

# 소프트웨어에 대한 종합군수지원(ILS) 적용 방안

## Software ILS Proposed

심행근\*      김인중\*\*      고재영\*\*  
 Shim, HangGeun, Kim, InJung, Koh, JaeYoung

### ABSTRACT

고품질의 소프트웨어를 개발하는 사업의 중요성이 나날이 증대되고 있는 시점에서 소프트웨어를 종합군수지원 관점에서 해석하려고 하는 연구가 최근에 진행되고 있다. 즉, 소프트웨어 설계자의 입장은 소프트웨어 개발도 중요한 문제이지만 소프트웨어를 얼마나 가용성 있게 유지하는가에 더 많은 관심을 갖게 된다. 하드웨어는 개발 시부터 종합군수지원이라는 기법을 적용하여 체계 배치 시 체계 장비가 최상의 가용도를 가질 수 있도록 하는 노력이 이루어지고 있으며, 그 결과도 현실화되고 있다.

본 논문에서는 하드웨어에서만 적용하고 있는 종합군수지원 개발 기법을 소프트웨어에 적용하여 소프트웨어에 대한 종합군수지원 적용방안을 제시하고, 향후 소프트웨어 개발/배치 시에 경제적이고 가용도가 높은 소프트웨어의 개발/활용이 가능하도록 군수지원 분석을 통한 종합군수지원 활용 방안을 제안한다.

주요기술용어 : MTBF ; Mean Time Between Failure(고장간 평균시간), CSCI ; Computer Software Configuration Item(소프트웨어 형상항목), SRGM ; Software Reliability Growth Models(신뢰도 성장 모델)

### 1. 서 론

독자적인 무기/정보체계의 연구개발이 활성화되고 정보 체계의 개발 부서가 다양해짐에 따라 발생하는 군수지원 제반 난제를 해결하는 최선의 대안으로 종합군수지원분야의 필요성이 대두 되고 있다. 미래 전쟁 수행에 핵심인 전자, 정보 기술의 급속한 발전 추세에 따라 다양성, 복잡성, 고가 특성을 갖는 현대 정보체계 개발의 종합군수지원을 보장하기 위해 개발 초기 단계부터 종합군수지원 분야의 참여를 통하여 전 수명 주기를 고려한 개발 및 획득 계획과 절차를

구체적으로 수립해야 한다.

하드웨어와 달리 소프트웨어의 유지/보수는 소프트웨어 수명주기에서 필요한 비용 중에서 가장 많은 부분을 차지하고 있다. 소프트웨어의 정비는 현실적으로 매우 많은 문제점과 문제점 해결에 관계되는 비용들이 표면에 노출되지 못하고 있다. 오늘날 체계 분석과 설계에 관해서는 많은 연구가 이루어진 반면에 소프트웨어 정비에 관한 한 모든 문제점을 해소하기 위한 노력에 대한 인식은 저조하다.

소프트웨어 정비의 기본적인 문제는 기존의 소프트웨어를 변경하고자 할 때에 전체 체계에 예기하지 못한 파급효과의 발생이다. 이러한 파급효과를 통제하기 위한 관리 및 기술적인 노력이 필요하다. 이러한

\* 현대전자산업주식회사 특수사업부

\*\* 국가보안기술연구소

노력은 단순히 소프트웨어 수명주기의 정비 단계에서만 적용되는 것이 아니고 최초에 체계가 정의되고 설계되는 단계에서부터 소프트웨어의 정비도를 고려하여 적용해야 한다.

본 논문에서는 종합군수지원의 개략적인 업무인 신뢰도 분석, 정비도 분석, 군수지원 분석에 대해 알아보고 소프트웨어에서의 적용가능한 지를 분석하여 소프트웨어 종합군수지원 개발 시 적용할 수 있는 방안을 제시한다. 본 논문의 제안을 토대로 소프트웨어 체계를 개발하게 되면 체계적, 합리적, 경제적인 성과를 이룰 수 있다.

## 2. 종합군수지원 개요

### 2.1 종합군수지원이란<sup>(1)</sup>

체계의 운용/유지에 필요한 제반 군수지원 요소를 효율적이고 경제적으로 적시에 획득, 배치할 수 있도록 군수지원요소를 개발하는 업무로 정의되며, 목적은 체계에 대한 군수지원 소요를 효율적이고 경제적으로 개발, 시험, 획득, 야전 배치 시까지 종합 관리함으로써 군수지원 소요를 최소화하고 체계의 불 가동 시간을 최소화하여 운용/유지비용의 최적화와 체계 배치 시 종합군수지원 요소를 개발배치하는데 그 목표가 있다.

### 2.2 소프트웨어에 대한 종합군수지원 정의

소프트웨어의 종합군수지원이란 “체계가 개발되어 사용자에게 인도된 다음에 이루어지는 소프트웨어에 관한 변경과 수정에 필요한 제반 군수지원 요소를 효율적이고 경제적으로 적시에 획득, 배치할 수 있도록 군수지원 요소를 개발하는 것”으로 다음과 같이 구분한다.<sup>(2)</sup>

- 오류와 설계 결함을 교정

- 설계 결과를 개선
- 기존 소프트웨어를 다른 하드웨어, 소프트웨어에 사용할 수 있도록 전환
- 다른 소프트웨어와의 접속
- 파일이나 데이터베이스를 변경
- 새로운 응용분야 수렴을 위한 성능 개선
- 운영체제의 발전에 따른 대책 수립

소프트웨어의 정비는 하드웨어의 정비와는 성격이 다르다. 하드웨어 정비는 통상 고장 부품을 교체하거나 또는 기계부품을 주유/세척하는 등의 근무작업을 포함하여 결함사항을 교정하고 설계를 개선하는 공학적인 작업과정으로 이루어진다. 하드웨어의 정비 과정은 하드웨어가 정상적으로 가동되는 상태에서는 체계에 어떠한 영향을 주지 않기 때문에 대부분 사용자들은 변경이나 수정한 내용을 알 수 없다. 하지만, 소프트웨어의 정비는 결함사항을 교정하고 설계를 개선하는 과정을 포함할 뿐만 아니라 프로그래밍의 처리를 개선하는 과정도 포함된다. 대부분의 소프트웨어 정비는 프로그램의 신뢰도를 다루는 일보다는 사용자 요구사항의 변화에 의하여 시행되고 있다.<sup>(3)</sup>

