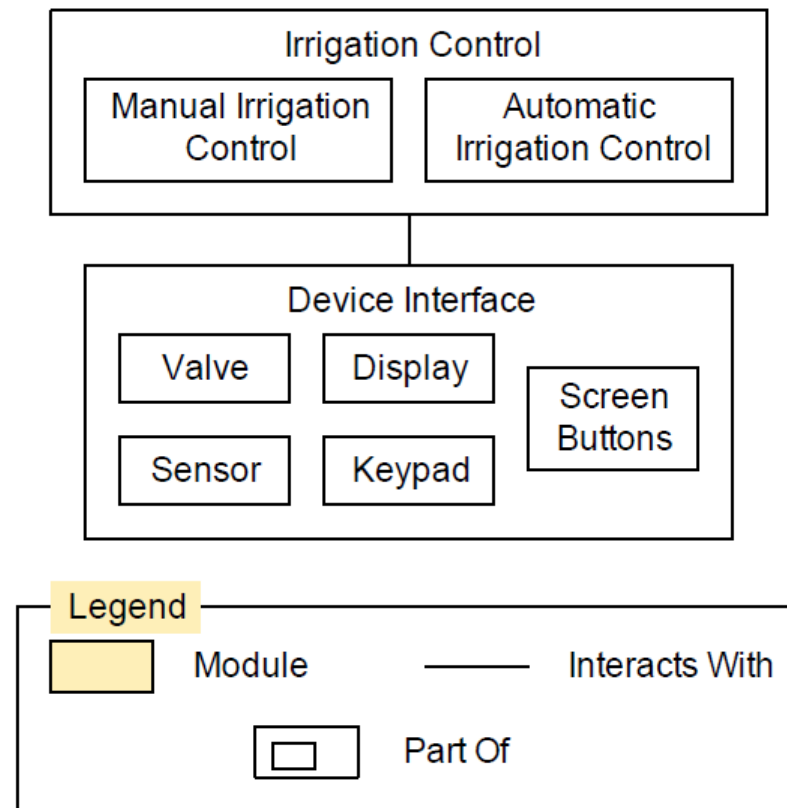
- Tamamen ve sadece bir işi yapar (uyum/cohesion)
- Basit, tutarlı, eksiksiz bir arabirime sahiptir (basitlik/simplicity)
- Programın geri kalanıyla gevşek şekilde bağlıdır (bağılılık/coupling)
- İmplementasyonunu saklar (bilgi saklama/information hiding)
- Asla değişmez (stabilite/stability)

Farklı gerçek aygıtlar veya sistemler için genellikle bir bileşen ailesi olarak gerçekleştirilir.

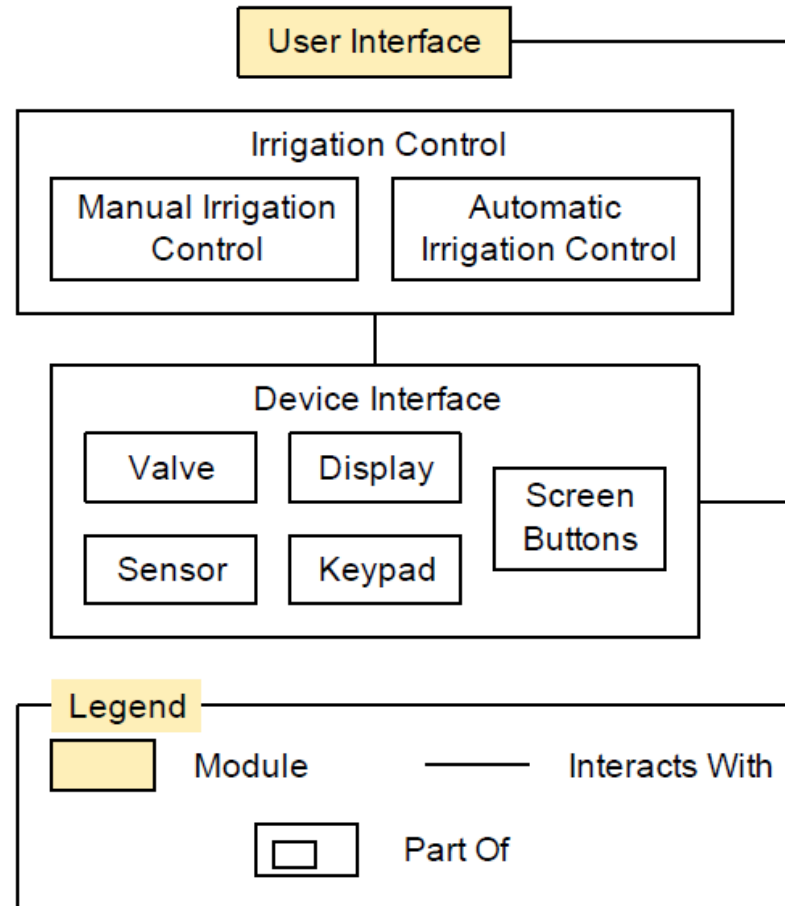
AquaLush Kalite Niteliklerine Göre Ayırıştırma 1



