



# AI 시스템 테스트와 TMMi

#SuccessWithTMMi

EDITOR: ERIK VAN VEENENDAAL

[www.tmmi.org](http://www.tmmi.org)

TMMi 는 적용되는 개발 수명주기나 테스트 대상 시스템의 유형에 구애받지 않고 독립적으로 적용될 수 있도록 설계되었습니다. 본 문서는 AI 시스템을 테스트할 때, TMMi 테스트 프로세스 개선 모델을 어떻게 활용하고 적용하여 혜택을 얻을 수 있는지 설명합니다. 본문에서는 TMMi 레벨 2 의 각 프로세스 영역과 그 목표들을 하나씩 상세히 다룹니다. 또한 이러한 요소들이 AI 시스템 테스트와 어떤 연관성이 있는지, 그리고 실제 프랙티스(실무 관행)가 통상적으로 어떤 형태를 띠게 될지를 명시하고 있습니다. 만약 특정 프랙티스가 부가가치를 창출하지 못하여 AI 시스템 테스트 시 수행될 필요가 없는 경우, 이를 명확히 언급하였습니다. 향후 개정될 문서 버전에서는 상위 TMMi 레벨의 프로세스 영역들도 다룰 예정입니다. TMMi 레벨 3 와 관련해서는 유연성, 적응성, 자율성, 편향성과 같은 AI 기반 시스템 고유의 품질 특성에 대한 테스트를 논의하게 될 것입니다. 참고로, 본 문서는 전체적인 AI 시스템 테스트 실라버스(Syllabus)를 대체하기 위한 것이 아닙니다. 오히려 AI 시스템 테스트 시 TMMi 의 목표들을 어떻게 해석해야 하는지 보여주고, 심화 학습을 위한 아이디어와 방향성을 제시하는 데 그 목적이 있습니다. 또한, 'AI 시스템 테스트와 TMMi' 문서는 전체 TMMi 모델을 대체하는 설명서가 아닙니다. 따라서 본 문서는 반드시 기존 TMMi 참조 모델 문서와 함께 참조하여 사용해야 합니다. 본 문서는 TMMi 참조 모델을 대체하는 것이 아니라 보완하는 자료입니다. 목표(goal)와 프랙티스(practice)에 대한 완전한 설명은 TMMi 참조 모델을 참고해야 합니다.

© TMMi Foundation 2011–25

관련 라이선스 문서에 명시된 경우를 제외하고, TMMi Foundation 의 사전 허가 없이는 본 간행물의 전체 또는 일부를 어떠한 형태나 수단으로든 대여, 판매, 양도, 복제 또는 전송할 수 없습니다. 관련 라이선스 문서의 조건에 따라 복제가 허용되는 경우라도, 해당 복제본에는 반드시 본 고지문이 포함되어야 합니다. 상표로 간주되는 용어에 대해서는 그에 따라 표기하였습니다. 그러나 이러한 표기의 유무가 해당 상표의 법적 지위에 어떠한 영향도 미치지 않습니다.

TMMi 는 TMMi Foundation (UK)의 등록 상표입니다.

## 기여자

Amanda Dean (Australia)  
Mattijs Kemmink (The Netherlands)  
Vipul Kocher (India)  
Poonam Jain (India)  
Qin Liu (China)  
Srinivas Padmanabhuni (India)  
Çiçek Tuna (Germany)  
Erik van Veenendaal (The Netherlands)  
Kelly Ye (Australia)

## 개정 이력

본 섹션은 본 문서의 릴리스별 주요 개정 사항을 요약하여 기술합니다. 본 섹션은 참고 목적으로만 제공됩니다.

| 릴리즈  | 개정 내용                                   |
|------|-----------------------------------------|
| V1.0 | 'Testing AI-systems and TMMi' 문서의 초기 버전 |
|      |                                         |
|      |                                         |

# 목차

|                                               |           |
|-----------------------------------------------|-----------|
| 기여자.....                                      | 2         |
| 개정 이력 .....                                   | 3         |
| 목차 .....                                      | 4         |
| <b>1 TMMi를 활용한 AI 기반 시스템 테스트 .....</b>        | <b>7</b>  |
| 1.1 인공지능.....                                 | 7         |
| 1.2 테스트 성숙도 모델 통합(TMMi) .....                 | 7         |
| 1.3 AI 기반 시스템 테스트와 TMMi.....                  | 9         |
| 1.3.1 AI 기반 시스템 테스트 vs. 테스트에서의 AI 활용 .....    | 9         |
| 1.3.2 AI 시스템 테스트의 난제 .....                    | 10        |
| 1.3.3 목적 및 접근 방식.....                         | 11        |
| <b>2 TMMi 레벨 2: 관리됨 (Managed).....</b>        | <b>13</b> |
| 2.1 프로세스 영역 2.1 테스트 정책 및 전략 (TPS).....        | 13        |
| 2.1.1 SG1 테스트 정책 수립.....                      | 13        |
| 2.1.2 SG2 테스트 전략 수립 .....                     | 14        |
| 2.1.3 SG3 테스트 성과 지표 수립 .....                  | 16        |
| 2.2 프로세스 영역 2.2 테스트 계획 (TPL) .....            | 17        |
| 2.2.1 SG1 리스크 평가 수행.....                      | 17        |
| 2.2.2 SG2 테스트 접근법 수립 .....                    | 19        |
| 2.2.3 SG3 테스트 추정치 수립 .....                    | 20        |
| 2.2.4 SG4 테스트 계획 수립.....                      | 21        |
| 2.2.5 SG5 테스트 계획에 대한 확약 획득.....               | 22        |
| 2.3 프로세스 영역 2.3 테스트 모니터링 및 제어 (TMC) .....     | 23        |
| 2.3.1 SG1 계획 대비 테스트 진척 모니터링 .....             | 23        |
| 2.3.2 SG2 계획 및 기대치 대비 제품 품질 모니터링 .....        | 23        |
| 2.3.3 SG3 시정 조치 관리 및 종결 .....                 | 24        |
| 2.4 프로세스 영역 2.4 테스트 설계 및 실행 (TDE) .....       | 24        |
| 2.4.1 SG1 테스트 설계 기법을 활용한 테스트 분석 및 설계 수행 ..... | 24        |

|            |                                                                           |           |
|------------|---------------------------------------------------------------------------|-----------|
| 2.4.2      | SG2 테스트 구현 수행 .....                                                       | 26        |
| 2.4.3      | SG3 테스트 실행 수행 .....                                                       | 27        |
| 2.4.4      | SG4 테스트 인시던트 종결 관리 .....                                                  | 27        |
| <b>2.5</b> | <b>프로세스 영역 2.5 테스트 환경 (TEV).....</b>                                      | <b>28</b> |
| 2.5.1      | SG1 테스트 환경 요구사항 개발 .....                                                  | 28        |
| 2.5.2      | SG2 테스트 환경 구현 수행 .....                                                    | 29        |
| 2.5.3      | SG3 테스트 환경 관리 및 제어 .....                                                  | 29        |
| <b>3</b>   | <b>TMMi 레벨 3: 정의됨 (Defined).....</b>                                      | <b>29</b> |
| <b>3.1</b> | <b>프로세스 영역 3.1 테스트 조직 (Test Organization).....</b>                        | <b>29</b> |
| 3.1.1      | SG1 테스트 조직 수립 .....                                                       | 30        |
| 3.1.2      | SG2 테스트 전문가를 위한 테스트 기능 수립 .....                                           | 30        |
| 3.1.3      | SG3 테스트 커리어 경로 수립 .....                                                   | 31        |
| 3.1.4      | SG4 테스트 프로세스 개선의 결정, 계획 및 실행 .....                                        | 32        |
| 3.1.5      | SG5 조직 테스트 프로세스의 배포 및 교훈의 반영 .....                                        | 33        |
| <b>3.2</b> | <b>프로세스 영역 3.2 테스트 훈련 프로그램 (Test Training Program).....</b>               | <b>34</b> |
| 3.2.1      | SG1 조직적 훈련 역량 수립 .....                                                    | 34        |
| 3.2.2      | SG2 필요한 테스트 훈련 수행 .....                                                   | 35        |
| <b>3.3</b> | <b>프로세스 영역 3.3 테스트 라이프사이클 및 통합 (Test Life Cycle and Integration).....</b> | <b>35</b> |
| 3.3.1      | SG1 조직 테스트 프로세스 자산 수립 .....                                               | 35        |
| 3.3.2      | SG2 테스트 라이프사이클 모델과 개발 모델의 통합 .....                                        | 37        |
| 3.3.3      | SG3 마스터 테스트 계획 수립 .....                                                   | 38        |
| <b>3.4</b> | <b>프로세스 영역 3.4 비기능 테스트 (Non-Functional Testing).....</b>                  | <b>39</b> |
| 3.4.1      | AI 기반 시스템의 품질 특성 .....                                                    | 39        |
| 3.4.2      | SG1 비기능 제품 리스크 평가 수행.....                                                 | 41        |
| 3.4.3      | SG2 비기능 테스트 접근법 수립.....                                                   | 43        |
| 3.4.4      | SG3 비기능 테스트 분석 및 설계 수행.....                                               | 44        |
| 3.4.5      | SG4 비기능 테스트 구현 수행 .....                                                   | 45        |
| 3.4.6      | SG5 비기능 테스트 실행 수행 .....                                                   | 46        |
| <b>3.5</b> | <b>프로세스 영역 3.5 동료 검토 (Peer Reviews) .....</b>                             | <b>47</b> |
| 3.5.1      | SG1 동료 검토 접근법 수립.....                                                     | 47        |
| 3.5.2      | SG2 동료 검토 수행 .....                                                        | 49        |

|                  |    |
|------------------|----|
| References ..... | 50 |
|------------------|----|

# 1 TMMi 를 활용한 AI 기반 시스템 테스트

## 1.1 인공지능

인공지능(AI)은 가장 획기적인 기술 혁명의 최전선에 자리 잡고 있습니다. AI 는 과거 인간 마음의 전유물이었던 사고, 학습, 적응 능력을 기계에 부여하고자 하는 컴퓨터 과학의 한 분야입니다. 산업, 사회, 그리고 우리의 일상을 재편할 수 있는 잠재력을 지닌 AI 는 수많은 첨단 혁신을 이끄는 핵심 원동력이 되었습니다. 본질적으로 AI 는 통상 인간의 지능이 요구되는 작업을 수행할 수 있는 컴퓨터 시스템을 개발하는 것을 의미합니다. 이러한 작업은 단순한 계산부터 복잡한 문제 해결, 패턴 인식, 언어 이해, 그리고 의사 결정에 이르기까지 광범위합니다. AI 시스템은 방대한 양의 데이터를 처리하고 패턴을 인식하며 경험을 통해 성능을 지속적으로 개선합니다. 이는 데이터 기반의 통찰력이 갈수록 중요해지는 현대 사회에서 AI 를 강력한 도구로 만들어줍니다.

AI 의 한 분야인 머신 러닝은 AI 시스템의 핵심 요소입니다. 머신 러닝은 방대한 데이터셋을 사용하여 알고리즘을 훈련시킴으로써, 별도의 명시적인 프로그래밍 없이도 패턴을 인식하고 예측을 수행할 수 있게 합니다. 이러한 역량은 데이터 기반의 의사 결정과 반복 업무의 자동화를 가능하게 하여 의료, 금융, 마케팅, 제조 등 다양한 분야에 변화를 일으키고 있습니다. 그러나 AI 의 잠재력에는 고유한 과제들이 뒤따릅니다. 윤리적 우려, 알고리즘 편향, 데이터 품질, 개인정보 보호, 해석 가능성 및 투명성은 AI 시스템에서 반드시 해결해야 할 주요 쟁점들입니다. 또한 AI 는 일자리와 사회 전반에 미치는 영향에 대해 중요한 화두를 던지고 있습니다.

## 1.2 테스트 성숙도 모델 통합(TMMi)

TMMi 프레임워크는 테스트 프로세스 개선을 위한 지침이자 참조 프레임워크로서 TMMi 재단에 의해 개발되었습니다. 이는 테스트 관리자, 테스트 엔지니어, 개발자, 그리고 소프트웨어 품질 전문가들에게 중요한 문제들을 다루고 있습니다. TMMi 에서 정의하는 '테스팅'은 가장 포괄적인 의미로, 소프트웨어 제품 품질과 관련된 모든 활동을 의미합니다.

TMMi 는 프로세스 평가와 개선을 위해 성숙도 레벨 개념을 활용합니다. 아울러 각 레벨별로 프로세스 영역, 목표, 그리고 프랙티스가 식별되어 있습니다. TMMi 성숙도 기준을 적용하면 테스트 프로세스가 개선될 뿐만 아니라, 제품 품질 향상, 테스트 엔지니어링 생산성 증대, 그리고 리드 타임 단축에 긍정적인 영향을 미치는 것으로 입증되었습니다. 즉, TMMi 는 조직이 자체 테스트 프로세스를 평가하고 개선할 수 있도록 지원하기 위해 개발되었습니다.

TMMi 는 프로세스 개선을 위해 단계적 아키텍처를 채택하고 있습니다. 이는 조직의 테스트 프로세스가 즉흥적이고 관리되지 않는 상태에서 시작하여, 점차 관리되고(Managed), 정의되며(Defined), 측정되고(Measured), 최적화된(Optimized) 상태로 발전해 나가는 단계들을 포함합니다. 각 단계를 달성한다는 것은 해당 단계의 모든 목표가 충족되었음을 의미하며, 이러한 개선 사항들은 다음 단계로 나아가기 위한 기반이 됩니다.

TMMi 의 내부 구조는 체계적으로 학습하고 적용할 수 있는 풍부한 테스트 프랙티스들로 구성되어 있어, 점진적인 단계를 통해 품질 높은 테스트 프로세스로 개선되도록 지원합니다. TMMi 에는 5 개의 레벨이 존재하며, 이는 테스트 프로세스 개선을 위한 성숙도 계층과 진화 경로를 제시합니다. 각 레벨에는 해당 성숙도에 도달하기 위해 조직이 반드시 구현해야 할 일련의 프로세스 영역들이 포함되어 있습니다. TMMi 의 각 성숙도 레벨별 프로세스 영역은 [그림 1]에 나타나 있습니다.

TMMi 의 주요 기본 원칙 중 하나는, 이 모델이 다양한 수명주기 모델과 환경에 적용 가능한 일반적인 모델이라는 점입니다. TMMi 에서 정의한 대부분의 목표와 프랙티스는 순차적 모델뿐만 아니라, 애자일 및 데브옵스를 포함한 반복적 수명주기 모델 모두에 적용 가능한 것으로 나타났습니다. 다만, 모델의 최하위 레벨로 내려가면, 제공되는 서브 프랙티스와 예시들은 적용되는 수명주기 모델에 따라 (매우) 달라질 수 있습니다. TMMi 내에서 목표는 필수 사항이지만, 프랙티스는 필수가 아니라는 점에 유의해야 합니다.

TMMi 모델은 TMMi 재단 웹사이트에서 무료로 이용할 수 있습니다. 이 모델은 스페인어, 프랑스어, 중국어를 포함한 다국어로 번역되어 있으며, 출판된 서적 형태로도 이용 가능합니다.

