
IREB Certified Professional for Requirements Engineering

- Foundation Level -

Syllabus

Version 2.2

Month Day 2014^[m1]

(based on the English Version 2.2)

본 실라버스 사용에 앞서:

1. 개인 및 교육 제공자는 이 용어집을 세미나의 기본 자료로 활용할 수 있으며, 이 경우 저작권이 분명하게 세미나 자료 내에 표시되어야 한다. 홍보 목적으로 본 용어집을 활용해야 할 경우 문서로 작성된 IREB의 승인을 받아야 한다.
2. 투고, 서적 및 기타 출판 매체의 기본 자료로 요구공학 용어집을 활용할 경우, 용어집의 저자 및 IREB e.V.의 저작권을 해당 매체 내에 표시하여야 한다.

© IREB e.V.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without either the prior written permission of the authors or IREB e.V.

알리는 글

본 실라버스는 다음에 열거된 위원회 멤버들이 집필했습니다: Karol Frühauf, Emmerich Fuchs, Martin Glinz, Rainer Grau, Colin Hood, Frank Houdek, Peter Hruschka, Barbara Paech, Klaus Pohl and Chris Rupp. 또한 위원회 멤버들은 다음에 열거된 IREB 멤버들의 지원을 받고 있습니다: Ian Alexander, Joseph Bruder, Samuel Fricker, Günter Halmans, Peter Jaeschke, Sven Krause, Steffen Lentz, Urte Pautz, Suzanne Robertson, Dirk Schüpferling, Johannes Staub, Thorsten Weyer and Joy Beatty.

한국어 번역은 김모세님이 담당하고 진행하여 완성되었습니다. 김모세님 포함 실라버스 제작에 참가해 주신 모든 분들의 헌신에 진심으로 감사드립니다.

Copyright © 2009 – 2012. 이 실라버스는 위 저자들의 저작권을 보호합니다. 저작권은 IREB 국제 요구공학 위원회 e.V.에 귀속됩니다.

서문

실라버스의 목적

이 실라버스는 국제 요구공학 위원회(International Requirements Engineering Board, IREB)가 제정한 “공인 요구공학 전문가”의 Foundation Level 인증 관련 내용을 정의합니다. IREB는 본 실라버스와 관련된 시험을 다양한 언어로 제공합니다. 본 실라버스는 교육 제공자가 교재를 작성하기 위한 기본 자료로 활용할 수 있습니다. 학습자들은 시험 준비를 위해 본 실라버스를 개인 학습 교재로 활용할 수 있습니다.

실라버스의 내용

Foundation Level은 요구공학 업무와 관련된 모든 사람들에게 필요한 내용을 담고 있습니다. 프로젝트 혹은 IT 관리자, 도메인 전문가, 시스템 분석가 및 소프트웨어 개발자 등이 포함됩니다.

내용 범위

Foundation Level은 모든 영역(즉, 임베디드 시스템, Safety-critical 시스템, 전통적인 정보 시스템 등)에 동일하게 적용되는 기본 사항을 다룹니다. 다시 말하면 특정 영역에 한정적으로 적용되는 접근법이나, 해당 영역의 특수성을 반영하는 내용은 교육 과정에서 다루지 않습니다.

계획 수립이나, 교육 과정 중 학습한 내용의 적용을 위한 특정 기법과 같은 특정한 수행 절차나 프로세스 모델은 실라버스에서 제공하지 않습니다. 요구공학의 특정 프로세스나 소프트웨어 공학 전체를 강조하는 것은 실라버스의 목적에 부합하지 않습니다.

요구공학 엔지니어에게 필요한 지식 체계는 정의하나, 소프트웨어 공학의 다른 원리들이나 프로세스와 직접적으로 연관되는 내용은 다루지 않습니다.

세부 레벨

본 실라버스는 세부적인 레벨을 사용해 범세계적으로 일관적인 교육과 시험이 수행됨을 목표로 합니다. 이와 같은 목표를 달성하기 위해, 실라버스는 다음의 내용을 포함하고 있습니다:

- ▶ 일반적인 교육 목표
- ▶ 교육 목표에 대한 설명을 포함한 콘텐츠
- ▶ (필요한 경우) 다른 교재 참조 링크

교육 목표 / 지식 기억 레벨

본 실라버스의 각 모듈에는 기억 레벨 값이 할당되어 있습니다. 상위 기억 레벨은 하위 기억 레벨을 포함합니다. 기억 레벨 1에 해당하는 교육 목표들은 “이해하기(Knowing)”이라는 용어를 사용하여 기술하며, 기억 레벨 2에 해당하는 교육 목표들은 “활용하기(Mastering and Using)”이라는 용어를 사용하여 기술합니다. 해당 용어들은 다음과 같은 의미를 내포하고 있습니다:

- ▶ L1 (이해하기/ knowing): 나열, 특성, 인지, 이름, 반영
- ▶ L2 (활용하기/ mastering and using): 분석, 사용, 실행, 정확화, 세부적인 기술, 판단, 표시, 설계, 개발, 완료, 설명, 예시, 도출, 정규화, 식별, 해석, 결론 도출, 할당, 차이점 기술, 비교, 이해, 제안, 요약

용어집에 정의된 용어들은 실라버스의 교육 목표에 직접 언급되지 않은 경우에도 L1로 간주됩니다.

이 실라버스에서는 요구공학을 RE(Requirements Engineering)으로 표기합니다.

실라버스의 구조

실라버스는 9개 주 챕터로 구성되어 있습니다. 챕터 별로 하나의 교육 단위(EU, Educational unit)를 다룹니다. 각 주 챕터의 제목은 해당 챕터의 기억 레벨을 포함하고 있으며, 하위 챕터의 기억 레벨 중 가장 높은 기억 레벨과 동일합니다. 또한 해당 챕터의 내용을 교육하는 데 필요한 최소한의 소요 시간을 제안합니다. 해당 챕터에 등장하는 주요 용어들은 챕터의 앞 부분에 리스트되어 있습니다.

예: EU1 도입(L1)

교육 시간: 1 ¼ 시간

관련 용어: 요구사항, 이해관계자, 요구공학, 기능 요구사항, 품질 요구사항, 제약사항

위의 예는 “챕터 1은 L1의 교육 목표를 가지고 있으며, 해당 챕터의 교육을 위해 최소 75분 정도가 필요함”을 의미합니다.



각 챕터는 여러 하위 챕터를 포함하고 있습니다. 하위 챕터의 제목들 역시 해당 챕터에서 다루는 기억 레벨 내용을 포함하고 있습니다.

교육 목표는 실제 본문이 시작되기 전에 나열됩니다. 교육 목표의 번호는 해당 내용이 어느 챕터에 포함되어 있는지를 표시합니다.

예: EO 3.1.2

위의 예는 “해당 교육 목표가 하위 챕터 3.1 에서 설명되어 있음”을 의미합니다.

자격 시험

본 실라버스는 foundation level 자격 시험을 위한 기본 자료입니다.



자격 시험에서는 한 문제가 여러 챕터의 내용을 동시에 포함하고 있을 수 있습니다. 모든 챕터(EU 1~EU 9)가 출제 범위에 포함됩니다.

자격 시험 문제는 다지선다형 객관식입니다.

자격 시험은 정규 교육 과정 종료 직후 실시되며, 경우에 따라 별도의 일정으로 실시될 수 있습니다(예: 공인 시험 기관에 의한 경우). 공인 시험 기관 리스트는 인터넷 홈페이지에서 확인할 수 있습니다(<http://www.certified-re.de/en>)

문서 버전 정보

| Version | 일자 | 내용 |
|---------|---------|----------------------------|
| 2.2 | 2014.10 | 최초 버전(영문 실라버스 2.2 기준으로 번역) |

목차

| | |
|--|----|
| 알리는 글 | 2 |
| 서문 | 2 |
| 문서 버전 정보..... | 5 |
| 목차 | 6 |
| EU 1 도입 및 기초 (L1)..... | 8 |
| EU 2 시스템과 시스템 컨텍스트(L2) | 10 |
| EU 2.1 시스템, 시스템 컨텍스트 및 시스템 바운더리 (L1) | 10 |
| EU 2.2 시스템 바운더리 및 컨텍스트 바운더리 결정 (L2) | 10 |
| EU 3 요구사항 도출 (L2) | 12 |
| EU 3.1 요구사항 소스(L1) | 12 |
| EU 3.2 카노 모델(Kano Model)에 따른 요구사항 분류 (L2)..... | 13 |
| EU 3.3 요구사항 도출 기법 (L2)..... | 13 |
| EU 4 요구사항 문서화 (L2) | 14 |
| EU 4.1 문서 설계 (L1) | 14 |
| EU 4.2 문서화 종류 (L1)..... | 14 |
| EU 4.3 문서 구조 (L1) | 15 |
| EU 4.4 요구사항 문서의 활용 (L1)..... | 15 |
| EU 4.5 요구사항 문서 품질 기준 (L2)..... | 16 |
| EU 4.6 요구사항 품질 기준 (L2) | 16 |
| EU 4.7 용어집 (L2)..... | 16 |
| EU 5 일상어를 통한 요구사항 문서화 (L2) | 18 |
| EU 5.1 언어 효과 (L2) | 18 |
| EU 5.2 템플릿을 활용한 요구사항 구조화 (L2) | 18 |
| EU 6 모델 기반 요구사항 문서화 (L2)..... | 20 |
| EU 6.1 모델 (L1) | 20 |
| EU 6.2 골 모델 (L2)..... | 21 |
| EU 6.3 유스케이스 (L2)..... | 21 |
| EU 6.4 요구사항을 바라보는 세 가지 관점 (L1)..... | 22 |
| EU 6.5 데이터 관점에서의 요구사항 모델링 (L2)..... | 23 |
| EU 6.6 기능 관점에서의 요구사항 모델링 (L2)..... | 23 |
| EU 6.7 행동 관점에서의 요구사항 모델링 (L2) | 24 |
| EU 7 요구사항 검증 및 협의 (L2) | 26 |
| EU 7.1 요구사항 검증 기초 (L1) | 26 |
| EU 7.2 요구사항 협의 기초 (L1) | 26 |
| EU 7.3 요구사항과 관련된 품질 측면 (L2) | 27 |

