

군용정보시스템의 소프트웨어 시험평가 전략 및 계획  
 -“차세대 한국형 워게임 모델”을 중심으로-  
 (Software Testing/Evaluation Strategy and Plan  
 for the Military Information Systems)

김 화 수\*

**Abstract**

This paper is aimed to propose an testing and evaluation strategy and plan in Military Information Systems, especially focusing on development an "New-generation Wargame Model". Through this research, we concluded that the effective and efficient testing/evaluation strategy/plan can aid the productivity, maintainability, availability, etc., of the Military Information Systems. Highlights of the proposed testing and evaluation strategy and plan for Military Information Systems are as follows. First, in the unit and module integration testing phase, hybrid of black-box and white-box testing techniques are available for Military Information Systems and progressive approach for module integration phase should be considered because of the complexity. Second, in the system testing phase, integrated module should be tested with respect to the function and performance that should be satisfied with the user requirements, specifications, risk analysis, etc.. Third, in the acceptance testing phase, reliability, interoperability, maintainability, availability, integrity, etc., must be considered in the actual or mini-operational environments for testing efficiently and effectively.

---

\* 국방대학원 전자계산학과

# 1. 서 론

‘차세대 한국형 위게임 모델’은 육군에서 기존에 활용하고 있는 미국의 위게임 모델인 CBS(Corps Battle Simulation)모델이 가지고 있는 여러 가지 제한사항-한국적 전장환경을 고려하지 않는 실전 묘사 등을 극복하기 위해 육군 교육사령부 전투지휘훈련단에서 군단 및 사단급 지휘관 및 참모들의 지휘소 훈련을 지원하고, 한국적 여건에 부합하는 독자적인 위게임 모델을 보유하기 위해서 자체적으로 개발을 시작하였다.

현재 개발완료 단계로서 개별 용역 개발업체와 협력하여 개발한 모델들이 군의 요구성능을 만족하는지 여부를 시험평가 중의 단계에 와 있다.

그러나, 개발된 모델의 시험평가는 국방부 훈령 제561호(국방부정보체계관리규정) 제32조에 의거 용역업체 지원 하에 전투지휘훈련단에서 주관하여 수행토록 되어 있으나, 전투지휘훈련단 자체 시험평가안으로 평가시 객관성 유지가 어렵고, 시험평가 계획 수립에는 전문성과 장시간의 연구가 요구됨으로 인해 ‘차세대 한국형 위게임 모델’의 시험평가계획에 대한 연구의 필요성이 제기되었다.

따라서, 이러한 필요에 의해 시작된 본 논문의 목적은 현재 개발된 ‘차세대 한국형 위게임 모델’의 모의논리 및 기술적인 측면에서 요구성능의 만족여부를 평가하기 위한 객관적인 시험평가 전략 및 방안을 도출하는데 있다.

본 논문에서는 육군 교육사령부 전투지휘훈련단에서 개발하여 시험평가 하고자 하는 ‘차세대 한국형 위게임 모델’을 대상으로 하여 이 모델의 단위 및 통합시험, 체계시험, 운용 및 채택시험을 위한 시험 전략과 계획을 수립하기 위해 국내에서 실시한 각종

시험사례(KNTDS, C3I 등) 및 관련자료와 시험용 공개 및 상용 자동화도구, 국내의 소프트웨어 시험평가 기관에 대한 조사 및 분석을 실시하고 ‘차세대 한국형 위게임 모델’의 특성과 소프트웨어공학적인 시험평가 기법을 조사 및 분석함으로써 ‘차세대 한국형 위게임 모델’에 적합한 시험전략과 시험계획을 연구한 연구 결과 보고서를 바탕으로 중요한 사항만 요약하였다.

# 2. 차세대 한국형 위게임 모델의 특징

“차세대 한국형 위게임 모델”은 전체적인 면에서 지상이동, 근접전투, 지휘, 정보, 탐지 및 정찰, 지형, 전자, 포병, 공병, 육군항공, 방공, 군수, 공수, 전술공군, 침투 및 특수작전, 동원, 화생방, 해상, 후방 등 전장 21개 기능별 상호작용으로 상호 연동되어 작동된다. 이러한 다양하고 복잡한 많은 임무기능을 수행해야 하는 “차세대 한국형 위게임 모델” 시스템은 일반적으로 다음과 같은 특성들을 가지고 있다고 할 수 있다.[1][4]

첫째, 구조가 매우 복잡하다. 차세대 한국형 위게임 모델은 복잡한 여러 서브 시스템들을 다룰 수 있도록 설계함에 따라서 이들을 지원하는 컴퓨터 시스템은 많은 구성품들과 접속해야 하고, 많은 신호와 행동들을 모니터링해서 상당한 양의 정적, 동적 자료들을 관리해야 한다. 뿐만 아니라 제어신호를 활성화시키거나 시스템 운용자와의 인터페이스를 통해 운용자 지시를 받아들이는 일 등의 기능들을 직시에 수행해야 한다. 따라서 이러한 시스템들은 많은 양의 하드웨어와 소프트웨어 구성품들 간의 상호작용이 필요하기 때문에 복잡한 설계가 요구된다.

둘째, 규모가 매우 크다. 차세대 한국형 위게임 모델과 같은 경우 많은 서브시스템들이 지형적으로 넓은 지역에서 운용되고 많은 외부 구성품들과 접속하게 되는데 그러한 시스템을 지원하는 컴퓨터 시스템은 기능과 성능을 고려하여 흔히 대형 분산 시스템으로 구성된다. 따라서 이러한 유형의 시스템들이 갖는 대규모적인 특성은 시스템을 설계, 구현, 시험, 검증, 운용, 유지보수 할 때 많은 어려움을 야기시킬 수 있다.

셋째, 소프트웨어 집약적이다. 차세대 한국형 위게임 모델 시스템에서는 복잡한 임무수행을 위해서 컴퓨터 시스템이 어떤 결정을 내리거나 행동을 취하기 위해서 엄청나게 많은 조건을 요구한다. 따라서 컴퓨터 시스템은 여러 구성품들의 상태를 정확히 유지하고 있어야 하는데 이런 상태들은 크고, 복잡한 자료구조에 의해서 정의될 수 있다. 위와 같은 기능들은 소프트웨어에 의해서만 구현 가능하기 때문에 시스템을 소프트웨어 집약적으로 만드는 원인이 된다.

넷째, 고도의 정확성이 요구된다. 차세대 한국형 위게임 모델의 임무와 관련된 기능들은 반드시 정확히 수행되어야 한다. 기능의 정확성은 군사적 응용 시스템의 중요한 요구조건이 되는데 시스템이 배치되기 전에 시험과 검증이 철저히 수행되어야 함을 의미한다.

다섯째, 실시간으로 작동한다. 일부분의 차세대 한국형 위게임 모델은 실시간 제약조건을 만족하면서 그 기능수행을 종료시간 이내에 완료하여야 한다. 실시간 작동은 크게 엄격한 실시간 작동과 완만한 실시간 작동으로 분류되는데 엄격한 실시간 작동은 종료시간을 만족시키지 못한 모든 기능수행을 모두 심각한 결함으로 간주하는 것이고, 완만한 실시간

작동은 종료시간에 시간 위약 기능을 배정해서 시스템 작동이 시간 위약기능을 최소화하는 방향으로 운용되는 것을 말한다.

여섯째, 신뢰성과 가용성이 요구된다. 차세대 한국형 위게임 모델에서 요구되는 중요한 특성 중 하나가 고도의 신뢰성과 가용성을 요구한다는 것인데 그것은 시스템의 하드웨어나 소프트웨어 구성품이 고장났다 할지라도 그 기능은 계속 수행해야 한다는 것을 의미한다. 물론 시스템의 어느 한 구성품이 고장났을 때 고장허용 기능에 의해서 시스템 기능이 점진적인 퇴보 현상을 보일 수 있으나 고장허용 기능이 있어야만 완벽하게 외부 간섭 없이 스스로 시스템을 복구시켜서 본래의 성능을 믿음만하게 계속 유지할 수 있는 것이다.