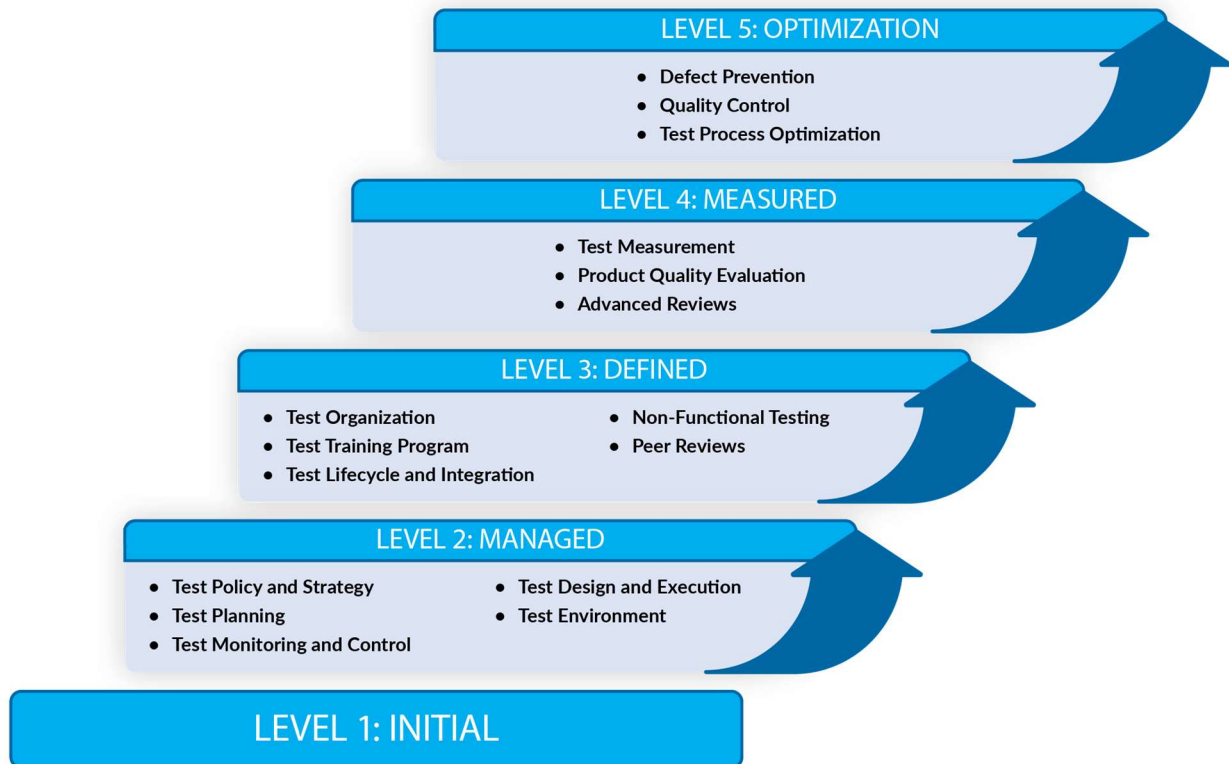


그림 1: TMMi 성숙도 레벨 및 프로세스 영역

## 1.3 AI 기반 시스템 테스트와 TMMi

### 1.3.1 AI 기반 시스템 테스트 vs. 테스트에서의 AI 활용

AI 시스템 테스트와 TMMi의 관계를 논의하고, AI 시스템 테스트의 맥락에서 TMMi를 어떻게 활용할지 설명하기에 앞서, 먼저 AI 시스템 테스트와 테스트에서의 AI 활용 간의 중요한 차이점을 짚고 넘어갈 필요가 있습니다. 이 둘은 소프트웨어 테스트 도메인에 인공지능(AI)을 적용하는 것과 관련하여 서로 다른 두 가지 개념입니다.

AI 시스템 테스트는 AI 컴포넌트를 포함하는 소프트웨어 애플리케이션이나 시스템의 성능, 기능 및 품질을 평가하고 검증하는 프로세스를 의미합니다. 주된 목표는 소프트웨어 시스템 내의 AI 알고리즘, 모델 및 기능들이 의도한 대로 작동하고, 적절한 결과를 도출하며, 원하는 품질 표준을 충족하도록 보장하는 것입니다. AI 시스템 테스트는 입력 데이터 테스트, 모델 테스트, 통합 테스트, 기능 테스트, 성능 테스트, 보안 테스트, 컴플라이언스(규제 준수) 테스트, 그리고 편향성이나 공정성과 같은 윤리적 고려사항에 대한 테스트 등 다양한 테스트 활동을 포함합니다. 아울러, AI 기반 시스템을 테스트할 때 AI를 활용하는 것, 즉 두 개념을 결합하는 것 또한 가능하다는 점에 유의해야 합니다.

테스팅에서의 AI 활용은 소프트웨어 테스팅 프로세스 자체(및 그 내부 활동)를 향상시키고 최적화하기 위해 인공지능 기법과 도구를 적용하는 것을 말합니다. 주된 목표는 AI 기술을 활용하여 다양한 테스트 활동을 지원하고 (부분적으로) 자동화하며, 데이터 기반의 의사결정을 내림으로써 테스팅의 효율성, 정확성 및 커버리지를 향상시키는 것입니다. 또한 AI 는 추정, 결함 예측, 테스트 분석, 테스트 케이스 생성, 테스트 데이터 생성, 테스트 실행, 테스트 결과 분석 등의 작업에도 활용될 수 있습니다. 아울러 리스크와 과거 데이터를 기반으로 테스트 케이스의 우선순위를 지정하는 데에도 AI 가 도움을 줄 수 있습니다.

요약하자면, “AI 시스템 테스팅”은 AI 기반 소프트웨어의 품질을 보장하는 것에 관한 것이고, “테스팅에서의 AI 활용”은 테스팅 프로세스 자체의 효율성과 효과성을 높이기 위해 AI 를 활용하는 것입니다. 본 문서의 범위는 AI 시스템 테스팅에만 국한되며, 이것이 TMMi 와 어떻게 관련되는지, 그리고 TMMi 가 어떤 도움을 줄 수 있는지에 초점을 맞춥니다. 테스팅에 AI 를 활용하는 주제는 차기 TMMi 버전(TMMi 2.0)의 테스팅 프랙티스로 고려될 가능성이 높습니다.

### 1.3.2 AI 시스템 테스팅의 난제

특히 가장 보편적인 형태인 머신러닝이 테스트에 반응하여 행동을 변화시킨다면, 과연 AI 시스템을 어떻게 테스트해야 할까요? 또한, 테스팅에서는 테스트의 통과(Pass) 또는 실패(Fail)를 명확히 판별할 수 있어야 합니다. 하지만 AI 시스템에서 “테스트를 통과했다”는 것은 도대체 무엇을 의미하는 것일까요? 복잡하지 않은 추론의 경우 수동으로 통과/실패 여부를 판단하는 것이 가능할 수도 있겠지만, 방대한 양의 데이터를 다룰 때는 불가능할 수 있습니다. 명확하고 완전한 명세서(Specification)가 존재하지 않을 수도 있고, 올바른 동작이 시간이 지남에 따라 변할 수도 있으며, 심지어 시스템이 알려주는 결과 이외에는 무엇이 정답인지 알 수 없는 경우도 있습니다.

AI 시스템 테스팅을 다룰 때 직면하는 몇 가지 주요 난제들은 다음과 같습니다.

- AI 시스템은 (기존 소프트웨어와 비교하더라도) 경우의 수가 막대하게 많은 복잡한 문제를 해결하는 데 사용되고 있습니다.
- 데이터는 편향을 가지고 있으며, 이러한 편향은 AI 시스템에 매우 쉽게 내재화될 수 있습니다.
- AI 기반 시스템은 모델 드리프트, 데이터 드리프트, 또는 개념 드리프트를 겪을 수 있습니다.
- 기대 결과를 확정할 수 있는 테스트 오라클이 부재합니다.
- AI 모델의 성능 지표는 통상적인 시스템 성능 지표와 매우 다르며, 많은 테스터들에게 생소한 개념입니다.

- 전통적인 테스트 기법들은 더 이상 적용할 수 없는 경우가 많습니다. 따라서 우리는 테스트 기법을 재고해야 하며, 테스트 커버리지를 측정하는 방식 또한 다시 생각해야 합니다.
- AI 시스템의 비결정론적(non-deterministic) 특성: 다수의 AI 기반 시스템이 지닌 이러한 특성으로 인해, 해당 시스템의 정확한 동작을 보장하는 것이 본질적으로 어렵습니다.
- 일부 AI 시스템이 지닌 자가 적응적(self-adaptive) 특성: 이러한 시스템은 테스트 도중에, 또는 테스트의 결과로 그 동작을 변경할 수 있습니다.

오늘날 AI 시스템을 테스트하는 것은 매우 도전적인 과제입니다. 이러한 모든 난제들을 해결할 수 있는 쉬운 해법은 존재하지 않습니다. 아마도 AI 기반 시스템 테스트에 대한 학습을 시작하는 흥미로운 방법 중 하나는 AiU United [AiU]와 ISTQB [ISTQB]에서 개발한 자격증 실라버스를 공부하는 것일 것입니다. 일반적으로, 조직이 AI 기반 시스템 테스트이라는 난제에 직면하기 위해서는 그전에 일정 수준의 테스트 성숙도를 갖추어야 합니다.

### 1.3.3 목적 및 접근 방식

본 문서는 AI 시스템 테스트와 TMMi 테스트 프로세스 개선 모델의 결합이, 더 높은 수준의 테스트 성숙도를 달성하고 궁극적으로 비즈니스 목표를 실현할 수 있는 유효한 경로임을 설명합니다. 또한 AI 시스템 테스트의 맥락에서 TMMi 를 어떻게 활용하고 적용해야 실질적인 혜택을 얻을 수 있는지에 대해 기술합니다. AI 시스템 테스트 시 적용되는 TMMi 프랙티스와 관련하여, 기존의 전통적인 테스트 접근법을 대체할 수 있는 검증된 대안들을 제시합니다.

본 문서는 조직의 현재 성숙도나 품질 향상을 위한 여정의 단계와 무관하게, AI 시스템 테스트를 수행하는 조직에게 TMMi가 어떠한 도움을 줄 수 있는지 다룹니다. 예를 들어, 조직의 성장이나 프로젝트 압박의 증가로 인해 간과하기 쉬운 핵심적인 테스트 프랙티스들을 다시금 상기시켜 주는 역할을 합니다.

각 TMMi 프로세스 영역과 그 세부 목표들을 하나씩 상세히 논의합니다. 이러한 요소들이 AI 시스템 테스트와 어떻게 연관되는지, 그리고 실제 테스트 프랙티스는 통상적으로 어떤 형태를 띠게 되는지 명시합니다. 만약 특정 TMMi 프랙티스가 부가가치를 창출하지 못하여 AI 시스템 테스트 시 수행될 필요가 없는 경우, 이를 명확히 언급하였습니다.

참고로, 본 문서에서는 AI 기반 시스템을 위한 테스트 프랙티스를 식별하고 이를 TMMi 의 프랙티스 및 목표와 매핑하였으나, 각 프랙티스에 대한 상세한 설명은 다루지 않습니다. AI 시스템 테스트를 위한 구체적인 프랙티스에 대해서는 시중에 나와 있는 훌륭한 참고 자료들(예: [Smith])을 활용하시기 바랍니다.

AI 시스템 테스트는 단순히 '또 다른 유형의 시스템'을 테스트하는 것이 아니므로, 프랙티스의 변화는 주로 테스트 엔지니어링 활동과 관련된 프로세스 영역(예: 테스트 계획, 테스트 설계, 비기능 테스트 등)에서 나타납니다. TMMi 모델에서 테스트 엔지니어링 프로세스 영역은 주로 TMMi 레벨 2와 3에 집중되어 있습니다. 따라서 본 문서는 AI 기반 시스템 테스트의 관점에서 TMMi 레벨 2(관리됨)와 TMMi 레벨 3(정의됨)만을 다룹니다.

본 문서는 기존 TMMi 모델과 독립적으로 사용되도록 작성된 것이 아닙니다. 이 문서는 AI 시스템 테스트 환경에서 테스트 프로세스 개선을 수행하는 실무자들에게, 다양한 TMMi 목표와 프랙티스를 어떻게 해석하고 이해해야 하는지에 대한 지침을 제공하는 것을 목적으로 합니다. 이 문서는 원본 TMMi 모델을 보완하는 역할을 하며, 반드시 상호 보완적인 문서로 활용되어야 합니다.

## 2 TMMi 레벨 2: 관리됨 (Managed)

### 2.1 프로세스 영역 2.1 테스트 정책 및 전략 (TPS)

*테스트 정책 및 전략 프로세스 영역의 목적은 테스트 정책을 개발하고, 테스트 유형 및 테스트 레벨과 같은 테스트 활동이 명확하게 정의된 전사적 또는 프로그램 차원의 테스트 전략을 수립하는 것입니다. 또한 테스트 성과와 테스트 활동의 가치를 측정하고 개선이 필요한 영역을 도출하기 위해, 테스트 성과 지표가 도입됩니다.*

#### 2.1.1 SG1 테스트 정책 수립

테스트 개선 프로젝트에 착수하는 모든 조직은 가장 먼저 테스트 정책을 정의해야 합니다. 테스트 정책은 테스트 및 테스트 전문가에 대한 조직의 전반적인 테스트 목적, 목표, 그리고 전략적 관점을 정의합니다. 테스트 정책은 조직의 전반적인 비즈니스(품질) 정책과 연계되는 것이 중요합니다. 테스트 개선은 명확한 비즈니스 목표에 의해 주도되어야 하며, 이는 다시 테스트 (개선) 정책에 문서화되어야 합니다. 테스트 정책은 조직 내 모든 이해관계자들 사이에서 테스트와 그 목적에 대한 공통된 견해를 확보하기 위해 필수적입니다. 이러한 공통된 견해는 조직 전체에 걸쳐 테스트 (프로세스 개선) 활동을 일치시키기 위해 요구됩니다. 유의할 점은 테스트 목적 그 자체가 최종 목표가 되어서는 안 되며, 이는 '작동하는 소프트웨어 확보' 및 '제품 품질'이라는 상위 목표로부터 도출되어야 한다는 것입니다.

물론 앞서 언급한 내용은 테스트 대상 시스템의 유형과 관계없이 모든 조직에 적용되는 사실입니다. 우선적으로 AI 를 이해하고, AI 가 가져오는 리스크, 그리고 그에 따른 다른 테스트 접근법이 필요하다는 점을 인지하는 것이 중요합니다. 윤리(Ethics), 편향(Bias), 데이터(Data), 규제(Regulations)와 같은 주제들은 새롭게 도입되거나 더욱 명확하게 집중해서 다뤄져야 할 수 있습니다. 또한 투명성(Transparency), 해석 가능성(Interpretability), 설명 가능성(Explainability)과 같은 AI 시스템 특유의 비기능적 특성들이 도입되어야 합니다. 결과적으로 테스트 목적과 목표, 그리고 테스트 및 테스트 프로세스 정의에 대한 기본적인 관점들 또한 재고되어야 할 것입니다.

대부분의 사항은 비(Non) AI 시스템과 크게 다르지 않겠으나, 분명한 차이점은 존재할 것입니다. 예를 들어, AI 시스템으로 인해 발생하는 리스크를 줄인다는 내용이 목표에 포함될 수 있으며, 프로세스 정의 시 AI 시스템은 동일한 입력에 대해서도 다른 결과를 도출할 수 있다는 사실이나, 통합 단계 이전에 '모델 테스트 단계'가 존재한다는 점을 언급해야 할 수도 있습니다. 이러한 모델 테스트 단계는 일반적으로 전통적인 시스템에는 존재하지 않는 단계입니다.

윤리측면에서는 테스터가 AI 기반 시스템에서 데이터를 어떻게 사용하는지에 대한 고려뿐만 아니라, 시스템 자체의 윤리적 행동과도 관련이 있습니다. AI 시스템이 현실 세계에서 생명과 자유에 심각한 결과를 초래할 수 있는 결정을 내리는 경우, 윤리적 판단은 매우 위태롭고 중요한 문제가 됩니다. AI에서의 윤리적 우려는 시스템이 결과로부터 학습을 하는지, 아니면 지정된 훈련 데이터셋으로부터만 학습하는지에 따라 달라질 것입니다.

데이터는 편향, 개인정보 보호, 보안과 같은 이슈들을 테스트 정책에서 고려해야 할 중요한 항목으로 부각시킵니다. 올바른 데이터 요구사항과 모집단에 대한 이해는 무엇보다 중요해지고 있습니다. 예를 들어, 안면 인식 시스템은 단순히 구하기 쉬운 데이터 세트가 아니라, 실제 영향을 받는 인구 집단을 대표할 수 있는 데이터로 훈련되고 테스트되어야 합니다. 다국적 제품의 경우, 단일 지역 인구 집단에서 얻을 수 있는 것보다 훨씬 광범위한 데이터로 테스트해야 함을 의미합니다. 편향 점검은 모델 훈련 이후에 멈춰서는 안 된다는 점에 유의해야 합니다. 모델이 실제로 예측을 수행하는 데 사용되는 시점(추론 시점, inference time)에도 편향을 모니터링해야 합니다.

품질은 안전 필수 시스템과 같은 영역을 제외하고는, 통상적으로 IT 시스템에서 법적 규제의 대상이 아닙니다. 그러나 사람에 대한 결정을 내리는 AI 기반 시스템의 사용이 증가하고, 사람들이 이러한 결정을 완전히 이해하지 못하는 문제가 발생함에 따라, 조직의 품질에 대해 법적 제약을 가하는 다양한 규제들이 진행 중이거나 이미 시행되고 있습니다.

## 2.1.2 SG2 테스트 전략 수립

테스트 전략은 테스트 정책을 따르며, 프로젝트 내 테스트 활동의 출발점 역할을 합니다. 테스트 전략은 일반적으로 전사적 차원 또는 프로그램 차원에서 정의됩니다. 통상적인 테스트 전략은 상위 레벨의 제품 리스크 평가를 기반으로 하며, 수행해야 할 테스트 유형과 테스트 레벨에 대한 설명을 포함합니다.