AquaLush Kalite Niteliklerine Göre Ayırıştırma 2



AquaLush Kalite Niteliklerine Göre Ayırıştırma 3



Alternatiflerin İyileştirilmesi

- **Alternatifleri Kombine Edin**—İki veya daha fazla alternatifin en iyi özelliklerini kombine edin
- **Bir Mimari Stilini Empoze Edin**—Bir tasarım stiline çok yakın olan bir mimaride değişiklik yaparak ilgili stile tam uygun olmasını sağlayın
- **Orta-düzey Tasarım Kalıplarını Uygulayın**—Orta-düzey tasarım kalıplarının avantajlarından yararlanmak üzere mimaride değişiklik yapın



Alternatiflerin Değerlendirilmesi

- Tasarımcılar bir mimari spesifikasyona göre oluşturulan bir programın gereksinimleri karşılayacağını program daha oluşturulmadan nasıl bilebilirler?
- Bunun nasıl garanti edileceğini kimse bilemez, ancak olasılığı arttırıcı bazı teknikler kullanılabilir.
- Mimarilerin değerlendirilmesi için senaryoların ve prototiplerin kullanımını inceleyeceğiz.



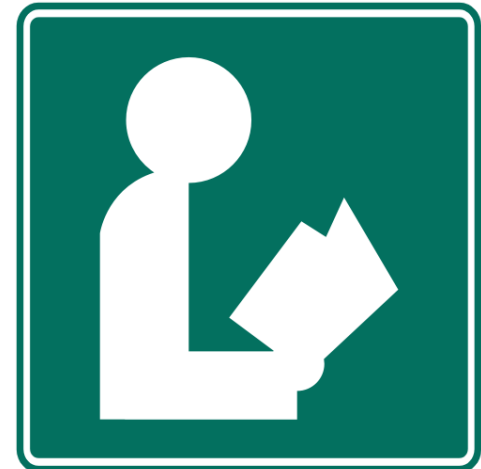
Senaryolar

Bir **senaryo** bir ürün ile belirli bireyler arasındaki bir etkileşimdir.

- Use-case örnekleri bir ürün ve aktörler arasındaki etkileşimlerdir
- Daha geniş bir bakış gerekli çünkü şimdi bir ürün ve herhangi bir birey arasındaki etkileşimleri dikkate alıyoruz.

Senaryo Yazma Kuralları 1

- Her bir senaryoyu açıklayıcı bir ifadeyle etiketleyin.
- Etken çatı ile basit, bildirimsel cümleler yazın.
- Senaryo açıklamalarını üç bölüm halinde yazın
 - Ürünün ve ortamının başlangıç (initial) durumu
 - Ürün ve bireyler arasındaki aktivite akışı (activity flow)
 - Ürünün ve ortamının son (final) durumu



Senaryo Yazma Kuralları 1

- Aktivite akışını tanımlayan cümlede, etkileşimin esas tarafını cümlenin öznesi yapın.
- Hem ürünün hem de ürünün ortamının başlangıç ve son durumlarını açıklayın.
- Mümkün olduğunca hedeflenen çıktı ölçütlerini belirtmeye çalışın.
- Açıklamayı tekrar okuyun ve hata varsa düzeltin.