일곱째, 안전한 작동이 요구된다. 차세대 한국형 위게임 모델은 비밀성, 무결성, 가용성을 보장하기 위해서 보안 요구조건을 만족시켜야 하는데 실시간 시스템인 경우에는 이런 보안 요구조건들이 실시간 운용 제약조건 아래서 만족해야 할 때도 있다.

여덟째, 과중한 정보를 취급한다. 차세대 한국형 위게임 모델은 운용자가 필요에 따라서 적시에 병합되고 반복되어야 하는 엄청난 양의 정보에 직면하게 됨으로 운용자가 시스템의 기능을 감독하고 제어하기에는 복잡한 시스템이다. 따라서 운용자가 시스템 운용상의 취약점이 되지 않기 위해서 인간-기계 인터페이스(MMI: Man-Machine Interface)를 잘 설계 함으로써 정확한 정보가 적절한 형태로 운용자에게 표현되어서 시스템에 의해 운용자에게 주어진 기능이 신뢰할 수 있게끔 만들어져야 한다. 이러한 과중한 정보를 어떻게 잘 표현하고 대표성 있는 정보로 축약해낼 것인가 하는 것도 쉬운 문제는 아니다.

### 3. 소프트웨어 시험평가 기법 분석

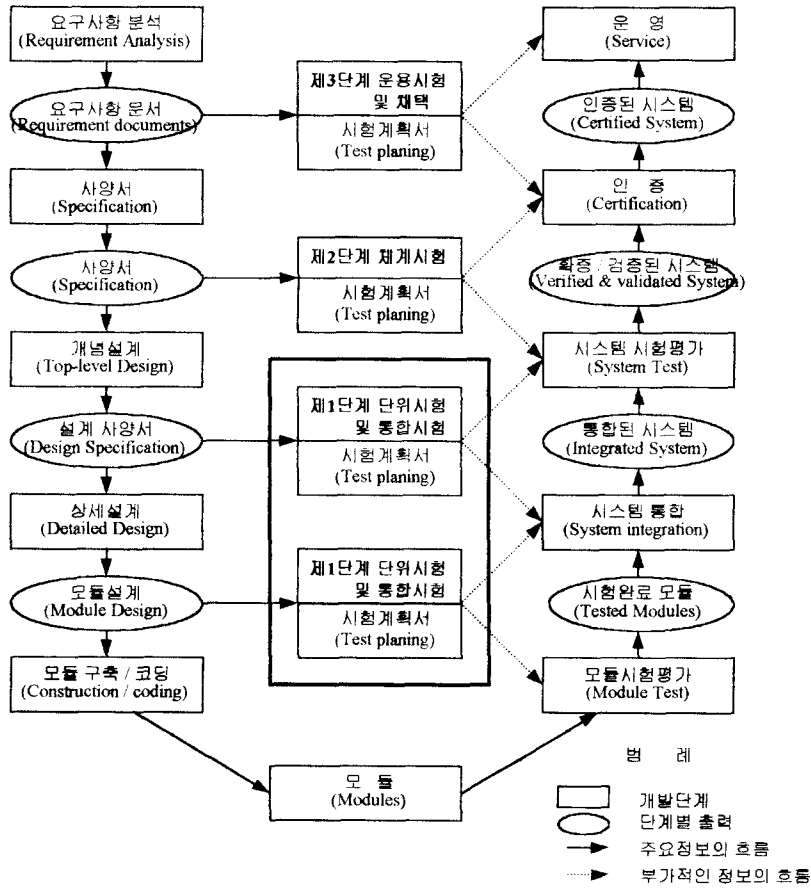
#### 3.1 소프트웨어 시험평가 개요

소프트웨어의 시험평가는 소프트웨어의 구조, 기능을 시험사례를 이용하여 소프트웨어를 수행시킨 후 그 결과를 분석하여, 소프트웨어에 내재되어 있을 오류를 제거함으로써, 최종 소프트웨어 제품의 품질 향상을 목적으로 실시한다. 이론적으로 소프트웨어의 완전한 시험은 소요비용과 시간의 관점에서 불가능하다. 따라서, 시험평가할 분야를 기술적으로 선

정하여 선택적인 시험을 실시하되, 특정기법에 의존하기보다는 프로젝트 특성에 적합한 기법을 선택하여 실시하는 것이 바람직하다는 것은 MIL-STD-498에서도 강조된 사항이다.[8]

#### 3.1.1 시스템 개발과정상의 시험평가 계획서 작성

일반적으로 소프트웨어 개발 생명주기 모델의 설명에 있어서 시험을 하나의 공정으로서 취급하고 있으나, 이것은 시험을 실시하는 공정을 가리키고 있는 것이며, 실제로 시험은 설계단계부터 시험설계를 하게 된다. 즉, 프로그래밍이 완전히 끝날 때까지 시험



<그림 1> "V" 개발 생명주기 모델

같은 것을 하지 않는다는 태도는 바람직하지 않으며, 대규모적인 시험평가는 시험평가 계획을 세워서 시험평가 방법을 정하고, 시험평가의 실제순서를 구체화하고 시험평가를 실시하여 그 결과를 가지고 소프트웨어의 좋고 나쁨을 판정하게 되는 것이다. 이것을 대표적인 소프트웨어 개발 생명주기 모델인 "V"모델을 대상으로 하여 도식하면 <그림 1>과 같다.[4]

### 3.1.2 시험평가 절차

한 개의 모듈 또는 여러 모듈들의 통합된 집합은 특정 소프트웨어 형상(코드)에 각 시험사례를 수행시킨 결과는 기대값과 비교된다. 결과치가 기대치와 다를 경우 이는 결함이 발견되었음을 뜻하며 디버깅 작업을 거쳐 수정되게 된다. 또한 결함이 얼마만큼 발견되었느냐하는 결함률은 신뢰도 측정 모형의 입력값이 되어 소프트웨어의 신뢰성을 예측하도록 해준다.

참고로 신뢰도 예측 절차 및 기대효과를 간략히 기술하면 다음과 같다.

첫째, 결함률(error rate)을 MTTF(Mean Time To Failure), 가용성 등의 소프트웨어 신뢰도 분석을 위한 메트릭을 결정한다.

둘째, 적합한 신뢰도 모형을 선정한다.

소프트웨어 신뢰도 모형은 소프트웨어를 블랙박스로 간주하고 시스템내의 모든 결함이 균일한 빈도로 고장발생에 영향을 미친다는 가정 하에 만들어진 수학 공식을 말하며 대표적인 신뢰도 모형은 Jelinski-Moranda Linear/Geometric De-trophication 모델, Bayesian 모델, Goal and Okumoto의 불안정한 디버깅 모델이 있다. 또한 포아송 프로세스를 기본 모델로 사용하는 Goel-Okumoto 모델, Musa 모델, Shoomen의 Exponential 모델 등이 있다.

셋째, 모델에 대한 분석 및 수정을 실시한다.

개발중인 시스템을 선정된 모형에 적용하고, 분석 결과 따라 신뢰도 할당이나 시스템 재설계가 이루어져야하며, 경우에 따라서는 모델 그 자체의 수정 또는 보완이 필요하며, 대부분의 경우 이러한 작업은 반복적으로 수행된다.

이와 같은 모형을 이용해 신뢰도를 예측하면 유사 프로젝트를 진행시 시험기간의 합리적인 설정, 시험기간동안 예측할 수 있는 고장발견수를 계산할 수 있게 됨으로써, 현재의 신뢰도 수준을 파악할 수 있게 되고, 목표수준 달성을 위해 얼마만큼의 시험이 추가로 진행되어야 하는가를 알 수 있게 됨으로써 프로젝트 관리에 큰 도움을 받을 수 있을 것이다.