소프트웨어 체계는 자료의 변경과 사용자 요구사항을 만족시키기 위하여 지속적으로 수정되어야 한다. 비록 한 시점에서 완벽하게 신뢰도를 보장하고 사용자 요구사항을 충족시켜 준다고 하더라도, 또는 그 체계가 구조적으로 잘 설계되었다 하더라도 정비 단계에서 때때로 변경과 수정을 시행할 수밖에 없다. 만일 소프트웨어 체계가 원래의 기능들이 모듈화 정형화되어 있지 않다면 이 체계의 정비를 하는데 시간과 노력, 비용이 더 많이 소모될 것이다. 즉, 소프트웨어 설계과정에서부터 정비 작업을 고려해야 한다.

정비 작업을 네 가지로 분류하면, 교정정비(corrective), 조정정비(adaptive), 예방정비(preventive), 완전정비(per-

fective)로 분류한다.

- 교정정비 : 체계의 사용 중에 오류가 발생하면 관련 소프트웨어의 진단과 오류를 교정.
- 조정정비 : 소프트웨어의 수명주기에서 발생하는 급속한 환경변화를 기존의 소프트웨어 체계에 수렴하기 위하여 수행하는 활동으로서, 기존의 소프트웨어를 조정하여 새로운 환경과 적절한 접속관계를 유지하는 활동.  
응용 소프트웨어의 수명주기는 운영체제의 수명과 좌우되며, 소프트웨어가 최초로 개발될 시점의 체계 환경은 체계를 운용하면서 운영체제, 장비 등이 새로운 형태로 변화하기 마련이므로 이를 수렴하는 것이 필요하다.
- 완전정비 : 기존의 소프트웨어가 성공적으로 그 기능을 수행할 때에 발생하는 활동으로서, 현재 수행중인 기능을 수정하여 새로운 능력을 보완하거나 사용자로부터 일반적인 개선의 요구를 받았을 때 시행.  
소프트웨어를 사용하면서 새로운 요구사항을 만족하기 위하여 수행하는 정비활동을 수정작업이라 하며 이 작업은 소프트웨어의 정비 단계에서 실시하는 모든 조력의 대부분을 차지한다.
- 예방정비 : 소프트웨어의 정비도를 향상하고 신뢰도를 개선하기 위하여 소프트웨어를 변경하는 활동. 예방정비는 소프트웨어의 정비 단계에서 실시하는 모든 노력의 대부분을 차지하고 있다.

### 2.3 소프트웨어 종합군수지원 업무

소프트웨어 종합군수지원 업무는 5가지로 분류가 가능하며, 신뢰도 분석, 정비도 분석, 군수지원 분석, 종합군수지원 요소 개발, 종합 군수지원 활용을 통해 수행된다.

- 신뢰도 분석 : 소프트웨어 체계의 신뢰도 측정 모델의 개념을 적용하여 개념설계 단계부터 양산배치 후까지 신뢰도 예측, 신뢰도 자료 수집/분석/데이터베이스 유지의 신뢰도 분석을 통해 장비의 가동률을 예측하며, 설계에 반영한다.<sup>(4)</sup>
- 정비도 분석 : 정비도 개념을 설계 초기에서부터 적용하여 소프트웨어 정비 시 복구가 얼마나 용이하며, 자동적으로 복구가 가능한지, 정비에 따른 파급효과 등을 분석하여 소프트웨어 정비에 소모되는 노력과 비용을 최소화 할 수 있는 방안을 모색한다.
- 군수지원 분석 : 군수지원분석 기법을 소프트웨어 개념에 적용하여 결합 유형 및 영향 분석, 치명도 분석, 수명주기비용 분석, 수리수준분석 등의 분석 업무를 통해 종합군수지원 개발 자료 산출한다.<sup>(5)</sup>
- 종합군수지원 요소 개발 : 국방 획득 관리 규정의 종합군수지원 업무 규정에 정의된 종합군수지원 요소인 "연구 및 설계반영, 표준화 및 호환성, 정비 지원, 지원장비, 보급지원, 인력운용, 교육훈련 및 교보재, 기술자료, 포장취급저장 및 수송, 시설, 군수관리 전산자료 지원" 요소를 개발한다.<sup>(1)</sup>
- 종합군수지원 활용 : 군수지원 분석 결과를 CALS 체계의 기본 데이터베이스인 MIL-STD-1388/2B LSA ADPS<sup>(18)</sup>에 데이터베이스화하여 군수지원 및 형상관리 데이터베이스로 활용하며, 체계와 함께 동시에 종합군수지원 요소 분석 및 개발자료를 제공하여 체계의 운용/관리 능력을 구축시킨다.

## 3. 신뢰도 분석

### 3.1 소프트웨어 신뢰도 정의

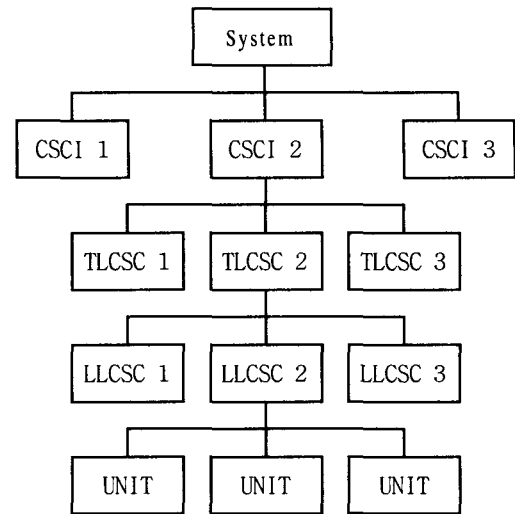
체계의 신뢰도는 추정 통계학에 의하여 특정된다. 즉 신뢰도란 "체계가 주어진 시간 동안에 명시된 환경

