

정보시스템 개발전략에 따른 사업관리 방안

심승배

한국국방연구원 정보화연구센터

The Method of Program Management for Information System Development Strategy

Sim, Seung Bae

KIDA (Korea Institute for Defense Analyses)

E-mail: sbsim@kida.re.kr

요 약

정보시스템의 개발전략에는 대표적으로 총괄적, 점증적, 진화적 개발전략이 있다. 현재 규모가 큰 사업의 경우, 점증적 및 진화적 개발전략을 사용하여 정보시스템을 개발하고 있지만, 개발전략을 적용하기 위한 세부적인 지침이 너무 일반적이거나 부재한 것이 현실이다. 또한, 개발전략에 따른 개발 수명주기 단계별 산출물 관리 체계의 효율성 저하로 사업관리에 어려움이 있었다. 따라서 효율적인 정보시스템 사업추진 및 관리를 위해서 개발전략에 대한 세부적인 지침과 개발 산출물에 대한 관리 체계의 필요성이 제기되고 있다. 본 논문에서는 점증적 및 진화적 개발전략에 대한 일반 지침을 제시하였고, 개발전략에 따른 개발 수명주기 단계 산출물을 효율적으로 관리하는 체계를 제안하였다.

1. 서론

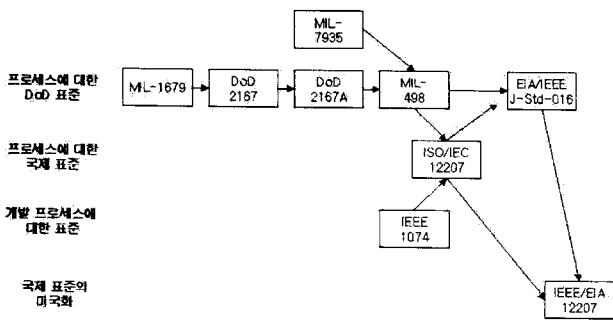
정보시스템을 개발하기 위한 전략에는 대표적으로 총괄적(Waterfall), 점증적(Incremental), 진화적(Evolutionary) 개발 전략이 있다. 총괄적 개발 전략은 기본적으로 “처음부터 끝까지 단 한번에, 각각의 단계는 단 한번만 수행(once-through, do-each-step-once)하는” 전략이다. 즉, 사용자의 요구를 결정하고, 요구사항을 정의하며, 시스템을 설계하고, 시스템을 구현하며, 테스트하고, 수정하고 납품하는 것이다. 반면에, 점증적

(Incremental) 개발전략은 사용자의 요구를 결정하여 시스템의 요구사항을 정의하고, 그 이후에 일련의 빌드를 통하여 개발의 나머지 부분을 수행한다. 첫 번째 빌드는 계획된 시스템 능력의 일부만을 구현하며, 시스템이 완전해 질 때까지 다음의 빌드에 계속적으로 추가적인 능력을 더해 간다. 그리고, 진화적(Evolutionary) 개발전략은 점증적 개발전략과 마찬가지로 빌드를 통하여 시스템을 개발하지만, 모든 요구사항을 처음부터 정의할 수 없다는 점이 점증적 개발전략과 다르다. 진화적 개발전략에서는 사용자 소요과 시스템 요구

사항이 처음에는 부분적으로만 정의되며, 후속적인 빌드가 진행됨에 따라 자세하게 정의된다. [1, 2, 3, 4, 5]

현재 국방정보시스템과 같이 규모가 큰 사업의 경우, 점증적·진화적 개발전략을 사용하여 정보시스템을 개발하고 있지만, 개발전략을 적용하기 위한 세부적인 지침이 너무 일반적이거나 부재한 것이 현실이다. [1, 2, 3]