### 3.1.3 시험평가 기법의 분류

소프트웨어의 시험평가 기법은 유형에 따라 여러 형태로 분류할 수 있으나, 일반적으로 소프트웨어 명세서에 정의된 기능의 유무 및 수행여부를 시험하는 기능시험과 소프트웨어 구조의 적절성 여부를 시험하는 구조시험, 시험평가시에 소프트웨어를 실행여부에 따라 정적시험과 동적시험으로 분류할 수 있으며, < 표 1 >은 시험평가 기법들간의 관계를 나타낸다.[6][12]

<표 1> 시험평가 기법간의 관계

분 류	기능시험	구조시험
내 용	· 소프트웨어 요구사항으로부터 시험내용 추출 · 명세서에 명시되지 않은 기능은 시험 불가	· 프로그램 구조로부터 시험내용 추출 · 기능상의 오류는 시험 곤란
적 용	· 단위시험 및 통합시험, 체계시험 및 운용 시험에 적용 · 동적시험 적용	· 단위시험 및 통합 시험에 적용 · 동적, 정적시험 적용
시험사례 설계기법	· 블랙박스 시험 · 점검표(Check list)	· 화이트박스 시험

## 3.2 시험평가 기법

앞에서 언급한 바처럼 소프트웨어에 대한 시험평가 기법은 분류 방법에 따라 다양하게 분류할 수 있으나, 소프트웨어 실행을 통해 시험평가를 실시하는가에 따라 소프트웨어를 실행해서 시험하는 동적 시험평가와 실행하지 않는 정적 시험평가로 구분할 수 있으며, 여기서는 정적 시험평가와 동적 시험평가에 대해서 시험평가 절차와 기법을 응용하는 방법에 대하여 다음과 같이 실 예를 조사 및 분석하고자 한다.

### 3.2.1 정적 시험평가

정적 시험평가 기법이란 프로그램 내용을 눈으로 보고 해석해서 에러를 발견하는 즉, 프로그램을 실행시키지 않고 시험평가를 실시하는 기법으로 다음과 같이 컴파일러에 의한 문법 체크 등 다양한 시험 기법이 있다.[18]

#### 가. 컴파일러에 의한 문법 체크 기법

소프트웨어의 상세설계를 기반으로 모듈을 구축한 후에 고급언어를 기계어로 번역하는 컴파일러를 이용하여 구축된 모듈의 문법(Syntax)을 체크하여, 발견된 오류를 수정할 수 있는 기법으로서 이러한 시험기법은 대규모 소프트웨어를 작성하던 소규모 소프트웨어를 작성하던 기본적으로 사용할 수 있는 시험기법의 일종이다.

#### 나. 변수의 일관성 체크 기법

이러한 시험의 한 예가 범위 체크 기법으로서, 이는 계산된 또는 저장된 변수의 값이 주어진 범위 내에 존재하는가 여부를 체크하는 기법으로, 시험평가에 있어서 매우 유용하게 활용될 수 있을 것이다.

#### 다. 내포된 루프(Nested loop)의 정도

정적 시험평가를 할 때에는 제어구조의 흐름을 분

석하여 불 필요가 대두된다. 일반적으로 절차적인 언어를 사용하여 소프트웨어를 개발할 때에는 내포된 루프가 3이하로 되는 것이 전체적인 소프트웨어의 가독성을 증진시킬 뿐만 아니라 유지보수의 용이성, 실행속도의 증진 등을 이룰 수 있다.

#### 라. 최적화 기법

최적화 기법이란 주어진 입력 프로그램과 의미적으로 동등하면서 좀더 효율적인 코드로 바꾸는 기법을 말한다. 전체적인 프로그램의 제어구조를 살펴보고 컴파일러에 의한 문법 체크에서 발견되지 않은 사항들을 검사하고 발견되면 최적화 기법으로 변경한다. 전체적으로 해당 시스템의 속도증진, 메모리 공간 사용의 효율성 등을 높일 수 있도록 시험평가를 하는 것이 바람직할 것이다.

### 3.2.2 동적 시험평가

동적 시험평가 기법이란 실제 시험대상 모듈의 프로그램을 실행시키면서 시험평가하는 기법이다. 이를 위한 시험사례 설계 기법에는 화이트박스 시험과 블랙박스 시험이 있다.

#### 가. 시험사례(Test case)

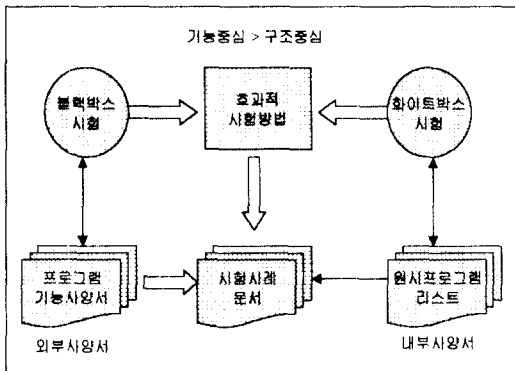
시험사례는 프로그램의 어떤 동작상태를 조사하기 위하여 여러 가지 시험자료를 경우에 따라 설정한 것으로 시험자료(test data)와 기대되는 출력결과를 포함한다. 좋은 시험사례의 작성은 소프트웨어 시험을 준비하는 궁극적인 목표로서 일종의 최적화 문제라고 할 수 있다. 즉, 모든 가능한 시험사례 중에서 결함 발견 확률을 가장 높일 수 있는 시험사례가 무엇인가를 찾는 문제로서, 이를 위한 방안으로 고안된 것이 시험사례 설계 방법이며, 여기에는 화이트박스 시험과 블랙박스 시험이 있으며, < 표 2 >에서 보는 것처럼 여러 가지 시험사례 설계기법들이

<표 2> 시험기법

전략	블랙박스 시험	화이트박스 시험
기법	• 동등분할	• 기본경로시험
	• 경계값 분석	• 자료흐름시험
	• 원인결과 그래프	• 조건시험
	• 오류예측	• 루프시험

포함되어 있다.

이들 시험방법 중 어느 한쪽만을 활용하여 시험사례를 설계하게 되면 충분한 시험을 실시할 수 없으며, 기본적으로 화이트박스 및 블랙박스 이들 2개의 시험방법을 함께 사용하여 실시하는 것이 바람직하다고 할 수 있다. 다음의 < 그림 2 >는 효과적인 시험을 위한 전체적인 개념을 나타낸 것이다.



<그림 2> 효과적인 시험의 개념도

### 나. 블랙박스 시험(Black box test)

블랙박스 시험방법은 소프트웨어의 기능적 요구사항에 초점을 맞추고, 이를 충분히 조사할 수 있는 일련의 입력조건을 유도해 낼 수 있게 해 주는 것으로, 기능시험이라고도 부른다. 즉 소프트웨어 내면을 알 수 없는 블랙박스로 규정하고 외부에서 그 기능과 성능을 시험하는 것으로 경우의 수를 만들어 입력자료에 대한 출력 값과 예상되는 출력 값을 비교

하여 명세서에 정의된 기능의 유무와 적절한 수행여부를 확인하는 시험이며, 블랙박스 시험의 종류, 특성은 다음과 같다.

#### (1) 동등분할(Equivalence partitioning) 기법

프로그램의 입력 도메인을 두 개이상의 동등클래스(equivalence class), 즉 유효동등클래스와 무효동등클래스로 분할하여 분할된 도메인의 대표 값을 이용하여 시험사례를 설계하는 방법으로, 유효한 시험사례만이 아니라 무효한 시험사례도 설정함으로써 프로그램의 여러 각도로부터 점검할 수 있게 한다.

#### (2) 경계값 분석(Boundary value analysis) 기법

명확하지 않은 이유 때문에, 대단히 많은 수의 오류들은 입력영역의 "중앙"에서 보다 입력영역의 경계에서 발생한다. 경계값 분석은 이런 이유 때문에 개발되었으며, 프로그램의 입력영역, 출력영역을 동등한 영역으로 분할하여, 각각의 영역의 경계에 가까운 점, 즉 on, above, below에 해당하는 값을 시험의 대상이 되도록 하는 기법이다. 경계값 분석은 동등분할을 보완해 주는 기법으로, 입력조건에 초점을 맞추기보다는 출력영역으로부터 시험사례를 유도한다.

