

Beykent Üniversitesi
Yazılım Mühendisliği Bölümü
Yazılım Mühendisliği Tasarım Projesi

Rapor 1
2021 – GÜZ

Proje İsmi (kısaltması ile birlikte)

Grup Elemanları

No, Adı Soyadı (Soyadına Göre Alfabetik Sırada)

I Proje Tanımı (16 font)

1 Projeye Genel Bakış (14 font)

Ayrıntılara girmeden geliştirilecek ürün ile ilgili kısa bir tanımlama yapılır. (12 font)

2 Projenin Amacı (14 font)

Takım bu projeyi niçin seçmiştir? Projenin kapsamı nedir?

Proje çalışması ile yapılması hedeflenenlerin özetlenmesi istenir. Bu bölümdeki açıklamalar projenin gerekçesini oluşturacaktır. Böylece problemin proje olarak seçiminin önemi açıklanmış olacaktır.

Özetle, niçin bu ürünün geliştirileceği açıklanır. Geliştirmenin her aşamasında burada açıklanan ifadelerin sağlanıp sağlanmadığının kontrol edilmesi gerekecektir.

3 Projenin Kapsamı (14 font)

3 a. Mevcut Durum

Geliştirilecek ürünün proje takımı tarafından ilk defa geliştirilmeyeceği kabul edilmektedir. Gerçek yazılım ürünlerinin pek çoğu da önceden geliştirilmiş ürünlerdir. İş analistleri her projede eski ürün yerine yenisinin niçin geliştirilmesi ya da önceden elle yapılan işlerin neden otomatikleşmesi gerektiğini araştırmalıdır.

3 b. Projenin İçeriği

Proje çalışmasına ait “component diagram” çizilmesi istenmektedir (<https://www.uml-diagrams.org/component-diagrams.html>).

UML diyagramlarının “visual paradigm community edition” kullanılarak tasarımı önerilmektedir. Çevrimiçi erişilebilen UML linkleri tasarım projenizin diyagramlarını çizmek için yeterli olmayacaktır. UML araçları kullanılmadan çözümlenmemelidir.

Bu diyagram ile projenin sınırları belirlenmiş olacak ve bu çalışmanın niçin yapıldığı görsel olarak betimlenecektir. Özetle geliştirilecek ürün tüm alt bileşenleri ile birlikte açıklanmış olacaktır.

Çizdiğiniz tüm diyagramlar şeklinizin altında açıklanmalıdır.

Örneğin:

Şekil 3.1:projesine ait “component” diyagramı

Anlamı tam olarak vermeyen Türkçe ifadeler nedeni ile diyagram isimlerinin orijinal olarak kullanılması önerilir.

3 c. Rakip Ürün Karşılaştırması

Geliştirilecek ürün mevcut iken niçin tekrar üretiliyor? Belli özelliklere sahip alternatifler kullanılıyor ise, mevcut sistemde ne gibi eksiklikler görüldü ki ürünün yeniden geliştirilmesi gereksinimi doğdu?

4 Ürün Senaryoları

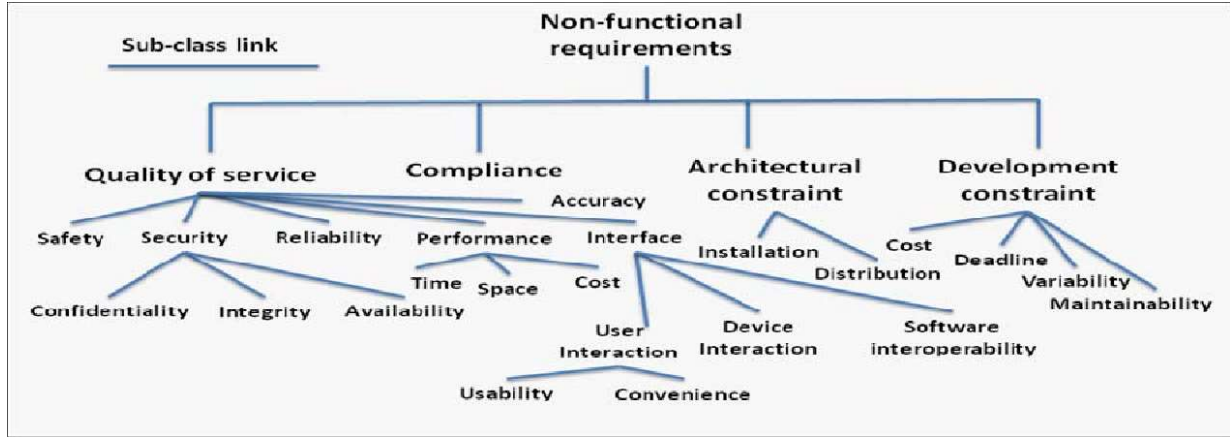
Senaryolar, kullanıcıların ürünün geliştirilmesi sonunda nasıl kullanacağını açıklayan formal olmayan hikâyelerdir ve ürün senaryoları listesi şeklinde (1,2,3,4,...olarak) sözel olarak açıklanacaktır.

