

시스템 요구사항 문서 (SRD)

프로젝트 제목: <YOLO 기반 화재 감지와 협동 로봇 Fire Guard AMR>

버전: 1.0

날짜: [2025.12.02]

팀명: Double F (F1 & F3)

1. 소개

이 시스템 요구사항 문서는 실내 화재 대응 협동 로봇 시스템(FireGuard)의 개발 및 구현을 위한 기술 사양을 정의합니다. FireGuard 시스템은 웹캠 기반 화재 1차 감지, 자율 이동 로봇(AMR)에 의한 현장 재확인 및 경보, 대피 안내와 위험 구역 분석 기능을 통합하여 초기 3~5분 골든타임 내 대응을 목표로 합니다. 본 문서에는 PC 기반 화재 감지 서버(PC1), UI·관제 서버(PC2), TurtleBot4 AMR(A/B), ROS2 DDS 네트워크, 데이터베이스 및 통합 인터페이스에 대한 하드웨어, 소프트웨어, 네트워킹 및 연동 요구사항이 포함됩니다.

2. 시스템 개요

FireGuard 시스템은 AI 기반 자율 이동 로봇(AMR) 두 대와 고정형 웹캠을 사용하여 실내에서 발생한 화재를 조기에 감지하고, 로봇이 화재 구역으로 이동해 현장을 재확인한 뒤 경보를 발령하며, 동시에 대피 안내와 위험 구역(Hazard Zone) 지도를 제공하는 통합 솔루션입니다. PC1은 웹캠 영상에 대한 YOLO 화재·연기 검출과 Zone 매핑을 수행하여 1차 의심 화재 이벤트를 생성하고, PC2는 ROS2 토픽을 통해 수집된 화재 이벤트, 로봇 상태, 위험 구역 정보를 실시간 UI와 SQLite DB로 시각화·저장합니다. Robot A는 의심 화재 Zone으로 자율 주행하여 온보드 카메라로 화재를 재감지하고 최종 확정 및 부저 경보를 수행하며, Robot B는 LiDAR 기반 SLAM과 사람 감지를 통해 대피 경로 안내 및 Hazard Zone 생성을 담당합니다. 이와 같이 네비게이션, 센서 데이터 처리, 실시간 경보, 로그 관리 기능이 ROS2 DDS 네트워크 상에서 유기적으로 통합되도록 설계됩니다.

2.1 주요 시나리오

- 1) PC1 웹캠이 실시간으로 영상을 수집하고, YOLO 화재·연기 검출을 수행한다.
- 2) 신뢰도 임계값을 초과하면 /fire_event(1차 의심 이벤트)를 발행한다.
- 3) Robot A는 /fire_event를 수신하여 해당 Zone을 목표로 /navigate_to_pose를 호출하고 출동한다.
- 4) Robot B는 /fire_event를 수신하여 Guide 모드로 전환하고, SLAM 및 Hazard Zone 분석을 준비한다.
- 5) Robot A가 화재 Zone 인근에 도착하면 온보드 RGB-D 카메라와 YOLO를 통해 /fire_local_event(현장 재검출 결과)를 발행한다.
- 6) /fire_local_event가 화재 확정인 경우, Robot A는 부저를 활성화하고, PC2 UI에는 화재 확정 상태가 붉은색 아이콘으로 표시된다.
- 7) Robot B는 Hazard Zone 정보를 /hazard_zone 토픽으로 주기적으로 발행하고, 사람을 감지할 경우 /tts_cmd 기반 음성 안내로 대피 방향을 안내한다.
- 8) 모든 이벤트(/fire_event, /fire_local_event, /hazard_zone, /robot*_status)는 SQLite DB에 기록되어 이후 분석 및 리플레이에 사용된다.

3. 하드웨어 및 OS 요구사항

3.1 PC

- **운영 체제:** Ubuntu 22.04 (ROS2 Humble 기반)
- **PC1 카메라:** USB 웹캠 2대 (서로 다른 시야 확보)
- **PC1 GPU:** 실시간 YOLO 추론을 위한 NVIDIA GPU 탑재(예: RTX 3060급 이상)
- **저장 장치:** SQLite 로그 저장을 위한 SSD (권장 256 GB 이상)
- **네트워크:** PC1, PC2, AMR 로봇 간 Wi-Fi 6 을 통한 고 대역폭 무선 연결 지원