#### (3) 원인-결과 그래프(Cause-effect graph) 기법

블랙박스 시험 중에서 동등분할이나 경계값 분석 기법의 단점은 입력환경의 복잡성을 완전하게 고려할 수 없다는 점이다. 입력 데이터간의 관계가 출력에 영향을 미치는 상황, 즉 입출력간의 논리적인 관계, 입력관계와 출력범위의 모든 한계값과 각각의 구조와 선택사항 등을 체계적으로 분석하여 효용성 높은 시험사례를 발견코자 원인-결과 그래프 기법이 제안되었으며, 입력데이터를 프로그램의 동작상태를 결정짓는 「원인」으로 간주하고, 출력데이터는 프

로그래머가 실행하여 실현하는 「결과」로 간주하여, 그 상관관계를 지정된 표기법으로 그리도록 추천한다. 그리고 각 노드는 1(Yes) 혹은 2(No)의 값만 갖는 것으로 해석된다. 이 방법을 사용하면 누구든지 일정한 조건을 만족하는 시험사례를 설계할 수 있다는 장점이 있다.

#### (4) 오차예측(Error guessing) 기법

오차예측 기법은 경계값 분석, 원인-결과 그래프 등 블랙박스 시험기법들이 놓칠 수 있을 만한 오류들을 감각과 경험으로 찾아보는 것이다. 흔히 입력 데이터의 오류를 시험하기 때문에 데이터 확인(data validation)기법으로 불리기도 한다.

#### (5) 블랙박스 시험 기법 분석

이미 기술한 동적 시험평가 기법 중 블랙박스 시험 기법들에 대해 '차세대 한국형 위게임 모델' 시험평가 적합성 여부를 분석하면 다음과 같다.

'차세대 한국형 위게임 모델'의 체계시험과 운용 및 채택시험에 적합하며, 이 경우 입력값이 어떤 범위를 가지고 있을 경우는 경계값 분석 기법, 입력값이 범위가 아닌 특수한 값을 요구하거나 집합의 구성요소 또는 논리조건 등이라면 동등분할 기법을, 그리고 위의 두 가지 기법이 놓칠 수 있는 오류들을 찾아내기 위해서는 추가적으로 오차예측 기법을 적용하는 것이 바람직하다고 판단된다.

그리고, 원인-결과 그래프 기법은 일정한 조건을 만족하는 시험사례를 누구나 설계할 수 있다는 장점이 있지만 위게임 모델의 방대한 모듈을 일일이 조사 및 분석하여 원인-결과 그래프화하여 완벽한 시험사례를 설계한다는 것은 시간 및 비용의 소요가 과다하다는 점에서 고려하기가 매우 곤란하다.

다. 화이트박스 시험(White-box test)

화이트박스 시험은 시험사례를 만들기 위해 소프트웨어 형상의 구조를 이용하는 것으로 구조시험이라고도 부른다. 즉, 프로그램의 내부구조인 알고리즘을 보면서 허용되는 모든 논리적인 경로(logical paths)를 파악하거나 경로들의 복잡도(complexity)를 계산하여 시험사례를 설계한다.

#### (1) 시험영역에 대한 이해

시험영역이란 시험대상이 가능한 경로를 어느 정도 통과하는지의 영역을 말하며, 여기에는 문장영역, 분기영역, 조건영역이 있다.

첫째, 문장영역 : 프로그램내의 각 원시코드 라인이 시험과정에서 단 한번이라도 수행되도록 시험사례들을 설계하는 것이다. 사실 개발자들은 많은 코드를 짜놓고 그중 상당부분(예를 들어 예러처리나 특수 예외조치를 고려한 코드)은 시험조차 하지 않고 만족해하는 경우가 있다.

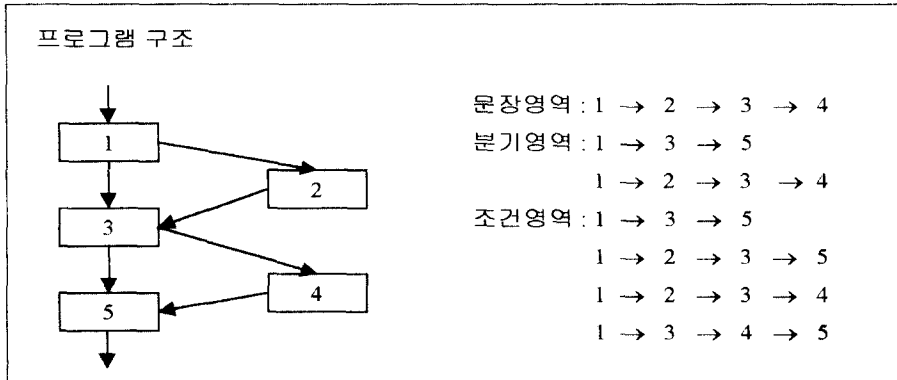
둘째, 분기영역 : 프로그램의 모든 경로가 단 한번이라도 수행되도록 시험사례들을 설계하는 것이다.

셋째, 조건영역 : 분기영역의 순서가 결과에 영향을 미친다는 가정하에 모든 논리적 경로들이 단 한번이라도 수행되도록 시험사례를 설계하는 것이다.

< 그림 3 > 은 조그만 프로그램의 시험영역을 보여주고 있다.[6]

화이트박스 시험은 각각의 기능적 요소를 검사하고, 처리순서가 올바른가를 확인하는 것이다. 그러나 대규모의 소프트웨어 개발에 있어서 그에 포함된 논리의 복잡성으로 인해 수행 가능한 경로가 너무 많아 그 모두를 시험하는 것이 매우 어렵기 때문에 최종 연구보고서에서 조사 및 분석된 자동화도구의 사용이 불가할 경우 화이트박스 시험은 별로 적합하지 않다.





<그림 3> 화이트박스 시험영역

(2) 화이트박스 시험 기법

(가) 기본경로 시험

기본경로 시험은 대표적인 화이트박스 시험기법으로서 프로그램의 논리적 복잡도를 측정 한 후 이 척도에 따라 수행시킬 기본경로들의 집합을 정의하는 것으로 프로그램의 모든 문장을 적어도 한번씩 실행하는 것을 보장받는다. 논리적 복잡도는 프로그램 구조의 순환적 복잡도(cyclomatic complexity)라고도 불리는데, 측정을 위해서는 프로그램 수행경로를 프로그램라인을 노드로, 수행방향을 간선으로 한 그래프로 나타낸 후 다음과 같은 공식을 적용시켜 계산한다.

$$\text{복잡도}(V) = \text{간선의 수}(E) - \text{노드의 수}(N) + 2$$

각 노드는 원칙적으로 프로그램 라인 한 줄에 해당하지만 복잡도에 영향을 미치지 않는 직선으로 수행되는 여러 라인의 집합으로 생각해도 무방하다. 예를 들어 < 그림 3 > 의 프로그램 구조가 지닌 복잡도는  $3(= 6 - 5 + 2)$  이 된다. 이렇게 계산된 값을 프로그램에서 독립적인 경로의 수를 정의하고 모든 문장이 적어도 한번은 실행되었음을 보장하기 위해 행해져야 할 시험회수의 상한 값을 제공해 준다. 이러한 기본경로 시험의 매력은 시험영역을 현실적

으로 최대화 시켜주고 독립경로들의 발견이 자동화될 수 있다는 것이다.

(나) 루프시험 기법

루프시험은 화이트박스 시험 기법으로서 프로그램의 루프구조에 국한해서 실시하는 기법이다. 따라서 기본경로시험과 병행사용이 가능하다. 루프시험은 다음과 같은 루프구조의 오류를 발견할 수 있다.

첫째, 초기화(initialization) 결함

둘째, 인덱싱(indexing) 및 증가(incrementing)의 결함

셋째, 루프의 경계선에서 나타나는 경계의 결함

(다) 조건시험

조건시험은 한 프로그램 모듈 안에 포함되어 있는 논리조건들을 조사하는 시험사례 설계방법이다. 이 시험의 목적은 프로그램의 조건에서 오류뿐만 아니라 프로그램의 다른 오류도 찾아내는데 있다. 즉, 한 시험전략이 한 조건에서 오류를 찾아내는데 효율적이라면, 이 시험전략은 프로그램에서 다른 오류를 찾아내는데도 역시 효율적일 것이다.