Özetle II. Bölümde, Gereksinimlerin Analizi Bölümünde, kullanılacak olan her bir use case diyagramına ait ürün senaryolarının (diyagram isimlerinin) listelenmesidir. Bu liste her bir senaryo için numaralandırılır. Böylece çalışma grubu ürün geliştirilirken kaç tane “use case”

diyagramı kullanılacağına da karar vemiş olacaktır. Bu bölümde ürün senaryolarını açıklamak üzere “use case” diyagramları listelenir, kesinlikle tasarımı yapılmaz.

5 Kısıtlar

Çözülecek probleme özgü fonksiyonel olmayan gereksinimlerin tanımı, gerekçeleri ile birlikte, yapılmalıdır. Aşağıdaki fonksiyonel olmayan gereksinimler sınıflandırmasından probleminiz için tanımlanması gerekli fonksiyonel olmayan gereksinimler seçilecek problem alanına göre tanımlanacaktır.



Axel Van Lamsweerde for Non-Functional Requirements on Software Services

6 Tanımlar Proje içerisinde kullanılacak simgeler ve kısaltmalar belirlenecek ve listelenecektir.

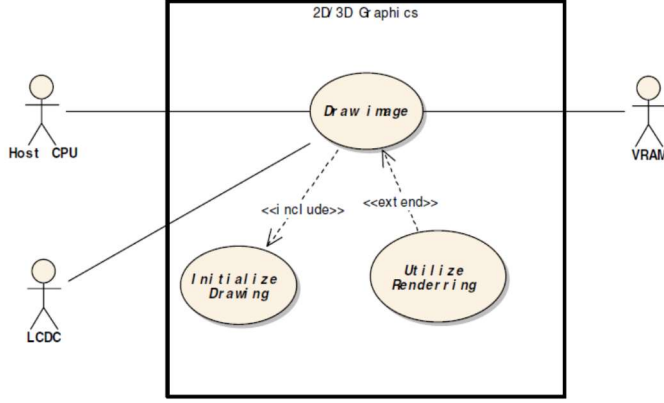
II Gereksinimlerin Analizi (16 font)

7 “Use Cases” ve Tablo Açıklamaları (14 font)

“Uses cases” (kullanım durumları), hem geliştireceğiniz yazılım sistemini ve sistemin sınırlarını ayrıntılı olarak tanımlamanızı, hem de fonksiyonel gereksinimleri tanımlamanızı sağlar. Böylece gereksinimlerin belirlenmesi ve bunların tablo üzerinde açıklanmasından sonra, sisteminizin sınıflarını ve nesnelerini tanımlayacağınız ve aralarındaki ilişkileri tasarlayacağınız tasarım aşamasına geçilecektir. Bu bölümde tanımlanmış ve açıklanmış olan “use case” diyagramları çözümlerinin tasarım aşamasındaki tüm diyagramlara değiştirilmeden yansıtılması zorunludur. Sonraki raporda bu rapordaki “use case” diyagramları ile örtüşmeyen çözümler değerlendirilmeye alınmaz.

Her bir “use case” diyagramı ve bu diyagramın açıklaması için aşağıda bir örnek verilmiştir. . (<https://www.uml-diagrams.org/use-case-diagrams.html>).

Use case diyagramları ve tablolar sıralı olarak numaralandırılmalıdır. Şekil 7.1, Tablo 7.1, Şekil 7.2, Tablo 7.2 , vs. ...