3.2 AMR 장치 (TurtleBot4 기반, Robot A/B 공통)

- **하드웨어:** TurtleBot4 Standard 플랫폼 (Raspberry Pi 4 기반 컨트롤러)
- **운영 체제:** Ubuntu 22.04 LTS + ROS2 (Nav2/SLAM 패키지 구동 가능)
- **카메라:** USB 카메라
- **센서:**
 - A. 2D LiDAR (실내 자율주행 및 장애물 회피용)
 - B. RGB-D 카메라 (화재 현장 재확인, 사람 감지용)
 - C. IMU 및 휠 오도메트리 센서 (EKF 기반 상태 추정용)
- **배터리:** 최소 4시간 배터리 수명

4. 소프트웨어 요구사항

4.1 PC 소프트웨어 패키지

- **Python3**: 화재 감지 노드, 데이터 처리, UI 연동 등을 위한 프로그래밍 언어
- **ROS2** (Robot Operating System 2 – Humble): AMR 제어 및 통신을 위한 주요 플랫폼
- **PC1**: 웹캠 영상 수집, YOLO 화재·연기 감지, Zone 매핑, 화재 의심 이벤트(/fire_event) 발행을 위한 ROS2 노드 실행
- **PC2**: 로봇 상태, 화재 이벤트(/fire_event, /fire_confidence), hazard_zone 등을 구독하여 관제 UI와 연동
- **OpenCV**: PC1에서 USB 웹캠 영상 캡처 및 전처리, YOLO 입력용 이미지 변환에 사용
- **Ultralytics (YOLO)**: 객체 탐지 모델을 사용하여 실시간 화재 감지 수행
- **SQLite3**: 데이터 저장 및 기록 보관을 위한 경량 데이터베이스
- **UI 프레임워크**: PC2에서 실시간 지도, 로봇 위치, 화재 상태, 위험 구역 정보를 시각화하고, 필요 시 로봇 제어 명령(/navigate_to_pose, /tts_cmd)을 발행하기 위한 인터페이스

4.2 AMR 소프트웨어 패키지

- **Python3**: ROS2 노드 구현(센서 드라이버, 상태 추정, 온보드 화재 검출, 대피 안내 로직 등)에 사용
- **ROS2 네비게이션 스택(Nav2)**:
 - A. Lidar, IMU, Wheel Odometry, AMCL(Localization) 정보를 활용하여 화재 의심 Zone까지 자율 주행 수행
 - B. /navigate_to_pose 액션 목표를 받아 경로 계획 및 장애물 회피, /cmd_vel 속도 명령 생성
- **OpenCV**: 이미지 처리 및 컴퓨터 비전 기능
 - A. Robot A: 온보드 RGB-D 카메라(OAKD) 영상 전처리 및 로컬 YOLO 화재 재감지에 사용
 - B. Robot B: 사람 검출 및 대피 안내용 영상 처리에 사용
- **Lidar**: 장애물 파악 및 맵 그리기
- **Ultralytics (YOLO)**: 객체 탐지 모델을 사용하여 정확한 화재 위치 파악
 - A. Robot A: 화재 의심 구역 도착 후 현장 재검출(/robotA/local_fire_event) 수행
 - B. Robot B: 사람 검출 기반 대피 유도 기능 구현
- **TTS(Text-to-Speech) 엔진**: Robot B에서 /tts_cmd를 받아 음성으로 대피 안내 방송 출력
- **ROS2 기반 시스템 유틸리티**:
 - A. publish/robot/status, tf2 : 로봇 링크·좌표계 관리
 - B. 진단/모니터링 노드: 배터리 상태, 도킹 상태, 동작 모드 등 /robotA_status, /robotB_status 토픽 발행

5. 네트워크 요구사항

5.1 AMR 및 서버 연결성

• 무선 네트워크:

- A. PC1(화재 감지 서버), PC2(UI 관제 서버), AMR Robot A/B는 동일 무선망에 연결되며 ROS2 DDS를 통해 통신한다.
- B. 최소 Wi-Fi 5, 권장 Wi-Fi 6 환경을 사용하며, 일반적인 실내 환경에서 평균 왕복 지연(RTT)을 50 ms 이하로 유지해야 한다.

• 네트워크 토폴로지: PC1을 ROS2 도메인의 중심 노드로 두고, PC2 및 AMR A/B는 동일 Domain ID에서 /fire_event, /fire_confidence, /robot*_status, /hazard_zone 등의 토픽을 주고받는다.