| | |
|----------------------------------|-----------|
| EU 7.4 요구사항 검증 원칙 (L2)..... | 27 |
| EU 7.5 요구사항 검증 기법 (L2)..... | 28 |
| EU 7.6 요구사항 협의 (L1)..... | 28 |
| EU 8 요구사항 관리 (L2) | 30 |
| EU 8.1 요구사항 속성 할당하기 (L1)..... | 30 |
| EU 8.2 요구사항 뷰 (L2) | 31 |
| EU 8.3 요구사항 우선 순위 선정하기 (L2)..... | 31 |
| EU 8.4 요구사항 추적성 (L2) | 32 |
| EU 8.5 요구사항 버전 관리하기 (L2)..... | 32 |
| EU 8.6 요구사항 변경 관리 (L2)..... | 33 |
| EU 8.7 요구사항 측정 (L1)..... | 34 |
| EU 9 도구 지원 (L1)..... | 35 |
| EU 9.1 도구의 종류 (L1) | 35 |
| EU 9.2 도구 도입하기 (L1)..... | 35 |
| EU 9.3 도구 평가하기 (L1)..... | 36 |

EU 1 도입 및 기초 (L1)

| | |
|-------|---|
| 교육 시간 | 1 ¼ 시간 |
| 관련 용어 | 요구사항, 이해관계자, 요구공학, 기능 요구사항, 품질 요구사항, 제약사항 |

교육 목표:

| | |
|--------|--|
| EO 1.1 | 요구공학을 적절하게 수행하지 못하는 이유와 그에 따라 발생하는 현상(혹은 증상)을 이해한다 |
| EO 1.2 | 요구공학의 주요한 네 가지 영역을 이해한다 |
| EO 1.3 | 요구공학에서 의사소통의 역할을 이해한다 |
| EO 1.4 | 요구공학 엔지니어에게 필요한 역량을 이해한다 |
| EO 1.5 | 요구사항의 세 가지 형태를 이해한다 |
| EO 1.6 | 품질 요구사항의 역할을 이해한다 |

요구공학을 적절하게 수행하는 것은 매우 중요하다. (프로젝트의 수명 주기에서 발생할 수 있는) 많은 오류(Error)들이 이 단계(요구사항을 정의하는 단계)에서 발생하고, 이러한 오류들은 프로젝트가 진행될 수록 더 많은 비용을 들여야만 조치가 가능하기 때문이다. 요구공학을 적절하게 수행하지 못하는 경우 요구사항이 누락되거나 모호하게 기술되는 것이 일반적인 현상이다. 요구공학을 적절하게 수행하지 못하는 이유는 주로 다음과 같다:

- ▶ 요구사항이 그 자체로 명백하기 때문에, 별도의 구체적인 기술이 필요하지 않다고 이해관계자들이 잘못 가정하는 경우
- ▶ 이해관계자들의 경험이나 지식의 차이로 인한 의사소통의 문제에 의한 경우
- ▶ 생산적인 시스템을 빠른 시간 내에 만들어 달라고 하는 등, 고객이 프로젝트 진척과 관련해 강한 압력을 가하는 경우

요구공학에서 수행되는 주요 활동 (요구사항)도출, (요구사항)문서화, (요구사항)검증/합의, (요구사항)관리의 네 가지로 분류한다. 이 활동들은 ISO/IEC/IEEE 29148:2011 표준에서 권장하는 특정한 프로세스에 맞추어 스케줄링 될 수 있다. 요구사항은 이해관계자 관점의 요구사항, 시스템 관점의 요구사항과 같이 다양한 레벨로 구분되기도 한다.

일상어(Natural Language)는 요구사항 의사소통에서 가장 중요한 수단이며, 공통으로 사용할 용어와 관련된 합의를 진행하는 경우에도 매우 중요한 역할을 담당한다. 또한 의사소통 수단—기록물이든 구술이든—으로서도 매우 중요한 역할을 하기 때문에, 의사소통에 있어 모든 참여자들은 충분한 주의를 기울여야 하며, 가장 단순한 형태로 다듬어진 용어를 사용하여야 한다.

요구공학에서 가장 중요한 역할을 하는 구성 요소는 당연히 요구공학 엔지니어이다. 요구공학 엔지니어는 의사소통 역량은 물론 다음과 같은 지식들을 반드시 습득하여 자유롭게 활용할 수 있어야 한다: 분석적 사고 능력, 감정이입 능력(empathy), 충돌 해소 능력, 중재 능력, 자신감 및 설득력

요구사항은 일반적으로 기능 요구사항, 품질 요구사항 및 제약사항의 세 가지 형태로 분류한다.

“비기능 요구사항”은 넓은 의미에서 품질 요구사항과 제약사항을 동시에 의미한다. 품질 요구사항은 분명하게 문서화되어야 하며, 이 경우 일반적으로 다음과 같은 측면을 고려하여야 한다:

- ▶ 성능(Performance)
- ▶ 보안성(Security)
- ▶ 신뢰성(Reliability)
- ▶ 사용성(Usability)
- ▶ 유지보수성(Maintainability)
- ▶ 이식성(Portability)

품질 모델과 관련된 보다 자세한 내용은 요구공학 literature, 혹은 ISO/IEC 25010:2011 표준 등을 참고한다.

품질 요구사항들이 일상어를 사용하여 대부분 문서화 되어 있는 경우라도, 해당 요구사항들이 다른 요소들과 어떻게 연관되어 있는지 확인할 수 있어야 하며, 정량적인 방법이나 추가적인 기능으로 변환하여 검증이 가능해야 한다.

EU 2 시스템과 시스템 컨텍스트(L2)

교육 시간: 1 ¼ 시간

관련 용어: 시스템 컨텍스트(System Context), 시스템 바운더리(System Boundary), 컨텍스트 바운더리(Context Boundary)

교육 목표:

EO 2.1 시스템 컨텍스트, 시스템 바운더리, 컨텍스트 바운더리를 이해한다

EO 2.2 시스템 바운더리, 컨텍스트 바운더리를 활용한다

EU 2.1 시스템, 시스템 컨텍스트 및 시스템 바운더리 (L1)

시스템 자원 및 시스템 요구사항의 타당성 여부는 대상 시스템의 시스템 컨텍스트를 통해 판단한다. 시스템 자원은 요구사항을 결정하는 데 영향을 미치거나, 요구사항 그 자체를 결정하는 것과 관련된 모든 컨텍스트의 집합으로 구성된다. 다음과 같은 (잠재적인) 요소들이 시스템 컨텍스트에 포함될 수 있다:

- ▶ 사람(People), 즉 이해관계자 혹은 이해관계자 그룹
- ▶ 운용중인 시스템(System in operation), 즉 기술적인 시스템, 소프트웨어 및 하드웨어
- ▶ 프로세스(Processes), 즉 기술적 혹은 물리적 프로세스, 비즈니스 프로세스
- ▶ 이벤트(Events), 즉 기술적 혹은 물리적인 이벤트
- ▶ 문서(Documents), 즉 법률이나 표준 및 시스템 문서

시스템 바운더리는 어떠한 영역이 계획된 시스템에 의해 다루어지며, 해당 시스템의 환경은 어떻게 구성될 것인지를 결정하는 역할을 한다. 또한 컨텍스트 바운더리는 개발 대상이 되는 시스템과 이를 연결하는 외부 시스템(혹은 환경)을 식별하는 역할을 한다.

EU 2.2 시스템 바운더리 및 컨텍스트 바운더리 결정 (L2)

시스템 바운더리는 요구사항 프로세스의 막바지에 이르러서야 명확하게 정의되는 경우가 일반적이다. 그 이전 단계까지는 시스템의 필요한 기능이나 품질은 불완전하게 인식(공유)되거나 전혀 알려지지 않은 상태로 머물러 있다. 따라서 시스템 바운더리를 정의해야 하는 그레이존(Grey zone)이라는 것이 만들어진다. 또한 시스템 바운더리를 그레이존 내에 설정한 경우라 하더라도, 요구사항 프로세스가 진행되는 동안 그레이존 자체가 변경될 수 있기 때문에(시스템 바운더리가 이동할 수 있기 때문에), 시스템을 둘러싼 환경에 대해 더욱 많은 주의를 기울여야 한다.