Profiller

Bir **profil** bir ürünün bir grup gereksinimi karşılayıp karşılamayacağını değerlendirmek için kullanılan senaryo kümesidir.

Örneğin: kullanım profili, güvenilirlik profili

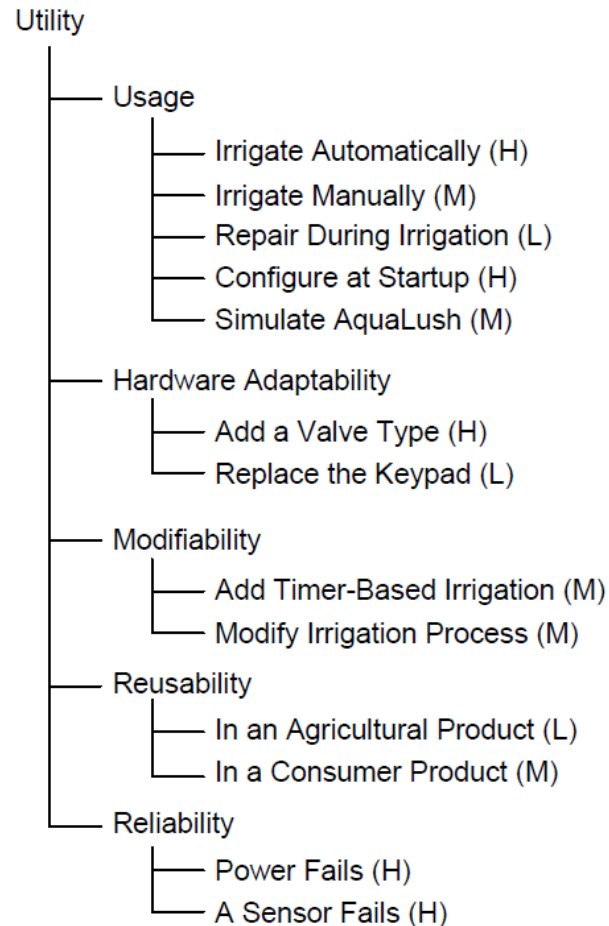
Profil içindeki senaryoların ağırlıkları olmalıdır

Profiller genellikle 3-10 arasında senaryodan oluşur.

Profil ve Senaryo Oluşturma

- Bir **fayda ağacı** (utility tree), dalları profiller ve yaprakları senaryolar olan bir ağaçtır.
 - Kökü “fayda” olarak etiketleyin.
 - Ürün gereksinimlerini yansıtacak şekilde isimlendirilmiş dallar ekleyin.
 - Her bir profil için senaryolar oluşturun.
 - Beyin fırtınası ile senaryolar oluşturun
 - Senaryo listesini rasyonelize edin
 - Her bir senaryoya bir ağırlık verin
 - Her bir profilde 3-10 arası senaryo kalana kadar düşük ağırlıklı senaryoları eleyin
 - Senaryo açıklamaları yazın.

Örnek Bir Fayda Ağacı



Senaryolarla Değerlendirme ve Seçim

- Her bir senaryonun üzerinden giderek
 - Tasarım alternatiflerinin senaryoyu ne kadar iyi desteklediğini değerlendirin.
 - Her bir senaryo için değerlendirme sonucunu kaydedin.

- Uygun bir teknik kullanarak alternatifler arasından seçim yapın.
 - Avantaj ve dezavantajlarına göre
 - Çok-boyutlu sıralama
 - Senaryo ağırlıkları normalize edilir
 - Değerlendirmeler sayısallaştırılır