조건 아래서 그에 할당된 기능을 수행하는 확률이다". 만일 우리가 컴퓨터에 근거한 체계를 고려할 때 하드웨어의 신뢰도는 고장간 평균시간(MTBF; Mean Time Between Failure)으로 측정할 수 있다.<sup>(6)</sup> 소프트웨어 신뢰도는 기술적 체계의 신뢰도 정의와 거의 유사한 의미로 사용될 수 있다. 하드웨어처럼 소프트웨어 신뢰도는 오류의 발견과 수정의 비율로 나타낼 수 있다. 하드웨어와 다른 점은 경험적 오류자료를 사용하여 소프트웨어 신뢰도를 예측할 수 있는 모델을 개발할 수 있는 증거가 더 적다는 것이다. 소프트웨어의 신뢰도의 수학적 모델을 개발하려는 시도의 결과로 수많은 좋은 모델들이 제안되었다.

### 3.2 신뢰도 모델링

소프트웨어는 하드웨어의 품목처럼 형상관리가 가능하다. 이것을 하드웨어 부품처럼 형상화한 것이 소프트웨어 형상 항목(CSCI; Computer Software Configuration Item)이며, 논리적이거나, 기능적으로 분할하여 모듈화한다. 소프트웨어의 신뢰도 모델은 소프트웨어 형상항목을 기준으로 그림 1과 같은 구조를 갖게 된다. 신뢰도 모델링은 이러한 구조를 기준으로 소프트웨어 형상항목을 정의하고 각 형상항목에 대한 신뢰도를 할당하여 군수지원 분석에서 활용하게 된다.

소프트웨어 신뢰도 모델링은 모델링 과정에서 각 형상항목들의 연관 구조를 고찰하여 직렬, 병렬, 대기, 중복 구조 등의 구조로 분류하게 된다. 이러한 방법을 통해서 각 형상항목 간의 관계가 규정되고 이러한 모델링을 통해서 군수지원분석에서의 영향 분석이 이루어지게 된다. 그림 1은 각 형상항목이 직렬로 구성되었음을 의미하는 구조를 표현하고 있다.<sup>(7)</sup>



CSCI : Computer Software Configuration Item  
 TLCSC : Top Level Computer Software Component  
 LLCSC : Low Level Computer Software Component

(그림 1) 소프트웨어 구조

### 3.3 신뢰도 성장모델

소프트웨어 테스트 시간 등이 경과함에 따라 소프트웨어 내에 잠재하는 결함, 수정, 제거되어 고장의 발생 확률이 감소하여 소프트웨어 신뢰도는 증가되고, 소프트웨어 고장의 발생 시간 간격이 길어진다. 이와 같은 현상을 수학적으로 표현한 것을 소프트웨어 신뢰도 성장 모델(SRGM; Software Reliability Growth Models)이라 하며, 표 1과 같이 분류한다.<sup>(8)</sup>

#### 3.3.1 지수형 SRGM

이 모델은 Goel<sup>(9)</sup> 과 Okumoto<sup>(10)</sup>에 의해 제안된 NHPP (Non-Homogeneous Poisson Process)이다. 기본적 가정에 의해 단위 시간당에 발견된 결함수가 임의의 테스트 시각에 있어서 소프트웨어 내에 잔존하는 결함수에 비례한다는 것이다. 이 모델의 평균치 함수는 식 (1)과 같다.

[표 1] 신뢰도 성장 모델 분류

순서	모델	설명
1	시간계측모델	소프트웨어 고장 발생 시간 혹은 소프트웨어 결함 발견 시간에 기초한 확률 통계 모델 (Jelinski-Moranda모델, Schick-Wolverton모델, Little Wood-Verrall모델, Moranda모델, LittleWood 모델, Goel-Okumoto모델)
2	개수계측모델	발생한 소프트웨어 고장 수 혹은 발견된 소프트웨어 결함 수에 기초한 확률 통계 모델 (Shooman 모델, Musa모델, Moranda 모델, Musa-Okumoto모델, NHPP모델)
3	가용시간모델	소프트웨어의 시간적 흐름을 소프트웨어 고장이 발생하지 않는 상태와 소프트웨어 고장이 발생하여 소프트웨어 실행이 중단 상태에 기초한 확률 통계모델(Shooman모델, Shooman-Trivedi모델, Okumoto-Goel 모델)
4	경향곡선모델	소프트웨어 내에 잠재하는 총 결함 수를 회귀 분석에 의해 추정하는 모델 (logistic 곡선, gompertz 곡선)

$$H_e(t) \equiv m(t) = a(1 - e^{-bt})$$

$$a > 0, b > 0 \tag{1}$$

여기서,  $a$ 는 테스트 전에 소프트웨어 내에 잔존하는 총 기대 결함 수이며,  $b$ 는 잔존 결함 1개당 결함 발견율이다.

### 3.3.2 지연 S자형 SRGM

이 모델은 고장의 발생 시각으로부터 그 고장의 원인인 결함을 제거하는 시각까지의 시간적인 지연

을 고려하고 있다. 시험에 의해 결함을 발견하기 위해서는 우선 소프트웨어 고장현상을 확인하고 그 후에 원인을 규명해야 한다는 개념으로 Yamada et al이 제안한 것으로, 이 모델의 평균치 함수는<sup>(11)</sup> 식 (2)와 같다.

$$H_d(t) \equiv M(t) = a(1 - (1 + bt)e^{-bt})$$

$$a > 0, b > 0 \tag{2}$$

### 3.3.3 숙련도 S자형 SRGM

이 모델은 소프트웨어 결함이 다른 결함과 상관없이 발견 가능한 것과 하나의 결함이 발견되기 위해서는 다른 결함이 그 이전에 발견되어 제거되어야 하는 것으로 분류하고, 테스트 요원의 숙련도를 고려하여 결함을 발견하여 제거한다면 남아 있는 결함이 발견될 가능성이 높은 것으로 Ohba에<sup>(12)</sup> 의해 제안된 것으로서, 평균치 함수는 식 (3)과 같다.

$$H_f(t) \equiv I(t) = \frac{a(1 - e^{-bt})}{(1 + ce^{-bt})}$$

$$a > 0, b > 0, c > 0 \tag{3}$$

여기서,  $c$ 는 결함 발견 능력에 관한 숙련 계수이다.