마찬가지로 컨텍스트 바운더리 역시 시간이 흐를수록 변경된다. 예를 들면, 초기 계획 당시 예상했던 것과는 다르게, 계획 초기에는 관련이 있을 것으로 구분되었던 법적인 요구사항이 개발 대상 시스템과 전혀 관련없는 것으로 판명된 경우, 법적 요구사항이 정의하는 영역의 시스템 컨텍스트는 제외된다.

컨텍스트 바운더리 역시 그레이존을 가진다. 그레이존은 특정 시점에서 식별된 환경 요소들 중, 개발 대상 시스템과의 관련성의 불분명한 요소들로 구성된다.

유스케이스 다이어그램(Use case diagrams)이나 데이터 흐름 다이어그램(Data flow diagram)을 사용해 일반적으로 시스템 컨텍스트(그 중에서도 시스템 바운더리 및 컨텍스트 바운더리)를 표현한다. 컨텍스트 모델링(Context modeling) 수행 시, 데이터 흐름 다이어그램에 기반하여 시스템 환경 내부의 자원과 각 자원 간 연결 구조를 모델링한다. 시스템 자원 및 자원 연결구조는 개발 대상 시스템과 시스템을 둘러싼 환경 사이에서의 데이터 흐름의 시작과 끝을 의미한다. 시스템 환경에서 액터(actor, 예를 들면 사람 혹은 다른 시스템) 및 개발 대상 시스템의 사용과 관련된 연관성은 유스케이스 다이어그램을 통해 모델링한다.

EU 3 요구사항 도출 (L2)

교육 시간: 1 ½ 시간

관련 용어: 없음

교육 목표:

- EO 3.1.1 다양한 요구사항 출처의 종류를 이해한다
- EO 3.1.2 요구사항 출처의 중요성과 요구사항 출처 누락의 결과를 이해한다
- EO 3.1.3 이해관계자 문서와 관련된 가장 중요한 정보를 이해한다
- EO 3.1.4 이해관계자와 관련된 중요한 원칙들을 이해한다(이해관계자의 권리와 의무(rights and duties))
- EO 3.2.1 카노 모델(Kano model)의 내용과 중요성을 이해하고 활용한다
- EO 3.3.1 (요구사항) 도출 기법 선택에 영향을 주는 요소들을 이해한다
- EO 3.3.2 다양한 (요구사항) 도출 기법의 장점과 단점을 이해한다
- EO 3.3.3 다양한 (요구사항) 도출 기법을 이해하고 활용한다: 서베이(Survey) 기법, 창의적(Creativity) 기법, 문서 중심(Document-centered) 기법, 관찰(Observation) 기법 및 지원(Supporting) 기법

EU 3.1 요구사항 **소스(출처)**(L1)

요구공학 활동에서 개발 대상 시스템과 관련된 요구사항을 도출하는 활동은 매우 중요하다. 요구사항은 시스템 컨텍스트와 요구사항 출처에서 도출한다. 수없이 다양한 요구사항 출처가 존재하는데, 일반적으로 이해관계자, 문서 및 기존 시스템 등이 이에 해당한다.

다양한 요구사항 출처로부터 구체적 목표와 요구사항을 수집해 내는 과정이 요구공학을 통해 이루어진다. 요구사항 출처가 누락되는 경우, 전체 프로젝트에 걸쳐 매우 심각하고 부정적인 결과를 초래할 수 있다. 이해관계자가 요구사항 출처인 경우, 요구사항 출처와 관련된 문서는 다음의 정보들을 포함하고 있어야 한다:

- ▶ 이름
- ▶ 기능(function/ role)
- ▶ 추가적인 인적 정보(personal and contact data)
- ▶ 프로젝트 공수 투입 가용성(temporal and spatial availability)
- ▶ 연관성(relevance of the stakeholder)
- ▶ 전문 영역과 레벨(area and extent of expertise)
- ▶ 프로젝트와 관련된 당사자의 구체적 목표 및 관심사(goals and interests)

조직 문화에 따라 다를 수 있으나 이해관계자화의 합의하에, 구두 혹은 문서를 통해 이해관계자의 업무(task), 책임(responsibility) 및 권한(authority) 등에 관해 정의해 두는 것이 바람직하다. 각 이해관계자 별로 책임과 의무에 대한 합의를 해야 한다. 이해관계자들을 효과적으로 통제하여 프로젝트에 대해 지속적인 관심과 동기를 유지하고 충돌이 최소한으로 발생하도록 해야 한다. 이해관계자들은 단지 프로젝트의 영향을 받는 것이 아니라 능동적으로 프로젝트에 참여해야 한다.

EU 3.2 카노 모델(Kano Model)에 따른 요구사항 분류 (L2)

요구사항을 도출하는 경우, 각 요구사항이 이해관계자들의 만족 여부에 얼마나 영향을 미치는지 파악하는 것이 매우 중요하다. 카노 박사(Dr. Kano)가 제안한 모델에 따르면, “만족도”는 다음 세 가지로 구분할 수 있다:

- ▶ Basic factors(동의어: Dissatisfiers)
- ▶ Performance factors(동의어: Satisfiers)
- ▶ Excitement factors(동의어: Delighters)

EU 3.3 요구사항 도출 기법 (L2)

요구사항 도출 기법은 이해관계자의 요구사항(의식적인, 무의식적인 혹은 잠재의식 속에 있는)을 찾아내기 위해 사용한다. 요구사항 도출 기법을 선정하는 경우 리스크 요소(risk factors), 사람이 미치는 영향(human influences), 조직이 미치는 영향(organizational influences), 기능-내용이 미치는 영향(function-content influences), 요구사항의 세부적인 단계와 관련된 의도(intended level of detail of the requirements)등을 고려해야 한다. 다양한 요구공학 산출물 작성을 위해서는 여러가지 기법들이 필요하다:

- ▶ 서베이(Survey) 기법(예, 인터뷰(interview)나 설문지(questionnaire))
- ▶ 창의적(Creativity) 기법(예, 브레인스토밍(brainstorming), 브레인 스토밍 역설(brainstorming paradox), 관점 변화(change of perspective), 유추 기법(analogy))
- ▶ 문서 중심(Document-centered) 기법(예: 시스템 고고학(system archaeology), 관점 기반 읽기(perspective-based reading), 요구사항 재사용(requirements reuse))
- ▶ 관찰(Observation) 기법(예: 필드 관찰(field observation), 견습(apprenticing))
- ▶ 지원(Support)기법(예: 마인드 매핑(mind mapping), CRC card, 음성 녹음 및 영상 녹화(audio and video recordings), 유스케이스 모델링(use case modeling), 프로토타입(prototype))

적절한 요구사항 도출 기법을 선택해서 적용하는 것이 프로젝트 경쟁력의 핵심이다. 최적의 결과를 달성하기 위해서는 다양한 요구사항 도출 기법을 효과적으로 조합하여 활용해야 한다.

EU 4 요구사항 문서화(명세화) (L2)

교육 시간: 2 시간

관련 용어: 요구사항 문서(Requirements Document), 요구사항 명세(Requirements specification)

교육 목표:

- EO 4.1.1 요구사항을 명세화해야 하는 핵심 이유를 이해한다
- EO 4.2.2 기능 요구사항의 세 가지 관점을 이해한다
- EO 4.2.2 일상어를 사용하여 요구사항을 명세화하는 경우의 장점과 단점을 이해한다
- EO 4.2.3 가장 중요한 모델 기반의 요구사항 명세화양식을 이해한다
- EO 4.2.4 요구사항 명세화 양식을 조합해 사용하는 경우의 장점을 이해한다
- EO 4.3.1 표준화 된 요구사항 명세화 양식의 장점을 이해한다
- EO 4.3.2 (일반적으로) 널리 사용되고 있는 문서 구조를 이해한다
- EO 4.3.3 표준 구조를 적절하게 커스터마이징하기 위해 중요하게 고려해야 할 요소를 이해한다
- EO 4.4.1 요구사항 명세화 활동을 이해한다
- EO 4.5.1 요구사항 문서의 품질 기준을 이해하고 활용한다
- EO 4.6.1 요구사항의 품질 기준을 이해하고 활용한다
- EO 4.6.2 요구사항의 가장 중요한 두 가지 스타일 규칙을 이해한다
- EO 4.7.1 콘텐츠, 용어집의 중요성을 이해하고 활용한다
- EO 4.7.2 용어집 활용의 규칙을 이해하고 활용한다

EU 4.1 문서 설계 (L1)

요구공학에서 모든 중요한 정보는 반드시 명세화 되어야 한다. 요구사항을 표현하는 형태가 갖추어져 있지 않은 것만, 단순한 기술에서 공식적인 표현 기법(semantics)을 사용한 다이어그램에 이르기까지, 모든 양식은 명세화 기법이라 명칭한다. 요구사항 명세화 수명 주기에는 많은 사람들이 명세화 작업에 관여한다. 작성된 문서는 커뮤니케이션에서 골 달성에 집중할 있도록 지원하는 기능을 한다. 요구사항들은 오래 지속되고, 법적으로 적절해야 하며 관련된 모든 이들이 접근할 수 있어야 하기 때문에, 문서를 통해 지원되어야 한다. 요구사항 문서는 대단히 복잡하다.