예를 들어, 각 프로젝트에서 단순히 '통합 테스트'와 '인수 테스트'를 수행하겠다고 명시하는 것만으로는 충분하지 않습니다. '통합'과 '인수' 테스트가 구체적으로 무엇을 의미하는지, 일반적으로 어떻게 수행되는지, 그리고 주요 목적이 무엇인지 정의해야 합니다. 경험에 따르면, 테스트 전략이 명확히 정의되고 준수될 때 다양한 테스트 활동 간의 중복이 줄어들어 테스트 프로세스가 더욱 효율적으로 운영됩니다. 또한, 다양한 테스트 유형과 레벨의 테스트 목적 및 접근 방식이 상호 정렬됨에 따라 테스트의 누락이 줄어들고, 결과적으로 더 효과적인 테스트 프로세스와 높은 수준의 제품 품질을 달성할 수 있습니다.

테스트 전략은 AI 시스템을 테스트할 때에도 매우 필수적인 문서입니다. AI 시스템 테스트는 많은 조직에게 비교적 새로운 분야이므로, 공통된 전사적 테스트 전략이 있어야 프로젝트들이 AI 테스트를

수행할 때 지침을 얻고 불필요한 시행착오를 방지할 수 있습니다. 또한 테스트 전략은 테스트에 참여하는 모든 인원이 전체적인 그림을 이해하도록 보장합니다.

전통적인 시스템의 경우, 테스트 전략은 수행할 테스트 레벨과 테스트 유형을 중심으로 구성되며, 이들은 서로 연관되어 식별되고 정의됩니다. 하지만 AI 시스템 테스트에서는 이러한 많은 부분이 다르게 적용됩니다.

AI 테스트 전략은 우선 크게 오프라인 테스트와 온라인 테스트이라는 두 가지 주요 단계로 구분됩니다.

오프라인 단계에서는 AI 모델이 목표를 달성했는지 평가하기 위해 정확도나 정밀도와 같은 지표를 사용하여 모델을 튜닝하고, 검증하며, 테스트합니다. 일반적으로 머신러닝 모델의 기능적 동작을 테스트할 뿐만 아니라, 학습 속도, 예측 속도, 사용된 컴퓨팅 리소스, 적응성(Adaptability), 투명성(Transparency) 등 격리된 모델 자체에 적합한 비기능 테스트도 수행됩니다. 모델 테스트 외에도 오프라인 단계는 입력 데이터 테스트라는 새로운 레벨의 테스트를 포함합니다. 입력 데이터 테스트는 시스템의 훈련과 예측에 사용되는 데이터가 최고 수준의 품질임을 보장하기 위한 것입니다.

온라인 단계에서는 훈련된 모델을 다른 AI 및 비(AI) AI 컴포넌트를 포함한 나머지 시스템과 통합하고, 운영 환경 내에서 테스트합니다. 이 단계에서는 기능 테스트와 성능과 같은 비기능 테스트가 모두 수행됩니다. 온라인 단계에서는 전통적인 테스트에서 적용되는 테스트 레벨과 유형을 사용하여 테스트 전략을 구조화할 수 있습니다.

- 컴포넌트 테스트: 사용자 인터페이스나 통신 모듈과 같은 모든 비(AI) AI 모델 컴포넌트에 적용됩니다.

- 컴포넌트 통합 테스트: 시스템 컴포넌트들(AI 및 비 AI 모두 포함)이 올바르게 상호작용하는지 확인하기 위해 수행됩니다.

- 시스템 테스트: 통합된 컴포넌트들(AI 및 비 AI 모두 포함)로 구성된 전체 시스템이 운영 환경과 밀접하게 유사한 테스트 환경에서 기능적 및 비기능적 관점 모두에서 기대한 대로 수행되는지 확인하기 위해 수행됩니다. 시스템에 따라, 이 테스트는 예상되는 운영 환경에서의 현장 시험 형태를 취할 수도 있습니다.

- 인수 테스트: 전체 시스템이 고객에게 수용 가능한지 판단하기 위해 사용됩니다. 다만, 추가적인 집중이 필요한 측면들이 있을 수 있습니다. 예를 들어, 시스템이 인간의 활동을 대체하도록 설계된 경우, 시스템의 동작을 인간의 동작과 비교하고 정의된 성능 기준과 대조하는 것이 필요할 수 있습니다.

- 지속적인 모니터링: AI 시스템 테스트에 적용 가능한 또 다른 테스트 레벨은 지속적인 기반 위에서 시스템의 성능을 모니터링하는 것입니다. AI 시스템은 실제 사용중인 시스템의 품질을 지속적으로 검토하고 재확인해야 하는 경우가 많습니다. 운영 환경에서 품질을 모니터링해야 하는 중요한 이유 중 하나는 편향 탐지입니다. 때로는 훈련 데이터와 실제 라이브 환경에서 모델이 노출되는 데이터 간의 잠재적 차이로 인해 테스트 환경에서 편향을 탐지하기 어려울 수 있으며, 지속적인 모니터링이 이에 도움이 될 수 있습니다. 지속적인 모니터링의 또 다른 중요한 이유는 드리프트입니다. AI 기반 시스템에서 모델 드리프트, 데이터 드리프트, 개념 드리프트는 모두 시간이 지남에 따라 고려하고 모니터링해야 할 요소들입니다. 드리프트의 징후를 조기에 감지하기 위해서는 모델 성능 지표를 지속적으로 모니터링해야 합니다. 드리프트는 입력 피처와 출력간의 관계 변화로 인해, 작동 중인 모델이 시간이 지남에 따라 저하될 수 있다는 사실을 의미합니다. 모델이 사용되는 동안 상황이 진화할 수 있으며, 모델은 의도된 성능에서 멀어지는 방향으로 표류할 수 있습니다.

참고로, AI 시스템을 조직 내부에서 직접 개발하지 않고 외부(예: Microsoft, Apple, Google 등)에서 개발된 AI 시스템을 (더 큰 구성의 일부로) 구현하는 경우도 있을 수 있습니다. 물론 이러한 경우에도 AI 테스트 전략은 영향을 받게 되며, 결과적으로 다른 시스템과의 통합, 조직의 필요에 맞는 시스템의 테일러링 및 구성에 초점을 맞춘 상용(COTS) 테스트 전략과 유사한 형태를 띠게 될 것입니다.

### 2.1.3 SG3 테스트 성과 지표 수립

테스트 정책에 정의된 테스트 개선의 비즈니스 목표들은 일련의 핵심 테스트 성과 지표로 구체화되어야 합니다. 테스트 정책과 이에 수반되는 성과 지표들은 명확한 방향성을 제시하며, 기대되는 테스트 성과 수준과 실제 달성된 성과 수준을 소통할 수 있는 수단이 됩니다. 성과 지표는 이해관계자들에게 테스트 및 테스트 프로세스 개선이 주는 가치를 명확히 보여주어야 합니다.

프로세스 개선에 대한 투자는 장기적인 경영진의 지원을 필요로 하므로, 경영진의 지속적인 동기를 유발하기 위해서는 개선 프로그램의 혜택을 정량적으로 측정하는 것이 매우 중요합니다.

유의할 점은, 이 TMMi 고유 목표가 제한된 수(예: 2~3 개)의 테스트 성과 지표를 정의하는 데 초점을 맞추고 있다는 것입니다. 이는 방대한 전체 측정 프로그램을 수립하고 이행하는 것이 아니라, 시간이 지남에 따라 그리고 다양한 전달(Delivery) 환경 내에서 테스트의 가치가 어떻게 변화하는지를 보여주는 핵심 지표 세트를 정의하는 것에 관한 것입니다.

만약 AI 시스템 테스트가 테스트 프로세스 개선 프로그램의 일부로서 정기적으로 수행되는 경우, 개선 프로그램의 결과를 입증하기 위해 AI 시스템 테스트에 특화된 별도의 성과 지표가 필요합니다. AI

시스템의 고유한 특성과 그에 따른 테스트 목적의 차이로 인해, 결함 탐지율과 같은 전통적인 성과 지표는 일반적으로 적용하기 어렵습니다.

AI 시스템을 위한 테스트 성과 지표는 훈련되거나 테스트된 모델이 얼마나 정확하고 신뢰성 있게 결과를 도출하는지를 측정하게 됩니다.

|        |               | Predicted                            |                                     |
|--------|---------------|--------------------------------------|-------------------------------------|
|        |               | Negative (N)<br>-                    | Positive (P)<br>+                   |
| Actual | Negative<br>- | True Negative (TN)                   | False Positive (FP)<br>Type I Error |
|        | Positive<br>+ | False Negative (FN)<br>Type II Error | True Positive (TP)                  |

그림 1: 혼동행렬

참고로, AI 시스템에 적용되는 성과 지표는 학습 유형(비지도 학습 또는 지도

학습)과 모델 유형(군집화, 연관, 분류 또는 회귀)에 따라 달라진다는 점에 유의해야 합니다. 예를 들어, 지도 학습 기반의 분류 AI 시스템의 경우 혼동 행렬 ([그림 1] 참조)을 활용할 수 있으며, 이를 기반으로 정확도(Accuracy) 및 정밀도(Precision)를 산출하여 성과 지표로 사용할 수 있습니다.

지도 학습 기반의 회귀(Regression) AI 시스템에서는 평균 제곱 오차(Mean Square Error, MSE)를 성과 지표로 활용할 수 있습니다. MSE 는 실제 값과 예측 값 사이의 차이를 제곱한 값들의 평균을 의미합니다. MSE 값은 항상 양수이며, 그 값이 0 에 가까울수록 더 우수한 회귀 모델임을 나타냅니다.

(테스트) 성과 지표로 활용 가능한 더 다양한 메트릭의 예시는 AiU Certified Tester in AI 실라버스 및 ISTQB Certified Tester AI Testing 실라버스 에서 확인할 수 있습니다.

## 2.2 프로세스 영역 2.2 테스트 계획 (TPL)

테스트 계획의 목적은 식별된 리스크와 정의된 테스트 전략을 기반으로 테스트 접근법을 정의하고, 테스트 활동의 수행 및 관리를 위한 타당성 있는 계획을 수립하여 유지하는 것입니다.

성공적인 테스트 계획의 핵심은 '테스트 계획서(문서)'를 작성하는 것 자체가 아니라, 사전에 충분히 고민하는 '활동' 그 자체에 있음을 유의해야 합니다.

### 2.2.1 SG1 리스크 평가 수행

완전한 테스트는 불가능합니다. 이는 AI 기반 시스템 테스트에서도 마찬가지이며, 항상 선택과 집중을 통한 우선순위 설정이 필요합니다. 따라서 이 TMMi 목표는 AI 기반 시스템을 테스트할 때도 동일하게 적용됩니다. 테스트 접근법은 AI 기반 시스템에 대한 리스크 분석을 토대로 수립되어야 하며, 전통적인 테스트뿐만 아니라 AI 컴포넌트 및 AI 기반 시스템 고유의 특성을 다루기 위한 보다 전문화된 테스트를 포함해야 합니다.

AI 기반 시스템의 제품 리스크를 평가하는 대표적인 기법으로는 브레인스토밍, 전문가 인터뷰, 체크리스트 등이 있습니다.

- 브레인스토밍: 커뮤니케이션을 핵심으로 하는 한 번 이상의 회의를 통해 진행됩니다. 프로젝트 내에서 AI 기반 시스템의 전체적인 작동 방식을 완벽하게 파악하고 있는 사람은 없는 경우가 많습니다. 따라서 프로젝트의 어느 한 측면도 주류를 이루지 않는 상황에서, 특히 AI 와 관련된 광범위한 전문 기술을 보유한 다분야적팀 구성이 프로세스의 품질을 위해 필수적입니다. 이 팀의 목표는 AI 기반 시스템과 그 운영의 모든 측면에 대해 다양한 관점과 인식을 제공하는 것입니다.
- 전문가 인터뷰: 귀중한 통찰력과 전문 지식을 제공하여 리스크 평가 및 식별 프로세스를 강화합니다. 인터뷰 기법을 활용함으로써 조직은 AI 전문가로부터 직접 정보를 수집할 수 있고, 이를 통해 잠재적 리스크와 그 영향에 대해 더욱 포괄적으로 이해할 수 있게 됩니다.
- 체크리스트: 리스크 식별을 위한 체크리스트는 이전 AI 프로젝트들의 경험을 집약한 것입니다. 이상적으로는 AI 기반 시스템의 일반적인 리스크 영역에 초점을 맞춰 목록을 구성해야 합니다. 그 목적은 자칫 간과하기 쉽지만 빈번하게 발생하는 제품 리스크를 상기시키고, AI 기반 시스템이 겪을 수 있는 문제 유형에 대한 아이디어를 제공하는 데 있습니다.

#### AI 시스템의 제품 리스크 예시:

- 입력 데이터의 품질 저하
- 훈련을 위한 데이터 양 부족
- 운영 데이터 파이프라인의 문제
- 모델 개발에 사용된 ML 워크플로가 불완전할 가능성
- ML 프레임워크, 알고리즘, 모델, 모델 설정 및/또는 하이퍼파라미터의 잘못된 선택
- 운영 환경에서 목표한 ML 기능 성능을 달성하지 못함
- 제공된 ML 기능 성능에 대해 사용자가 불만족함

#### AI 시스템의 제품 리스크 예시 (계속):

- 자가 학습 시스템이 사용자가 기대하는 서비스를 제공하지 않음
- 사용자가 시스템의 의사결정 방식을 이해하지 못해 답답해함

- 입력 피처와 출력(Output) 간의 관계 변화로 인해 시간이 지남에 따라 운영 중인 모델의 성능이 저하됨(흔히 콘셉트 드리프트'라고 함)
- 사전 훈련된 모델이 자체적으로 알 수 없거나 문서화되지 않은 편향 및 결함을 가지고 있음
- 할루시네이션: 부실한 훈련 데이터, 그럴듯한 텍스트로 공백을 채우려는 모델의 경향, 또는 프롬프트 오해 등으로 인해 발생할 수 있습니다. AI 가 조작된 출력을 높은 확신을 가지고 제시하는 경우가 많기 때문에 심각한 문제가 될 수 있습니다.

## 2.2.2 SG2 테스트 접근법 수립

테스트 접근법은 식별되고 우선순위가 지정된 제품 리스크를 완화하기 위해 정의됩니다. 또한 테스트 접근법은 정의된 조직 차원의 테스트 전략(2.1.2 항 참조)에 따라 수립됩니다. 관련 당사자들에게 무엇을 해야 하며, 테스트가 어떻게 수행될지를 명확히 해야 합니다. 테스트 전략과 마찬가지로, 상세 테스트 접근법은 애플리케이션의 오프라인 테스트와 온라인 테스트이라는 두 가지 주요 단계로 구분됩니다. 수행될 각 테스트 레벨과 테스트 유형별로 테스트 접근법이 정의될 것입니다.

오프라인 단계는 데이터 준비, 전처리 및 데이터 입력 테스트(초기 테스트 레벨)으로 시작됩니다. 수행되는 활동에는 데이터 조작, 데이터 필터링 및 데이터 전처리가 포함됩니다. 데이터 전처리에는 데이터 임퓨테이션(결측값 처리), 전체적인 파악을 위한 데이터 시각화 및 이상치 처리, 상관 분석 및 차원 축소가 포함됩니다. 그 후 AI 모델에 대한 평가와 분석이 이루어집니다. 여기에는 가용한 입력 데이터셋을 훈련, 검증 및 테스트 데이터셋으로 분할하는 작업도 포함됩니다. 훈련 세트에는 모델 훈련에 사용되는 데이터가 포함됩니다. 검증 데이터셋은 ML 알고리즘이 훈련의 효과성을 평가하는 데 사용됩니다. 훈련 후 단계에서는 테스트 데이터셋을 사용하여 ML 모델을 평가합니다.

온라인 단계에서는 비(Non) AI 컴포넌트가 테스트되며, AI 부분과 비 AI 부분 간의 상호작용이 테스트됩니다. 이를 위해서는 일반적으로 AI 애플리케이션의 아키텍처에 대한 상세한 연구와 이해가 필요합니다. 마지막으로 완전히 통합된 AI 시스템이 테스트됩니다. 탐색적 테스트 외에도, AI 기반 시스템의 테스트 시나리오를 식별하고 정의하는 데 널리 사용되는 기법으로 메타모픽 테스트가 있습니다.

'준비 완료 정의(특정 프랙티스 2.3)'라고도 불리는 진입 기준과, '완료 정의(특정 프랙티스 2.4)'라고도 불리는 종료 기준 또한 정의된 테스트 접근법의 일부입니다. 특정 테스트 단계의 진입 기준은 일반적으로 이전 테스트 단계의 종료 기준 충족을 의미하지만, 테스트 준비 완료 및 테스트 환경 가용성과도 관련이 있습니다. 테스트 단계에 따라, 일부 종료 기준은 테스트 커버리지가 미해결 결함과 같이 전통적인 시스템의 기준과 유사할 수 있습니다. 또한 전통적인 시스템에서 사용되는 종료 기준과는 매우 다른, 종료 기준 정의의 기초로 사용할 수 있는 메트릭들도 존재합니다.