Şekil 1: Use case diyagramı örneği (hani çözüm olduğu açıklanır)

Tablo1 : Use case diyagramı açıklaması örneği (hangi çözümün açıklaması olduğu yazılır)

Use Case	Draw Image
Pre-condition	<ul style="list-style-type: none"> • Frame buffer must be cleared • Reset must switch off
Post-condition	<ul style="list-style-type: none"> • Image data output to VRAM
Basic path	<ol style="list-style-type: none"> 1. This use case starts while Host CPU has already prepared the display list for drawing image. 2. Host CPU writes display list system's FIFO. 3. 2D/3D Graphics reads display list from FIFO and starts drawing image according to commands of display list. 4. 2D/3D Graphics outputs image data to VRAM.
Alternative path	<p>At step 2 of the basic path, Host CPU writes display list that includes SYNC command to system's FIFO.</p> <p>At step 3 of the basic path, 2D/3D Graphics stops after reading SYNC command, and then 2D/3D graphics restarts drawing image after receiving a blank pulse from LCDC.</p>
Exceptional path	<p>At step 2 of the basic path, Host CPU writes display list includes undefined commands.</p> <p>At step 3 of basic path, 2D/3D graphics raises an error interrupt to Host CPU and clears system's FIFO.</p>

Not: Şekil ... açıklaması şeklin altına, Tablo ... açıklaması tablonun üstüne yapılır.

“use cases” (kullanım durumları) sayısı, grup elemanlarının sayısına göre ve bu bağlamda gerçekleştirilecek projenin kapsamına göre 15 -20 arasında olmalıdır. Bu nedenle de her bir ürün kullanım durumunun (use case) bir listesini yapmak, her birini ayrı ayrı modellemek veya tanımlamak proje çalışmasının analiz safhasının takibini kolaylaştıracaktır.

Kullanım durumu diyagramlarının geliştirilecek sistem üzerinde iki amacı vardır. Her bir use case (kullanım durumunu) listeleyen tablo, hem bir diyagramın sistemin bir parçası olarak neler içerdiği (hangi eylemleri) ve sınırlamaları ifade etmek üzere nelerin sistemde olmadığına listesidir.

Her bir “use case” diyagramının problemin ilgili parçasının kavramsal çözümünde “**include**” ve/veya “**extend**” ilişkilerini içermek zorundadır.

Aktörlerin canlı ve/veya cansız bir varlık olması mümkündür.

Çözdüğünüz problemin karmaşıklığına bağlı olarak, bazı kullanım durumlarını (use cases) listelemek için birden fazla diyagram kullanılması mümkündür. Örneğin herhangi bir kullanım durumu (use case) için birden fazla şema gerekirse, kullanım alanları birkaç yola ayrılabilir: normal işlemlere karşı göz önüne alınmayan durumlar veya günlük işlemlere karşılık aylık işlemler veya kullanıcı görevlerine karşılık yönetim işleri gibi.

Proje grubu “use cases” ile ilgili olarak geliştireceği ürünün sınırının ne olması gerektiğine karar vererek ürün kullanım durumlarını (product use cases), diğer bir ifade ile “use cases” tasarlayarak probleminin gereksinimler analizini tamamlayacaktır. Bu sınırın ne olacağı (ne büyüklükte olacağı, “use case” diyagramlarının sayısı ve birbirine bağlılığı) çalışma konusundaki bilgilerinize ve gereksinim kısıtlamalarınıza göre belirlenmelidir.

8 Fonksiyonel Gereksinimler

Her bir olay (event) ya da diğer bir ifade ile “use case” kullanımı listesi için fonksiyonel gereksinimler aşağıdaki tabloya yazılacaktır. Her bir fonksiyonel gereksinime tek (unique) bir sayı verilecektir. Bu bölüm çalışmanın izlenebilirliğini (traceability) ve geliştirme aşamasını kolaylaştıracaktır.

Tablo 2 Türkçe verilmemiştir. Gruplar bu bölümü Türkçe gelecek zaman (shall) cümleleri ile tasarlayacaklardır.

Tablo 2: Problem Gereksinimleri Tanımı

Section/ Requirement ID	Requirement Definition (Gereksinim Tanımı)
FR1.0.	The system shall [parent requirement group 1].
FR1.1	The system shall [child/parent requirement].
FR1.1.1	The system shall [child requirement].
FR1.1.2	The system shall [child requirement].