일반적으로 공공 및 민간의 대규모 사업에서 정보시스템 개발 프로세스를 효율적으로 관리하기 위해서 소프트웨어 개발 수명주기 모델을 사용하고 있다. 소프트웨어 개발 수명주기 프로세스에 대한 국제 표준에는 대표적으로 MIL-STD-498, ISO/IEC 12207, IEEE/EIA 12207 등이 있다. 현재 가장 널리 쓰이고 있는 표준은 미 국방부에서 개발한 MIL-STD-498이며, 점차 IEEE/EIA 12207을 표준으로 채택하거나 활용하고 있다. <그림 1>은 소프트웨어 수명주기 표준에 대한 동향을 나타낸다. [4, 5, 6, 7, 8]



<그림 1> 소프트웨어 수명주기 표준 동향

현재 많은 공공 및 민간 사업에서 앞에서 언급한 소프트웨어 수명주기 표준을 사용하여 정보시스템 사업관리를 수행하고 있지만, 개발전략을 적용하기 위한 세부적인 지침이 너무 일반적이거나 부재한 것이 현실이다. 또한, 개발전략에 따른 개발 수명주기 단계별 산출물 관리 체계의 효율성 저하로 사업관리에 어려움이 있었다. 따라서 효율

적인 정보시스템 사업추진 및 관리를 위해서 개발 전략에 대한 세부적인 지침과 개발 산출물에 대한 관리 체계의 필요성이 제기되고 있다.

본 논문의 나머지 구성은 다음과 같다. 2 장에서는 MIL-STD-498, IEEE/EIA 12207 에 제시된 정보시스템 개발전략을 설명하고, 3 장에서는 점증적, 진화적 개발전략에 대한 일반적인 적용 지침을, 4 장에서는 개발전략에 따른 개발 수명주기 단계 산출물을 효율적으로 관리하는 체계를 제안하였다.

2. 정보시스템 개발전략

일반적으로 정보시스템 사업관리자는 다음과 같은 사업 특성에 따라 개발전략을 결정한다.

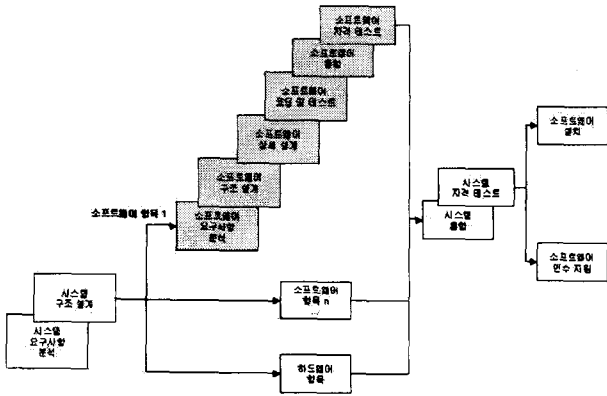
- 모든 요구사항을 초기에 정의하는가?
- 복수의 개발 주기를 사용하는가?
- 시스템 개발의 중간 단계에서 소프트웨어를 사용하는가?

2.1 총괄적 개발전략

총괄적 개발전략은 기본적으로 “처음부터 끝까지 단 한번에, 각각의 단계는 단 한번만 수행 (once-through, do-each-step-once) 하는” 전략이다. 즉, 사용자의 수요를 결정하고, 요구사항을 정의하며, 시스템을 설계하고, 시스템을 구현하며, 테스트하고, 수정하고 납품한다. 총괄적 개발전략의 세부 수명주기 단계는 시스템 요구사항 분석, 시스템 구조 설계, 소프트웨어 요구사항 분석, 소프트웨어 구조 설계, 소프트웨어 상세 설계, 소프트웨어 코딩 및 테스트, 소프트웨어 통합, 소프트웨어 자격 테스트, 시스템 통합, 시스템 자격 테스트, 소프트웨어 설치, 소프트웨어 인수 지원으로 구성되며, <그림 2>는 총괄적 개발전략의 수명 주기 단계를 설명하고 있다.

총괄적 개발전략은 주로 다음과 같은 특성을 갖는 정보시스템 사업에 적용한다.