조건시험에는 여러 가지 방법이 있지만 여기서는 분기시험법, 영역시험법에 대해서 조사 및 분석하기로 한다.

첫째, 분기시험은 가장 간단한 조건시험법으로서, 한 복합조건 C에 대해 C의 참과 거짓 분기들과 C에 있는 모든 단순조건을 적어도 한 번 실행되게 하는 것이다.

둘째, 영역시험은 한 관계식을 유도하는데 세 개 또는 네 개의 시험이 요구된다. 예를 들어,

E1 <관계-연산자> E2

위와 같은 관계식이 있다고 할 때, E1의 값을 E2값보다 크거나(>), 같게(=), 작게(<) 만드는데 세 개의 시험이 요구된다. 만약에 <관계-연산자>는 부정확하고 E1과 E2가 정확하다면, 이들 세 개의 시험은 관계연산자 오류를 찾아내는 것을 보장한다. E1과 E2에서 오류를 찾아내기 위해 E1의 값을 E2값보다 크거나 작게 만드는 하나의 시험은 두 값의 차이를 가능한 한 작게 만들어야 한다. 이 방법은 논리연산자, 변수, 괄호오류들을 찾아 낼 수 있지만, n개의 변수를 가진 논리수식에 대해 모두 2<sup>n</sup>개의 시험이 요구되기 때문에 n이 작을 때만 실용적이다.

조건시험은 일반적으로 조건의 시험범위 측정이 간단하다는 장점을 가지고 있다.

(라) 자료흐름시험 기법

자료흐름시험 기법은 한 프로그램에서 변수의 정의와 사용의 위치에 따라 프로그램의 시험 경로를 선택한다. 이 기법은 중첩 if 혹은 루프문을 포함하는 프로그램의 시험경로를 선택하는데 유용하다. 또한 프로그램 안의 문장들은 변수의 정의와 사용에 따라 서로 관련이 있으므로 자료흐름시험 접근법은 오류방지에 효율적이다. 그러나 자료흐름 검사를 하기 위한 시험범위를 측정하고 시험경로들을 선택하는 문제는 조건시험을 하기 위해 시험범위를 측정하고 시험경로 등을 선택하는 문제보다 더욱 어렵고

복잡하다.

### (3) 화이트박스 시험 기법 분석

이미 기술한 동적 시험평가 기법중 화이트박스 시험 기법들에 대해 '차세대 한국형 위게임 모델' 시험 평가 적합성 여부를 분석하면 다음과 같다.

화이트박스 시험에 의한 시험사례 설계 기법은 결국 소프트웨어의 구조를 시험하는데 있어서 시험영역을 어떻게 설정할 것이냐의 문제이다. '차세대 한국형 위게임 모델'은 훈련부대 쌍방에 대해서 현재 직면한 상황과 각종 전투력 요소 등 복잡한 요소들을 고려하여 전투피해를 산출하여 승패를 결정함으로써, 상황을 판단하거나, 피해를 산출하여 승패를 결정하는 기능을 수행하는 모듈이나 서브시스템은 모든 논리구조를 검사하는 조건영역의 시험을 실시하는 것이 바람직하다고 할 수 있다. 그러나, 조건영역으로 시험평가를 하는 것은 위게임 모델이 대규모 시스템이고, 자체 논리의 복잡성으로 인해 수행 가능한 경로가 너무 많아 모두를 시험하는 것은 주어진 자원을 고려할 때 매우 어려울 것으로 판단되며, 최종 보고서에서 조사 및 분석하여 제시한 공개되거나 상용화된 기본경로 시험용 자동화도구[4] 등을 이용할 수 있다면 기본경로 시험을 실시하는 것도 바람직할 것으로 판단된다. 특히 중요 기능을 수행하는 단위모듈에 대해서는 정적으로 논리경로를 분석한 후 시험사례를 작성하여 예상출력 결과와 비교 분석함으로써 부분적으로나마 소프트웨어의 구조에 대한 시험평가의 목적 달성이 가능할 것으로 판단된다.

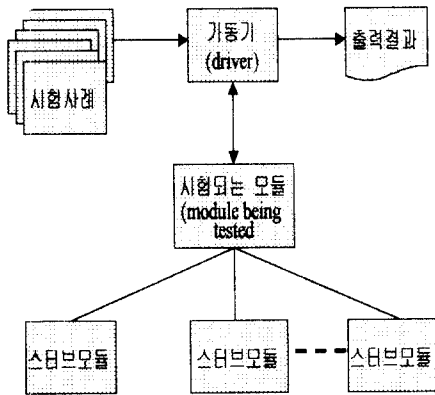
## 3.3 단위모듈 통합 기법

단위모듈의 통합은 단위시험이 끝난 모듈을 체계적으로 결합하여 설계된 소프트웨어 구조로 만들어

가는 과정으로, 여기에는 빅뱅 테스트, 단계별 테스트(상향식, 하향식)가 있다.

### 3.3.1 의사모듈(Dummy module)

단위시험이나 통합시험 모두에서 시험대상이 되는 모듈은 독자적으로 운용되는 프로그램이 아닌 시스템의 일부이기 때문에 모듈을 시험평가하기 위해서는 모듈을 가동시키는 가동기(Driver)와 타 모듈들을 흉내내는 스템(Stub) 모듈들이 필요하다. 이러한 모듈들을 개발함은 추가비용을 발생시키지만 간단한 형식만 갖추으로써 값싸게 마련할 수 있다. 가동기는 모듈을 호출(Call)하는 메인 프로그램에 불과하며 가짜 모듈들은 입출력 흉내내 내는 무기능 모듈이면 된다. < 그림 4 >는 이러한 모듈의 시험환경을 보여주고 있다.[12]



<그림 4> 모듈시험 환경

### 3.3.2 빅뱅 테스트(Big-bang test)

이 방식은 단위시험이 끝난 모듈을 모두 한꺼번에 결합하여, 완전한 시스템으로 구성하여 전체 시스템을 대상으로 시험하는 방법이다. 따라서 시험과정에서 생기는 오류를 찾기 위해서는 모든 모듈을 조사하여야 하기 때문에 어느 모듈이 원인인가를 파악하

기가 어렵고, 일단 이러한 오류가 수정된다고 하더라도 새로운 오류가 나타나고, 끝이 없이 이러한 과정이 지속됨으로 인해 혼란스럽고, 오히려 통합기간이 훨씬 많이 소요된다. 하지만 모듈구조가 잘 설계되어 인터페이스가 번잡하지 않은 소규모적인 프로그램에는 효과가 있다고 할 수 있다.

### 3.3.3 단계별 테스트(Progressive test)

빅뱅 테스트와 달리 단위시험이 끝난 모듈들을 하나씩 연결시켜 가면서 시험하는 기법으로서 오류를 격리시키고 수정하는 것이 쉽고, 인터페이스를 완전하게 시험할 수 있으며, 대규모 실시간 시스템 등에서는 단계별 테스트 기법을 이용하는 것이 바람직할 것으로 판단되나 통합시의 복잡성과 모듈통합 단계에서 많은 시간과 경비가 소요되는 단점이 있다. 단계별 테스트에는 하향식 통합과 상향식 통합이 있으며, 그 모듈들을 흉내내 주는 가짜모듈들인 스템브가 필요하다. 통합이 진행되면서 이 스템브들의 자리에는 실제 모듈들이 차례로 하나하나 들어서게 된다. 또한, 상향식 통합은 소프트웨어 계층구조의 최하위부터 점진적으로 모듈들을 통합시켜 나가는 기법이다. 스템브의 필요성은 없어졌으나 시험가동기는 역시 필요한데, 이를 드라이버라고 하며 마지막 통합단계에 이를 때까지 주 프로그램은 아니다.