아래 표는 AI 모델 유형별로 사용할 수 있는 종료 기준(메트릭)의 예시를 제공합니다 [AiU].

| 학습유형   | 모델 타입 | 사용되는 종료 기준                                                                                                     |
|--------|-------|----------------------------------------------------------------------------------------------------------------|
| 비지도 학습 | 군집화   | 관성 (Inertia),<br>조정 랜드 지수 (Adjusted Rand Score)                                                                |
|        | 연관    | 지지도 (Support),<br>신뢰도 (Confidence),<br>향상도 (Lift)                                                              |
| 지도 학습  | 분류    | 정확도 (Accuracy),<br>정밀도 (Precision),<br>재현율/민감도 (Recall/Sensitivity),<br>특이도 (Specificity),<br>F1-점수 (F1-score) |
|        | 회귀    | 평균 제곱근 오차 (Root-mean-square-error, RMSE),<br>R-제곱 오차 (R-square error)                                          |

표 1: AI 모델 유형별 종료 기준 및 메트릭

특정 프랙티스 2.5 '중단 및 재개 기준 정의'가 AI 기반 시스템 테스트와 관련이 있는지 여부는 적용되는 수명주기와 접근 방식에 따라 달라집니다. 전통적인 접근 방식을 따를 경우에는 이 프랙티스가 적용 가능할 것입니다. 하지만 애자일 환경에서는 그 관련성이 낮을 가능성이 큼니다.

애자일에서는 통상적으로 현황 및 진척 점검 회의에서 이러한 기준들이 암묵적으로 다루어지기 때문에, 각 테스트 단계마다 이를 사전에 명시적으로 정의하는 것은 불필요한 과잉 작업이 될 수 있습니다. 진행에 잠재적 또는 실질적인 위협이 될 수 있는 블로킹 이슈가 발생하면, 이는 회의를 통해 논의됩니다. 해당 회의에서 팀은 이슈 해결을 위해 필요한 조치가 무엇인지 결정하게 됩니다. 따라서 애자일 프로젝트 관리 루틴 자체가 이 특정 프랙티스에 대한 대안 프랙티스 역할을 수행한다고 볼 수 있습니다.

### 2.2.3 SG3 테스트 추정치 수립

테스트 노력 추정은 프로젝트나 팀의 테스트 목적을 달성하는 데 필요한 테스트 관련 업무량을 예측하는 활동을 의미합니다. 이해관계자들에게 이 추정치가 여러 가지 가정에 기반하고 있으며, 항상 추정 오차(Estimation error)가 발생할 수 있음을 명확히 하는 것이 중요합니다. 이는 조직이 특정 유형의 시스템(예: AI 기반 시스템)에 대한 테스트 경험이 부족할 때 특히 더 그렇습니다.

일반적으로 다음과 같은 두 가지 추정 접근법을 사용할 수 있습니다.

비율 기반 추정 (Estimation based on ratios). 이 메트릭 기반 기법에서는 조직 내 이전 프로젝트 들로부터 수치를 수집하며, 이를 통해 유사한 프로젝트에 적용할 수 있는 "표준" 비율을 도출할 수 있습니다. 조직 자체 프로젝트의 비율(예: 과거 데이터에서 산출된 비율)은 일반적으로 산정 과정에서 활용하기에 가장 좋은 자료입니다. 이렇게 도출된 표준 비율은 이후 신규 프로젝트의 테스트 노력을 산정하는 데 사용될 수 있습니다. AI 기반 시스템 테스트는 업계와 대다수 조직에 있어 비교적 새로운 분야이므로, 활용 가능한 메트릭이 있다 하더라도 그 수가 적어, 이 산정 접근법은 현재로서는 AI 기반 시스템 테스트에 아직 선호되지 않는 접근법입니다.

전문가 기반 추정 (Expert-based estimation), 예: 와이드밴드 델파이 이 기법에서는 '전문가'들이 경험을 바탕으로 추정을 수행합니다. 작업 분할을 기초로 각 전문가는 개별적으로 각 태스크에 대한 노력을 추정합니다. 결과를 취합한 후, 합의된 범위를 벗어나는 편차가 있는 경우 전문가들이 현재의 추정치에 대해 논의합니다. 플래닝 포커는 와이드밴드 델파이의 변형으로, 애자일 소프트웨어 개발에서 흔히 사용됩니다. 메트릭을 사용할 수 없는 상황에서는, 이 기법이 AI 기반 시스템 테스트와 관련된 노력 및 비용을 추정하는 데 선호되는 방법으로 보입니다.

프로젝트나 팀에 참여한 인원, 그리고 조직 내 다른 AI 프로젝트에 참여해 본 경험이 있는 인원들이 추정 팀의 일원이 될 수 있습니다. 이 프로세스를 따르면서 그들은 최선의 추측을 내놓고, 추정 시 전제한 가정 사항들도 문서화하여 테스트 노력과 비용을 산출하게 됩니다. 와이드밴드 델파이 추정 세션에는 정의된 테스트 접근법에 기반하여 개발될 테스트 작업 산출물 목록과, 이와 관련된 수행해야 할 테스트 태스크 목록 등이 입력물로 제공됩니다.

AI 시스템 테스트를 위한 노력 추정은 전통적인 소프트웨어 테스트보다 일반적으로 더 복잡합니다. 다음은 고려해야 할 AI 특유의 요소들에 대한 몇 가지 예시입니다.

- 데이터 복잡성: AI 시스템/애플리케이션 테스트는 정확하고 공정한 예측을 보장하기 위해 데이터 검증, 편향 탐지, 드리프트 모니터링에 추가적인 노력이 요구됩니다.
- AI 모델 동작: 전통적인 소프트웨어와 달리 AI 는 학습하고 진화하므로, 지속적인 테스트, 설명 가능성 확인, 그리고 튜닝이 필요합니다.
- 지속적인 모니터링 : AI 는 성능을 유지하기 위해 실시간 추적, 성능 확인, 모델 재훈련이 필요합니다.
- 규제 준수: AI 테스트는 윤리적이고 합법적인 사용을 보장하기 위해 GDPR, AI 법(AI Act), 그리고 산업별 규칙을 준수해야 합니다.

## 2.2.4 SG4 테스트 계획 수립

테스트 계획은 항상 두 주요 그룹, 즉 (1) 테스터(무엇을 해야 할지 알아야 함)와 (2) 이해관계자(진행될 테스트를 이해하고, 참여하며, 자원에 동의하고, 관련 리스크를 파악해야 함)를 염두에 두고 수립되어야

합니다. AI 기반 시스템 테스트에서 테스트 계획의 상세 수준은 적용되는 수명주기에 따라 달라지는데, 예를 들어 애자일 방식의 업무 환경은 보다 경량화된 테스트 계획을 의미하게 됩니다.

AI 시스템을 위한 테스트 계획은 AI 의 데이터 주도적이고, 적응형이며, 확률적인 특성으로 인해 전통적인 소프트웨어 테스트의 범위를 넘어서야 합니다. 기존 애플리케이션과 달리 AI 모델은 학습하고, 진화하며, 확률적인 의사결정을 내리므로 전문화된 테스트 전략과 집중적인 테스트 계획이 요구됩니다. AI 의 복잡성과 진화하는 특성을 고려할 때, 테스트는 지속적이고 자동화되어야 하며, 파이프라인을 통해 지속적인 검증을 보장해야 합니다.

AI 기반 시스템 테스트는 많은 조직에게 비교적 새로운 분야이기 때문에, 테스터들이 수행해야 할 업무를 이해하고 이해관계자들이 충분한 정보를 얻을 수 있도록 문서화된 테스트 계획이 필요할 것입니다. 테스트 계획은 목적에 부합한다면 문서, 프레젠테이션, 또는 마인드맵 등의 형식을 취할 수 있습니다. 테스트 계획에는 이전 테스트 계획 활동의 결과와 우선순위가 지정된 제품 및 프로젝트 리스크 목록이 문서화되며, 오프라인 단계(데이터 입력 테스트, 모델 테스트)와 온라인 단계의 테스트 레벨 및 테스트 유형에 대한 접근법과 활동이 상세히 기술됩니다.

테스트 일정도 개발됩니다. 테스트는 개발의 필수적인 부분으로 수행되므로, 통상적으로 별도의 테스트 일정이 존재하기보다는 테스트 활동을 포함하는 전체 일정이 수립됩니다. 일정은 순차적 수명주기에서와 같이 상세하게 문서화된 일정 형식을 취할 수도 있고, 애자일 수명주기에서와 같이 우선순위가 있는 태스크 보드형식을 취할 수도 있습니다.

테스트 계획의 일환으로, 오프라인 및/또는 온라인 테스트를 수행하는 데 필요한 지식과 기술을 갖춘 필수 테스트 인력 자원의 가용성에 대한 계획이 수립됩니다. 물론 이러한 종류의 테스트를 수행할 수 있는 지식과 기술을 갖춘 테스터는 일반적으로 희소한 자원이기 때문에 이는 도전적인 과제가 될 수 있습니다. 따라서 테스트 인력이 테스트 업무를 완수할 수 있는 역량을 갖추도록 보장하기 위해, 테스트 계획의 일부로 필요한 교육을 정의해야 할 수도 있습니다.

## 2.2.5 SG5 테스트 계획에 대한 요약 획득

마지막으로, 테스트 계획과 정의된 테스트 접근법에 대해 이해관계자들의 합의가 필요합니다. 이해관계자들은 우선순위가 지정된 제품 리스크 목록과 수행해야 할 테스트 완화 조치에 대해 이해하고 동의해야 합니다. 이들의 이해와 확약은 예를 들어 제품 리스크, 테스트 접근법 및 그 근거, 그리고 소요 예산을 설명하는 짧은 프레젠테이션과 이어지는 토의를 통해 확보할 수 있습니다. 테스트 계획과 테스트 접근법에 대한 논의는 개발될 ML 모델 또는 AI 시스템의 목적이라는 맥락 안에서 이루어져야 하며, 비즈니스 우선순위와 부합하는지 확인해야 합니다. 물론 인수 기준 또한 이해관계자들과 논의하고 합의해야 할 중요한 사항입니다.

만약 테스트 계획과 테스트 접근법을 개발하고 수립하는 데 있어 보다 애자일한 접근법이 사용된다면, 이는 이미 (팀원 중 한 명인) AI 테스트 전문가가 주도하는 팀 기반의 활동으로 진행되었을 것입니다. 제품 품질은 팀 전체의 책임입니다. 따라서 팀이 올바른 프로세스를 따랐다면, (테스트) 접근법과 (테스트) 계획에 대한 확약은 팀 활동의 자연스러운 결과로서 이미 내재되어 있는 것입니다. 물론, 팀 외부의 다른 이해관계자들로부터 테스트 계획에 대한 확약을 받아야 할 수도 있습니다. 이 경우, 앞서 언급한 프레젠테이션 및 토의 세션은 여전히 중요하며 반드시 수행되어야 합니다.

## 2.3 프로세스 영역 2.3 테스트 모니터링 및 제어 (TMC)

*테스트 모니터링 및 제어의 목적은 테스트 진척 상황과 제품 품질에 대한 이해를 제공함으로써, 테스트 진척이 계획에서 상당히 벗어나거나 제품 품질이 기대치에서 크게 벗어날 경우 적절한 시정 조치를 취할 수 있도록 하는 것입니다.*

### 2.3.1 SG1 계획 대비 테스트 진척 모니터링

테스트 진척을 모니터링하고, 리뷰를 수행하며, 테스트 진척 상황을 보고하는 것은 AI 기반 시스템 테스트라고 해서 전통적인 시스템 테스트와 다르지 않습니다. 테스트 진척 모니터링이 구현되고 수행되는 방식은 적용되는 수명주기에 크게 좌우됩니다. 순차적 V-모델 유형의 수명주기에서 테스트 진척 모니터링은 통상적으로 보다 공식적이며 문서화의 비중이 높습니다. 반면 애자일 유형의 수명주기에서 테스트 진척 모니터링은 훨씬 덜 공식적이며, 태스크 보드등을 통해 지원되며 문서화가 적습니다. 이에 대한 자세한 내용은 [TMMiAgile]을 참조하십시오.

### 2.3.2 SG2 계획 및 기대치 대비 제품 품질 모니터링

제품 품질을 모니터링할 때, AI 기반 시스템 테스트는 전통적인 시스템 테스트와는 다르다는 점을 이해하는 것이 중요합니다. 이러한 차이는 모니터링해야 할 제품 품질의 측면에도 영향을 미칩니다. 전통적인 시스템에서 "테스트"는 결함을 찾고, 테스트를 실행하며, 성능을 측정하는 등의 다양한 프로세스를 지칭합니다. 그러나 AI에서의 테스트는 전용 데이터셋에 대한 시스템 성능의 '통계적 평가'를 구체적으로 의미합니다. 기본적으로 모니터링되고 보고되는 종료 기준의 유형 자체가 본질적으로 다릅니다 (이에 대한 몇 가지 예시는 2.2.2 항을 참조하십시오).

요약하자면, 제품 품질 모니터링은 식별된 제품 리스크와 정의된 종료 기준에 대해 중점적으로 수행됩니다. 결함(Defects)에 대한 모니터링은 시스템의 비(Non) AI 부분에 대해서는 충분히 수행될 수 있습니다. 제품 품질 모니터링을 구현하는 방식에 대해서는 앞선 단락을 참조하십시오.

AI 시스템의 제품 품질 모니터링에 있어 또 하나의 차이점은 시스템 성능을 지속적인 기반 위에서 모니터링한다는 점입니다. AI 시스템에서는 실제 사용 중인 시스템의 품질을 지속적으로 검토하고 재확인해야 할 필요가 빈번합니다. 본 문서에서 앞서 언급했듯이, AI 기반 시스템에서는 모델 드리프트, 데이터 드리프트, 그리고 콘셉트 드리프트가 시간이 지남에 따라 고려하고 모니터링해야 할 요소들입니다. 드리프트의 징후를 조기에 감지하기 위해서는 정확도와 같은 모델 성능 지표를 지속적으로 모니터링해야 합니다. 릴리스 이후에도 제품 품질을 계속 모니터링해야 하는 또 다른 중요한 이유로는 적대적 공격, 데이터 중독, 그리고 자가 학습 진화 행동을 방지하는 것 등이 있습니다.

### 2.3.3 SG3 시정 조치 관리 및 종결

테스트 진척 모니터링(위 내용 참조)과 마찬가지로, 시정 조치를 관리하고 종결하는 것 또한 AI 기반 시스템 테스트가 전통적인 시스템 테스트와 다르지 않습니다. 하지만 이슈의 '유형'은 다를 가능성이 높습니다. AI 시스템 테스트 시 예상되는 전형적인 이슈는 입력 데이터 품질 영역에서 발생합니다. 데이터 품질 이슈는 프로젝트에 중대한 영향을 미치거나 해결하는 데 오랜 시간이 걸릴 수 있습니다.

시정 조치 관리 및 종결이 구현되고 수행되는 방식은 적용되는 수명주기에 크게 좌우됩니다. 순차적 V-모델 유형의 수명주기에서는 진척과 관련된 이슈 프로세스를 통상적으로 테스트 관리자나 프로젝트 관리자가 관리합니다. 애자일 유형의 수명주기에서는 스크럼 마스터가 진척 방해 요소를 처리하는 책임을 맡는 경우가 많습니다. 이러한 유형의 시정 조치에 대한 상태는 태스크 보드에 게시되며 일일 스탠드업 미팅에서 논의됩니다.

## 2.4 프로세스 영역 2.4 테스트 설계 및 실행 (TDE)

*테스트 설계 및 실행의 목적은 테스트 설계 명세서를 수립하고, 테스트 설계 기법을 사용하며, 구조화된 테스트 실행 프로세스를 수행하고, 테스트 인시던트를 종결될 때까지 관리함으로써, 테스트 설계 및 실행 단계의 테스트 프로세스 역량을 향상시키는 것입니다.*

*비록 상위 레벨에서의 목적 진술은 여전히 유효하지만, 테스트 조건과 테스트 케이스를 정의하고 이를 실행하는 구체적인 프로세스는 이하 설명과 같이 매우 다릅니다. 모델이 생성되면(즉, 훈련, 평가 및 튜닝이 완료되면), 합의된 기능적 성능 기준(2.2.2 항 참조)이 충족되는지 확인하기 위해 독립적인 테스트 데이터셋에 대해 테스트되어야 합니다.*

### 2.4.1 SG1 테스트 설계 기법을 활용한 테스트 분석 및 설계 수행

AI 시스템을 테스트할 때, 테스트 조건과 테스트 설계의 개념은 전통적인 소프트웨어 테스트와

근본적으로 다릅니다. 명시적인 입력-출력 쌍(input-output pairs)을 도출하는 대신, 훈련된 모델의 행동, 정확성, 견고성을 의미 있게 평가할 수 있는 데이터셋을 구축하고 검증하는 데 초점이 맞춰집니다.