## 3.4 신뢰도 분석 절차

### 3.4.1 데이터 수집 및 분석

신뢰도 데이터를 수집/분석하는 과정으로 소프트웨어 고장 발생 시간 또는 소프트웨어 결함 발생 건수를 취급할 것인가, 그리고 테스트 시간으로서 켈린더 시간, CPU 시간, 테스트 케이스 수를 취급할 것인가를 결정한다. 표 2는 분석 대상 소프트웨어의 데이터 수집 자료의 예이다.

[표 2] 신뢰도 데이터 수집 예

Tt	Fc	Fcc	Tt	Fc	Fcc	Tt	Fc	Fcc
1	1	1	14	333	2,047	27	58	4,736
2	16	17	15	319	2,366	28	65	4,801
3	2	19	16	362	2,728	29	51	4,852
4	11	30	17	369	3,097	30	81	4,933
5	20	50	18	353	3,450	31	66	4,999
6	94	144	19	288	3,738	32	60	5,059
7	122	266	20	160	3,898	33	63	5,122
8	179	445	21	120	4,018	34	67	5,189
9	211	656	22	242	4,260	35	66	5,255
10	222	878	23	199	4,459	36	51	5,306
11	344	1,222	24	88	4,547	37	27	5,333
12	248	1,470	25	77	4,624	38	15	5,348
13	244	1,714	26	54	4,678			

Tt:시험시간 (일), Fc:고장발견 건수, Fcc:고장발견 누적건수

### 3.4.2 신뢰도 모델 선정

적절한 NHPP 모델을 선정하는 단계로 이때 테스트 환경이 신뢰도에 미치는 요인을 충분히 고려해야 한다. 일반적으로 모델의 선택 기준 혹은 비교 기준을 예측 타당성, 유용성, 가정의 특성, 적용성, 간결성 등을 고려해야 한다.<sup>(13)</sup>

### 3.4.3 파라미터 값 추정

우선 NHPP 모델의 파라미터 값을 최우 추정법에 의해 추정하기 위하여 일정한 테스트 시간  $t_k$ 까지 발견된 총 결함수가  $y_k$ 에 관한  $n$ 쌍의 측정 데이터  $(t_k, y_k)$  ( $k=1, 2, \dots, n$ ) 가 관측되는 것으로 한다. 이때 이 데이터에 대한 평균치 함수  $H(t)$ 를 가지는 NHPP 모델의 우도 함수는

$$L = P_r[N(t_1) = y_1, N(t_2) = y_2, \dots, N(t_n) = y_n] = \exp\left[\frac{-H(t_n) \prod_{k=1}^{n-1} (H(t_k) - H(t_{k-1}))}{H(y_k - y_{k-1})!}\right]_{y_k - y_{k-1}} \quad (4)$$

에 의해서 주어진다. 여기서  $t_0 = 0, y_0 = 0$ 으로 한다. 식 (4)의 양변에 자연 대수를 취할 때

$$\ln L = \sum (y_k - y_{k-1}) \ln [H(t_k) - H(t_{k-1})] - \sum \ln [(y_k - y_{k-1})!] \quad (5)$$

로 된다. 식 (1), 식 (2), 식 (3)의 평균치 함수를 대입하고, 각 모델의 파라미터에 대해 편 미분하여 이들을 수학적으로 해결하면 각 모델의 파라미터 추정치를 구할 수 있다.

위의 식 (5)와 표 2를 이용하여 각 모델의 파라미터 값의 최우 추정치를 구하면 표 3과 같다.

[표 3] 각 모델의 파라미터 값의 예

구분	지수형	지연 S자형	숙련도 S자형	비 고
a	10844.6	6132.15	6083.09	
b	0.01788	0.09415	0.06623	고장율( $\lambda$ )
C			0.56545	

### 3.4.4 평균치 함수 $H(t)$ 추정

선택된 모델의 평균치 함수로 표현된 식 (1), 식 (2), 식 (3)에 모델의 파라미터 추정치  $a, b, c$ 를 대입하면, 각 평균치 함수의 추정치  $H_e(t), H_d(t), H_i(t)$ 를 구할 수 있다.

표 3에 나타난 각 모델의 파라미터 추정 값을 대입하면 평균치 함수는 식 (6-8)과 같다.

$$H_e(t) = 10844.6(1 - e^{-0.01788t}) \quad (6)$$

$$H_d(t) = 6132.15(1 - (1 + 0.09415t)e^{-0.09415t}) \quad (7)$$

$$H_i(t) = 6083.09 \left[ \frac{1 - e^{-0.06623t}}{1 + 0.56545e^{-0.06623t}} \right] \quad (8)$$

3.4.5 신뢰도 평가 척도 추정

기대 잔존 결함 수  $n(t)$ , 소프트웨어 신뢰도  $R(x|t)$ , 순간 MTBF 또는 누적 MTBF 등의 소프트웨어 신뢰도 평가 척도를 추정 및 예측한다. 그리고 과거의 경험이나 실적을 참조하여 소프트웨어 신뢰도 달성 비율과 테스트 진척 현황을 판단하여 소프트웨어 릴리즈 시기 예측이나 프로젝트 관리에 반영한다.