### 3.3.4 단위모듈 통합기법 비교 분석

'차세대 한국형 위게임 모델'은 대단히 복잡한 대규모의 실시간 시스템이기 때문에 모델의 특성을 고려하여 단계별 테스트 기법 중 하나를 선택하는 것이 바람직하겠지만 위게임 모델 개발과 관련된 현재 상황과 여러 가지 자원을 고려할 때 주요 기능모듈 간의 결합에는 하향식 통합 및 시험평가를 실시하고 기타 모듈간의 결합에는 빅뱅 테스트 방법으로 모듈

들을 결합하면서 시험평가하는 것이 바람직할 것으로 판단된다.

부가해서 체계 시험평가 및 운용/채택 시험평가 단계에 대한 시험평가 기법 조사 및 분석은 점검표(Check list)에 의해서 기능, 성능 중심으로 거의 모든 부분을 시험평가하기 때문에 본 논문에서는 기술하지 않았고, '차세대 한국형 위게임 모델'에 적합한 체계 시험평가 및 운용 및 채택 단계 시험평가를 위한 점검표를 직접 제시하였다.[14]

## 4. 단위 및 통합모듈 시험평가

### 4.1 목적

차세대 한국형 위게임 모델 시험평가의 단위시험 및 통합시험의 목적은 사양서, 개념설계, 상세설계를 바탕으로 코딩된 단위모듈의 중요한 제어경로 및 모듈의 경계 내에 있는 오류를 발견하고, 모듈간 인터페이스와 관련된 오류를 발견하여 설계에서 지시된 프로그램 구조로 만들어 내는 것이다.

### 4.2 절차 및 범위

시험절차 및 범위는 차세대 한국형 위게임 모델을 구성하는 각각의 모듈에 대해서 시험사례 및 시험절차서와 단위 및 통합테스팅을 위한 데이터를 준비한 후 단위시험을 실시하고, 필요한 경우 재시험을 실시하여 발견된 오류를 근거로 모듈을 개조하고 시험결과를 기록 및 분석한 후, 통합시험을 단위시험과 동일한 절차로 실시한다.

### 4.3 단위시험

차세대 한국형 위게임 모델의 단위모듈에 대한 시험평가는 모듈의 기능과 구조를 시험하기 위해 정적 시험평가 기법과 동적 시험평가 기법을 병행하여 수행하지만 동적 시험평가 기법 중에서도 화이트박스 시험기법이 중심이 되어 실시한다.

따라서, 차세대 한국형 위게임 모델의 단위시험에서는 이미 기술한 시험평가 기법의 조사 및 분석에 따라 정적 시험평가 기법과 동적 시험평가 기법 중 화이트박스 시험 기법의 하나인 기본경로 시험을 실시하되, 기본경로 시험은 공개 및 상용화된 시험용 자동화도구 중 McCabe Tool set을 이용하여 실시하는 것이 경제적이고 효율적이며 시험결과에 대한 신뢰성을 가질 수 있을 것으로 판단된다.

#### 4.3.1 정적 시험평가 기법

##### 가. 내포된 루프의 정도

차세대 한국형 위게임 모델에서 단위모듈에 대한 정적시험평가지 모듈의 루프(제어)구조를 분석하여 복잡할 경우에는 일반적으로 내포된 루프가 3이하로 될 수 있도록 루프 구조를 조정하여 소프트웨어의 가독성과 실행속도의 증진, 유지보수를 용이하게 한다.

##### 나. 최적화 기법

##### (1) 루프융합(Loop fusion) 기법

루프의 범위가 같아서 한 개의 루프로 재구성할 수 있으면 한 개의 루프로 바꾸는 방법을 루프융합 기법이라고 앞서 말하였다.

##### (2) 공통 부분식의 제거 기법

프로그램 내에서 공통된 부분이 수식에 반복해서 나타나는 경우에 오직 한 번만 계산되도록 함으로써 코드를 효율적으로 만드는 방법이다.

##### (3) 대수학적 간소화 기법

수학적인 대수법칙을 응용하여 식을 간소화하는

방법을 대수학적 간소화라 한다. 교환법칙을 만족하는 연산자는 필요할 때마다 적용하여 식을 간소화할 수 있다. 예를 들면 다음과 같은 대수법칙은 코드 최적화에 유익한 기능을 제공한다. 이러한 예는 무심코 프로그램을 코딩하는 개발자의 실수에 의하여 작성될 수 있으며, 대규모의 실시간 소프트웨어에서 이러한 것이 많으면 많을수록 속도에 많은 영향을 끼치게 되므로 차세대 한국형 위게임 모델에서도 이러한 기법을 적용하여 오류가 발견된다면 조치하여야 할 것이다.

#### 4.3.2 동적 시험평가 기법

앞장에서 블랙박스 시험 기법과 화이트박스 시험 기법의 분석에서 살펴본 차세대 한국형 위게임 모델에 적용될 수 있는 기본경로 시험 기법에 대해 구체적인 시험절차와 실 예를 들어 설명하면 다음과 같다.

##### 가. 기본경로 시험

차세대 한국형 위게임 모델의 소프트웨어 구조를 시험하기 위한 시험사례 설계 기법으로 다음과 같은 절차를 거쳐 시험사례를 설계하게 되며, 이러한 기본 지침 및 절차를 참고로 하여 차세대 한국형 위게임 모델의 시험평가에 필요한 시험사례를 작성하여 시험평가를 해야 할 것이다.

첫째, 상세설계나 원시코드에 노드번호를 부여하고, 이를 기초로 논리흐름도를 그린다.

둘째, 복잡도를 계산 후 프로그램 구조에서 직선으로 독립된 경로의 수를 제시한다. 이때, 한 독립경로는 그 경로가 정의되기 전에 순회되지 않은 하나 이상의 간선을 따라 이동해야 한다.

셋째, 모든 독립경로마다 적절한 시험사례를 준비한다.

##### 나. 자동화 도구(tool)를 이용한 논리구조 시험

차세대 한국형 위게임 모델의 특성상 논리구조 시험이 필요하나, 이 경우 위의 기본경로 시험 기법과 같이 인적요소를 투입하여 시험할 수도 있으나 차세대 한국형 위게임 모델이 대규모 시스템이며 논리구조가 복잡하여 적절한 시험의 실시가 곤란하다는 것은 이미 앞에서 기술하였다.

따라서 차세대 한국형 위게임 모델에 필요한 논리구조 시험을 실시하기 위해서는 상용 및 공개된 자동화도구의 도움이 필요하게 되며, 도구를 사용하게 될 경우 필요한 기능을 제공하는 공개된 자동화도구 몇 가지를 결합하여 시험을 실시할 수도 있다. 그러나 필요한 기능을 모두 제공하면서 보다 다양한 기능을 제공하고 완전한 시험이 가능한 상용 자동화도구의 사용이 더욱 바람직할 것으로 판단된다.

## 4.4 통합시험

앞장에서 통합시험을 위한 차세대 한국형 위게임 모델의 단위모듈을 통합하는데에는 여러 가지 환경과 자원을 고려하여 빅뱅 접근방법을 기본으로 하고, 필요시 단계별 접근방법 중 하향식 통합방법으로 단위모듈 통합을 제안한다. 통합시의 절차와 시험시에 필요한 스테브 모듈의 예를 들어 설명함으로써, 차세대 한국형 위게임 모델의 시험평가에 활용할 수 있도록 하였다.

#### 4.4.1 하향식 통합 순서

하향식 통합은 다음과 같은 5단계를 거치게 되며 차세대 한국형 위게임 모델에서도 이러한 기본 5단계를 바탕으로 꼭 필요한 중요 컴포넌트에 대해서는 스테브 모듈을 작성하여 하향식으로 통합하는 과정에서 시험평가를 실시하여야 할 것이다.

첫째, 주 프로그램을 시험가동기로 사용되고, 이

모듈에 종속적인 모듈들은 스테브로 대신한다.

둘째, 통합방식(깊이 우선 혹은 넓이 우선)에 따라 하위 스테브들을 하나씩 실제 모듈로 대체한다.

셋째, 각 모듈이 통합될 때마다 시험을 실시한다.

넷째, 시험이 통과할 때마다 또 다른 스테브가 실제 모듈로 대체된다.