- 모든 요구사항을 초기에 정의
- 단일 개발 주기를 사용
- 시스템 개발 종료 후에 소프트웨어 사용



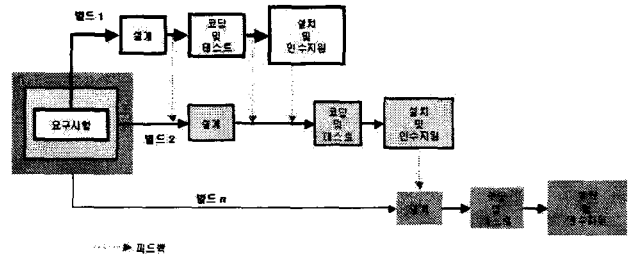
<그림 2> 총괄적 개발전략

2.2 점증적 개발전략

점증적(Incremental) 개발전략은 사용자의 소요를 결정하여 시스템의 요구사항을 정의하고, 그 이후에 일련의 빌드를 통하여 개발의 나머지 부분을 수행한다. 첫 번째 빌드는 계획된 시스템 능력의 일부만을 구현하며, 시스템이 완전해 질 때까지 다음의 빌드에 계속적으로 추가적인 능력을 더해 간다. 점증적 개발전략에서는 초기에 정의된 요구사항을 후속 빌드에서 점차 확장하면서 설계, 코딩, 테스트, 통합에 적용한다. 각 빌드에서 요구사항, 설계, 코딩 및 테스트, 설치 및 인수지원의 세부 절차는 총괄적 개발전략의 프로세스를 따른다. <그림 3>는 점증적 개발전략의 개념과 수명 주기 단계를 설명하고 있다.

점증적 개발전략은 주로 다음과 같은 특성을 갖는 정보시스템 사업에 적용한다.

- 모든 요구사항을 초기에 우선 정의
- 복수의 개발 주기를 사용
- 보통 시스템 개발의 중간 단계에서 소프트웨어를 사용



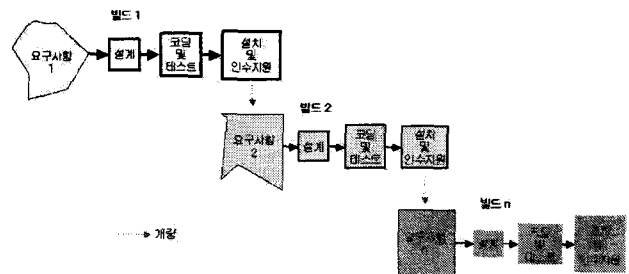
<그림 3> 점증적 개발전략

2.3 진화적 개발전략

진화적(Evolutionary) 개발전략은 점증적 개발전략과 마찬가지로 빌드를 통하여 시스템을 개발하지만, 모든 요구사항을 처음부터 정의할 수 없다는 점이 점증적 개발전략과 다르다. 진화적 개발전략에서는 사용자 소요와 시스템 요구사항이 처음에는 부분적으로만 정의되며, 후속적인 빌드가 진행됨에 따라 자세하게 정의된다. <그림 4>는 진화적 개발전략의 개념과 수명 주기 단계를 설명하고 있다.

진화적 개발전략은 주로 다음과 같은 특성을 갖는 정보시스템 사업에 적용한다.

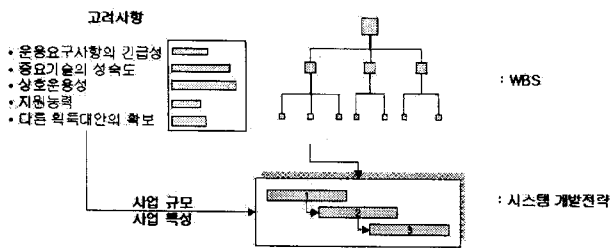
- 요구사항을 초기에 부분적으로 정의
- 복수의 개발 주기를 사용
- 대부분의 시스템 개발의 중간 단계에서 소프트웨어를 사용