Prototiplerle Değerlendirme ve Seçim

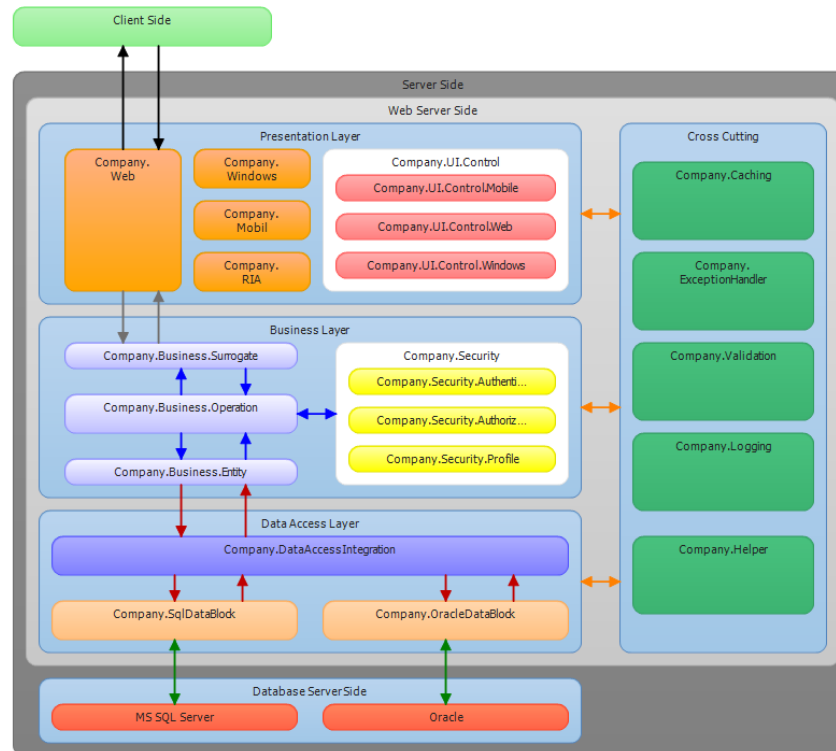
- Tasarım alternatiflerini sınamak için prototipler oluşturulabilir.
- Senaryoların üzerinden ilerlerken prototiplendirme ihtiyacı ortaya çıkabilir.
- Prototipler seçim yapabilmek için somut bir dayanak oluşturabilir.
 - Avantaj ve dezavantajlarına göre
 - Çok-boyutlu sıralama

Özet

- Mimari alternatifler oluşturmak ve iyileştirmek için birbirini tamamlayan çeşitli teknikler kullanılabilir.
- Ağırlıklandırılmış senaryolar içeren profiller oluşturmak ve bunlar üzerinden ilerlemek mimari alternatiflerini değerlendirmek için kullanılabilecek güvenilir bir tekniktir.
- Prorotipler de mimari değerlendirme için veri sağlayabilir.

KISIM 2

Yazılım Mimarilerinin Tamamlanması



Amaçlar

SAD kalite karakteristiklerini sunmak

Çeşitli gözden geçirme (review) türlerini araştırmak

Örnek bir mimari inceleme kontrol listesi sunmak

Aktif tasarım gözden geçirme sürecini ayrıntılı olarak sunmak

Tasarım sürecinde sürekli gözden geçirmenin (continuous review) önemini vurgulamak



İçerik

- SAD kalite karakteristikleri
- Gözden geçirmeler (Reviews)
- Gözden geçirme türleri
- Bir mimari inceleme kontrol listesi
- Aktif tasarım gözden geçirme süreci
- Sürekli gözden geçirme



SAD Kalite Karakteristikleri 1

- **Yapılabilirlik (Feasibility)**—SAD gerçekleştirilebilir (yapılabilir) bir programı belirtir.
- **Yeterlilik (Adequacy)**—SAD gereksinimleri karşılayacak bir programı belirtir.
- **İyi Biçimlendirilmişlik (Well-Formedness)**—SAD içindeki notasyonlar doğru şekilde kullanılmıştır.



SAD Kalite Karakteristikleri 1

- **Tamlık/Bütünlük (Completeness)**—SAT gereken tüm bölümleri içerir; tasarımı açıklamak için gereken tüm modelleri içerir; ve önemli tüm bileşen karakteristiklerini, ilişkilerini, etkileşimlerini vb. belirtir.
- **Netlik (Clarity)**—Probleme ve notasyonlara aşina herkes SAD dokümanını anlayabilir.
- **Tutarlılık (Consistency)**—Tek bir program SAD ile belirtilenleri karşılayabilir.

Gözden Geçirmeler (Reviews)

Gözden geçirme (review), bir ürünün ya da sürecin kalifiye elemanlardan oluşan bir ekip tarafından incelenmesi ve değerlendirilmesidir.