3.4.6 신뢰도 할당

앞의 표 3에서 계산된 고장율( $\lambda$ )은 전체 체계의 고장율( $\lambda; 0.01788$ )을 나타낸다. 체계의 고장율을 식 (9)를 이용하여 계산하면 MTBF(고장간 평균시간; Mean Time Between Failure)를 구할 수 있다. (1일 10시간 사용 기준)

$$MTBF = 1/\lambda \times 10\text{시간} \quad (9)$$

$$\approx 560 \text{ 시간}$$

식 (9)와 그림 1의 소프트웨어 구조를 이용하여 하위 형상 항목으로 신뢰도를 할당할 수 있다. 그 예는 표 4와 같다.

[표 4] 신뢰도 할당의 예

순위	형상항목	할당율(%)	고장률	MTBF	비고
1	CSCI1	25%	0.000445	2,247	
2	CSCI2	50%	0.00089	1,124	
2-1	TLCSC1	25%	0.0002225	4,494	
2-2	TLCSC2	50%	0.000445	2,247	
2-2-1	LLCSC1	30%	0.0001335	7,491	
2-2-2	LLCSC2	40%	0.000178	5,618	
2-2-3	LLCSC3	30%	0.0001335	7,491	
2-3	TLCSC3	25%	0.0002225	4,494	
3	CSCI3	25%	0.000445	2,247	

4. 군수지원 분석

군수지원 분석 절차는 대상품목 선정, 관리번호 부여, 계통도 작성, 기능그룹부호 할당, 고장 유형/영향 및 치명도 분석 등의 업무가 있으나, 여기서는 MIL-STD-1629A의 고장 유형/영향 및 치명도(FMECA) 분석 방법에 대해서 설명한다.<sup>(5,14)</sup>

4.1 고장 유형 및 영향 분석(FMEA)

분석 대상품목이 가질 수 있는 모든 고장 유형을 식별하고 식별된 고장 유형에 대한 원인과 고장이 체계운용에 미치는 영향을 분석하여 정비 기능별 소요 업무를 식별하는데 기초 자료로 활용한다. 세부적인 적용 방법은 표 5와 같으며, 설명은 아래와 같다.

[표 5] 고장 유형 및 영향분석 SHEET

장비명 :		작성일 :		작성자 :				
단계(1) :		참조자료(2) :		임무(3) :				
(4) L C N	(5) 품명	(6) 기능	(7) 고장 유형 및 원인	(8) 고장 및 손상 영향	(9) 고장 탐지 방법	(10) ) 기본 복구 활동	(11) 위험 도 부호	(12) 비 고

- 1) 단계 : 계통상의 레벨을 기술한다.
- 2) 참조 도면(자료) : 분석대상품목의 도면(자료) 번호를 기술한다.
- 3) 임무 : 분석대상품목이 어떤 임무를 수행 중일 때 고장을 일으키는가의 예상되는 임무를 기록한다.

- 4) 군수지원분석 관리번호(LCN) : 분석대상품목의 군수지원분석 관리번호를 기록.
- 5) 품명 : 분석대상품목의 품목 명칭을 기록한다.
- 6) 기능 : 분석대상품목이 체계에서 수행되고 있는 본래의 기능을 기술한다.
- 7) 고장 유형 및 원인 : 고장이 발생하는 형태로 분석되는 각 소프트웨어 모듈의 모든 예측 가능한 고장이 식별 기술되어야 한다.
- 8) 고장 및 손상 영향 : 분석대상품목의 운용, 기능, 상태에 대한 각 고장/손상 유형의 결과들을 식별하여 기술한다.
- 9) 고장 탐지 방법 : 고장 유형의 발생이 운용/정비요원에 의해 탐지되는 방법으로 적용할 수 있는 경보장치와 운용/요원이 체계의 고장이라고 판단을 갖도록 하는 다른 징후들을 기술한다.
- 10) 기본 복구 활동 : 대상품목에 고장이 발생해서 기능을 상실하였을 때 탐지된 고장을 복구하는 기본적인 정비활동을 기술한다.
- 11) 위험도 부호 : 분석대상품목이 고장을 일으켰을 때 체계가 필수 임무를 수행하는데 어느 정도 치명적인 영향을 미치는가를 분석하여 기술한다.(1:재난, 2:치명, 3:보통, 4:경미)
- 12) 비고 : 설계 개선사항, 기타 필요한 추가사항 등을 기록한다.

#### 4.2 치명도 분석(CA)

고장 유형 및 영향분석에서 식별된 고장 유형 및 원인에 대한 상대적인 치명도 값을 산출하여, 상대적으로 위험도가 높은 항목을 선정, 설계에 반영하고 고장 복구 절차를 개발하는데 기초자료로 활용한다. 세부적인 적용 예는 표 6과 같으며, 설명은 아래와 같다.

[표 6] 치명도 분석 SHEET

장비 명 :		작성일 :			작성자 :					
단계(1) :		참조자료(2) :			임무(3) :					
(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)
L	품명	기능	고장 유형 및 원인	위험도 부호	고장 영향 확률	고장 유형 비	고장 율	운용 시간	고장 유형별 치명도	비고
C										
N										

- 1) 단계 : 계통상의 레벨을 기술한다.
- 2) 참조 도면(자료) : 분석대상품목의 도면(자료) 번호를 기술한다.
- 3) 임무 : 분석대상품목이 어떤 임무를 수행 중일 때 고장을 일으키는가의 예상되는 임무를 기록한다.
- 4) 군수지원분석 관리번호(LCN): 분석대상품목의 군수지원분석 관리번호를 기록한다.
- 5) 품명 : 분석대상품목의 품목 명칭을 기록한다.
- 6) 기능 : 분석대상품목이 체계에서 수행되고 있는 본래의 기능을 기술한다.
- 7) 고장 유형 및 원인 : 고장이 발생하는 형태로 분석되는 각 소프트웨어 모듈의 모든 예측 가능한 고장이 식별 기술되어야 한다.
- 8) 위험도 부호 : 분석대상품목이 고장을 일으켰을 때 체계가 필수 임무를 수행하는데 어느 정도 치명적인 영향을 미치는가를 분석하여 기술한다.(1:재난, 2: 치명, 3:보통, 4:경미)
- 9) 고장 영향 확률( $\beta$ ) : 고장 발생 시 고장 발생의 영향으로 인하여 할당된 위험도 부호에 해당하는 결과를 일으킬 조건부 확률로, 분석 자 판단에 의해 산정 한다.(100%: 상실, 10%~100%:상실 예상, 0%~10%:상실가능, 0%:무 영향)