<그림 4> 진화적 개발전략

3. 점증적, 진화적 개발전략에 대한 지침

정보시스템 사업은 사업 규모와 더불어 운용요구사항의 긴급성, 중요 기술의 성숙도, 상호운용성, 지원능력, 다른 획득대안의 확보 등을 고려하여 시스템 획득 전략을 결정한다. 정보시스템 사업은 WBS(Work Breakdown Structure)를 사용하여 하위 구성요소로 분해한 후에, 구성요소에 대하여 관리한다. 구성요소 중에서 비개발대상을 제외한 개발 대상에 대하여 개발전략을 2장에서 설명한 기준에 따라 결정한다. [1] <그림 5>는 이러한 정보시스템 사업관리 프로세스를 나타낸다.



<그림 5> 정보시스템 사업관리 프로세스

특히, 점증적, 진화적 개발전략을 적용하는 경우, 다수의 빌드를 통해 시스템을 완성하기 때문에 사업 초기에 빌드의 구성이나 내용에 대한 개략적인 정의가 필요하다. 하지만, 대부분의 정보시스템 사업은 빌드에 대한 구분을 명확히 정의하지 않고 진행되는 경우가 빈번하며, 점증적 또는 진화적 개발 전략에 대한 개념을 효과적으로 반영하여 개발하지 못하고 있다.

본 연구에서는 WBS 구성요소 중에서 개발 대상으로 식별된 소프트웨어에 대하여 점증적, 진화적 개발전략을 적용할 때, 빌드에 대한 개발 내용을 간략하게 명시하는 문서로써 빌드계획서의 도입을 제안하였다. WBS와 빌드계획서는 시스템의 운용 개념에서부터 최종개발까지 점점 구체화하여 작성 및 관리해야 한다.

빌드계획서의 기본적인 가정은 다음과 같다.

- 각 빌드는 기본적으로 년 단위로 구분하여 개발하는 것을 원칙으로 한다. 필요할 경우 6개월 단위로 구분하여 개발할 수 있지만, 빌드 간격이 줄어들수록 빌드계획서의 도입 효과는 감소되며, 전체 사업의 효과성 및 효율성이 떨어진다.
- 점증적 개발전략을 사용하는 경우, 전체 개발 기능을 분해하고 각 기능을 각 빌드에서 중복되지 않게 분할하여 개발한다.
- 진화적 개발전략을 사용하는 경우, 각 빌드가 동일한 기능을 반복적으로 개발한다.

그리고, 점증적, 진화적 개발전략을 적용하는 경우에 일반적으로 다음과 같은 사항을 고려해야 한다.

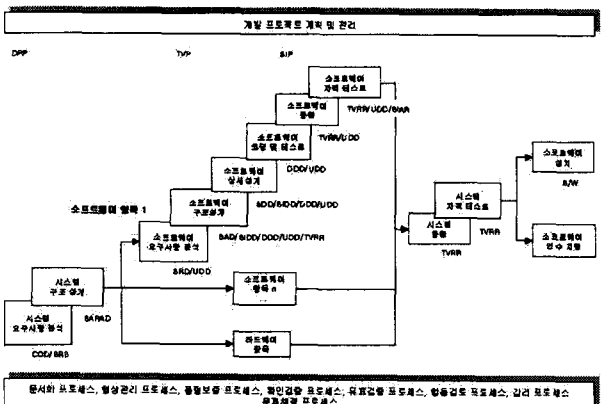
- 점증적, 진화적 개발전략은 다수의 빌드를 중심으로 적용한다. 빌드의 간격은 일반적으로 년 단위로 구분하며, 빌드의 수는 해당 사업 기간과 개발 대상의 규모를 분석하여 결정한다.
- 점증적, 진화적 개발전략은 운용요구사항의 긴급성, 중요 기술의 성숙도, 상호운용성, 지원능력, 다른 획득 대안의 확보 가능성 등을 고려하여 적용한다.
- 점증적 개발전략을 사용하는 경우, 최초의 빌드에서는 초기 능력의 정의, 개발, 배치/전력화를 담당하는 것이 일반적이며, 이후 빌드에서는 추가적인 능력의 개발 및 배치/전력화를 담당한다.
- 진화적 개발전략을 사용하는 경우, 최초의 빌드에서는 현재까지 식별된 초기 능력의 정의, 개발, 배치/전력화를 담당하는 것이 일반적이며, 이후 빌드에서는 반복적인 피드백을 통한 요구사항의 보완, 개발 및 배치/전력화를 담당한다.