모든 AMR은 하나의 AP 또는 동일한 ESS 내 AP들에 접속하여 이동 중에도 세션이 유지되어야 한다.

• 대역폭 요구사항:

- A. 웹캠 원본 영상은 PC1 내부 처리용으로만 사용하며, 무선망을 통해 전송되는 데이터는 주로 ROS2 메시지(이벤트/상태 정보)로 한정한다.
- B. 로봇 1대당 ROS2 토픽 전송량은 약 1 Mbps 이내로 유지되도록 설계하며, 두 대의 AMR과 PC2가 동시에 동작하더라도 네트워크 병목이 발생하지 않아야 한다.

• 데이터 전송 및 보안 :

- A. 데이터 전송 속도: 실시간 비디오 스트리밍 및 원격 측정을 위한 AMR 당 최소 1 Mbps
- B. 네트워크 보안: Wi-Fi 암호화를 위한 WPA3, 서버 통신의 IP 화이트리스트 및 방화벽 보호

6. 기능 요구사항

6.1 실시간 데이터 처리

• 웹캠 영상 처리(PC1):

- A. PC1은 2대의 USB 웹캠으로부터 최소 5 FPS(프레임 간격 0.2초 이내) 이상으로 영상을 수집해야 한다.
- B. YOLO 화재·연기 검출 모델의 1프레임당 추론 시간은 0.1초 이내로 제한하며, 웹캠 영상 입력부터 1차 화재 의심 이벤트 생성까지의 총 지연시간은 1초 이내여야 한다.
- C. 웹캠으로부터의 영상 처리는 PC1 내부에서만 수행하며, 무선망에는 이벤트·상태 정보만 전송되도록 한다.

- **로봇 현장 재확인(Robot A):**

- A. Robot A는 화재 의심 이벤트 수신 후 1초 이내에 목표 위치를 설정하고 Nav2 /navigate_to_pose 액션을 호출해야 한다.
- B. 화재 의심 Zone 도착 후, 온보드 RGB-D 카메라와 YOLO 모델을 사용하여 2초 이내에 현장 재검출을 수행하고, 결과에 따라 화재 확정/오탐 여부를 상위 시스템에 전송해야 한다.
- C. 화재가 확정된 경우, 확정 메시지 생성 후 1초 이내에 로봇 부저 및 경고음이 작동해야 한다.

- **위험 구역 분석 및 UI 반영(Robot B, PC2):**

- A. Robot B는 Lidar 기반 SLAM 및 센서 데이터를 사용하여 위험 구역(hazard_zone)을 판단하고, 1초 주기 이내로 hazard zone 정보를 갱신해야 한다.
- B. PC2 UI는 화재 의심 및 확정 이벤트, hazard_zone, 로봇 상태 정보를 수신한 후 0.5초 이내에 지도 및 상태 패널에 반영해야 한다.

6.2 저장 및 보관

- **데이터 로그 기록:**

- A. 시스템은 이벤트, 화재 발생 위치(Zone), 발생·확정 시각, Robot A/B의 위치 및 도착 시각, hazard_zone 정보, 배터리 및 도킹 상태 등을 모두 로그로 기록해야 한다.
- B. 각 로그 항목에는 타임스탬프와 로봇 ID, Zone ID, 신뢰도(Confidence) 등 분석에 필요한 메타데이터가 포함되어야 하며, 이벤트 발생 후 1초 이내에 DB(SQLite)에 저장되어야 한다.

- **데이터 아카이빙:**

- A. 모든 로그 데이터는 최소 1년간 보관 가능해야 하며, 월별 또는 시나리오별로 백업 파일을 생성하여 외부 저장 장치로 내보낼 수 있어야 한다.
- B. 저장된 로그를 기반으로 “1차 감지 → 로봇 출동 → 현장 확정 → 대피 안내 및 hazard_zone 변화”의 시나리오를 재구성할 수 있어야 하며, 이를 통해 시스템 성능 평가 및 오탐/미탐 사례 분석이 가능해야 한다.

6.3 ROS2 interface 정의

FireGuard 시스템은 ROS2 DDS 상에서 다음과 같은 주요 토픽을 사용하여 PC1, PC2, Robot A/B 간 정보를 교환한다. 특히 /fire_event(1차 의심 화재 이벤트) 와 /fire_local_event(로봇 현장 재확인 이벤트) 는 역할이 명확히 구분된다.