주된 목표는 AI 모델이 작동할 다양한 조건의 범위를 대표하는 데이터셋을 정의하고 설계하는 것입니다. 여기에는 훈련, 검증 및 테스트에 사용될 데이터의 유형, 출처, 품질을 식별하는 작업이 포함됩니다. 통상적으로 공통의 데이터 소스로부터 통계적 대표성을 갖는 세 가지의 별도 데이터셋이 구축됩니다.

- 훈련 데이터셋: AI/ML 모델을 훈련시키는 데 사용됩니다.
- 검증 데이터셋: 모델 파라미터를 평가하고 튜닝하는 데 사용됩니다.
- 테스트 데이터셋(또는 홀드아웃 데이터셋): 튜닝된 모델의 일반화 성능을 평가하는 데 사용됩니다.

TMMi 맥락에서 테스트 분석 및 설계 단계는 AI 시스템의 목적, 요구사항, 리스크 평가로부터 도출된 테스트 데이터 조건과 평가 메트릭을 식별하고 우선순위를 지정하는 활동을 포함합니다. 여기서 테스트 베이스에는 명세서나 설계 문서뿐만 아니라 데이터 수집 전략, 라벨링 가이드라인, 모델 성능 기준까지 포함됩니다.

적절하고 편향되지 않은 데이터가 제한적인 경우가 많으므로, 훈련 및 검증 데이터셋은 단일 통합 데이터셋에서 파생되는 경우가 빈번하지만, 테스트 데이터셋은 반드시 격리되어야 합니다. 이러한 분리는 객관적인 모델 평가를 보장하고 데이터 누수를 방지하기 위함입니다. 데이터셋 분할에 대한 만능 비율은 없으나, 일반적으로 훈련, 검증, 테스트에 대해 60:20:20의 비율이 가이드라인으로 사용됩니다. 데이터 분할은 무작위로 수행되어야 하지만, 특정 층화나 대표성 제약 조건(예: 소규모 데이터셋 또는 중요한 하위 모집단)이 존재하는 경우는 예외로 합니다.

AI에서의 테스트 설계는 현실적인 시나리오와 엣지 케이스 시나리오를 반영하는 테스트 케이스 및 조건을 선정하고 우선순위를 정하는 데 중점을 둡니다. 여기에는 다음이 포함될 수 있습니다.

- 모델의 약점을 파악하기 위한 적대적 또는 스트레스 테스트 데이터 설계
- 데이터 공간 내에서의 동등 분할 및 경계 테스트 사용 (예: 피처 분포의 엣지 값 활용)
- 조합 테스트 기법(예: 페어와이즈 테스트) 사용. 이는 AI 기반 시스템의 관심 파라미터 수가 극도로 많을 수 있기 때문이며, 특히 시스템이 빅데이터를 사용하거나 외부 세계와 상호작용할 때 더욱 그렇습니다.
- 메타모픽 테스트 사용. 이는 이미 통과한 원본 테스트 케이스를 기반으로 변형된 테스트 케이스를 생성하는 기법으로, AI 시스템 테스트에 활용될 수 있습니다.

- 테스트 조건 세트의 일부로 편향(bias), 공정성(fairness), 설명 가능성(explainability) 테스트 적용
- 데이터 소스 및 모델 요구사항에서부터 테스트 데이터셋 및 성능 결과에 이르는 추적성(traceability) 확립

이 목표의 산출물에는 데이터셋 명세서, 테스트 데이터 커버리지 문서, 그리고 정의된 인수 기준(예: 정확도 임계값, 신뢰 구간, 편향 한계치)이 포함됩니다. 이러한 산출물들은 AI 모델 테스트가 구조화된 접근 방식을 따르고 있음을 보장합니다.

## 2.4.2 SG2 테스트 구현 수행

AI 시스템을 위한 테스트 구현 단계에서는 테스트 설계 시 정의된 바에 따라 테스트 데이터와 환경을 획득하고, 준비하며, 확정하는 활동을 수행합니다. AI 테스트는 훈련된 모델의 성능, 견고성(Robustness), 그리고 요구사항 준수 여부를 검증하는 데 사용될 데이터셋을 확보하고 정제(Curation)하는 데 중점을 둡니다.

획득한 데이터는 수치형, 범주형, 이미지, 표, 텍스트, 시계열, 센서, 지리공간, 비디오, 오디오 등 다양한 형태를 띌 수 있으며, 실제 테스트에 사용하기 위해서는 일반적으로 광범위한 전처리(Pre-processing) 과정을 거쳐야 합니다. 주요 전처리 활동에는 정제, 변환, 증강, 샘플링이 포함됩니다.

- 정제: 부정확하거나 중복된 데이터, 혹은 일관성 없는 데이터를 식별하여 수정하거나 제거합니다. 누락된 데이터는 임putation(결측치 대체) 기법을 사용하여 처리하며, 개인정보나 민감 정보 보호를 위해 익명화(Anonymization) 또는 가명화(Pseudonymization)를 적용합니다. 이상치(Outliers)는 AI 시스템의 운영 맥락과의 관련성에 따라 제거하거나 유지할 수 있습니다.
- 변환: 데이터의 형식이나 구조를 테스트 대상 모델에 적합한 일관된 형태로 변환합니다 (예: 범주형 데이터를 수치형으로 변환, 수치형 피쳐의 정규화, 또는 이미지 포맷 재인코딩 등). 이러한 변환 과정은 테스트 실행 전반에 걸쳐 비교 가능성과 안정성을 보장합니다.
- 증강: 커버리지와 견고성을 향상하기 위해, 새로운 샘플이나 수정된 샘플을 생성하여 데이터셋을 인위적으로 확장할 수 있습니다 (예: 회전된 이미지, 패러프레이징된 텍스트, 노이즈가 주입된 데이터 등). 증강에는 시스템이 작은 변화나 공격에 얼마나 저항력이 있는지 테스트하기 위해 고안된 적대적 예제가 포함될 수도 있습니다.
- 샘플링: 대규모 데이터셋에서 대표성 있는 부분집합을 선택하여 테스트의 실행 가능성을 확보하는 동시에 통계적 유효성을 유지합니다. 샘플링 방식(무작위, 층화, 또는 리스크 기반)은 테스트 설계 시 식별된 다양성과 핵심적인 특성들을 보존해야 합니다.

모든 전처리 구현 단계는 문서화되어야 하며 데이터셋 명세서와의 추적성(Traceability)이 확보되어야 합니다. 구현된 테스트 데이터, 테스트 스크립트, 그리고 환경 구성 정보는 테스트 실행 및 모델 평가에 필요한 테스트 작업 산출물(Test work products)을 구성합니다.

### 2.4.3 SG3 테스트 실행 수행

AI 시스템을 위한 테스트 실행은 준비된 테스트 데이터셋과 스크립트를 구동하여, 훈련된 모델이 정의된 인수 기준 대비 어떻게 수행되는지에 대한 증거를 수집하는 활동입니다. 이 활동의 목표는 모델이 의도한 대로 동작하고, 기대 수준의 성능을 달성하며, 앞선 단계에서 식별된 다양한 데이터 조건에서 신뢰성 있게 작동함을 보장하는 것입니다. 테스트는 AI 모델의 훈련, 튜닝, 검증이 완료되고 정의된 기준을 충족한 후에만 시작됩니다. 이는 테스트 실행을 시작하기 위한 진입 기준으로 간주될 수 있습니다.

AI 시스템 테스트에서 테스트를 실행한다는 것은 일반적으로 훈련 및 검증된 모델에 테스트 데이터셋을 입력하고, 그 출력값(예측, 분류, 점수, 또는 행동)을 기록하는 것을 의미합니다. 이러한 출력값은 기대 결과 또는 참조 결과(즉, 정답 데이터/Ground truth)와 비교되며, 혼동 행렬(Confusion matrices) 및 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), F1 점수(F1 score)와 같은 정량적 성능 지표를 사용하여 평가됩니다 (2.2.2 항 참조).

AI 시스템이 본질적으로 지니는 확률적 특성으로 인해, 개별 테스트 실행은 약간씩 다른 결과를 낼 수 있습니다. 따라서 테스트는 통제된 조건하에서 반복 수행되어야 하며, 그 결과는 통계적으로 분석되어야 합니다 (예: 성능 지표의 평균과 분산 확인). 이를 통해 도출된 결론이 재현 가능하고 안정적인 증거에 기반하고 있음을 보장할 수 있습니다.

모든 테스트 결과, 이상 징후, 그리고 기대 성능과의 편차는 문서화되어야 하며, 정의된 해당 테스트 데이터 및 조건과의 추적성이 확보되어야 합니다. 테스트 인시던트는 그 원인이 데이터 품질 문제인지, 모델의 한계인지, 혹은 구현상의 결함인지를 판단하기 위해 분석됩니다. 이 특정 목표의 산출물에는 실행된 테스트 로그, 캡처된 결과, 성능 보고서, 인시던트 보고서가 포함됩니다.

### 2.4.4 SG4 테스트 인시던트 종결 관리

테스트 인시던트를 관리하고 종결하는 프로세스와 프랙티스는 AI 시스템과 관련된 인시던트 및 결함이라고 해서 다르지 않습니다. 물론 더 상세한 기술적 수준으로 들어가면 발견되는 결함의 유형이나

결함 분석 수행 방식 등에는 차이가 있습니다. 예를 들어, 상세 데이터 품질 분석을 통해 정확도 저하, 편향된 모델, 손상된 모델과 관련된 결함이 드러날 수 있습니다.

인시던트 관리 프로세스 또한 적용되는 개발 수명주기에 크게 좌우됩니다. 순차적인 전통적 프로젝트와 비교할 때, 애자일 환경의 프로젝트에서는 인시던트가 문서화되고 종결될 때까지 관리되는 방식에 상당한 차이가 있습니다 [TMMiAgile].

## 2.5 프로세스 영역 2.5 테스트 환경 (TEV)

*테스트 환경 프로세스 영역의 목적은 테스트를 관리 가능하고 반복 가능한 방식으로 실행할 수 있는 적절한 환경을 수립하고 유지하는 것입니다.*

*테스트 환경 프로세스 영역을 통해, 일반 테스트 데이터를 포함하여 테스트를 관리 가능하고 반복 가능한 방식으로 실행할 수 있는 적절한 테스트 환경이 수립되고 유지됩니다. AI 기반 시스템은 매우 다양한 운영 환경에서 사용될 수 있으며, 이는 곧 테스트 환경 또한 그만큼 다양함을 의미합니다. 통상적으로 테스트 환경을 전통적인 시스템과 다르게 만드는 AI 기반 시스템의 특징으로는 자가 학습, 자율성, 멀티 에이전트, 빅데이터 등이 있습니다.*

*또한, AI 기반 시스템을 테스트할 때는 가상 테스트 환경이 흔히 사용됩니다. 가상 환경을 사용하면 위험하거나, 이례적이거나, 극한의 상황, 또는 시간이 많이 소요되는 시나리오를 테스트하기가 훨씬 수월해집니다. 또한 가상 테스트 환경은 테스트 환경에 대해 훨씬 더 큰 제어성을 제공합니다. 가상 테스트 환경을 통한 하드웨어 시뮬레이션은 실제로는 가용하지 않을 수 있는 하드웨어 컴포넌트들을 (시뮬레이션하여) 시스템과 함께 테스트할 수 있게 해줍니다.*

*참고로, 통상 AI/ML 시스템 워크플로의 큰 비중을 차지하는 데이터 준비는 '테스트 설계 및 실행' 프로세스 영역(2.4 항 참조)에 속한다는 점에 유의해야 합니다. 원칙적으로 데이터는 튜닝된 모델을 훈련, 검증, 테스트하는 데 사용될 (테스트) 케이스를 포괄합니다. 따라서 (테스트) 데이터는 전통적인 시스템에 비해 AI 기반 시스템 테스트에서 매우 중요한 역할을 수행합니다.*

### 2.5.1 SG1 테스트 환경 요구사항 개발

AI 기반 시스템을 위한 테스트 환경 요구사항 명세는 프로젝트 초기에 수행되어야 합니다. 이를 위해서는 테스트 환경 요구사항을 정의하는 인원이 개발될 AI 시스템에 대해 상세히 이해하고 있어야 합니다. 왜냐하면 앞서 언급한 시스템의 특징들이 무엇이 필요한지를 결정짓기 때문입니다. 요구사항 명세는 그 정확성, 적합성, 실현 가능성, 그리고 실제 운영 환경을 얼마나 정확하게 대변하는지를 확인하기 위해 검토됩니다.

초기에 요구사항을 명세화하면 필요한 테스트 환경과 컴포넌트를 획득하거나 개발할 시간을 더 많이 확보할 수 있다는 장점이 있습니다. 특히 AI 기반 시스템의 경우, 테스트 환경이 단순하지 않고 꽤 복잡할 수 있기 때문에 이는 매우 중요합니다.

## 2.5.2 SG2 테스트 환경 구현 수행

이 TMMi 고유 목표 단계에서는 테스트 환경 요구사항이 구현되고, AI 모델 테스트에 사용할 수 있도록 테스트 환경이 가용해집니다. 적절한 경우 테스트 환경 컴포넌트에 대해 단위 테스트가 수행되며, 구현 단계의 마지막에는 테스트 환경이 AI 모델 테스트에 사용될 준비가 되었는지 판단하기 위해 확인 테스트가 수행됩니다. 환경 자체를 테스트하는 동안에는 일반 테스트 데이터가 사용됩니다.

## 2.5.3 SG3 테스트 환경 관리 및 제어

테스트 환경에 대한 관리와 제어는 AI 프로젝트 전반에 걸쳐 이루어집니다. 종종 프로젝트 팀이 직접 테스트 환경을 구축하고 유지보수해야 하는 경우가 있지만, 팀 외부의 별도 조직이 이를 수행할 수도 있습니다. 만약 이 활동을 팀이 직접 수행한다면, 당연히 팀이 이러한 작업을 수행할 수 있는 충분한 기술적 지식을 갖추고 있음을 전제로 합니다.

또한 이는 환경 관련 작업들이 프로젝트의 일부가 됨을 의미합니다. 예를 들어, 계획 세션에서 다루어져야 하고, 태스크 보드에 게시되어야 하며, 이슈가 발생할 경우 일일 스탠드업 미팅에서 논의되어야 합니다.

결론적으로 '테스트 환경' 프로세스 영역의 고유 목표와 프랙티스들은 AI 시스템에서도 모두 적용 가능하며 본질적으로 변하지 않습니다. 달라질 수 있는 점은 테스트 환경의 유형, 그리고 필요한 테스트 데이터와 그 컴포넌트들입니다.

# 3 TMMi 레벨 3: 정의됨 (Defined)

## 3.1 프로세스 영역 3.1 테스트 조직 (Test Organization)

*테스트 조직(Test Organization) 프로세스 영역의 목적은 테스트를 책임지는 고도로 숙련된 인력 그룹을 식별하고 조직하는 것입니다. 이 테스트 그룹은 테스트뿐만 아니라, 조직의 현행 테스트*

프로세스 및 테스트 프로세스 자산이 지닌 강점과 약점에 대한 철저한 이해를 바탕으로 이를 개선하는 활동도 관리합니다.

### 3.1.1 SG1 테스트 조직 수립

테스트 조직은 흔히 오해받는 프로세스 영역입니다. 많은 사람들이 이를 TMMi 가 독립적인 테스트를 수행하는 별도의 테스트 그룹, 심지어 별도의 부서를 요구하는 것으로 해석합니다. 그러나 AI 시스템에 가장 흔히 적용되는 테스트 모델은 애자일이며, 자동화와 모니터링의 지원을 받아 AI 개발팀 내에 내재화된 지속적 테스트입니다. 애자일에서, 그리고 따라서 AI 시스템 테스트에서도 테스트 조직은 테스트 프로세스와 전문성(스킬 및 커리어 경로 포함)의 조직적 개선에 초점을 맞춘 테스트 역량 센터(test competence centre)의 형태를 취합니다. 이미 테스트 역량 센터를 운영 중인 조직이라면 원칙적으로 그 센터가 AI 시스템 테스트도 함께 다루어야 합니다. 주된 변화는 (뒤이어 나오는 SG2 참조) 요구되는 역량과 책임에서 나타납니다.