- 10) 유형 비( $\alpha$ ) : 분석대상품목의 고장 유형에 대한 부품 모듈 고장율의 비율을 나타내며, 고장 유형 비는 품목 또는 하부 모듈의 식별된 고장 유형으로 고장이 발생하는 비율을 소수로 표시한다.
- 11) 고장 율( $\lambda$ ) : 분석대상품목이 특정기간(시간) 동안 발생한 고장 회수를 총 운용기간으로 나눈 값을 기입한다.
- 12) 운용시간( $t$ ) : 체계의 연간 운용소요시간을 기입한다.
- 13) 고장 유형별 치명도( $c_m$ ) : 고장 유형별 치명도 값은 품목 치명도 값의 일부로써 특정 위험도 분류하의 분석대상품목의 고장 유형 중 특정 고장 유형에 대한 치명도 값이다.

분석대상품목의 각 고장 유형 및 원인에 대한 치명도 값( $c_m$ )은 식 (10)과 같다.

$$c_m = 1,000,000(\beta \times \alpha \times \lambda \times t) \quad (10)$$

여기서,  $c_m$  : 고장 유형별 치명도 값  
 $\beta$  : 고장 영향 확률  
 $\alpha$  : 고장 유형 비  
 $\lambda$  : 고장 율  
 $t$  : 운용시간

### 5. 소프트웨어에 대한 ILS 적용 방안

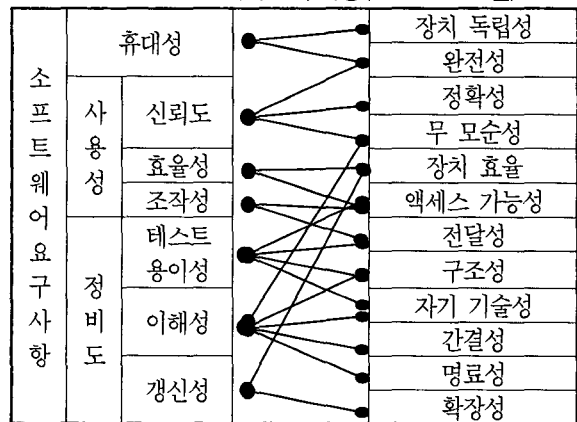
소프트웨어에 대한 종합군수지원은 하드웨어에 대한 종합군수지원과는 상이하다. 그러나 종합군수지원의 많은 개념이 소프트웨어에도 적용된다. 종합군수지원의 11개 항목인 "연구 및 설계반영, 표준화 및 호환성, 정비지원, 지원장비, 보급지원, 인력운용, 교육훈련 및 교보재, 기술자료, 포장취급저장 및 수송, 시

설, 군수관리 전산자료 지원"의 항목이 적용된다. 소프트웨어 체계가 지원 요구조건을 만족시키기 위해서는 각 종합군수지원 요소들은 소프트웨어체계 개발과정에서 분석 적용되어야 한다.

### 5.1 연구 및 설계반영

소프트웨어체계 종합군수지원 연구를 위하여 관련 체계의 경험 제원에 의한 종합군수지원 자료를 수집하며, 필요 시 국외 업체를 방문하여 체계별 종합군수지원 소요를 사전에 연구한다. 설계반영은 소프트웨어체계의 소요 기획, 연구개발, 시험평가, 야전배치 후 운영 능력 평가에 이르기까지 지속적으로 소프트웨어체계의 종합군수지원 요소 및 요구사항을 설계에 반영한다. 특히 설계반영 시에는 표 7과 같은 항목에 대해서 중점적으로 검토한다.<sup>(15)</sup>

(표 7) 소프트웨어 요구사항(Boehm 모델)



### 5.2 표준화 및 호환성

소프트웨어체계 개발 시 타 체계와 최대한 공통성을 유지하도록 하여 체계간의 군수지원이 용이하게 군수지원 소요를 단순화한다.

- 1) 유사 운용체제와 호환/공유 가능분야를 판단하여 표준화/호환성 유지 및 모듈화.
- 2) 여러 종류의 운영체제에서도 실행이 가능하도록 프로그래밍(Windows, UNIX, Linux)
- 3) 각 정비 계단에 대한 종합군수지원 요소 개발 시 표준화 및 호환성 반영(표준 소프트웨어 사용 및 표준 소모재 사용)
- 4) 소프트웨어 형상 항목 정의 및 개발 절차 표준화

### 5.3 정비지원

소프트웨어체제의 정비지원이 용이하도록 필요한 지원요소를 분석/개발한다.

- 1) 고장 정의 : 어떤 체계가 최초 설계에서 의도된 기능을 수행할 수 없을 때 고장이라고 하며<sup>7)</sup>, 소프트웨어 고장은 “소프트웨어 실행 불가, 소프트웨어 메뉴 항목의 실행 불가, 관련 소프트웨어와 연동 불가 등”으로 정의하고 이에 따른 정비 방안을 수립한다.
- 2) 정비 지원 수준 : 소프트웨어의 특성상 정비 수준을 크게 부대정비, 야전정비, 창정비로 구분할 수 있으며, 정비 지원개념은 표 8과 같다.
- 3) 정비 업무량 추정 및 분석을 통해 정비 시 예측, 정비 업무량을 예측한다.
- 4) 소프트웨어 기술인원의 주특기 소요 및 주특기별 기술수준을 정의한다. 현재 전자계산기 운용 주특기(3114)가 반영되어 있으며, 소프트웨어 정비 또는 프로그래머 주특기가 추가로 신설되어야 할 것으로 판단된다.
- 5) 복구용 예비소프트웨어를 준비하여 고장(에러) 시 즉각 교체가 가능하도록 하는 절차가 필요하다.
- 6) 소프트웨어의 재 설치 및 예방정비를 위한 자료의 백업 및 복구 방안, 소프트웨어 점검 방안을 수립한다.