다섯째, 각 시험은 회귀시험을 실시하는 것이 바람직하다. 회귀시험이란 새로운 결합발생의 가능성에 대비하여 이미 실시했던 시험사례들의 전부 혹은 일부를 재실시하는 것이다.

위 과정은 둘째 단계로부터 전체 프로그램구조가 구축될 때까지 계속 반복된다.

#### 4.4.2 하향식 통합을 위한 스테브 모듈

하향식 통합은 단순한 방법 같으나 실제로는 논리적인 문제들을 유발시킨다. 문제들의 대부분은 상위 모듈들의 시험에 필요한 하위 모듈이 스테브로 대체되어 있어 요구하는 데이터를 공급해 주지 못하기 때문이다.

즉, 어떠한 중요 자료도 프로그램 구조 내에서 위쪽 방향으로 흐를 수가 없다는 것이다. 이 문제를 해결하기 위한 방법으로는 다음과 같은 두 가지 기법을 사용할 수 있으며, 개발 주관 기관에서 이것을 참고하여 시험평가를 할 수 있을 것이다.

첫째, 스테브가 실제모듈로 대체될 때까지 시험을 지연시킨다. 그러나 이 방법은 발생한 결합의 원인을 파악하지 못하게 하고 하향식의 철학을 무시하는 셈이 된다.

둘째, 스테브가 실제 모듈을 '어느 정도' 흉내내어 기능을 발휘하도록 개발하여 사용한다. 타당한 방법이지만 이와 같은 개발엔 투자가 필요하며 인적요소의 자질, 시간의 여유 등이 보장되어야 한다.

차세대 한국형 위게임 모델에서의 모듈통합 단계에서의 시험은 원칙적으로 실시하지 않는 것이 바람직하며, 제 2 단계인 체계시험에서 시험평가를 하는 것으로 대체하는 것이 바람직하다.

그러나 기본적으로 빅뱅 접근방법으로 모듈을 통합하는 가운데서 꼭 필요한 경우 혹은 여러 자원을 고려하여 중요 모듈간의 통합에는 단계별 접근방법 중 하향식 방법으로 통합하면서 시험평가를 하는 것으로 제안하였으므로 상기 둘째 방법을 선정하여 부분적으로 시험평가하는 것도 하나의 대안이라 판단된다.

## 5. 체계 시험평가 및 운용/채택 시험평가

### 5.1 체계 시험평가

#### 5.1.1 목적

모듈이 통합된 전체시스템이 사용자 요구사항, 사양서, 위험분석문서, 설계사양서 등에서 제기된 모든 요구를 하나의 시스템으로서 완벽한 기능과 성능을 발휘하는가를 여부를 시험한다.

#### 5.1.2 절차 및 범위

개발된 차세대 한국형 위게임 모델을 사용자 요구사항과 사양서를 참고하여 일치하지 않는 점을 발견하기 위하여 동적 시험평가 방법 중 동등 분할 기법, 오차 예측 기법, 경계값 분석 기법 등 블랙박스 시험 기법을 이용한 시험사례를 설계하여 시험한다. 또한 점검표에 의한 시험평가를 적용하되 이는 반드시 입력값과 기대값, 사후조치 등을 포함하는 시험사례가 갖추어야 할 양식으로 문서화하여야 할 것이며, 사용자 요구사항 및 사양서에 설정된 요구범위를 넘는 것은 시험범위에 포함하지 않아야 할 것이다.

### 5.1.3 각 분야별 점검내용

#### 가. 기능 시험

사용자 요구사항과 사양서에 기술된 기능을 실제로 만족하고 있는지 여부를 판정하는 것으로 적합한 입력에 따른 예상 출력과 실제 출력한 값을 비교하여 시험한다. 이 시험평가 제안서는 기능시험 단계의 공통 점검 사항은 별도로 작성하지 않고 각 분야의 기능 위주로 점검하도록 작성하면 된다.

#### 나. 성능 시험

성능중심 시험평가 점검표는 전체 시스템에 대한 실용성 시험, 용량 시험, 내구성 시험, 보안 및 호환성 시험을 중심으로 다음과 같은 점검표에 의해서 성능시험 공통 점검사항과 분야별 점검사항으로 구분하여 다음과 같이 시험평가를 실시한다.

각 분야별 점검 사항에 포함되지 않은 사항으로 각 분야에 공통으로 적용하여 다음과 같이 성능 시험을 하며 세부적인 점검표는 참고 문헌[4]을 참고 바란다

첫째, 사용자의 입장에서 차세대 한국형 위게임 모델의 소프트웨어가 사용하기 편리한가를 시험하는 실용성 시험이다.

둘째, 물리적 제한요건 내에서 해당 시스템의 요구성능 발휘 여부를 결정하는 시험하는 것으로 기억용량, 처리용량시험, 스트레스시험 등을 점검하는 용량시험이다.

셋째, 차세대 한국형 위게임 모델의 오류복구 능력, 신뢰성을 점검하는 기능을 수행하는 내구성 시험이다.

넷째, 차세대 한국형 위게임 모델에서 다음과 같은 보안 및 호환성 기능을 제공하는가를 시험한다.

그리고 분야별 점검사항은 지상이동, 근접전투, 지

휘, 정보, 전자, 포병, 공병, 군수 등 각 기능분야별로 체크리스트를 작성하여 시험평가를 실시하며 상세한 내용은 참고문헌[4]을 참고 바란다.

## 5.2 운용 및 채택 시험평가

### 5.2.1 개요

MIL-STD-498을 바탕으로 이 단계에서는 기술 시험평가와 운용 시험평가를 실시하는데 기술 시험평가란 체계개발단계에서 제작된 제품에 대하여 기술상의 성능으로 문서화, 확장성, 이식성, 재이용성을 측정하고 설계상의 중요한 문제점이 해결되었는가를 재확인 평가하여 기술적 개발목표가 충족되었는지를 결정하기 위해 작전 운용환경 하에서 수행하는 시험을 말한다. 기술 시험평가는 각 분야별로 시험평가할 필요성은 없으며, MIL-STD-498에 의거 전체 시스템에 모두 적용되는 공통적인 사항에 대해서 시험평가를 한다.[2][3]

운용 시험평가는 차세대 한국형 위게임 모델 시스템에 대하여 각종 작전환경 또는 이와 동등한 조건에서 작전운용성능 충족여부를 확인하고, 교리, 편성, 교육훈련, 종합군수지원요소 등에 대하여 신뢰성, 이용성, 유지보수성, 무결성, 가용성, 상호운용성 등을 평가하는 시험을 말한다.

차세대 한국형 위게임 모델은 최종적으로 사용자 요구사항을 만족하는가를 확인하기 위해서 점검표와 점검 기준을 정의하고 모든 점검 항목들이 제시한 차세대 한국형 위게임 모델 품질평가 방안의 기준(Threshold)을 통과했다면 올바르게 만들어졌다고 판단할 수 있을 것이다.[7]

본 논문에는 기술시험과 운용시험에서는 공통 점검사항 및 분야별로 시험평가하여야 할 항목들을 제

시하였다. 그러나, 공통 점검사항은 차세대 한국형 위게임 모델의 기능과 성능에 대하여 전체적으로 그리고 완벽한 이해는 여러 제한이 있으므로 부분적으로 일반적인 시험항목을 제시하였다. 필요시 개발 주관 기관에서는 이러한 일반적인 시험항목을 바탕으로 하여 좀더 구체적인 시험평가 점검표를 작성하여 시험평가할 수 있도록 하여야 할 것이다. 그러나, 본 논문에서는 지면 관계상 위게임 모델의 품질평가 방안을 기술하지 못하였으며 세부적인 내용은 참고 문헌[4]을 참고 바란다.

## 5.2.2 기술 시험

### 가. 문서화 시험평가

사용자가 시스템을 이해하고 운용과정에서 발생하는 문제점을 쉽게 해결하며, 개발자가 정비가 용이도록 하며 유지보수요원이 교체될 경우에 대비하여 문서의 정확성과 명확성을 시험한다. 문서화 시험평가에 포함되는 중요 기준요소는 문서적합성과 문서표현성 등이 있다.