Gözden Geçirme Türleri

- **Desk Check**—Tasarımın bir tasarımcı tarafından değerlendirilmesidir
- **Walkthrough**—Tasarımın bir tasarımcı ekibine resmi olmayan bir biçimde sunulmasıdır
- **Inspection**—Eğitimli bir inceleme ekibi tarafından yapılan resmi bir gözden geçirmedir
- **Audit**—Tasarım ekibinin dışından çeşitli uzmanlar tarafından gerçekleştirilen gözden geçirmedir
- **Active Review**—Tasarımla ilgili spesifik soruları cevaplayan uzmanlar tarafından yapılan incelemedir

Bir Mimari İnceleme Kontrol Listesi

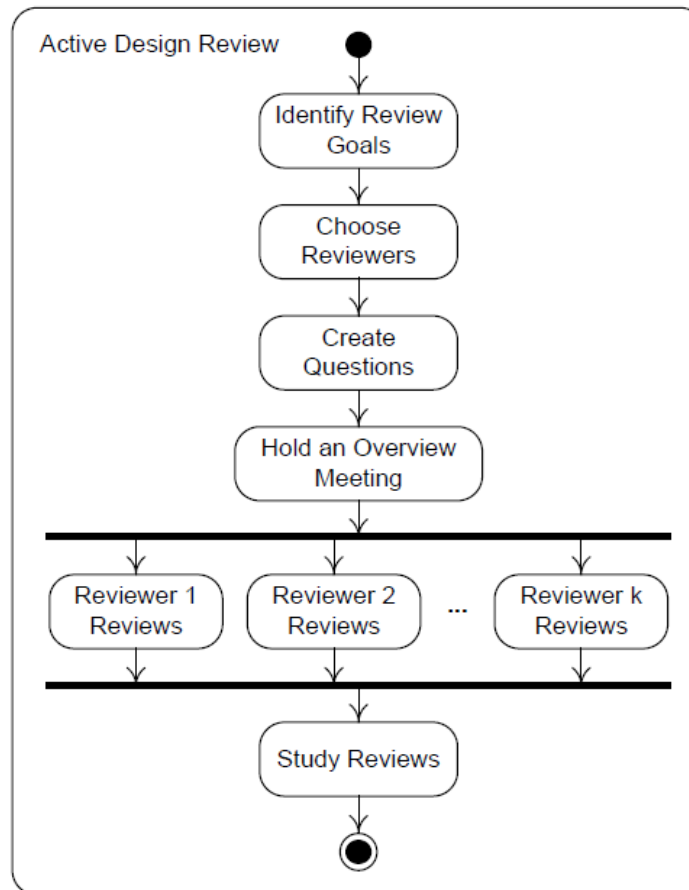
- ☐ The notations used for each model are correct.
- ☐ Every required section of the SAD is present.
- ☐ The SAD specifies the program's main components.
- ☐ The SAD specifies the states and state transitions for all components with important states.
- ☐ The SAD specifies important or complex component collaborations.
- ☐ The SAD specifies each component's responsibilities.
- ☐ The SAD specifies each component's interface.
- ☐ The SAD specifies each component's important properties.
- ☐ The SAD specifies each component's important relationships to other components.
- ☐ The SAD clearly states the connections between different architectural models.
- ☐ The SAD states the rationale for all important design decisions.
- ☐ Each design rationale states the problem to be solved and the constraints on the designer.
- ☐ Each design rationale summarizes the major design alternatives and their evaluations.
- ☐ Each design rationale explains why the final design was selected.
- ☐ All specifications are clear.
- ☐ No specification contradicts any other specification in the SAD.

Aktif Tasarım Gözden Geçirme

- Geleneksel gözden geçirmelerin problemlerine çare olur
 - Uzmanlık eksikliği
 - Gelişigüzel gözden geçirme
- Gözden geçirenleri dokümanın kendi uzmanlıklarıyla ilgili bölümlerini iyi anlamaya zorlar; bu kişiler tasarımın ayrıntılarıyla ilgili spesifik cevaplar verebilmelidir



Aktif Tasarım Gözden Geçirme Süreci



Gözden Geçirmeye Hazırlık

- **Gözden Geçirme Amaçlarını Belirleyin**—Tasarımcılar tasarımın kontrol edilmesi gereken yönlerini belirler.
- **Gözden Geçirme Ekibini Seçin**—Tasarımcıları 2-4 kişilik kalifiye ekip elemanı belirleyip inceleme yapmak üzere onaylarını alır.
- **Sorular Hazırlayın**—Tasarımcılar gözden geçiriciler tarafından cevaplanmak üzere sorular oluşturur.
 - Gözden geçiricileri tasarımı anlamaya zorlamak için
 - Gözden geçiricilere problem çözmeleri veya bir şeyi açıklamaları için vb. sorular sorulur.