4. 정보시스템 개발전략에 따른 수명주기 산출물 관리체계

다양한 개발전략에 따라서 개발 수명주기 산출물을 관리하는 체계도 달라진다. 4장에서는 MIL-STD-498에서 제시하고 있는 개발전략에 대한 산출물 관리 방법론을 최신 표준인 IEEE/EIA 12207의 흐름에 맞게 재구성하여 제안하였다.

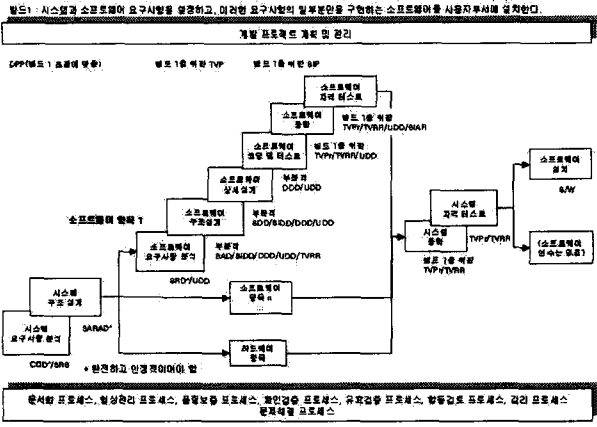
4.1 총괄적 개발전략에 대한 산출물 관리체계
 총괄적 개발전략을 적용할 때는 <표 2>과 같이 소프트웨어 수명주기 단계에 따른 IEEE/EIA 12207의 단계별 산출물을 참고하여 적용한다.

이것을 정리하여 MIL-STD-498에서 제시한 방법론에 따라서 수명주기 단계와 단계별 산출물을 표현하면 다음의 <그림 6>과 같다.

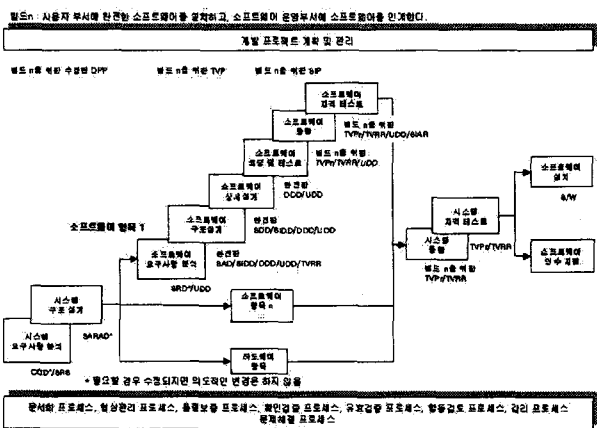


<그림 6> 총괄적 개발전략의 산출물 관리체계

4.2 점증적 개발전략에 대한 산출물 관리체계
 점증적 개발전략을 적용할 때는 다수의 빌드에 따라 산출물을 관리하며, 해당 빌드에 대한 산출물 관리는 총괄적 개발전략을 참고한다.
 마찬가지로 점증적 개발전략에 대하여 MIL-STD-498에서 제시한 방법론에 따라서 수명주기 단계와 단계별 산출물을 표현하면 다음의 <그림 7>과 <그림 8>과 같다. 예를 들어, 국방 분야의 자동화정보체계 개발 수명주기 단계에 이것을 <표 1>과 같이 적용할 수 있다. [1]