### 나. 확장성 시험평가

차세대 한국형 위게임 모델의 능력이나 성능이 현 기능을 향상시키면서 증가될 수 있는 정도를 시험한다. 확장성 시험평가에 포함되는 중요 기준요소는 모듈성, 범용성, 원시코드 해독성 등을 말한다.

### 다. 이식성 및 재이용성 시험평가

이식성이란 시스템 구성품들이 임의의 소프트웨어 환경에서 다른 소프트웨어 환경으로 전이될 수 있는 정도를 말하며, 재이용성이란 시스템 구성품이 다른 특정 응용에 이용될 수 있는 정도를 말한다. 이식성 및 재이용성 시험평가에 포함되는 중요 기준요소는 모듈성, 자기기술성, 기계독립성, 응용독립성, 원시코드 해독성 등을 말한다.

## 5.2.3 운용 시험

작전 운용 환경 하에서 실시하는 운용시험은 각 분야에서 공통으로 점검하여야 하는 사항과 각 분야 별로 점검하는 사항으로 분류하여 시험평가를 한다. 다시 한 번 강조하는데 여기에서 중요한 사항은 체계시험평가에서 실시하였던 내용 중에서 중요한 사항은 작전 운용 환경 하에서 중복하여 실시할 수 있다는 사실이다.

이러한 운용시험의 목표는 첫째, 의도되지 않은 결과 발생제거, 둘째, 사용자 요구사항과 사양서에 명시된 대로 시스템이 운용되는지를 확인, 셋째, 지정된 장비 및 환경 하에서 소프트웨어의 제기능 수행여부를 확인하는데 있다.

### 가. 신뢰성 시험평가

시스템이 특정 조건동안 실제 작전 환경 하에서 명시된 조건 아래서 정확하고 일관된 결과를 제공하여 요구된 기능을 수행하는 시스템의 능력을 나타내며, 이는 과부하 상태에서의 지속적인 테스트, 비정상적인 상황/자료에 대한 시험 등과 같은 요소들을 검증함으로써 확인될 수 있다. 신뢰성 시험평가에 적용되는 중요 기준요소는 일관성, 완전성, 오류허용성, 시험적합성, 기억장치 효율성, 실행효율성, 응답 속도성 등을 말한다.

### 나. 이용성 시험평가

쉽게 배우고 사용할 수 있는 시스템의 능력으로 사용자가 훈련받은 후에 오류 없이 특정 조건 아래서 실제 작전 운용 환경 하에서 시스템을 조작 할 수 있는가에 대해 시험한다. 이용성 시험평가에 포함되는 중요 기준요소는 운용성, 교육훈련성, 의사전달성, 출력결과 적합성 등을 말한다.

### 다. 유지보수성 시험평가



소프트웨어의 유지보수성은 소프트웨어를 이해, 수정, 수용, 향상시키는 것이 얼마나 용이한가를 의미하는 것으로, 주어진 환경에서 시스템이 특정 시간 이내에 특정 조건으로 복구될 수 있을 확률에 대해 시험한다. 유지보수성 시험평가에 포함되는 중요 기준요소는 일관성, 구조적 단순성, 모듈성, 원시코드 해독성 등을 말한다.

#### 라. 무결성 시험평가

무결성은 시스템이 고장 없이 수행되고 불법적인 접근으로부터 자료를 보호할 확률에 대한 시험을 한다. 무결성 시험평가에 포함되는 중요 기준요소는 사용통제성, 사용감사성 등이 있다.

#### 마. 가용성 시험평가

가용성(Availability)은 특정 시간에 임무 수행이 요구될 때 임무 시작과 동시에 시스템이 운용 가능한 상태로 있을 확률에 대한 시험을 한다. 가용성 시험평가에 적용되는 중요 기준요소는 MTTF(Mean Time To Failure)와 MTTR(Mean Time To Repair) 등이 있다.

#### 바. 상호 운용성 시험평가

두 개 이상의 시스템이 특정 조건 아래서 상호 정보교환이 가능하고 교환된 정보를 이용할 수 있는지의 여부를 시험한다. 상호 운용성 시험평가에 포함되는 중요 기준요소는 모듈성, 통신공통성, 자료공통성, 원시코드 해독성 등을 말한다.

## 6. 결 론

군에서는 전투력 향상과 업무의 효율성을 달성하기 위하여 다양한 소프트웨어를 개발 또는 구매하여 왔으나 많은 소프트웨어가 사용자 요구사항을 충분

히 만족치 못하여 그 활용도가 낮을 뿐 아니라 환경 변화에 따른 변경가능이나 추가에도 과다한 노력이 소요되어 유지보수에 엄두를 내지 못하는 경우가 많았다. 이는 소프트웨어 확보 당시 사업화 일정에 급급하여 개발업무 자체에 엄매인 나머지 품질확보를 위한 노력을 소홀히 했기 때문이다.

또한 군사복적용 시스템의 자체 개발이 주로 미국 장비의 모방 개발이었기 때문에 개발 중의 시험평가는 단지 성능을 확인하는 수준에 머물러서 시험평가 규정이나 체도가 미약한 실정이며, 시험평가 자체가 광범위하고 주관적인 요소가 많았다.

따라서, 과학기술이 급진적으로 발전해 나가고, 군도 정교하고 복잡한 고도로 정밀한 시스템을 요구함에 따라 최첨단 과학 기술이 적용된 시스템을 개발하여야 하겠으며, 개발된 응용시스템을 시험평가하기 위한 체계를 정립하여야 할 것이다.

차세대 한국형 위게임 모델을 중심으로 군용 정보 시스템의 시험평가 전략 및 계획을 고찰함으로써, 응용 시스템의 특성에 부합되게 시험평가 전략 및 계획을 수립하고 객관적인 평가를 통하여 고품질의 시스템을 개발 가능할 것이다. 또한 각종 소프트웨어의 시험평가 방법에 대한 지침을 제공할 것이다.

## 참 고 문 헌

1. 국방과학연구소, 『군용 소프트웨어 시험평가에 관한 연구』, 1994.
2. 국방부, 『국방 정보화 관계규정』, 국방부 정보체계국, 1996.
3. 국방부훈령 제568호, 『시험평가 업무규정』, 국방부, 1997.8.16.

4. 김화수, 윤인환, 이강희, 『육군 전투지휘 훈련단 위게임 모델 시험평가 연구』, 국방대학원, 1998.
5. 송병남 외, 『프로그램 품질보증 기준에 관한 연구』, 과학기술처, 1989.12.
6. 송영재, 『C언어로 구현한 소프트웨어 엔지니어링』, 홍릉과학출판사, 1992.
7. 윤창섭 외, 『소프트웨어 프로세스와 품질』, 홍릉과학출판사, 1997.
8. 이기현 외, 『시스템 분석과 설계』, 도서출판 대림, 1996.
9. 이성기, 『소프트웨어 품질측정을 위한 소프트웨어 품질메트릭 방법론과 적용연구』, 한국군사운영분석학회 22권, 1996.12.
10. 이성기 외, 『국방표준 프로토콜 시험평가 기술 연구』, 국방정보 체계연구소, 1992.8.
11. 이원희 외 『국방전산망 표준획득 및 관리체계 연구』, 국방정보체계연구소, 1994.12.
12. 이주현, 『실용 소프트웨어 공학론』, 법영사, 1993.
13. (주)LG소프트, 삼성SDS, 『지휘소 자동화체계 소프트웨어 시험 기술서』, 1997.12.
14. 총무처 정부전자 계산소, 『소프트웨어 품질보증 과정』, 1991
15. 한국 통신 품질보증단, 『소프트웨어 품질보증업무 지침서』, 1992.
16. Republic Of Korea Department Of The Navy, 『KNTDS SYSTEM TEST PLAN』, Litton, 1994.
17. Richard A. DeMillo, 『Software Testing and Evaluation』, The Benjamin/Cummings Publishing Company, 1987.
18. Neil Storey, 『Safety-critical Computer Systems』, ADDISON\_WESLEY, 1996.