EU 4.2 명세화 종류 (L1)

요구사항 문서는 기능 요구사항을 포함하고 있으며, 기능 요구사항은 일반적으로 시스템과 관련된 다음 세 가지 관점을 다룬다:

- ▶ 데이터 관점
- ▶ 행동 관점
- ▶ 기능 관점

일상어로 기술된 요구사항은 세 가지 관점은 동시에 다를 수 있으나, 개념적 모델(conceptual model)은 특정한 한 가지 관점에서의 기술을 위해 특성화되어 있다. 효과적으로 적용가능한 명세화 양식은 다음과 같다:

- 일상어의 형태로 기술한 요구사항 문서
- 개념적 요구사항 모델 (예, 유스케이스 다이어그램, 클래스 다이어그램, 액티비티 다이어그램, 혹은 상태 다이어그램(LE6 참조))
- 조합된 요구사항 명세화 양식

EU 4.3 문서 구조 (L1)

요구사항 문서의 핵심 요소는 개발 대상 시스템의 요구사항이다. 요구사항 문서는 요구사항 이외에, 해당 문서의 목적에 따라 시스템 컨텍스트, 인수 조건(acceptance conditions), 구현과 관련된 기술적 특성 (characteristics of the technical implementation) 등과 관련된 내용도 포함할 수 있다. 요구사항 문서는 효과적인 관리를 보장하기 위해, 가장 효과적인 형태로 구조화 해야한다.

요구사항 문서 작성시 문서 표준 구조를 문서 템플릿으로 참조하여 작성할 수 있으며, ISO/IEC/IEEE 29148:2011 표준 등에서 제공하는 문서 표준 구조를 제공한다.

표준 구조를 활용하여 요구사항 문서를 작성하면 실질적으로 긍정정인 효과를 얻을 수 있다. 예를 들어 표준 구조를 활용하는 경우, 이후 개발 활동에서 요구사항 문서를 간단하게 활용할 수 있다(테스트 케이스를 정의하는 경우 등). 일반적으로 표준 구조를 모든 요구사항 문서에 그대로 적용할 수는 없으며, 도메인, 조직 및 프로젝트 환경에 맞게 적절히 수정하여 사용해야 한다.

EU 4.4 요구사항 문서의 활용 (L1)

요구사항 문서는 프로젝트 수명 주기상에서 많은 액티비티의 베이스로 활용된다. 액티비티의 예는 다음과 같다:

- 계획
- 아키텍처 설계
- 구현
- 테스트
- 변경 관리
- 시스템 사용 및 유지보수
- 계약 관리

EU 4.5 요구사항 문서 품질 기준 (L2)

프로젝트 수명 주기상에서 계속되는 액티비티의 베이스 역할을 하기 위해서, 요구사항 문서는 특정한 품질 기준을 만족해야 한다:

- ▶ 명백함 및 일관성(Unambiguity and consistency)
- ▶ 명확한 구조(Clear structure)
- ▶ 수정 가능성 및 확장성(Modifiability and extensibility)
- ▶ 완전성(Completeness)
- ▶ 추적성(Traceability)

EU 4.6 요구사항 품질 기준 (L2)

각 요구사항은 특정한 품질 기준을 만족해야 한다:

- ▶ 합의됨(agreed)
- ▶ 분명함(unambiguous)
- ▶ 필수적임(necessary)
- ▶ 일관성 있음(consistent)
- ▶ 확인 가능함(verifiable)
- ▶ 명확함(feasible)
- ▶ 추적 가능함(traceable)
- ▶ 완전함(complete)
- ▶ 이해 가능함(understandable)

요구사항은 상기 언급된 품질 기준을 만족해야 함은 물론, 일상어로 기술된 경우 다음 두가지 기본 스타일 규칙 추가적으로 만족해야 한다:

- ▶ 짧은 문장이나 단락으로 기술되고
- ▶ 한 문장은 하나의 요구사항만을 담아야 한다

EU 4.7 용어집 (L2)

요구공학에서 발생하는 충돌의 대부분은 관계된 당사자(이해관계자)들이 요구사항으로 정의된 용어를 서로 다르게 이해하고 있는 데 기인한다. 이러한 문제를 사전에 방지하기 위해 프로젝트에서 사용되는 관련된 용어를 용어집으로 정리해야 한다. 용어집은 프로젝트에서 사용되는 용어의 정의들의 집합이며, 용어들은 다음에 기반하여 그 의미가 정의된다:

- ▶ 특정 컨텍스트 기반의 기술 용어(context-specific technical terms)
- ▶ 축약어(abbreviations and acronyms)
- ▶ 주어진 컨텍스트에서 특정한 의미로 사용되는 일상적인 컨셉(everyday concepts that have a special meaning in the given context)
- ▶ 동의어(synonyms)
- ▶ 동일어(homonyms)

용어집을 정의하는 경우 다음 규칙을 준수하여야 한다:

- ▶ 용어집은 집중화되어 일괄적으로 관리되어야 함
- ▶ 용어집 관리의 책임(주체)가 정의되어야 함
- ▶ 프로젝트 전체에 걸쳐 용어집은 유지보수 되어야 함
- ▶ 용어집에는 (프로젝트에 관련된) 모든 사람이 접근 가능해야 함
- ▶ 용어집은 의무적으로 사용해야 함(용어집의 사용은 강제성을 가지고 있어야 함)
- ▶ 용어집은 해당 용어의 출처를 포함해야 함
- ▶ 모든 이해관계자가 용어집에 합의해야 함
- ▶ 용어집 접근에 대한 일관적인 구조를 제공해야 함

용어집은 프로젝트 수명 주기 중 조기에 작성할수록 프로젝트 진행 및 후반부에 투입되는 공수를 줄일 수 있다.

EU 5 일상어를 통한 요구사항 문서화 (L2)

교육 시간: 1 hour

관련 용어: 요구사항 템플릿(Requirements Template)

교육 목표:

- EO 5.1 일상어를 인식하고 기록하는 과정에서 발생하는 다섯 가지 변환 프로세스(transformational processes)와 해당 프로세스가 요구사항의 변환에 미치는 결과(consequences)를 이해하고 활용한다
- EO 5.2 요구사항 템플릿을 사용하여 요구사항을 변환하는 다섯 단계를 이해하고 활용한다

EU 5.1 언어 효과 (L2)

일반적으로 일상어는 모호하고 자의적인 해석이 가능하기 때문에 (요구공학에서 일반적인) 언어를 사용하는 경우에는 특별히 주의를 기울여야 한다. 언어를 인식하고 기록하는 과정에서, 소위 “변환 프로세스”가 발생한다. 요구공학 엔지니어는 이 “전환 프로세스”가 특정한 규칙을 따른다는 점을 활용해서 요구사항을 제시한 사람이 원하는 바를 정확하게 도출해 낼 수 있다. 요구공학에서 다루는 중요한 다섯 가지 변환 프로세스는 다음과 같다:

- ▶ 정규화(Normalization)
- ▶ 참조 인덱스가 없는 (일반)명사(Nouns without reference index)
- ▶ 전체 정량자(Universal quantifiers)
- ▶ 불완전하게 기술된 조건들(Incompletely specified conditions)
- ▶ 불완전하게 기술된 프로세스 언어들(Incompletely specified words)

EU 5.2 템플릿을 활용한 요구사항 구조화 (L2)

요구사항 템플릿은 학습을 통해 습득하기 쉬우며, 요구사항의 변환 과정에서 (자연)언어에 의한 영향을 줄일 수 있는 접근 방법을 제공한다. 요구사항 저자는 요구사항 템플릿을 활용해 높은 품질의 요구사항을 작성할 수 있다.

요구사항 템플릿을 활용해 (높은 품질의) 요구사항을 작성하는 다섯 단계는 다음과 같다:

- ▶ 법적 준수 요건 결정(Determine legal obligation)
- ▶ 요구사항 핵심 결정(Determine the core of the requirement)
- ▶ 시스템 액티비티 성격 정의(Characterizes the activity of the system)
- ▶ 객체 삽입(Insert objects)
- ▶ 논리 및 임시 조건 결정(logical and temporal conditions)

(요구사항의) 준수 여부와 관련된 기준은 “shall”, “should”, “will”, “may” 등을 포함해 수립한다. 준수 여부가 변경되는 경우, 요구사항역시 변경되어야 하며, (특정한) 속성을 사용해서 요구사항의 준수 여부를 문서화할 수 있다.

적합한 요구사항을 정의하기 위해서는 단순히 템플릿을 그대로 활용는 방법보다는 해당 방법에 대한 교육을 제공하고, 템플릿은 보조적으로 사용하는 것이 활용하는 것이 바람직하다.