6.3.1 화재 관련 토픽

- **/fire_event:** 1차 화재 의심 이벤트(publish PC1, 웹캠에서 카메라가 감지됐는지 확인한 뒤 UI에 화재 발생 띄우기)

- `/fire_local_event`: 로봇 현장 재확인 이벤트 (Robot A → 시스템, AMR에서 불을 감지하면 알람 발생역할)
- `/cam/fire_detection_status`: 현재 화재 Zone 정보
- `/publish/cam*/yolo/compressed` :
- `/cam/fire_cam*_detection` : webcam fire detection

6.3.2 로봇 상태 및 제어 토픽

- `/robot*/status`: 로봇 상태 정보
- `/robot*/oakd/rgb/camera_info` : robot 카메라
- `/robot*/oakd/rgb/image_raw/compressed` :
- `/robot*/oakd/stereo/image_raw` :
- `/cam/fire_detection/status`:
- `/navigate_to_pose`: Nav2 목표 위치 액션
- `/robot4/cmd_audio`(화재 발견 시)
- `/robot6/cmd_audio`(사람 발견 시): 음성 안내 명령(경고 부저)

6.3.3 주요 ROS2 토픽 인터페이스

토픽 이름	메시지 타입	주기(Hz)	publisher	Subscriber	설명
<code>/fire_event</code>	<code>std_msgs/Bool</code>	비동기(감지 시)	PC1(yolo)	robotA, robotB, UI	웹캠 화재 감지
<code>/fire_local_event</code>	<code>std_msgs/Bool</code>	비동기(감지 시)	robotA	PC1, PC2	robotA 화재 감지
<code>/cam/fire_detection_status</code>	<code>std_msgs/String</code>	0.2초마다	PC1	robotA, robotB, UI	위험 구역 정보
<code>/robot4/amcl_pose</code> <code>/robot6/amcl_pose</code>	<code>geometry_msgs/PoseStamped</code>	1-5 Hz	robotA	PC1, PC2, robotB	robotA 위치 전송
<code>/navigate_to_pose</code>	<code>nav2_msgs/action/NavigateToPose</code>	요청 시	robotA/B client	robotA/B Nav2 server	Nav2 목표 위치 액션
<code>/robot4/cmd_audio</code>	<code>irobot_create_msgs/msg/AudioNoteVector</code>	필요 시	robotA	Create3 robot 내부 시스템	화재 감지 부저
<code>/robot6/cmd_audio</code>	<code>irobot_create_msgs/msg/AudioNoteVector</code>	필요 시	robotB	Create3 robot 내부 시스템	대피 안내 부저

6.4 시스템 상태/모드 정의

FireGuard 시스템은 다음과 같은 상위 상태(State)를 가진다.

- **IDLE**:
 - 화재 이벤트가 없는 기본 대기 상태.
 - PC1은 웹캠 감지를 계속 수행하지만, `/fire_event`는 발행되지 않는다.
 - Robot A/B는 Docked 상태 또는 순찰 모드에 머무른다.
- **화재 발생**:

- PC1에서 /fire_event가 발생한 상태.
- Robot A는 Fire Mode로 전환하여 해당 Zone으로 출동하고, Robot B는 Guide Mode를 준비한다.

• **화재 진압:**

- Robot A의 /fire_local_event를 통해 화재가 확정된 상태.
- Robot A 부저가 동작하며, Robot B는 Hazard Zone 기반으로 대피 안내를 수행한다.
- PC2 UI에는 '화재 발생'이 글자로 표시되며, 관련 로그가 우선적으로 강조 표시된다.

• **ERROR / DEGRADED :**

- 네트워크 장애, 센서 장애 등으로 시스템 기능이 저하된 상태.
- 해당 상태에서는 로봇 속도를 제한하고, Teleop 모드 우선, 경보 기능만 유지하는 등 안전 모드로 전환한다.

7. 비기능 요구사항

7.1 성능

• **네비게이션 응답성:**

- PC1 또는 PC2에서 로봇 목표 위치 명령을 전송했을 때, Nav2 액션 서버가 이를 수신하고 경로 계획을 시작하기까지의 응답 시간은 200 ms 이내여야 한다.
- Robot A/B의 주행 중 위치 갱신 주기(AMCL 기준)는 10 Hz 이상을 유지하여, UI에서 실시간에 가까운 위치 추적이 가능해야 한다.