[표 8] 소프트웨어 정비 지원수준

정비 계단	정비형태	정비 내용
부대 정비	조정정비	임무 변수 입력 및 검증
	교정정비	새로 보급된 소프트웨어 설치
	완전정비	해당 없음
야전 정비	조정정비	알려진 모든 임무 변수를 처리하는 패키지 준비
	교정정비	관리자의 결정에 따라 000라인 이하의 변경을 수반하는 설계, 논리, 코딩 오류의 정정
	완전정비	관리자의 결정에 따라 XXX라인 이하의 변경을 수반하는 성능 향상 반영
창 정비	조정정비	신규/다른 업무를 처리하도록 요구되어 지는 곳에 새로운 조정 코드의 산출
	교정정비	야전 정비에서 처리하지 못하는 모든 오류의 지원
	완전정비	야전 정비에서 구현하지 못하는 다른 모든 향상의 설계, 개발, 구현

- 7) 소프트웨어 고장 진단 및 복구 시 도움말 기능과 하이퍼텍스트 기능을 이용하여 복구가 용이하도록 지원하는 방안을 강구한다.

### 5.4 지원장비

지원장비는 소프트웨어의 운용, 유지에 필요한 운용지원 소프트웨어 및 부수 장비이다.

- 1) 소프트웨어 고장 진단 및 복구 시 도움말 기능과 하이퍼텍스트 기능을 이용하여 복구가 용이하도록 지원하는 방안을 강구한다.
- 2) 소프트웨어 운용 유지에 필요한 추가 소프트웨어 획득 방안 검토/수립하며, 각 정비 계단에서 기초적으로 필요한 소프트웨어 지원도구의 예는 표 9와 같다.

- 3) 재 설치 및 백업/복구에 필요한 소모품(디스크, 백업용 테이프 등)을 검토한다.
- 4) 파일서버 체계를 구축하여 수시로 소프트웨어 관련 뉴스, 갱신 소프트웨어를 제공할 수 있는 방안을 강구한다.

(표 9) 소프트웨어 지원장비

소프트웨어 도구	부대	야전	창
Text Editor	X	O	O
Compiler	X	O	O
Assemblers	X	O	O
S/W Library Manager	X	O	O
Link Editor	X	O	O
Program Analysis Aids	X	X	O
Document Generators	X	O	O
Soft Copy Read	O	O	O
Soft Copy Built	X	O	O
CASE Tool	X	X	O

### 5.5 보급지원

보급지원은 체계와 동시에 획득 보급되어야 할 초도 보급소요와 운영유지를 위한 물자, 재원 등을 포함한다.

- 1) 군수지원분석을 통해 전산 소모품 등의 보급품목, 방법 등을 결정한다.
- 2) 서버를 이용한 그룹웨어 또는 웹 서버를 통하여 적용 가능한 게시판을 활용하여 네트워크를 통해 체계의 변경사항을 사용자에게 제공하고 필요한 소프트웨어도 지속적으로 제공하는 방안을 강구한다.
- 3) 뉴스 메일 기능을 활용하여 네트워크 상에서 사용자가 등록되면 등록된 사용자에게 개선된 소프트웨어를 제공하는 방안을 검토한다.

### 5.6 인력 운용

군수지원분석을 통해 인원 및 주특기 소요를 검토하고 기존 인원/주특기의 현황과 비교하여 추가 인원 및 주특기 소요를 판단한다.

- 1) 운용 및 정비 인원 판단
- 2) 소요 주특기 및 추가 소요 주특기 판단

### 5.7 교육훈련 및 교보재

운용 및 정비요원에 대한 교육훈련은 새로운 체계를 운용하기 위한 초도 배치 전 교육훈련과 배치 후 손실인력을 충당하기 위한 양성교육으로 이루어진다.

- 1) 교육과정, 일정, 방법, 소요인원, 교보재 소요를 포함한 교육 계획을 수립한다.
- 2) 교육을 실시하며, 교육결과를 보고한다.
- 3) 네트워크 상에서 활용 가능한 교육용 CBT (Computer Based Training) 프로그램을 개발한다.

### 5.8 기술자료

종합군수지원 업무에 필요한 제반 기술문서와 자료를 말하며, 사용자교범, 기술교범, 기술자료 묶음(TDP) 등이 포함된다.

- 1) 군수지원분석 결과를 적용하여 기술교범(사용자; 정비, 보급)을 개발한다. 기술교범은 사용자가 이해하기 쉽고 체계 보수 및 개조가 수월하게 개발되어야 한다.
- 2) CD-ROM 형태 또는 네트워크 상에서 활용이 가능한 형태의 IETM(대화형 전자 교범)을 개발한다.

### 5.9 포장 취급 저장 및 수송

야전에 지원되는 수리부속에 대해 취급, 저장, 수송 시 손상을 최소화하기 위하여 포장, 취급, 저장 재원 표를 개발한다. 특히 저장 매체의 민감도가 충분히 고려되어야 한다.

- 1) 군수지원분석 결과를 근거로 대상품목을 선정하고, 재원 산출 및 재원 표를 개발한다.(소프트웨어 패키지 및 소모품)
- 2) 필요 시 포장 시제를 개발한다.
- 3) 네트워크 통신을 이용하여 소프트웨어를 전송할 때를 예상하여 통신 신뢰도 및 보안성을 감안한 소프트웨어적인 포장/저장/수송 대책이 필요하다.

### 5.10 시설

시설은 체계의 운용, 정비, 교육을 지원하는데 필요한 공간과 관련 설비를 파악하고 필요 시 시설의 추가/개조를 검토한다. 특히 재해를 대비한 백업 및 안전 시설의 필요성을 검토한다.