EU 6 모델 기반 요구사항 문서화 (L2)

교육 시간: 5 시간

관련 용어: Model

교육 목표:

- EO 6.1.1 “모델(model)”의 다양한 모델의 속성을 이해한다
- EO 6.1.2 개념적 모델링 언어의 요소의 정의를 이해한다
- EO 6.1.3 요구사항 모델들의 장점을 이해한다
- EO 6.2.1 요구공학에서 목표의 중요성을 이해한다
- EO 6.2.2 두 가지 종류의 목표 분해(goal decomposition)을 이해한다
- EO 6.2.3 And/ Or trees 로 목표 관계를 모델링하는 방법을 이해하고 활용한다
- EO 6.3.1 유스케이스 다이어그램 모델링 방법을 이해하고 활용한다
- EO 6.3.2 유스케이스 명세 사양을 이해하고 활용한다
- EO 6.4.1 요구사항과 관련된 세 가지 관점을 이해한다
- EO 6.5.1 요구사항에 대해 데이터 관점이 다루는 영역을 이해한다
- EO 6.5.2 객체 관계 다이어그램(entity relationship diagram)과 UML 클래스 다이어그램(UML class diagram)을 이해하고 활용한다
- EO 6.6.1 요구사항에 대해 기능 관점이 다루는 영역을 이해한다
- EO 6.6.2 데이터 흐름 다이어그램(data flow diagram)과 UML 액티비티 다이어그램(UML activity diagram)을 이해하고 활용한다
- EO 6.7.1 요구사항에 대해 행동 관점이 다루는 영역을 이해한다
- EO 6.7.2 UML statechart 를 이해하고 활용한다

Note

본 챕터에서 L2 레벨(“Mastering and Using”)은 “생성(create)”, “설계(design)”, “개발(develop)” 및 “공식화(formulate)”는 포함하지 않는다(단, 해당 모델은 반드시 이해해야 한다. 모델의 생성과 관련된 내용은 IREB advanced level 의 “요구사항 모델링(requirements modeling)에서 다룬다.)

EU 6.1 모델 (L1)

모델을 사용하면 특정한 사실들과 해당 사실들 사이의 관련성을 선택적으로 쉽게 이해할 수 있다. 또한 그것들을 빠르게 기록하고 분명하게 기록할 수 있다. 모델이란 이미 존재하는 현실 혹은 구현해야 할 대상으로서의 현실을 추상화 시킨 것이다(요구공학이 적용되는 대부분의 영역에서 이 정의를 적용할 수 있지만, 이 모델은 좁은 의미의 모델을 의미한다. 일반적으로 모델이란, 이미 존재하는 개체(entity) 혹은 구현해야 하는 대상 개체를 추상화 한 것이다. 여기서 개체란 현실의 일부일 수 있고, 인지 가능한 현상들의 집합일 수도 있으며, 다른 모델을 포함할 수도 있다. 모델과 비교해서 개체라는 개념은 원본(the original)이라고 명칭한다.)

모델은 다음과 같은 세 가지 중요한 속성을 가진다:

대표적 속성(Representation property): 모델은 현실을 반영한다

감소적 속성(Reduction property): 모델은 반영된 현실을 감소시킨다

실용적 속성(Pragmatic property): 모델은 특정한 하나의 목적을 위해 만들어진다

요구공학에서는 개념적 모델을 자주 사용한다. 개념적 모델은 일반적으로 몇몇 그래픽 요소들을 사용해 현실을 모델링한다. 개념적 모델에서는 구문(syntax, 모델링 요소와 해당 요소의 유효한 조합)과 시맨틱(semantics, 모델링 요소의 의미)로 구성된 개념적 모델링 언어를 사용한다. 요구사항 모델은 구현되어야 할 시스템과 관련된 요구사항을 정의한 개념적 모델이다. 개념적 모델이 제공하는 형태로 표현된 요구사항 문서는 일상어로 기술된 요구사항 문서와 비교해 볼 때 다음과 같은 장점을 가진다:

- ▶ 그림으로 표현된 정보는 더욱 쉽게 이해하고 기억할 수 있다
- ▶ 요구사항 모델을 사용하면 해당 요구사항을 특정한 관점으로 (제한하여) 모델링 할 수 있다
- ▶ 특정 목적을 위해 모델링 언어를 정의함으로써, 현실을 적절하게 추상화할 수 있다

일상어와 요구사항 모델을 조합해서 사용하면 두 문서가 가진 장점을 모두 활용할 수 있다.

EU 6.2 골 모델 (L2)

골(Goal)이란 구현되어야 할 시스템 혹은 관련된 개발 프로젝트의 특정한 속성을 의도적으로 (이해관계자의 바람) 기술한 것이다. 골은 일상어는 물론 모델링 언어 모두로 문서화될 수 있다. 골을 문서화하는 과정에서 필수적으로 골 사이의 관계를 다듬어야 하며, 이는 상위 차원의 골과 하위 차원의 골을 분해(decomposition)하는 작업이다. 이 과정에서 두 가지 종류로 분해 작업이 분류된다:

- ▶ “필수 분해(AND decomposition, 필요 조건 분해)” 상위 골을 달성하기 위해서는 해당 골의 하위 골들이 모두 달성되어야 함
- ▶ “선택 분해(OR decomposition, 충분 조건 분해)” 상위 골을 달성하기 위해서는 해당 골의 하위 골들 중 최소한 하나의 골이 달성되어야 함

골들 사이의 분해 관계는 and/ or 트리의 형태로 문서화된다.

EU 6.3 유스케이스 (L2)

유스케이스는 사용자 관점에서(users perspective) 구현 대상 시스템 혹은 기존 시스템을 평가하고 문서화하는 데 도움을 준다. 유스케이스를 사용한 접근법은 다음 두 가지 상호보완적인 기법에 근거한다:

- ▶ 유스케이스 다이어그램(Use case diagrams)
- ▶ 유스케이스 명세(Use case specification)

유스케이스 다이어그램은 단순한 모델로 시스템의 기능, 시스템의 기능들 사이의 관계, 시스템의 기능들과 시스템 컨텍스트들 사이의 관계들을 문서화한다. 유스케이스 다이어그램의 일반적인 모델링 요소는 다음과 같다:

- ▶ 시스템 컨텍스트에 포함되는 액터(Actors) (사람 혹은 다른 시스템)
- ▶ 시스템 바운더리
- ▶ 유스케이스
- ▶ 위에서 언급한 모델링 요소들 사이의 다양한 관계

유스케이스 명세는 각 유스 케이스들이 필수적인 특성을 더욱 구체적으로 기술하여, 개략적인 형태로 나타나는 유스케이스 다이어그램을 보완한다. 이를 위해, 사전에 정의된 템플릿을 일반적으로 사용하며 각 유스 케이스에 대한 정보들을 기록한다. 템플릿은 일반적으로 다음과 같은 요소를 포함한다:

- ▶ 고유한 유스케이스 식별자(Unique designation of the use case)
- ▶ 유스케이스 이름(Name of the use case)
- ▶ 유스케이스 설명(Descriptions of the use case)
- ▶ 트리거링 이벤트(Triggering event)
- ▶ 액터(Actors)
- ▶ 결과(Results)
- ▶ 사전/ 사후 조건(Pre- and post-conditions)
- ▶ 다양한 시나리오들. 시나리오는 유스케이스가 성공적으로 실행되는 이벤트의 흐름(주 시나리오(main scenarios), 대체 시나리오(alternative scenarios))을 기술하거나, 유스케이스 실행중에 특별히 주의를 기울여 취급되어야 하는 예외적인 상황(예외 시나리오(exceptional scenarios))를 기술한다.

EU 6.4 요구사항을 바라보는 세 가지 관점 (L1)

모델에 기반해 문서화하는 경우, 구현 대상 시스템에 대한 요구사항은 다음 세가지 모델링 관점(각각의 관점은 서로 겹치는 부분이 있을 수 있음)에 따라 모델링된다:

- ▶ 데이터 관점(Data perspective)
- ▶ 기능 관점(Functional perspective)
- ▶ 행동 관점(Behavioral perspective)

데이터 관점의 경우, 개념적 모델링 언어의 요소들 중 개체 관계 모델(entity relationship models)과 UML 클래스 다이어그램(UML class diagram)을 주로 사용한다. 기능 관점의 경우, 데이터 흐름 다이어그램(data flow diagram) 혹은 UML 액티비티 다이어그램(UML activity diagrams, 액션들 사이의 객체의 흐름을 포함)을 주로 사용한다. 행동 관점의 경우 상태 전이 오토마타(finite state automata) 혹은 상태 전이 차트(statechart)를 주로 사용한다.

EU 6.5 데이터 관점에서의 요구사항 모델링 (L2)

데이터 관점에서는 데이터 구조 및 용도, 시스템 컨텍스트 상에서 데이터들의 관계를 기술한다. 일반적으로 데이터 관점에서는 개체 관계 다이어그램을 주로 사용하며, 해당 다이어그램은 다음 세 가지 모델링 요소를 활용하여 현실을 모델링한다:

- ▶ 개체 종류(Entity types)
- ▶ 관계 종류(Relationship types)
- ▶ 속성(Attributes)

관계에서 한 개체 종류의 인스턴스(entity)가 참여하는 빈도는 카디널리티(cardinality)를 사용하여 기술한다.