AI 모델 및 AI 기반 시스템 테스트와 관련하여 테스트 조직에 새롭게 부여되는 구체적 책임의 예는 다음과 같습니다:

- AI 테스트 전략 정의
- AI 테스트 프로세스 및 표준 정의
- AI 테스트 모범 사례(best practice) 제공
- 팀의 AI 모델 품질 및 데이터 편향 평가 지원
- 모니터링 및 배포 후 검증 관행 정의

### 3.1.2 SG2 테스트 전문가를 위한 테스트 기능 수립

테스팅은 가치 있는 전문 직군 분야로 여겨집니다. AI 시스템 테스트에는 특정한 지식과 스킬이 요구되므로, 테스터의 직무 기술서 또한 갱신되어야 합니다. 이 갱신은 단순히 AI 시스템 테스트에

대한 책임을 추가하는 것에 그치지 않으며, 더 중요하게는 요구되는 지식과 역량을 확장하는 것을 의미합니다. 테스터는 여전히 전통적인 테스트 지식과 스킬을 필요로 하지만, 이에 더해 AI 시스템에 특화된 역량도 갖추어야 합니다. 여기에는 지도학습, 비지도학습, 강화학습과 같은 AI 및 머신러닝 개념에 대한 기본적 이해, 훈련·검증·테스트 데이터셋에 대한 이해, 그리고 과적합(overfitting), 과소적합(underfitting), 모델 드리프트와 같은 이슈에 대한 이해가 포함됩니다. 또한 테스터는 데이터 프로파일링과 탐색, 완전성·편향·노이즈·이상치·중복 여부 점검, 데이터 파이프라인에 대한 이해 등 데이터 품질 및 데이터 테스트 관행에 대한 확고한 인식을 갖추어야 합니다. AI의 결과는 이분법적인 경우가 드물기 때문에, 테스터는 정밀도(precision), 재현율(recall), F1 점수, 신뢰구간, 오탐(false positive)과 미탐(false negative), 표본 추출 기법과 같은 개념에 대한 지식을 바탕으로 통계적·확률적 사고를 적용해야 합니다. 아울러 AI 시스템 테스트에는 편향, 공정성, 윤리, 설명 가능성, 해석 가능성과 같은 새로운 품질 측면과 관련된 지식과 스킬이 요구됩니다. 마지막으로, AI 시스템은 새로운 소프트웨어 릴리스가 없어도 시간이 지남에 따라 변화할 수 있으므로, 테스터는 지속적인 모니터링과 릴리스 이후 테스트에도 대비해야 합니다.

AI 시스템 테스트는 팀 차원의 노력입니다. 테스터는 데이터 엔지니어와 데이터 과학자, ML 엔지니어, 도메인 전문가 등 테스터가 아닌 인력들과 긴밀히 협업하게 됩니다. 테스트를 전담하지 않는 다른 직무의 직무 기술서에도 그들에게 기대되는 테스트 활동, 역할, 책임, 그리고 요구되는 테스트 지식과 스킬을 구체화하여 반영하는 것이 중요합니다.

### 3.1.3 SG3 테스트 커리어 경로 수립

테스트 커리어 경로는 테스트 전문가들이 지식, 스킬, 지위, 보상을 향상시켜 자신의 커리어를 발전시킬 수 있도록 수립됩니다. 정의된 테스트 커리어 경로를 바탕으로, 테스트 조직의 모든 구성원에 대해 개인별 테스트 커리어 개발 계획이 수립되고 유지·관리됩니다. 조직 내에 이미 마련되어 있는 커리어 경로는 대체로 큰 변화 없이 유지됩니다. 다만 AI 비중이 높은 조직에서는 AI 시스템 테스트가 테스트 활동의 핵심 부분이 되면서, 테스트 경로에 남아 AI 시스템에 집중하고자

하는 이들을 위한 커리어 경로로서 AI 테스트 전문가(specialist)가 흔히 새로운 기회로 제시됩니다. AI 테스트 전문가는 AI(테스팅) 측면에 대한 심층적인 지식과 스킬을 갖추게 될 것으로 기대되며, 통상 AI 테스트 전략에 대한 오너십을 갖습니다.

### 3.1.4 SG4 테스트 프로세스 개선의 결정, 계획 및 실행

테스트 프로세스 개선의 목표(리스크 감소, 신뢰 향상)는 전통적인 시스템의 경우와 동일합니다. 그러나 AI 시스템에서는 '품질', '커버리지', 심지어 '결함'이 의미하는 바 자체가 달라지므로, 개선점을 식별하는 방식도 함께 진화해야 합니다. 전통적인 소프트웨어의 경우 테스트 개선은 대개 누락된 결함, 테스트 커버리지의 공백, 테스트 실행 메트릭, 근본 원인 분석으로부터 도출됩니다. 이러한 요인들은 AI 시스템에도 여전히 적용되지만, 그것만으로는 충분하지 않습니다. AI 시스템은 전통적인 테스팅으로는 대응되지 않는 새로운 실패 유형(failure mode)을 초래합니다. AI 테스팅에서는 모델 성능 저하, 데이터 관련 실패, 편향·공정성·윤리 이슈, 그리고 재현 불가능한 테스트 결과로부터도 테스트 프로세스 개선점이 도출됩니다.

테스트 프로세스 개선의 트리거가 서로 다른 경우가 많은 만큼, 이에 따라 취해야 할 개선 조치 또한 달라집니다. AI 시스템 테스팅에서 전형적으로 나타나는 테스트 프로세스 개선 사례 몇 가지를 표 2에 제시합니다.

| 트리거             | 전형적인 테스트 개선 조치                                    |
|-----------------|---------------------------------------------------|
| 모델 성능 저하        | 데이터 드리프트 탐지 추가<br>모니터링 기반 테스팅 개선<br>재훈련 검증 게이트 도입 |
| 데이터 관련 실패       | 데이터 커버리지 분석 확대<br>합성 데이터 생성 추가<br>데이터 검증 테스트 개선   |
| 편향, 공정성 및 윤리 이슈 | 공정성 테스팅 추가                                        |

| 트리거            | 전형적인 테스트 개선 조치                                                   |
|----------------|------------------------------------------------------------------|
|                | 보호 속성(protected attribute)별 평가 세분화<br>윤리적 리스크 검토의 공식화            |
| 재현 불가능한 테스트 결과 | 무작위성 통제(시드값 고정)<br>통계적 신뢰도 중심으로 초점 전환<br>분포 기반 어서션(assertion) 사용 |

표 2: AI 시스템 테스트에서의 테스트 개선 사례

필요한 테스트 프로세스 개선사항이 결정되면, 이를 계획하고 이어서 실제로 개선할 수 있습니다. 이 세부 목표의 해당 부분에는 표준 TMMi 모델의 세부 프랙티스(SP4.3 테스트 프로세스 개선 계획 수립, SP4.4 테스트 프로세스 개선 실행)가 그대로 적용됩니다.

### 3.1.5 SG5 조직 테스트 프로세스의 배포 및 교훈의 반영

AI 시스템 테스트에 관여하는 프로젝트나 팀 역시, 필요하고 부가가치가 있는 범위 내에서 AI 시스템 테스트를 위한 조직 표준 테스트 프로세스와 관련 테스트 프로세스 자산에 접근하고 이를 재사용할 수 있어야 한다는 점이 중요합니다. 테스트 조직은 AI 시스템 테스트에 관여하는 모든 팀에 테스트 프로세스와 테스트 프로세스 자산이 배포되도록 보장합니다. 회고 미팅(retrospective meeting)은 가치 있는 교훈을 식별하고, 팀이 파악한 테스트 개선사항을 테스트 조직에 전달하는 데 활용될 수 있습니다. 이렇게 얻은 교훈과 테스트 개선사항은 이후 테스트 프로세스 개선 제안으로 제출되어 그에 따라 관리될 수 있습니다.

전반적으로 이 세부 목표와 그 세부 프랙티스들은 변경 없이 AI 시스템 테스트에도 그대로 적용됩니다. 유일한 차이는 프로세스가 테스터뿐만 아니라 데이터 과학자와 ML 엔지니어에게도 배포되어야 한다는 점입니다. AI 시스템의 동작을 실질적으로 좌우하는 것이 데이터이기 때문에, 이들 역시 테스트에 관여하게 됩니다. 그 결과, 이들은 훈련 데이터의 품질, 편향, 누출(leakage),

대표성을 검증하는 데 핵심적인 역할을 담당하며, 정확도, 정밀도와 재현율, 견고성, 드리프트와 같은 메트릭을 활용해 초기 모델을 테스트하는 데에도 핵심적인 역할을 합니다.

## 3.2 프로세스 영역 3.2 테스트 훈련 프로그램 (Test Training Program)

*테스트 훈련 프로그램(Test Training Program) 프로세스 영역의 목적은, 테스트 과업과 역할을 효과적이고 효율적으로 수행할 수 있도록 인력의 지식과 스킬 개발을 촉진하는 훈련 프로그램을 개발하는 것입니다.*

테스트 훈련 프로그램의 주된 목적은 테스터와 그 밖의 팀원들에게 필요한 (테스트) 훈련을 제공하는 것입니다. 양질의 훈련 프로그램은 테스트에 관여하는 인력이 테스트 지식과 스킬을 지속적으로 향상시키고, 필요한 도메인 지식과 테스트 관련 기타 지식 및 스킬을 습득하도록 보장합니다. 품질과 테스트는 통상 팀 전체의 책임이므로, AI 테스트 관련 훈련 역시 테스트 전문가에게만 제공되는 것이 아니라 팀 전체에 제공되어야 합니다.

### 3.2.1 SG1 조직적 훈련 역량 수립

이 프로세스는 조직의 전략적 테스트 훈련 니즈를 식별하는 것에서 시작됩니다. 테스트 설계 기법이나 커버리지 측정과 같은 테스터를 위한 전통적인 훈련 니즈에 더해, 이제는 AI 시스템 테스트의 특수성(예: ISTQB 자격 인증 체계[ISTQB]를 통해)에 관한 지식과 스킬을 습득할 필요성도 존재합니다. 그 예로는 AI 및 머신러닝 개념에 대한 지식, 훈련·검증·테스트 데이터셋에 대한 이해, 과적합·과소적합·모델 드리프트와 같은 이슈에 대한 이해 등이 있습니다. 테스터는 또한 데이터 품질 및 데이터 테스트 관행에 관한 지식과 스킬, 그리고 데이터 파이프라인에 대한 이해도 필요로 합니다. 앞서 언급했듯, 테스트 훈련 니즈의 식별은 테스터에게만 국한되지 않으며, 다른 팀원들을 위한 AI 테스트 관련 훈련 니즈 또한 함께 구체화하는 것이 매우 중요합니다.

테스트 훈련 니즈는 (경량화된) 훈련 계획으로 구체화됩니다. 이 훈련 계획은 검토를 거쳐야 하며, 이에 대한 약속(commitment)도 수립되어야 합니다. 일부 스킬은 OJT(on-the-job training), 점심시간

훈련 세션, 코칭과 멘토링과 같은 비공식적 수단을 통해 효과적이고 효율적으로 전달될 수 있는 반면, 다른 스킬은 공식적인 훈련을 필요로 한다는 점에 유의해야 합니다. AI 테스트 훈련 니즈를 충족하기 위한 적절한 접근 방식은 지식·스킬 영역별로 결정되어야 합니다.

### 3.2.2 SG2 필요한 테스트 훈련 수행

조직의 훈련 계획을 바탕으로, 자신의 테스트 역할을 효과적으로 수행하기 위해 AI 테스트 훈련이 필요한 테스트 전문가와 그 밖의 팀원들이 식별됩니다. 이후 필요한 자원을 포함하여 훈련 일정이 수립되고 훈련이 실시됩니다. 앞서 언급한 바와 같이 비공식적인 훈련 수단도 함께 활용될 수 있습니다.

비공식적이든 공식적이든, 실시된 테스트 훈련에 대한 기록이 작성되며, AI 테스트 훈련의 효과성이 평가됩니다. 참석한 (테스트) 훈련은 회고 미팅의 일환으로 평가될 수도 있는데, 예를 들어 참석자들의 지식과 스킬이 AI 시스템 테스트와 관련된 테스트 과업을 수행하기에 이제 더 적합해졌는지에 대한 논의가 이루어질 수 있습니다.

## 3.3 프로세스 영역 3.3 테스트 라이프사이클 및 통합 (Test Life Cycle and Integration)

*테스트 라이프사이클 및 통합(Test Lifecycle and Integration)의 목적은, 사용 가능한 조직 테스트 프로세스 자산(예: 표준 테스트 라이프사이클)과 업무 환경 표준의 집합을 수립하고 유지하며, 테스트 라이프사이클을 개발 라이프사이클과 통합·동기화하는 것입니다. 이러한 통합된 라이프사이클은 프로젝트 내에서 테스트가 조기에 관여하도록 보장합니다. 테스트 라이프사이클 및 통합의 목적은 또한, 식별된 리스크와 정의된 테스트 전략을 바탕으로 여러 테스트 레벨에 걸친 일관된 테스트 접근법을 정의하고, 정의된 테스트 라이프사이클을 바탕으로 전체 테스트 계획을 제공하는 것입니다.*

### 3.3.1 SG1 조직 테스트 프로세스 자산 수립

테스트 조직의 중요한 책임 중 하나는 조직의 테스트 정책 및 목표에 부합하는 표준 테스트 프로세스를 정의, 문서화, 유지하는 것입니다. 이 표준 테스트 프로세스에는 무엇보다도 수행할 다양한 테스트 유형과 테스트 레벨에 대한 설명이 포함됩니다. AI 시스템 테스트는 전통적인 시스템 테스트와는 매우 다른 프로세스이므로, 조직 내에 AI 시스템 테스트를 위한 전용 표준 테스트 프로세스를 별도로 개발할 필요가 있습니다. 조직 내에 AI 시스템 테스트를 위한 표준 테스트 프로세스를 갖추는 것은 또한 각 프로젝트가 매번 '바퀴를 재발명(re-invent the wheel)'하는 것을 방지해 줍니다.

AI 시스템 테스트는 테스트 대상이 되는 AI 시스템의 유형에 따라 프로젝트마다 달라질 수 있으므로, '무엇을 해야 하는가'와 '어떻게 하는가'에 관한 정보를 분리할 것을 강력히 권장합니다. 테스트 대상 AI 시스템의 유형과 무관하게 모든 프로젝트에 이를 의무화하기로 결정한 경우가 아니라면, 테스트 프로세스 자체에는 '어떻게(how-to)'에 관한 정보나 도구 관련 정보를 포함시키지 않아야 합니다. '어떻게'에 관한 정보는 작업 지침(work instruction) 등 별도의 가이드라인에 담습니다. '어떻게'에 관한 정보의 경우, 기존에 존재하는 서적이나 논문을 그대로 참조하도록 결정할 수도 있다는 점에 유의합니다. 특히 빠르게 진화하는 AI 영역의 특성상, 이는 새로운 가이드라인을 수시로 다시 개발해야 하는 부담을 방지하는 데 매우 유익할 수 있습니다.

프로세스 자산의 선택은 각 조직의 재량에 달려 있지만, 앞선 문단의 취지에 따라 프로세스 자산은 단 두 개의 레벨로 구성할 것을 권장합니다. 이는 프로세스 수행 시 '무엇을 해야 하는가'를 담은 정책 선언문과 관련 프로세스 설명을 하나로 통합함으로써 이루어집니다. 두 번째 레벨에는 프로세스를 수행하고 테일러링하기 위한 '어떻게'에 관한 가이드라인이 담깁니다. 이 레벨은 프로세스 테일러링을 돕는 보조 수단으로 볼 수 있으며, 정의된 프레임워크 내에서 정확히 어떻게 작업할지를 프로젝트나 팀이 자율적으로 결정할 수 있도록 하고, 통상 이를 뒷받침하는 템플릿도 포함합니다. AI 시스템을 테스트하는 대부분의 조직에서는 단계별 절차가 도구 가이드와 훈련/멘토링으로 대체됩니다.

조직의 표준 AI 시스템 테스트 프로세스 집합은 프로젝트별로 테일러링되어 각자의 구체적인 정의된 프로세스로 만들어질 수 있습니다. 'SP 1.3 테일러링 기준 및 가이드라인 수립'의 맥락에서, 오늘날 대부분의 조직은 이른바 '테일러 다운(tailor down)' 접근법을 사용합니다. 이 접근법에서는 무언가가 왜 필요하지 않은지를 설명하는 데 노력을 들여야 합니다. 그 결과 비교적 단순한 AI 프로젝트에서도 불균형적으로 많은 노력이 소요되어, 이 접근법의 실효성이 떨어지게 됩니다. 테일러 다운을 한다면 어디까지 줄일 수 있습니까? 정의된 최소치가 존재합니까? 반면, 모든 AI 프로젝트가 반드시 수행해야 하는 최소 활동 집합에서 출발한다면, '반드시 해야 할 것(must do)'이 테일러링 과정에서 빠질 위험을 없앨 수 있습니다. '테일러 업(tailoring up)' 접근법은 AI 프로젝트에 필요한 유연성에 훨씬 더 잘 부합하며, TMMi 에도 완전히 부합합니다. 간단한 기준의 집합이 있으면 테일러 업 시 올바른 질문을 던지는 데 도움이 될 수 있습니다. 도움이 될 기준이 없으면, 일부는 지나치게 많이 테일러 업하는 경향을 보입니다.