Gözden Geçirmenin Gerçekleştirilmesi

- **Ön bilgilendirme Toplantısı Düzenleyin**—Tasarımcılar mimariyi, süreçleri vb. açıklarlar, son teslim tarihi belirlerler, vb.
- **Gözden Geçirme İşlemini Yapın**—Gözden geçiriciler işlerini kendi başlarına yaparlar.
 - Soru sormak, açıklama ve açıklık getirmek gibi amaçlarla tasarımcılarla görüşebilir, e-posta ile haberleşebilir vb.
 - Tamamlanınca sonuçları iletirler.

Gözden Geçirmenin Tamamlanması

- **Gözden Geçirme Sonuçlarını Çalışın**—Tasarımcılar gözden geçirme sonuçlarını çalışırlar.
 - Gözden geçiricilerle buluşarak ya da eposta ile sonuçlar üzerinde görüşebilirler.



Sürekli Gözden Geçirme

- **Bir hata (defect) ne kadar çabuk düzeltilirse o kadar ucuza düzeltilir.**
- Tasarım artefact'ları tamamlandığında son kalite kontrol için mutlaka gözden geçirme yapılmalıdır.
- Gözden geçirmeler olası hataları erken yakalamak için tasarım süreci boyunca da yapılmalıdır.
- Tasarım sürecinin farklı aşamalarında farklı tür gözden geçirmeler uygulanabilir.



Özet

- Bir SAD dokümanı büyük ölçüde tamamlandığında mutlaka gözden geçirilmelidir:
 - Yapılabilir ve yeterli bir mimari belirtiyor mu?
 - İyi biçimlendirilmiş modellerden oluşuyor mu?
 - Tam, açık ve tutarlı mı?
- Çeşitli gözden geçirme türleri kullanılabilir.
- Aktif gözden geçirme oldukça etkili bir gözden geçirme şeklidir.
- Tasarım artefact'ları tamamlandıkça ve tasarım süreci boyunca gözden geçirmeler mutlaka yapılmalıdır.

Kaynaklar

“Software Engineering A Practitioner’s Approach” (7th. Ed.), Roger S. Pressman, 2013.

“Software Engineering” (8th. Ed.), Ian Sommerville, 2007.

“Guide to the Software Engineering Body of Knowledge”, 2004.

” Yazılım Mühendisliğine Giriş”, TBİL-211, Dr. Ali Arifoğlu.

”Yazılım Mühendisliği” (2. Basım), Dr. M. Erhan Saridoğan, 2008, İstanbul: Papatya Yayıncılık.

Kalıpsız, O., Buharalı, A., Biricik, G. (2005). Bilgisayar Bilimlerinde Sistem Analizi ve Tasarımı Nesneye Yönelik Modelleme. İstanbul: Papatya Yayıncılık.

Buzluca, F. (2010) Yazılım Modelleme ve Tasarımı ders notları (<http://www.buzluca.info/dersler.html>)

Hacettepe Üniversitesi BBS-651, A. Tarhan, 2010.

Yazılım Proje Yönetimi, Yrd. Doç. Dr. Hacer KARACAN

YZM211 Yazılım Tasarımı – Yrd. Doç. Dr. Volkan TUNALI

http://www.cclub.metu.edu.tr/bergi_yeni/e-bergi/2008/Ekim/Cevik-Modelleme-ve-Cevik-Yazilim-Gelistirme

http://wiki.expertiza.ncsu.edu/index.php/CSC/ECE_517_Fall_2011/ch6_6d_sk

<http://dsdmofagilemethodology.wikidot.com/>

<http://caglarkaya.piququestion.com/2014/07/01/244/>