요구사항을 데이터 관점에서 모델링하는 경우 UML 클래스 다이어그램을 주로 사용한다. 하나의 클래스 다이어그램은 다수의 클래스와, 해당 클래스 사이의 관계들을 포함하고 있다, UML 클래스 다이어그램의 주요 모델링 요소는 다음과 같다:

- ▶ 클래스(Classess)
- ▶ 관계(Associations)
- ▶ 집합 및 포함 관계(Aggregation and composition relationships)
- ▶ 일반화 관계(Generalization relationships)

EU 6.6 기능 관점에서의 요구사항 모델링 (L2)

기능적 관점에서는 (시스템을 둘러싼) 환경으로부터 전달받은 입력값을 시스템의 환경으로 내보내는 출력값으로 변환하는 과정과 관련된 요구사항을 다룬다. 기능적 관점에서의 모델링 접근법은 기능 모델(function models)를 포함한다. 일반적으로, Tom DeMacro 의 “구조화된 분석(Structured Analysis)”에서 제시된 바와 같이, 데이터 흐름 다이어그램을 기능 모델로 사용한다. 시스템 및 해당 시스템의 컨텍스트를 포함하여 그래픽으로 표현한 것을 컨텍스트 다이어그램(context diagram)이라고 하며, 특별히 시스템 바운드리를 정의하기 위해 사용된 컨텍스트 다이어그램을 데이터 흐름 다이어그램이라 한다.

데이터 흐름 다이어그램은 다음 모델링 요소를 포함한다:

- ▶ 프로세스(Processes)
- ▶ 데이터 흐름(Data flows)

- ▶ 데이터 저장(Data stores)
- ▶ 데이터 소스와 연결구조(Sources/Sinks, 데이터 흐름의 시작과 끝)

데이터 흐름 다이어그램의 제어 흐름(control flow)이나 프로세스 내부에서 발생하는 업무를 표현하지 않기 때문에, 데이터 흐름 다이어그램은 추가적으로 구조화된 형태의 설명을 필요로 한다. 예를 들어 “구조화된 분석(structured analysis)”에서는 간단한 명세(mini-specification)를 사용해 프로세스의 내부 행동을 정의하였다.

UML 2.0의 경우, 액티비티 다이어그램에서 명시적으로 객체 흐름을 모델링하는 방법으로 데이터 흐름을 표시하기 때문에 데이터 흐름 다이어그램으로 사용하기에 매우 적합하다. 액티비티 다이어그램은 액티비티 노드(node)와 액티비티 노트들 사이의 제어 흐름을 모델링한다. 객체 흐름(Object flow)은 제어 흐름의 특수한 한 가지 형태이다. 액티비티 다이어그램에서는 동기화 바(Synchronization bars)를 사용해 동시에 발생하는 제어 및 오브젝트 흐름을 모델링할 수 있다. 대체 흐름(alternative control)과 객체 흐름은 결정 노드(decision node)를 사용하여 기술한다.

UML 2.0 액티비티 다이어그램의 핵심적인 모델링 요소는 다음과 같다:

- ▶ 액션(Actions)
- ▶ 시작 및 종료 노드(Start and end nodes)
- ▶ 제어 흐름(Control flow)
- ▶ 객체 흐름(Object flow)
- ▶ 결정 노드(Decision nodes)
- ▶ 대체 제어 흐름 통합(Merge of alternative control flow)
- ▶ 포크(Fork¹, 동기 분기점)
- ▶ 조인(Join², 동기 집합점)
- ▶ 계층 구조 요소(Hierarchization elements)

EU 6.7 행동 관점에서의 요구사항 모델링 (L2)

요구사항 모델링에서, 시스템의 동적인 행동은 행동 관점을 통해 모델링한다. 행동 관점은 시스템의 상태(state)와 해당 상태로의 변화(change)를 유발시키는 이벤트를 주로 다룬다. UML 상태 다이어그램(UML state diagrams)은 finite state machines의 원리에 기반하고 있으며, 다음 모델링 요소를 사용한다:

- ▶ 상태(State)
- ▶ 시작 및 종료 상태(Start and end states)

¹ 동기화 바(synchronization bar)의 시작점

² 동기화 바(synchronization bar)의 종료점

- ▶ 상태 전이(State transition)
- ▶ 동시성(Concurrency)

EU 7 요구사항 검증 및 협의 (L2)

교육 시간: 2 ½ 시간

관련 용어: 없음

교육 목표:

- EO 7.1.1 요구사항 검증의 중요성을 이해한다
- EO 7.2.1 요구사항과 관련된 충돌의 중요성을 이해한다
- EO 7.3.1 요구사항과 관련된 세 가지 품질 측면을 이해한다
- EO 7.3.2 콘텐츠(content), 문서화(documentation), 합의(agreement) 등 세 가지 품질 측면의 검증 기준을 이해하고 활용한다
- EO 7.4.1 요구사항 검증과 관련된 여섯 가지 원칙을 이해한다
- EO 7.4.2 요구사항 검증과 관련된 원칙을 활용한다
- EO 7.5.1 요구사항 검증과 관련된 기법을 이해한다
- EO 7.5.2 요구사항 검증 기법을 활용한다: 코멘팅(commenting, expert opinion), 인스펙션(inspection), 워크스루(walkthrough), 관점 기반 읽기(perspective-based reading), 프로토타입을 통한 검증(validation via prototypes) 및 체크리스트 활용(use of checklists)
- EO 7.6.1 요구사항 협의와 관련된 활동을 이해한다
- EO 7.6.2 요구사항 충돌의 종류를 이해한다
- EO 7.6.3 다양한 충돌 해소 기법을 이해한다
- EO 7.6.4 충돌 해소와 관련된 문서화 방법을 이해한다

EU 7.1 요구사항 검증 기초 (L1)

요구사항 검증의 목적은 가능한 조기에 오류를 발견하고 수정하기 위해 해당 요구사항들이 정의된 품질 기준(EU 4.6 참조)의 만족 여부를 검증하는 것이다. 요구사항 문서는 이후 진행되는 모든 개발 활동의 베이스 역할하기 때문에 요구사항 문서에 담긴 오류는 모든 개발활동에 영향을 미치고, 개발 단계가 진행됨에 따라 발견되지 않은 요구사항의 오류로 인한 이차적인 오류의 수는 기하급수적으로 증가한다. 때문에 요구사항 문서 자체의 오류뿐만 아니라 해당 요구사항 문서를 베이스로 하는 모든 산출물, 예를 들면 아키텍처 설계, 구현, 테스트 케이스 등을 재작업해야 한다.

EU 7.2 요구사항 협의 기초 (L1)

시스템 요구사항에서 발생할 수 있는 충돌이 해결되지 않은 채로 남아있다는 것은 특정한 이해관계자 그룹의 요구사항이 구현되지 않게 되거나, 운영 시스템이 일부 인수되지 않거나, 인수되었다 하더라도 그 사용빈도가 현저히 저하될 것임을 의미한다. 요구사항 협의의 목적인 관련된 이해관계자들 사이에서 개발 대상 시스템에 대한 요구사항을 동일하게 이해하고 합의하는 것이다.

EU 7.3 요구사항과 관련된 품질 측면 (L2)

요구사항과 관련된 세가지 품질 측면(컨텐츠(content), 문서(documentation) 및 합의(agreement))는 서로 구분되어 있다. 그러나 요구사항 혹은 요구사항 집합과 관련된 품질—개별적인 개별적인 품질 측면에 대한 부분—은 일련의 검증 기준을 통해 각각 평가될 수 있다.