• **다중 로봇 운용성:**

- 두 대의 AMR(TurtleBot4 A/B)이 동시에 움직이는 상황에서도 충돌 없이 경로 계획이 수행되어야 하며, 서로의 위치 및 Hazard Zone 정보를 반영하여 회피 경로를 생성할 수 있어야 한다.
- 네트워크 및 연산 부하 증가 시에도 화재 감지 및 로봇 제어 기능은 최소 95% 이상의 가용성을 유지해야 한다.

• **신뢰성:**

- ROS2 노드 장애 발생 시 재시작 전략(예: systemd, 라이프사이클 노드)을 통해 핵심 노드(화재 감지, 네비게이션, 통신)는 자동 복구될 수 있어야 한다.
- 네트워크 지연 또는 패킷 유실이 발생하더라도 화재 확정 이벤트 및 경고 알림은 중복 전송 또는 재시도 메커니즘을 통해 손실 없이 전달되어야 한다.

8. 준수 요구사항

• 데이터 보호 및 개인정보:

- A. 개인 및 비디오 데이터 처리를 위한 GDPR 수준의 데이터 보호 규정 준수 목표
- B. 웹캠 및 로봇 카메라로 수집되는 영상 데이터는 화재 감지 및 시스템 성능 분석을 위한 용도에 한정하여 사용하며, 외부로 반출 시에는 얼굴 등 개인 식별 정보가 포함되지 않도록 비식별화 조치를 수행해야 한다.
- C. 관련 법령(개인정보 보호법, 정보통신망법 등)에서 요구하는 영상정보 처리기기 운영 기준을 준수해야 한다.

• 안전 및 규제 준수:

- A. AMR 로봇은 실내 환경에서의 운용을 고려하여 전기·배터리 안전, 충전기 안전 등에 관한 국내·국제 규격(예: KC, CE, UL 등)을 준수해야 한다.
- B. 무선 통신 장비는 전파법 등 관련 규정을 준수하여 인증된 모듈을 사용해야 하며, 화재 안전 관련 시험이나 실증 환경에서는 소방 관련 법규 및 시설 관리 규정을 준수해야 한다.

• 운영 안정성:

- A. 시스템 업그레이드 또는 소프트웨어 패치 시, 화재 감지 및 로봇 제어 기능이 중단되지 않도록 단계적 배포(rolling update) 또는 사전 검증 환경을 통해 테스트를 수행해야 한다.
- B. 장애 발생 시 로그와 알림을 통해 원인을 추적할 수 있어야 하며, 장애 처리 절차(재부팅, 네트워크 복구, 로봇 회수 등)에 대한 운영 매뉴얼을 제공해야 한다.

9. 수락 기준

9.1 시스템 성능

• 화재 감지·전파 성능:

- A. 웹캠 영상 한 프레임이 PC1에서 YOLO 화재·연기 검출을 거쳐 1차 화재 의심 이벤트(/fire_event)로 전파되기까지의 지연 시간은 1초 이내여야 한다.

- B. /fire_event가 발행된 시점부터 Robot A가 현장 재검출을 수행하고 화재 확정 이벤트(/fire_local_event)를 발행하기까지의 전체 소요 시간은 실내 20 m 이내 환경에서 60초 이내여야 한다.
- C. /fire_event가 발행된 시점으로부터 1초 이내에 PC2 UI에 화재 확정 상태가 표시되고, Robot A의 부저가 동작해야 한다.

•데이터 로그 기록 신뢰도:

- A. /fire_event, /fire_confidence, /hazard_zone, /robot*_status 등 핵심 이벤트·상태 토픽의 99% 이상이 SQLite DB에 정상 저장되어야 하며, 로그 누락률은 1% 미만이어야 한다.
- B. 로그 타임스탬프의 오차(실제 발생 시각 대비 기록 시각)는 0.5초 이내여야 하며, 동일 이벤트에 대한 PC1·PC2 간 시각 차이는 1초 이내여야 한다.

9.2 Navigation / SLAM

• 위치 정밀도 및 도달 성능:

- A. Robot A/B는 사전 구축된 맵 상에서 자율 주행 시, AMCL 기준 위치 추정 오차가 ± 10 cm 이내(정지 시 기준)로 유지되어야 한다.
- B. /navigate_to_pose 목표가 주어졌을 때, 장애물 회피를 수행하면서 목표 지점 0.5 m 이내까지 도달하는 성공률이 95% 이상이어야 한다.