### 5.11 군수관리 전산자료 지원

체계의 수명 주기에 관련되는 부서 및 기관의 관리자에게 의사 결정에 필요한 관리정보를 제공할 수 있도록 전산장비의 확보, 제반 프로그램 개발 및 그 장비의 운영인력 등에 대한 각종 문서 등을 지원하는 활동이다.

- 1) 신뢰도 분석 자료, 군수지원분석 자료 등의 자료 제공
- 2) 컴퓨터 자원 종합 지원 문서(CRISD; Computer Resource Integrated Support Documentation), 종합군수

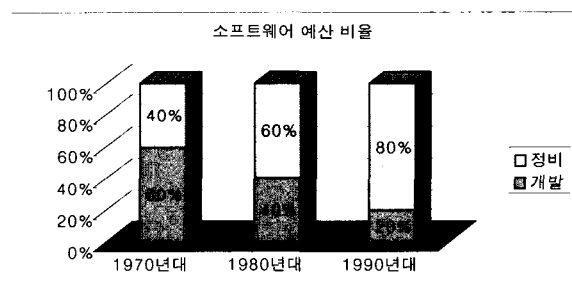
### 지원계획서 제공

## 6. 결 론

현대 무기체계에서 소프트웨어가 차지하는 비중은 점점 더 커지고 있는 실정이다. 또한 독립적인 정보체계의 소요도 더욱 확대되고 있는 실정이다. 소프트웨어는 개발하는 시점에서는 성능 구현에 주안점을 두고 개발하지만 실제 운용 과정에서 새로운 기능의 추가, 새로운 환경에 따른 소프트웨어 변경, 에러 발생 등의 여러 가지 문제가 발생되어 개발에 소요되는 비용보다 정비에 소요되는 비용이 더 많이 소요되는 실정이다. 개발비용과 정비 비용의 추세를 보면 그림 2와 같은 추세를 나타내고 있다.<sup>(4)</sup> 또한 정비 과정에서 예상치 못한 파급 효과가 발생하여 중대한 문제로 발전할 가능성도 잠재하고 있는 실정이다.

본 논문에서는 이러한 문제점을 방지하고 소요비용(인원 소요)을 최소화 할 수 있는 대안으로 소프트웨어의 종합군수지원 방안을 제시하였다. 종합군수지원 초기 개발 과정에서부터 신뢰도, 정비도, 군수 지원성을 충분히 고려하여 개발되므로 배치후의 문제점을 대부분 해결할 수 있는 방안으로 활용될 수 있다.

또한 신뢰도 분석, 정비도 분석, 군수지원분석 등의 분석 기법은 대부분 무기체계에서 다루었던 기법이지만 소프트웨어 체계에 적용할 수 있다는 것을 제시하



(그림 2) 정비 비용

였다. 이러한 분석을 바탕으로 종합군수지원을 적용하면 향후 소프트웨어 체계의 배치 후에 발생하는 많은 문제점들이 개선되는 효과를 가져올 수 있을 것으로 기대 된다.

일부 내용은 개략적인 면도 있지만 새로운 소프트웨어 체계의 개발에 적용하면 효과적인 결과를 얻을 수 있을 것으로 기대된다.

향후 소프트웨어의 종합군수지원 개발 산출물을 기초로 초고속 정비지원, 보급지원, 훈련, 교육, 기술자료(IETM) 활용 방안 등을 국방 전산망에 활용할 수 있도록 연구하여 국방 CALS와 연동될 수 있는 방안이 연구 과제이다.

### 참 고 문 헌

1. 국방부, 국방획득 관리 규정, 국방부 훈령 제610호, 1999. 1. 2.
2. McCracken, D., "Software in the 80s-Perils and Promises." Vol.14, No. 38, 8.17, 1980.
3. Boehm, B "Software engineering R&D Trend and Defense needs." In Research Directions in the Software Technology, p. Wegner, (ed.), MIT Press 1979.
4. 이철희 외 2인, 소프트웨어 공학, 홍릉 과학 출판사, 1998. 8.
5. 육군 교육사령부, 군수지원 실무지침서, 교육사령부 발간 물, 1998.10.
6. 박경수, 신뢰도 공학, 탑 출판사, 1996.
7. James V. Jones, Integrated Logistics Support Handbook, McGraw-Hill, 1998.1.
8. 山田茂, "ソフトウェアの信頼性評価法", "ソフトウェア.センター ; 東京, 1985.
9. A. L. Goel "Software Error -Detection Model with Applications", J. System and Software Vol. 1, No. 3, 1980.
10. A. L. Goel and K, Okumoto "Time -Defendent Error-Detection Rate Model for Software Reliability and other Performance Measures", IEEE Trans. Reliability Vol. R-28, No. 3, 1989.
11. S. Yamada, M Ohba, S. Osaki, "S-Shaped Reliability Growth Modeling for Software Error Detection" IEEE Trans. Reliability Vol. R-32, No. 5, 1983.
12. M. Ohba, "Inflection S-Shaped Software Reliability Growth Model" in Stochastic in Reliability Theory, S. Osaki and Y.Hatoyama(ed.) Springer-Verlag, 1984.
13. J. D. Musa, A.Iannino, and K.Okumoto "Software Reliability : Measurement, Prediction, Application", McGraw-Hill, 1987.
14. MIL-STD-1629A, Failure Mode Effects and Criticality Analysis.
15. 송 영재, 소프트웨어 엔지니어링, 홍릉과학출판사, 1992. 3.
16. MIL-STD-756B, Reliability Modeling and Prediction.
17. 이 경환 외 6인, 소프트웨어 공학. 청문각, 1994.8.
18. MIL-STD-1388/2B, DoD Requirements for a Logistics Support Analysis Record, 1991.
19. 김 인중, 심 행근, "보안 소프트웨어 종합군수지원 (ILS)에 관한 연구", 제 11회 정보보호와 암호에 관한 학술대회, WISC'99 논문집, 1999.