“컨텐츠(content)”와 관련된 품질 측면은 다음 여덟 가지 기준을 고려한다:

- ▶ 요구사항 문서의 완전성(Completeness of the requirements document)
- ▶ 개별 요구사항의 완전성(Completeness of the individual requirements)
- ▶ 추적성(Traceability)
- ▶ 정확성 및 적합성(Correctness and adequacy)
- ▶ 일관성(Consistency)
- ▶ 충분한 자료가 뒷받침되지 않은 설계 결정 금지(No premature design decision)
- ▶ 검증 가능성(Verifiability)
- ▶ 필요성(Necessity)

“문서화(documentation)”와 관련된 품질 측면은 다음 네 가지 기준을 고려한다:

- ▶ 문서 양식 및 구조 준수(Conformity to document format and document structures)
- ▶ 이해가능성(Understandability)
- ▶ 명료성(Unambiguity)
- ▶ 문서화 규칙 준수(Conformity to documentation rules)

“합의(agreement)”와 관련된 품질 측면은 다음 세가지 기준을 고려한다:

- ▶ 합의됨(Agreed)
- ▶ 변경 후 합의됨(Agreed after changes)
- ▶ 충돌이 해소됨(Conflicts resolved)

EU 7.4 요구사항 검증 원칙 (L2)

요구사항 검증은 다양한 원칙에 근거한다. 이 원칙들은 요구사항 검증 과정이 진행되는 동안 요구사항이 가지고 있는 결함을 가능한 많이 발견하도록 보장한다. 요구사항 검증과 관련된 여섯 가지 원칙은 다음과 같다:

- ▶ 올바른 이해관계자의 참여(Involvement of the correct stakeholder)
- ▶ 오류 진단과 오류 수정의 분리(Separating the diagnosis and the correction of errors)

- ▶ 다양한 시각을 통한 검증(Validation from different vies)
- ▶ 적절한 문서화 종류 변경(Adequate change of documentation type)
- ▶ 요구사항에 기반한 개발 산출물 작성(Construction of development artifacts that are based on requirements)
- ▶ 반복적인 검증(Repeated validation)

EU 7.5 요구사항 검증 기법 (L2)

요구사항을 체계적으로 검증하는 다양한 기법이 존재하며, 요구사항이 정의된 검증 기준을 충분히 만족하는지 판별하기 위해, 추가적인 기법들과 함께 사용할 수 있다. 요구사항 검증과 관련된 기법은 다음과 같다:

- ▶ 코멘팅(Commenting, expert opinion)
- ▶ 인스펙션(Inspection)
- ▶ 워크스루(Walkthroughs)

위의 기법과 부가적으로 함께 사용할 수 있는 기법은 다음과 같다:

- ▶ 관점 기반 읽기(Perspective-based reading)
- ▶ 프로토타입을 통한 검증(Validation through prototypes)
- ▶ 체크리스트 활용(Usage of checklishts)

EU 7.6 요구사항 협의 (L1)

요구사항 협의란 관련된 개발 대상 시스템과 관련된 요구사항을 모든 이해관계자들이 동일하게 이해하도록 하는 과정이다. 요구사항 협의 과정은 다음과 같다:

- ▶ 충돌 사항 식별(Conflict identification)
- ▶ 충돌 사항 분석(Conflict analysis)
- ▶ 충돌 사항 해소(Conflict resolution)
- ▶ 충돌 사항 해소 결과 문서화(Documentaion of conflict resolutions)

충돌 사항 분석 과정에서 요구사항 별로 다양한 종류의 충돌이 식별되며, 그에 따른 적절한 충돌 해소 전략을 수립하고 적용해야 한다. 다양한 충돌의 종류는 다음과 같다:

- ▶ 이익 충돌(Interest conflicts) - 이해관계자 혹은 개인 간의 이익이 상충되는 경우 발생한다(이해관계자의 실질적인 이익 충돌이 발생하는 경우 객관적 이익 충돌, 개인간의 이익 충돌이 발생하는 경우 주관적 이익 충돌이라고 명칭한다).

- ▶ 자료 충돌(Data conflicts) - 이해관계자들 사이에 동일한 정보를 다른 의미로 해석하거나, 서로 동일하지 않은 정보를 소유한 경우 발생한다(영문판 CPRE 컴패니언 북(ISBN-13: 978-1933953879)에서는 Data conflicts 를 Subject conflicts 로 표기하고 있으나, 다음 버전에서는 Data conflicts 로 정정될 예정이다).
- ▶ 가치 충돌(Value conflicts) - 이해관계자들이 가진 가치와 선호도가 상이한 경우 발생한다.
- ▶ 관계 충돌(Relationship conflicts) - 이해관계자들 사이에 정서적인 문제가 있는 경우 발생한다.
- ▶ 구조적 충돌(Structural conflicts) - 조직 내 이해관계자들의 위치와 결정 권한의 차이가 있는 경우 발생한다.

실제 상황에서 충돌은 여러가지 원인들이 복합되어 발생한다. 충돌을 해소하기 위해서는 모든 관련된 이해관계자들이 참여해야 한다. 충돌 해소를 위한 기법은 다음과 같다:

- ▶ 합의(Agreement)
- ▶ 타협(Compromise)
- ▶ 투표(Voting)
- ▶ 편차 정의(Definition of variants)
- ▶ 기각(Overruling)
- ▶ 모든 요소 고려(Consider-all-facts)
- ▶ PMI(Plus-minus-interesting)
- ▶ 결정 매트릭스(Decision matrix)

충돌을 해소한 이후, 해당 충돌은 적절하게 문서로 기록하여야 한다. 특히 해당 충돌의 원인, 관련된 이해관계자, 다양한 이해관계자의 의견, 충돌 해소 방법, 가능한 대안, 결정 내용과 해당 결정의 이유 등을 반드시 기록해야 한다.

EU 8 요구사항 관리 (L2)

교육 시간: 2 ½ 시간

관련 용어: 없음

교육 목표:

- EO 8.1.1 속성 정책(attribute scheme)의 목적과 정의를 이해한다
- EO 8.1.2 요구사항과 관련된 중요 속성의 종류를 이해한다
- EO 8.2.1 요구사항에 대한 뷰를 활용한다
- EO 8.3.1 요구사항의 우선 순위를 선정하는 방법을 이해한다
- EO 8.3.2 요구사항 우선 순위 선정 기법을 활용한다
- EO 8.4.1 요구사항 추적성의 이점을 이해한다
- EO 8.4.2 추적성 관계 클래스를 활용한다
- EO 8.4.3 추적성 관계 표현 양식을 활용한다
- EO 8.5.1 요구사항 버전 관리를 활용한다
- EO 8.5.2 요구사항 형상관리를 활용한다
- EO 8.5.3 요구사항 베이스라인 관리를 활용한다
- EO 8.6.1 요구사항 변경의 중요성을 이해한다
- EO 8.6.2 요구사항 변경 통제 위원회(CCB, Change Control Board)의 기능과 구성원을 이해한다
- EO 8.6.3 요구사항 변경 요구 요소를 활용한다
- EO 8.6.4 다양한 변경 요구 클래스를 활용한다
- EO 8.6.5 변경 요구 처리 프로세스를 활용한다
- EO 8.7.1 요구사항 측정의 중요성을 이해한다

EU 8.1 요구사항 속성 할당하기 (L1)

시스템 요구사항을 시스템 수명 주기에 걸쳐 관리하기 위해서는 요구사항 정보를 가능한 구조화된 속성의 형태로 수집해야 한다. 요구사항과 관련된 속성 구조는 속성 정책(attribute scheme)을 통해 정의하며, 형태적으로는 테이블 혹은 정보 모델(information model)로 표현된다.

일반적인 속성의 종류는 다음과 같다:

- ▶ 식별자(Identifier)
- ▶ 이름(Name)
- ▶ 설명(Description)
- ▶ 소스(Source)
- ▶ 안정성(Stability)
- ▶ 위험성(Rick)

- ▶ 우선순위(Priority)

“법적 준수사항(legal obligation)” 역시 속성의 하나로 요구사항 정보에 추가적으로 포함될 수 있다.

속성 정책은 특정 조건들에 기반하여 프로젝트에 따라 정의하고 적용한다. 특정 조건들은 다음과 같다:

- ▶ 해당 프로젝트의 특정한 속성(Specific properties of the project)
- ▶ 조직 내 제약 사항(Constraints of the organization)
- ▶ 도메인 규칙(Domain rules)
- ▶ 개발 프로세스 제약 사항(Constraints of the development process)

EU 8.2 요구사항 뷰 (L2)

실제, 프로젝트와 관련된 요구사항의 수와 해당 요구사항들 사이의 의존성은 지속적으로 증가해 왔다. 요구사항의 복잡성을 프로젝트 스태프가 관리할 수 있도록 유지하기 위해 진행 중인 task에 따라 적절하게 요구사항에 접근할 수 있는 방법의 제공이 필요하다. 다음과 같은 두 가지 뷰 형태를 제공할 수 있다:

- ▶ 선택적 뷰(Selective views): 정의된 선택 기준에 따라 선정된 요구사항으로부터 속성값들의 하위 셋을 표시한다
- ▶ 요약적 뷰(Condensed views): 정의된 선택 기준에 따라 선정된 요구사항으로부터 속성 값의 요약 정보를 표시한다

EU 8.3 요구사항 우선 순위 선정하기 (L2)

요구사항의 우선 순위는 다양한 액티비티들이 진행되는 동안, 여러가지 기준에 의해 수시로 변경된다. 요구사항 우선 순위 선정은 다음과 같은 간단한 시스템에 기반하여 대비한다:

- ▶ 해당 우선 순위 선정의 목적과 제약 사항 결정(Determining the goals and constraints for the prioritization)
- ▶ 우선 순위 선정 기준 결정(Determining the prioritization criteria)
- ▶ 관련된 이해관계자 결정(Determining the relevant stakeholders)
- ▶ 우선 순위 선정 대상 산출물 선택(Selection of the artifacts to be prioritized)

위에 언급된 활동의 결과에 기반해, 하나 혹은 그이상의 우선 순위 선정 기법을 선택하고 우선 순위를 선정한다. 우선 순위 선정 기법은 다음과 같다:

- ▶ 상위 10 개 우선 순위 결정 기법(Ranking and top-tem technique)
- ▶ 단일 기준 분류(Single-criterion classification)

- ▶ 카노 분류(Kano classification)
- ▶ 비거의 우선 순위 매트릭스(Prioritization Matrix According to Wiegiers)

EU 8.4 요구사항 추적성 (L2)

요구사항 관리를 위해서 해당 요구사항의 추적성 정보를 기록하고, 조직화하며 지속적으로 유지 보수해야 한다.