•SLAM 및 Hazard Zone 반영 성능 (Robot B):

- A. Robot B는 SLAM 기반으로 실내 지도를 구축·갱신할 때, 주요 구조 변경(출입문 개폐, 장애물 추가 등)이 발생한 후 최대 10초 이내에 맵과 경로 계획에 반영할 수 있어야 한다.
- B. Hazard Zone 정보(/hazard_zone)는 센서 입력 변화(연기·열·장애물 감지) 발생 후 1초 이내에 갱신되어야 하며, Robot A와 PC2 UI에서 최신 Hazard Zone을 기준으로 경로 계획 및 시각화를 수행할 수 있어야 한다.

9.3 Teleop 키 조작

- **원격 조작:** PC2 관제 UI 또는 Teleop 노드에서 발행하는 수동 조작 명령(키보드/조이스틱 등)에 대해, AMR의 반응 시간은 300 ms 이내여야 하며, 지연 없이 사용자가 의도한 방향으로 움직여야 한다.
- **경로 추종:** Teleop 모드에서 사용자가 지정한 방향으로 5 m 이상 연속 주행 시, 로봇은 장애물 회피 기능과 함께 95% 이상의 성공률로 충돌 없이 주행을 완료해야 한다.
- **모드 전환:** 자율 주행 중이라도 비상 상황에서 Teleop 명령이 들어오면 1초 이내에 자율 주행을 중지하고 수동 조작 모드로 전환되어야 한다.

9.4 통합 테스트

- **통신 및 네트워크:** PC1, PC2, AMR Robot A/B 간 ROS2 DDS 통신이 안정적으로 유지되고, 화재 의심/확정 이벤트 및 로봇 상태 메시지의 손실률이 1% 미만임을 통합 테스트를 통해 검증해야 한다.
- **기능 연동:** Navigation, SLAM, YOLO 기반 화재 감지, Hazard Zone 생성, UI 시각화 기능이 하나의 시나리오에서 통합적으로 동작해야 하며, 사용자는 관제 대시보드에서 3초 미만의 지연으로 화재 상태·로봇 위치·위험 구역을 모니터링할 수 있어야 한다.
- **시나리오 검증:** “1차 감지(PC1) → Robot A 출동 및 현장 확정 → Robot B Hazard Zone 생성 및 대피 안내 → UI 및 로그 기록”의 전체 플로우를 최소 10회 이상 반복 시험하여, 전체 시나리오 성공률이 90% 이상이어야 한다.

9.5 안전 및 보안

- **운행 안전:** AMR은 사람, 장애물, 계단·낙하 구역 등과의 충돌을 회피하며 동작해야 하고, 통합 테스트 환경에서 사람과의 물리적 접촉 사고가 발생하지 않아야 한다.
- **비상 정지:** 로봇 전면 비상 스위치 또는 소프트웨어 E-Stop 명령 입력 시, 로봇은 1초 이내에 구동을 정지해야 한다.
- **정보 보안:** 무선 네트워크는 WPA2 이상 암호화를 사용하고, ROS2 네트워크는 도메인 구분 및 방화벽 설정을 통해 외부 비인가 접속이 차단되어야 한다. 로그 및 영상 데이터는 권한을 가진 사용자만 접근할 수 있도록 계정·권한 관리가 적용되어야 한다.

9.6 최종 목표 달성

- **화재 상황 감지 성능:** 테스트 시나리오(실내 모형 화재, 연기 발생 등)에서 실제 화재 상황을 90% 이상의 확률로 감지하고, 그 중 85% 이상은 Robot A의 현장 재확인까지 포함하여 /fire_confidence 이벤트로 올바르게 확정되어야 한다.
- **초기 대응 시간:** 골든타임(3~5분) 내 대응을 목표로, 화재 발생 시점 기준 90초 이내에 Robot A가 화재 의심 Zone 인근에 도달하고, 120초 이내에 최초 현장 영상·상태가 PC2 UI에 표시되어야 한다.
- **시스템 가용성:** 테스트 기간 동안 FireGuard 시스템(PC1, PC2, AMR A/B, 네트워크)의 전체 가동률은 99% 이상이어야 하며, 계획된 점검·업데이트 시간을 제외하고 예기치 않은 다운타임으로 인한 서비스 중단이 발생하지 않아야 한다.