표준 AI 테스트 프로세스의 배포를 지원하기 위해, 흔히 위키(wiki) 형태로 전용 테스트 프로세스 자산 라이브러리가 구축되고 유지되며, 여기서 테스터는 AI 테스트 활동에 도움이 되는 템플릿, 모범 사례, 체크리스트를 찾을 수 있습니다. 또한 산정치, 커버리지, 품질 측정치 등 AI 테스트 데이터를 수집하여 팀들이 활용할 수 있도록 하는 전용 테스트 프로세스 데이터베이스도 마련합니다. 팀 간 학습과 경험 공유를 위해 수집·제공되는 모든 자료가 팀들에게 실질적인 부가가치를 지니도록 해야 합니다.

### 3.3.2 SG2 테스트 라이프사이클 모델과 개발 모델의 통합

AI 테스트 라이프사이클 모델은 다양한 테스트 레벨과 테스트 유형에 대한 주요 활동과 산출물을 정의합니다. AI 테스트 라이프사이클 모델은 AI 개발 라이프사이클 모델과 정렬되어, 단계 구분, 마일스톤, 산출물, 활동의 측면에서 테스트 활동을 통합합니다. 라이프사이클 통합은 AI 프로젝트에서 테스트가 조기에 관여하도록 보장하는 방식으로 이루어집니다. AI 에서는 개발과 테스트가 라이프사이클 내에서 통합되어야 하며, 테스트는 가능한 한 일찍 시작되어야 합니다. 테스터는 팀의

일원이므로, 계획 수립, 리스크 분석, 인수 기준 정의, 데이터 준비 및 데이터 테스트와 같은 선행 활동에도 참여합니다.

테스팅이 어떻게 조기에 관여하고 수행될 수 있는지를 정의함으로써 공통의 이해를 형성하는 것이 중요합니다. 이것이 적절히 이루어진다면 AI 개발과 테스트는 완전히 통합되며, 이를 어떻게 수행하는지에 대한 공통의 이해가 존재하게 됩니다. 이 세부 목표는 매우 중요하며, AI 개발과 테스트 활동이 정렬되도록 보장해야 합니다.

### 3.3.3 SG3 마스터 테스트 계획 수립

TMMi 레벨 3에서 테스트는 마스터 테스트 계획 수립(master test planning)을 다루는데, 이는 모든 테스트 레벨과 테스트 유형에 걸쳐 테스트 과업, 책임, 테스트 접근법을 조율하는 것을 의미합니다. 이를 통해 여러 테스트 레벨 사이에서 불필요한 중복이나 테스트 누락을 방지할 수 있으며, 전체 테스트 프로세스의 효율성과 품질을 크게 높일 수 있습니다. 마스터 테스트 계획은 특정 프로젝트에 대한 테스트 전략의 적용 방식을 기술합니다.

보다 복잡하고, 규제 대상이며, 리스크와 관련된 AI 시스템을 테스트할 때는 마스터 테스트 계획을 수립하는 것이 유익합니다. 마스터 테스트 계획은 여러 모델, 데이터 파이프라인, 환경, 그리고 컴플라이언스 및 리스크 영역에 걸친 전체 테스트 전략을 다루며, 그 아래에 하위 테스트 계획들을 둘 수도 있습니다. 이는 진화하는 모델, 데이터에 따라 달라지는 동작 변화, 영역을 넘나드는 리스크(편향, 드리프트, 컴플라이언스), 그리고 다수의 테스트 활동에 대해 중앙집중적인 거버넌스를 제공한다는 점에서 AI 시스템 테스트에 유리하며, 이는 전통적인 단일 레벨 테스트 계획으로는 효과적으로 관리할 수 없는 것입니다.

거버넌스 관점에서, AI 시스템 테스트를 위한 마스터 테스트 계획은 통상 데이터(파이프라인) 테스트, 모델 검증, 회귀 테스트, 편향 테스트, 통합 테스트, 시스템 테스트, 모니터링, 모델 드리프트를 일관되게 다루는 것을 목표로 합니다. 마스터 테스트 계획을 단순히 테스트 실행을 계획하기 위한 것이 아니라, 테스트 거버넌스 및 조율 도구로 바라보아야 합니다. 따라서 마스터 테스트 계획을

수립하는 것(즉 이 TMMi 세부 목표)은 AI 시스템 테스트에도 충분한 근거를 가지고 그대로 적용됩니다.

### 3.4 프로세스 영역 3.4 비기능 테스트 (Non-Functional Testing)

*비기능 테스트(Non-Functional Testing) 프로세스 영역의 목적은, 테스트 계획 수립, 테스트 설계 및 실행 과정에 비기능 테스트를 포함하도록 테스트 프로세스 역량을 향상시키는 것입니다.*

비기능 테스트 프로세스 영역은 비기능 제품 리스크 평가를 수행하고, 식별된 리스크 집합을 바탕으로 테스트 접근법을 정의하는 것을 포함합니다. 또한 비기능 테스트 조건과 테스트 케이스를 도출하고 선정하는 데 필요한 테스트 설계 및 테스트 구현 활동도 다룹니다. 비기능 테스트를 지원하기 위한 특정 테스트 데이터의 식별과 생성도 포함합니다. 비기능 테스트에 대한 테스트 실행, 테스트 로깅, 테스트 인시던트 보고 역시 이 프로세스 영역의 일부입니다.

#### 3.4.1 AI 기반 시스템의 품질 특성

제품 품질은 이해관계자의 니즈를 충족시키는 것을 중심으로 합니다. 이러한 니즈는 기능(제품이 '무엇을' 하는가) 요구사항과 비기능(제품이 '어떻게' 하는가) 요구사항으로 전환되어야 합니다. 국제표준화기구(ISO)는 소프트웨어 제품이나 시스템의 품질을 정의하는 데 사용되는 비기능 특성의 집합을 정의한 바 있다[ISO 25010]. 그러나 AI 기반 시스템의 고유한 측면을 다루기 위해서는 전통적인 소프트웨어 품질 모델 및 품질 특성에 대한 확장이 필요합니다.

ISO는 또한 AI 기반 시스템의 품질 특성을 다루는 전용 표준[ISO 25059]도 발간하였습니다. ISO/IEC 25010을 기반으로 하는 이 표준에서는 AI 시스템에 특화된 다수의 수정된 품질 특성과 새로운 품질 특성이 식별·정의되었습니다. 일부 하위 특성은 ISO/IEC 25010 모델과 비교했을 때 그 의미나 맥락이 다릅니다. ISO/IEC 25059에 포함된 수정 및 신규 품질 하위 특성은 다음과 같습니다:

- **기능적 정확성(Functional correctness) (수정됨):** 확률적 특성을 지닌 AI 시스템에서는 완벽한 정확성을 달성하는 것이 일반적으로 불가능합니다. 대신, 적절한 평가 메트릭을

사용하여 시스템의 정확도, 오류 특성, 정밀도를 정량화하고, 이것이 의도된 용도의 요구사항을 충족하는지 확인함으로써 품질을 입증합니다. (기능적 정확성은 기능 적합성(Functional suitability)의 하위 특성입니다)

- **기능적 적응성(Functional adaptability) (신규):** AI 시스템의 기능적 적응성이란, 시스템이 배포되어 있는 변화하는 동적 환경에 적응하는 능력을 말합니다. 이 특성을 평가하기 위해서는 여러 환경 조건과 장기간에 걸친 시스템 동작을 평가할 수 있는 테스트 접근법이 필요합니다. AI 시스템은 새로운 훈련 데이터, 운영 데이터, 그리고 시스템이 과거에 취한 행동의 결과로부터 학습할 수 있습니다. (기능적 적응성은 기능 적합성의 하위 특성입니다)
- **환경적 지속가능성(Environmental sustainability) (신규):** 이 하위 특성은 조직이 AI 시스템과 관련하여 각기 고유한 지속가능성 측면을 지닌 세 가지 주요 영역, 즉 경제적 지속가능성, 사회적 지속가능성(예: 신뢰성(trustworthiness)), 환경적 지속가능성을 고려할 것을 요구합니다. (환경적 지속가능성은 성능 효율성(Performance efficiency)의 하위 특성입니다)
- **사용자 제어 가능성(User controllability) (신규):** 사용자 제어 가능성은 통제자가 시의적절하게 AI 시스템의 작동에 개입할 수 있도록 하는 AI 시스템의 속성입니다. 예상치 못한 동작을 완전히 방지할 수 없어 부정적인 결과를 다루기 위한 개입이 필요한 경우, 이러한 통제 개입이 중요합니다. (사용자 제어 가능성은 상호작용 능력(Interaction capability)의 하위 특성입니다)
- **투명성(Transparency) (신규):** 투명성은 AI 시스템에 관한 적절한 정보가 이해관계자에게 전달되는 정도와 관련됩니다. AI 시스템의 투명성은 잠재적 사용자가 자신의 요구사항에 맞는 시스템을 선택하는 데 도움을 줄 수 있으며, AI 시스템의 적용 가능성과 한계에 대한 이해관계자의 지식을 높이고, AI 시스템의 설명 가능성을 뒷받침합니다. (투명성은 상호작용 능력의 하위 특성입니다)

- **견고성(Robustness) (신규):** 이는 처음 접하는 데이터, 편향되거나 유효하지 않은 데이터 입력, 외부 간섭, 환경 조건 등 어떠한 상황에서도 시스템이 기능적 정확성의 수준을 유지하는 능력을 설명하는 데 사용되며, 일반화, 회복력(resilience) 또는 신뢰성, 그리고 개발자가 의도한 대로 시스템이 올바르게 동작하는 것과 관련된 속성들을 포괄합니다. (견고성은 신뢰성(Reliability)의 하위 특성입니다)
- **개입 가능성(Intervenability) (신규):** 개입 가능성의 정도는 AI 시스템이 사용될 수 있는 시나리오에 따라 결정될 수 있습니다. 개입 가능성의 핵심은 상태를 관찰하고, 안전하지 않은 상태에서 안전한 상태로 전환할 수 있도록 하는 것입니다. 운용성(operability)과 비교할 때 개입 가능성은 품질 관점에서 더 근본적인 특성이며, AI 시스템이 해악이나 위험을 초래하는 것을 방지하는 데 목적이 있습니다. (개입 가능성은 보안성(Security)의 하위 특성입니다)

위에서 열거한 것들은 모두 제품 품질 특성인데, 이에 더해 ISO 25059 는 새로운 사용 품질(quality in use) 특성도 식별하였습니다: 사회적·윤리적 리스크 완화(Societal and ethical risk mitigation)('위험 회피성(Freedom from risk)'의 새로운 하위 특성). 이는 사회적·윤리적 리스크를 완화하기 위한 여러 영역, 즉 책임성(accountability), 공정성과 비차별, 전문가적 책임, 인간 가치의 증진, 개인정보 보호, 안전 및 보안, 기술에 대한 인간의 통제, 지역사회 참여와 발전, 인간 중심 설계, 법치주의 존중, 국제 행동 규범 존중, 환경적 지속가능성, 노동 관행 등을 고려합니다.

### 3.4.2 SG1 비기능 제품 리스크 평가 수행

테스트 계획 프로세스 영역의 세부 목표 'SG1 리스크 평가 수행'에서 식별·기술된 브레인스토밍, 전문가 인터뷰, 체크리스트와 같은 제품 리스크 기법들은, 이제 비기능적 측면과 리스크까지 명시적으로 포함하도록 확장됩니다. 프로젝트 초기에, 이제는 비기능 테스트까지 포함하는 제품 리스크 평가를, 정의된 제품 비전, 모델 목표, 시스템 요구사항, 인수 기준을 바탕으로 수행할 수 있습니다. AI 에 대한 제품 리스크 평가는 개발과 운영 전반에 걸쳐 계속 진화하는 살아있는

산출물(living artifact)로 다루는 것이 가장 바람직하며, 이는 릴리스 이전의 비기능 테스트와 운영 중의 지속적인 모니터링 모두에 정보를 제공합니다.

AI 시스템의 비기능 테스트를 위한 제품 리스크 평가는, 확률적 동작, 모델 드리프트, 데이터 의존성, 불확실성과 같은 AI 모델 고유의 속성을 고려함으로써 전통적인 소프트웨어 리스크 평가를 확장합니다. 그 목표는 (예: ISO 25059 에 따른) 어떤 비기능 특성이 가장 큰 비즈니스적 또는 운영상의 리스크를 제기하는지를 식별하여, 테스트 노력의 우선순위를 정할 수 있도록 하는 것입니다. 일부 비기능 영역에서는 전문가의 지원이 필요할 수 있습니다.

제품 리스크 평가 과정에서 고려해야 할, AI 시스템의 비기능 제품 리스크 예시:

- 성능 효율성(Performance efficiency): AI 시스템이 예상 부하 및 피크 부하 상황에서도 응답 시간과 처리량(throughput) 요구사항을 일관되게 충족합니까?
- 견고성(Robustness): 예상치 못한, 불완전한, 악의적인, 또는 학습 분포를 벗어난(out-of-distribution) 입력이 주어졌을 때 AI 가 예측 가능하게 동작합니까?
- 보안성(Security): AI 가 프롬프트 인젝션, 적대적 공격(adversarial attack), 무단 접근, 모델 오남용, 데이터 유출로부터 보호되고 있습니까?
- 호환성(Compatibility): AI 시스템이 장애를 유발하지 않으면서 다른 시스템, API, 모델, 서비스와 올바르게 상호운용될 수 있습니까?
- 상호작용 능력(Interaction Capability): 사용자가 AI 의 출력, 신뢰도 수준, 설명을 이해하고 통제하며 적절히 해석할 수 있습니까?
- 안전성(Safety): 장애나 예상치 못한 출력이 발생했을 때, AI 시스템이 사람·조직·환경에 대한 용인할 수 없는 피해를 방지하거나 완화할 수 있습니까?
- 이식성(Portability): 용인할 수 없는 수준의 성능 저하 없이, AI 솔루션이 서로 다른 환경, 클라우드 플랫폼, 하드웨어 가속기 전반에 걸쳐 배포·이전·실행될 수 있습니까?
- 투명성(Transparency): AI 라이프사이클 전반에 걸쳐, 의사결정을 설명하고 모니터링·감사할 수 있으며, 공정성·견고성·개인정보 보호·책임성 측면에서 평가할 수 있습니까?
- 공정성 및 책임 있는 AI(Fairness & Responsible AI): 관련 사용자 그룹 전반에 걸쳐 모델의 편향, 일관성, 공평한 성능이 평가되었습니까?

- 관찰 가능성 및 모니터링(Observability & Monitoring): 장애, 성능 저하, halusinasi(hallucination), 드리프트, 비정상적인 동작을 탐지·기록·조사할 수 있습니까?

### 3.4.3 SG2 비기능 테스트 접근법 수립

식별되고 우선순위가 매겨진 비기능 제품 리스크를 완화하기 위해, 관련 비기능 품질 특성에 대한 테스트 접근법이 정의됩니다. AI 시스템 개발 과정에서 새로운 비기능 제품 리스크가 드러나 추가 테스트가 필요해질 수 있습니다. 추가 테스트가 필요한 새로운 비기능 제품 리스크와 같은 이슈는 통상 일일 스탠드업 미팅에서 논의됩니다. 비기능 리스크를 완화하기 위해 정의되는 비기능 테스트 접근법은, 통상 비기능 리스크의 수준과 유형에 따라 적절한 비기능 테스트 방법과 테스트 기법을 식별하는 것을 다룹니다. 또한 통상적으로 지원 도구의 사용과 비기능 테스트를 위한 테스트 자동화 접근법도 다룹니다. 비기능 품질 특성, 특히 모델 품질 및 드리프트와 관련된 특성은, 관찰 가능성(observability)에 기반한 텔레메트리(telemetry)를 활용하여 운영 환경 배포 이후에도 테스트되고 측정된다는 점에 유의합니다. 이는 테스트 접근법의 일부로 문서화되어야 합니다.