- ▶ 요구사항 추적성을 통해 얻을 수 있는 이점은 다음과 같다:
- ▶ 검증의 단순화(Simplification of verifiability)
- ▶ 시스템 내 불필요한 속성 식별(Identification of unneeded properties in the system)
- ▶ 불필요한 요구사항 식별(Identification of unneeded requirements)
- ▶ 영향 분석 지원(Support of impact analysis)
- ▶ 재사용성 지원(Support of reusability)
- ▶ 책임 사항 결정 지원(Support of determination of accountability)
- ▶ 유지 보수 및 관리 지원(Support for maintenance and administration)

요구사항 추적성 관계와 관련하여, 세가지 추적성 관계 클래스로 구분한다:

- ▶ 전 요구사항 명세 추적성(Pre-requirements-specification traceability)
- ▶ 후 요구사항 명세 추적성(Post-requirements-specification traceability)
- ▶ 요구사항 간 추적성(Traceability between requirements)

명백하게 사용 관계가 있는 경우 해당 정보는 반드시 기록되어야 한다. 요구사항 추적성은 다양한 방법으로 표현될 수 있다. 일반적으로 다음과 같이 표현된다:

- ▶ 문장 기반 참조 및 하이퍼링크(Text-based references and hyperlinks)
- ▶ 추적 매트릭스(Trace matrices)
- ▶ 추적 그래프(Trace graphs)

EU 8.5 요구사항 버전 관리하기 (L2)

시스템 혹은 제품의 수명주기에 걸쳐 요구사항과 관련된 특정 개발 단계 및 요구사항 문서에 접근하기 위해서는 요구사항의 버전 및 형상(versioning and configuration)관리가 필요하다. 요구사항의 버전 번호는 최소한 다음의 두 가지요소를 포함하여 구성된다:

- ▶ 버전(Version)

- 증가치(Increment)

요구사항 형상은 논리적으로 관련이 있는 요구사항들의 집합으로 정의되며, 하나의 형상 단위에는 각 요구사항의 버전 중 최대 하나만 포함될 수 있다. 요구사항 형상의 형태는 다음의 두가지 측면으로 정의된다:

- 제품 측면(Product dimension): 요구사항 기초와 관련된 개별 요구사항
- 버전 측면(Version dimension): 요구사항의 다양한 변경 상태

요구사항 형상은 다음과 같은 여러 종류의 속성을 가진다:

- 하나의 형상 내에 존재하는 요구사항 사이의 논리적 연결
- 하나의 요구사항 형상 내에 존재하는 요구사항의 일관성
- 해당 요구사항 형상의 고유 식별자
- 해당 요구사항 형상 내에 존재하는 요구사항의 불변성
- 해당 요구사항 베이스의 이전 버전으로의 원복을 위한 베이스스

요구사항 베이스라인은 특정한 요구사항 영상이며, 베이스라인은 요구사항의 안정화 된 버전들로 구성되며, 일반적인 경우 대상 시스템의 배포 가능한 버전(system release)을 정의한다.

EU 8.6 요구사항 변경 관리 (L2)

요구사항은 시스템 전체 수명주기에 걸쳐 변경된다. 요구사항의 변경은 체계적인 변경 관리 프로세스에 따라 관리되고 진행되어야 한다. 변경 관리 프로세스에서, 변경 통제 위원회(CCB, Change Control Board)는 요구사항 변경 요청에 관한 처리를 담당해야 한다. 변경 통제 위원회의 업무는 다음과 같다:

- 요구사항 변경 요청의 구분
- 요청된 변경을 수행하는 데 투입되는 공수 결정
- 공수 투입 대비 이익 관점에서 요청된 변경 평가
- 요구사항 변경 요청에 기반한 새로운 요구사항 정의
- 승인된 변경 요청 사항의 우선 순위 선정
- 프로젝트 변경을 위한 승인된 변경 요청 사항 할당

변경 통제 위원회의 일반적인 구성원은 변경 관리자(change manager), 계약자(contracting), 아키텍트 설계자(architect), 사용자 대표(user representative), 품질 관리자(quality manager) 및 요구사항 엔지니어(requirement engineer)이다.

요구사항 변경이 필요하다고 판단되는 경우, 변경 요청사항의 형태로 해당 내용을 작성하여 변경 통제 위원회에 제출하여야 한다. 변경 요청은 최소한 다음의 정보를 포함하고 있어야 한다:

- ▶ 변경 요청 식별자(Identifier of the change request)
- ▶ 변경 요청 제목(Title of the change request)
- ▶ 필요한 변경 내용 설명(Description of the necessary change)
- ▶ 변경 필요성 증명(Justification of the need for the change)
- ▶ 날짜(Data field)
- ▶ 제출자(Applicant)
- ▶ 제출자가 판단하는 변경의 우선 순위(Priority of the change as seen by the applicant)

변경 요청은 다음과 같이 세 가지로 분류한다:

- ▶ 정정 변경(Corrective changes)
- ▶ 적용 변경(Adaptive changes)
- ▶ 예외 변경(Exceptional changes)

변경 관리 프로세스는 다음의 단계로 진행된다:

- ▶ 변경의 영향 분석 및 평가
- ▶ 변경 요청의 우선 순위 선정
- ▶ 변경 대상 프로젝트에 해당 변경 할당
- ▶ 변경 요청의 수락/ 거부 합의

EU 8.7 요구사항 측정 (L1)

에러, 속성, 변경 내용과 같은 요구사항 검증 및 관리 정보를 근거로 요구사항 문서의 품질과 프로세스를 분석할 수 있다. 이러한 과정을 통해 품질 및 프로세스 개선의 가능성을 식별할 수 있다. 측정시 일반적으로 다음과 같은 사항들을 고려한다:

- ▶ 요구사항 변경률 Requirements change rates
- ▶ 요구사항 삭제율 Requirements volatility
- ▶ 요구사항 오류율 Requirements errors

EU 9 도구 지원 (L1)

교육 시간: 1 hour

관련 용어: 없음

교육 목표:

- EO 9.1 요구사항 관리 도구와 관련된 여덟 가지 기능을 이해한다
- EO 9.2 요구공학 도구 도입시 고려할 다섯 가지 측면을 이해한다
- EO 9.3 요구공학 도구와 관련된 일곱 가지 관점을 이해한다

EU 9.1 도구의 종류 (L1)

요구공학에서는 테스트 관리 도구, 혹은 형상 관리 도구, Wikis, 오피스 소프트웨어 혹은 시각화 소프트웨어 와 같은 여러 개발도구를 활용할 수 있다. 요구공학에서는 데이터를 모델로 생성하고 분석할 필요가 있기 때문에, 모델링 도구들 역시 중요한 요소로 작용한다. 요구사항 관리 도구는 요구공학에 특화된 도구이다. 요구사항 관리 도구는 다음과 같은 특성을 가져야 한다:

- ▶ 다양한 정보 관리
- ▶ 정보들 사이의 논리적인 관계 관리
- ▶ 산출물을 고유하게 식별
- ▶ 접근 제어 등을 통해 데이터 접근에 대한 유연성과 보안성을 제공
- ▶ 데이터에 대한 뷰 지원
- ▶ 속성의 할당과 hierarchy 형태를 통한 정보의 조직
- ▶ 해당 정보들에 대한 보고서 생성
- ▶ 해당 정보들로부터 문서 생성

일반적인 오피스 도구들도 위의 기능들을 어느정도 제공하나, 이를 위해 특성화된 도구들은 이를 더욱 심화하여 지원한다(예: 추적성 관리 등)

EU 9.2 도구 도입하기 (L1)

요구공학 관련 절차나 기법 등이 도입되고 나면, 적절한 도구를 찾아낼 수 있다. 도구의 도입을 위해서는 분명한 요구공학의 책임 사항 및 절차들이 선행되어야 한다. 해당 프로세스에서 다음과 같은 관점을 고려해야 한다:

- ▶ 필요한 자원 고려
- ▶ 파일럿 프로젝트를 통한 리스크 최소화
- ▶ 정의된 기준에 따른 평가

- ▶ 라이선스 비용을 포함한 실질적 비용 계산
- ▶ 피고용인(도구 사용자) 교육

EU 9.3 도구 평가하기 (L1)

요구공학에서 사용할 도구를 평가하는 과정에서 고려되어야 할 다양한 측면은 다음 일곱 가지 관점을 고려해야 한다:

- ▶ 프로젝트 뷰(프로젝트 지원)
- ▶ 사용자 뷰(특히 사용성)
- ▶ 제품 뷰(기능성)
- ▶ 프로세스 뷰(방법론 지원)
- ▶ 공급자 뷰(벤더 서비스)
- ▶ 기술적 뷰(상호운용성, 확장성)
- ▶ 경제적 뷰(비용)

각 뷰에 대해 명료한 기준이 정의되어야 한다.