비기능 AI 시스템 테스트 접근법은 전체 테스트 접근법의 일부이며, 팀/프로젝트 위키에 유지·게시됩니다. 비기능 종료 기준은 이른바 '완료의 정의(Definition of Done, DoD)'의 일부입니다. DoD 에 비기능 테스트와 관련된 구체적인 기준이 포함되는 것이 중요합니다. 예를 들면, 보안 테스트에서 기밀 지침의 노출로 이어지는 프롬프트 인젝션이 성공하지 않음을 입증하는 것, 시스템이 (신뢰성 측면에서) 잘못된 형식의 입력이나 적대적 입력을 충돌 없이 안전하게 처리하는 것, 또는 (공정성 측면에서) 평가 대상 인구통계학적 그룹 간 분류 정확도 차이가 5%를 넘지 않는 것 등입니다. AI 시스템 개발은 합의된 비기능 요구사항 집합의 구현으로 귀결되어야 하며, DoD 에 정의된 비기능 (테스트) 종료 기준을 충족해야 합니다. 그러나 테스트는 AI 시스템이 릴리스된다고 해서 끝나지 않습니다. AI 시스템의 맥락에서는 시프트 라이트(shift-right) 접근법, 즉 정의된 기준에 대비하여 지속적인 모니터링을 통해 운영 환경에서 제품 품질 특성을 측정하는 것에 큰 비중을 둡니다. 시프트 라이트 테스트는 테스트 활동을 운영 환경에 더 가깝게 이동시켜, 실제 환경에서의 성능 평가와 지속적인 품질 모니터링을 가능하게 하는 것을 포함합니다.

### 3.4.4 SG3 비기능 테스트 분석 및 설계 수행

이 세부 목표는 테스트 설계 및 실행 프로세스 영역의 세부 목표 'SG1 테스트 설계 기법을 활용한 테스트 분석 및 설계 수행'과 대체로 동일한 프랙티스를 따르되, 이제는 비기능적 관점에서 수행된다는 점이 다릅니다. AI 기반 시스템의 경우, 비기능 테스트 분석 및 설계는 비기능 제품 리스크 평가와 수립된 비기능 테스트 접근법의 일환으로 선정된 품질 특성에 초점을 맞춥니다. AI 기반 시스템의 유형에 따라, 이러한 특성에는 성능 효율성, 견고성, 투명성, 설명 가능성, 공정성, 적응성, 자율성, 개인정보 보호, 보안성, 회복력, 그 밖의 AI 특유의 품질 특성이 포함될 수 있습니다.

테스트 분석 및 설계 활동은 관련 테스트 조건을 식별하고, 필요한 테스트 데이터를 결정하며, 선정된 비기능 품질 특성을 평가하기 위한 적절한 인수 기준과 메트릭을 정의해야 합니다. AI 기반 시스템은 데이터에 의해 좌우되며 확률적이거나 비결정론적인 동작을 보일 수 있으므로, 대표성을 지닌 테스트 데이터셋과 현실적인 운영 시나리오를 설계하는 것이 이 세부 목표에서 중요한 측면인 경우가 많습니다. 아울러 AI 모델의 동작에 부정적인 영향을 미칠 수 있는 엣지 케이스와 상황도 테스트 설계 과정에서 고려되어야 합니다.

비기능 테스트 분석 및 설계에 사용되는 기법은 선정된 비기능 품질 특성과 관련 제품 리스크에 따라 달라집니다. 그 예로는 견고성 테스트, 적대적 테스트, 공정성 및 편향 테스트, 설명 가능성 테스트, 회복력 테스트를 위한 기법들이 있습니다. AI 기반 시스템에 적용 가능한 기법, 메트릭, 테스트 설계 접근법에 관한 지침은 ISTQB 공인 테스터 AI 테스트 실러버스[ISTQB]에서 찾아볼 수 있습니다. 운영 환경에서 비기능 특성을 측정하는 데 도움이 되도록, 관찰 가능성 및 텔레메트리 요구사항과 스토리를 식별하는 것 역시 비기능 분석 및 설계의 산출물이 될 수 있습니다.

비기능 테스트의 실행을 뒷받침하는 데 필요한 특정 테스트 데이터가 식별됩니다. AI 기반 시스템의 경우, 필요한 테스트 데이터는 대부분 선정된 비기능 품질 특성에 따라 달라지며, 대표 데이터, 엣지 케이스 데이터, 합성 데이터, 또는 적대적 데이터가 포함될 수 있습니다. 일부 테스트 데이터는 테스트 실행 중에 준비될 수 있지만, 자동화되고 반복 가능한 비기능 테스트는 통상 필요한 데이터셋과 인수

기준이 사전에 명시되어야 합니다. 일부 비기능 특성은 운영 환경과 유사하거나 실제 운영 데이터를 사용하여 평가되므로, 테스트 과정에서 운영 데이터가 사용될 수도 있습니다. 이러한 경우에는 개인정보 보호, 보안, 그리고 GDPR 과 같은 해당 규제를 고려해야 합니다.

이 목표의 산출물에는 통상 비기능 테스트 조건에 대한 명세, 필요한 테스트 데이터 및 환경, 그리고 관련 인수 기준과 품질 메트릭이 포함됩니다. 이러한 산출물은 이후의 비기능 테스트 구현 및 실행 활동의 기반을 제공합니다.

### 3.4.5 SG4 비기능 테스트 구현 수행

테스트 구현이란 이전 테스트 단계에서 식별되고 설계된 비기능 테스트를 구현하여, 테스트 실행을 시작하는 데 필요한 모든 것을 갖추는 것을 말합니다. AI 기반 시스템의 경우, 여기에는 선정된 AI 특유의 품질 특성을 평가하는 데 필요한 테스트 데이터, 테스트 환경, 도구, 테스트 절차를 준비하는 것도 포함됩니다.

비기능 테스트 구현은 테스트 설계 및 실행 프로세스 영역의 세부 목표 'SG2 테스트 구현 수행'에서 이미 기술된 프랙티스 상당 부분을 따릅니다. 구체적으로 무엇을 해야 하는지, 그리고 비기능 테스트 구현을 어떻게 수행하는지는 정의된 테스트 접근법, 사용 중인 기법, 테스트해야 할 비기능 특성과 그 수준에 크게 좌우됩니다. 예를 들어, 준비가 덜 필요한 탐색적 테스트는 사용성 테스트에는 적합할 수 있지만, 광범위한 신뢰성 및 성능 테스트에는 대체로 덜 적합합니다.

AI 기반 시스템에 대한 비기능 테스트 구현은 전통적인 시스템에 비해 데이터 집약도와 기술적 난이도가 더 높은 경우가 많습니다. 테스트 대상 비기능 특성에 따라, 대표성을 지니거나 때로는 특화된 데이터셋을 확보·생성·준비해야 합니다. 예를 들어 견고성, 공정성, 투명성, 설명 가능성과 같은 특성을 테스트하려면 현실적인 상황, 경계 상황, 예외적인 상황을 포괄하는 전용 테스트 데이터셋이 필요할 수 있습니다. 아울러 테스트 환경은 운영 조건을 시뮬레이션하거나, 대용량의 테스트 데이터와 반복적인 테스트 실행을 지원해야 할 수도 있습니다.

전형적인 구현 활동에는 테스트 데이터셋의 준비 및 전처리, 테스트 환경과 시뮬레이션 플랫폼의 구성, 자동화된 테스트 스크립트의 구현, 테스트 결과를 수집·분석하는 메커니즘의 구축 등이 포함됩니다. 일부 상황에서는 드물거나 확보하기 어려운 시나리오에 대한 커버리지를 높이기 위해 합성 또는 증강된 테스트 데이터가 사용될 수 있습니다. 구현 활동은 또한 선정된 품질 특성을 반복 가능하고 관리 가능한 방식으로 측정할 수 있도록 보장해야 합니다.

전통적인 시스템과 마찬가지로, 구현된 모든 테스트 산출물은 식별된 비기능 제품 리스크, 테스트 조건, 인수 기준으로 추적 가능해야 합니다. 구현 활동은 전통적인 시스템에서 사용되는 것과 크게 다를 수 있지만, TMMi 목표의 취지는 변함이 없습니다: 비기능 테스트가 적절히 준비되어 실행할 수 있는 상태가 되도록 보장하는 것입니다.

### 3.4.6 SG5 비기능 테스트 실행 수행

이 프로세스 영역의 앞선 목표와 마찬가지로, 본 문서 앞부분에서 다룬 테스트 설계 및 실행 프로세스 영역의 관련 세부 목표 'SG3 테스트 실행 수행'을 주로 참조합니다. 이 세부 목표의 목적은 구현된 비기능 테스트를 실행하고, AI 기반 시스템이 선정된 비기능 품질 특성에 대해 정의된 인수 기준을 충족하는지 평가하는 것입니다. AI 시스템 테스트에서 비기능 테스트를 실행하고, 테스트 인시던트를 보고하며, 테스트 로그를 작성하는 프랙티스는 기능 테스트의 경우와 기본적으로 동일합니다. 비기능 특성 또한 배포 이후에도 테스트/측정된다는 점에 유의합니다.

비기능 테스트 실행은 계획 수립 단계에서 정의된 우선순위에 따라 이루어집니다. 일부 테스트는 문서화된 테스트 절차를 사용하여 실행될 수 있지만, 통상 많은 비기능 테스트는 탐색적 테스트와 세션 기반 테스트를 프레임워크로 삼아 실행됩니다. AI 시스템과 운영 중 지속적인 모니터링의 특성상, 자동화된 회귀 테스트와 지원 도구를 활용하여 비기능 테스트에 대해서도 회귀 테스트를 체계적으로 조직하고 구조화해야 할 필요성이 큼니다.

AI 기반 시스템에 대한 비기능 테스트 실행은, 확률적이며 때로는 비결정론적인 동작 특성으로 인해 전통적인 시스템과 다른 경우가 많습니다. AI 시스템의 유형과 평가 대상 비기능 품질 특성에 따라,

테스트 실행에는 대표성을 지니거나 이전에 접해보지 못한 데이터셋에 대해 모델을 실행하고, 테스트 실행을 반복하며, 얻어진 결과를 통계적으로 분석하는 작업이 포함될 수 있습니다. 비기능 요구사항의 달성 여부를 판단하려면 통상 정량적 메트릭의 사용이 요구됩니다. 테스트 실행 과정에서 평가될 수 있는 비기능 특성의 전형적인 예로는 견고성, 투명성, 설명 가능성, 공정성, 적응성, 회복력, 운영 성능이 있습니다. 선정되는 메트릭과 인수 기준은 AI 기반 시스템의 성격과, 프로세스 앞 단계에서 식별된 관련 제품 리스크 모두에 따라 달라집니다.

전통적인 시스템과 달리, AI 기반 시스템의 일부 비기능 특성은 릴리스 이전 테스트 단계에서 항상 완전히 평가될 수 있는 것은 아닙니다. 편향, 드리프트, 모델 성능 저하와 같은 특성은 시스템이 시간이 지나면서 실제 운영 데이터에 노출되어야만 비로소 드러날 수 있습니다. 따라서 비기능 테스트 실행에는 통상, 배포 이후 AI 기반 시스템의 운영상 동작에 대한 지속적인 평가와 모니터링을 뒷받침하는 활동도 포함됩니다.

실행 프랙티스는 전통적인 시스템에서 사용되는 것과 다르지만, TMMi 목표의 취지는 변함이 없습니다. 그 목표는 AI 기반 시스템이 정의된 비기능 품질 요구사항을 충족하며, 의도된 운영 맥락에서 충분히 적절하게 동작한다는 객관적인 증거를 제공하는 것입니다.

### 3.5 프로세스 영역 3.5 동료 검토 (Peer Reviews)

*동료 검토(Peer Review) 프로세스 영역의 목적은, 작업 산출물이 명시된 요구사항을 충족하는지 검증하고, 선정된 작업 산출물로부터 결함을 조기에 효율적으로 제거하는 것입니다. 중요한 부수적 효과로는, 작업 산출물과 예방 가능한 결함에 대한 더 나은 이해를 얻는 것을 들 수 있습니다.*

#### 3.5.1 SG1 동료 검토 접근법 수립

동료 검토는 사용되는 라이프사이클 모델이나 개발 중인 제품의 유형과 무관하게, 모든 개발 과정에서 중요한 결함 탐지 기법입니다. 이러한 이유로 동료 검토는 AI 시스템의 개발과 테스트에도 유익한 기법입니다. AI 기반 시스템을 위한 동료 검토 접근법은 경량화되고 리스크 기반이어야 하며,

반복적인(iterative) 개발 주기에 AI 시스템을 통합하는 동시에, 비결정성(non-determinism), 데이터 의존성, 확률적 결과와 같은 AI 고유의 특성을 명시적으로 다루어야 합니다. 동료 검토는 ML 워크플로우에 내재된 지속적인 활동으로 계획되어야 하며, 전통적인 테스트 산출물과 AI 특유의 산출물 모두에 대한 조기 결함 탐지와 품질 관리에 중점을 두어야 합니다.

동료 검토 접근법을 정의하는 작업은 리뷰 대상이 될 가장 핵심적인 작업 산출물을 식별하고 선정하는 것에서 시작됩니다. AI 시스템의 개발 및 테스트 과정에서 리뷰 대상이 되는 작업 산출물에는 통상 AI/ML 요구사항 및 인수 기준(특히 통계적 또는 임계값 형태로 정의된 것), 데이터 관련 산출물(훈련·검증·테스트 데이터셋, 데이터 준비 절차, 레이블링 가이드라인), 모델 문서(가정, 한계, 성능 메트릭 포함), 테스트 전략, 테스트 접근법, 테스트 케이스(편향·견고성·적대적 조건에 대한 것 포함)가 포함됩니다.

AI 시스템이 지닌 탐색적이고 계속 진화하는 특성을 고려할 때, 리뷰 접근법에는 다음과 같은 요소가 포함되는 것이 바람직합니다:

- 반복적인 변경사항에 대한 경량화된 동료 검토와 데스크 체크(desk check)
- 데이터 파이프라인이나 모델 동작과 같은 복잡한 산출물에 대한 워크스루(walkthrough)
- 고위험 요소(예: 안전이 중요한(safety-critical) 모델, 윤리적 리스크 완화, 규제 준수 등)에 대한 인스펙션(inspection)

리뷰 참여자는 AI 시스템의 다기능적(cross-functional) 특성을 반영하여, 테스터, 데이터 과학자, ML 엔지니어, 도메인 전문가, 그리고 관련이 있는 경우 윤리 또는 컴플라이언스 전문가를 포함하는 다학제적 팀으로 구성되어야 합니다. 리뷰어의 역할은 기술적 정확성(예: 모델 성능, 데이터 품질)뿐만 아니라, 투명성·견고성·공정성과 같은 AI 특유의 품질 특성과의 정합성도 함께 보장해야 합니다.

AI 시스템에 대한 동료 검토 접근법은, 리뷰 대상 산출물, 리뷰 유형, 참여자를 명시한 간단한 표로 문서화할 수 있습니다. 이를 통해 모든 관계자가 리뷰 접근법과 자신의 관여 범위에 대한 전체적인 개요를 파악할 수 있도록 하는 커뮤니케이션이 가능해집니다.

### 3.5.2 SG2 동료 검토 수행

다른 모든 접근법과 마찬가지로, 이는 정의되고 합의된 접근법으로 존재하는 데 그쳐서는 안 되며, 실제로 실천에 옮겨져야 합니다. 동료 검토는 동적 테스트만으로는 이후에 탐지하기 어려운 AI 특유의 리스크를 식별하는 데 특히 중요합니다. 그 예로는 편향되거나 불완전한 데이터셋을 들 수 있습니다. 앞 절에서 언급했듯, 리뷰에는 흔히 더 폭넓은 범위의 이해관계자가 참여합니다.

애자일 소프트웨어 개발에서와 마찬가지로, AI 프로젝트 내의 동료 검토는 통상 경량화되어 있고, 협업적이며, 반복적입니다. AI 모델과 데이터셋이 계속 진화하는 특성으로 인해, 리뷰는 흔히 일상적인 개발 활동에 통합되어 라이프사이클 전반에 걸쳐 지속적으로 수행됩니다. 그 목표는 변함이 없습니다: 조기 결함 탐지와 공유된 이해를 통해 품질을 향상시키는 것입니다. 다만 AI 시스템의 경우, 그 중점은 결정론적 동작만을 리뷰하는 것에서, 다양한 운영 조건 하에서 AI 시스템이 책임감 있고 신뢰할 수 있게, 그리고 용인 가능한 리스크 범위 내에서 동작하는지를 평가하는 것으로 옮겨갑니다.

## References

- [AiU] Artificial Intelligence United (2019), AiU Certified Tester in AI (CTAI) Syllabus V1.0, Artificial Intelligence United
- [ISTQB] International Software Testing Qualifications Board (2026), Certified Tester AI Testing (CT-AI) Syllabus V2.0, ISTQB
- [ISO 25010] ISO/IEC 25010 (2023), *Software and systems engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Product quality model*, International Organization for Standardization
- [ISO 25059] ISO/IEC 25059 (2023), *Software and systems engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Quality models for AI systems*, International Organization for Standardization
- [Smith] A.L. Smith, R. Black, J. Davenport, J. Olszeweska, J. Rößler and J. Wright (2022), *Artificial Intelligence and Software Testing*, BCS Learning and Development,
- [TMMiAgile] E. van Veenendaal (ed.) (2020), *TMMi in the Agile world V1.4*, TMMi Foundation