<그림 7> 점증적 개발전략의 산출물 관리체계 (빌드 1)



<그림 8> 점증적 개발전략의 산출물 관리체계 (빌드 n)

<표 1> 수명주기 산출물 관리체계 적용 예 (국방 자동화정보체계, 점증적 개발전략)

| 개발수명주기단계 | 산출물 (원형) | 빌드 1 | 빌드 2 | ... | 빌드 n-1 | 빌드 n |
|------------------|--|-------------|------|-----|--------|--------------------------------|
| 체계요구사항 분석 및 체계설계 | 운동개념기술서 체계규격서 체계설계서 인터페이스요구사항명세서 | 안전하고 인용적 | | | | 필요한 경우 수정되지만 의도적인 변경은 하지 않음 |
| 소프트웨어 요구사항 분석 | 소프트웨어요구사항명세서 인터페이스요구사항명세서 | 안전하고 인용적 | | | | 필요한 경우 수정되지만 의도적인 변경은 하지 않음 |
| 소프트웨어 설계 | 소프트웨어설계명세서 인터페이스설계명세서 데이터베이스설계명세서 | 부분적 | | | | 수정 및 개선 |
| 소프트웨어 구현 및 시험 | 개발일지에 대한 단위소프트웨어시험결과 | | | | | 해당 빌드에 대한 산출물 |
| 체계통합 및 시험 | 개발일지에 대한 체계통합시험기술서 체계통합시험결과보고서 | | | | | 해당 빌드에 대한 산출물 |
| 시험평가 | 개발일지에 대한 기술시험평가계획서 기술시험평가세부계획서 기술시험평가결과보고서 운동시험평가계획서 운동시험평가세부계획서 운동시험평가결과보고서 | | | | | 해당 빌드에 대한 산출물 |
| 체계 검증 및 인계 인수 | 개발일지에 대한 실행가능한 S/W, 원시파일 사용지침서, 운영 매뉴얼 소프트웨어 방편기술서 새부속함월차 | 초기 | | | | 수정 및 개선 안전함 |

<표 2> 소프트웨어 수명주기 단계 및 단계별 산출물

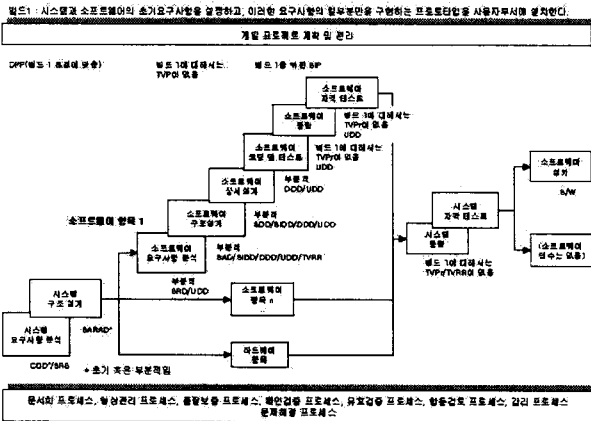
| 수명주기 단계 | 주요 산출물 |
|---------------|--|
| 체계요구사항 분석 | <ul style="list-style-type: none"> 개발 프로세스 계획서 (Development Process Plan) 운용 개념 기술서 (Concept of Operations Description) 체계 요구사항 명세서 (System Requirement Specification) |
| 체계 구조 설계 | <ul style="list-style-type: none"> 체계 구조 및 요구사항 할당 기술서 (System Architecture and Requirements Allocation Description) |
| 소프트웨어 요구사항분석 | <ul style="list-style-type: none"> 소프트웨어 요구사항 기술서 (Software Requirements Description) |
| 소프트웨어 구조 설계 | <ul style="list-style-type: none"> 소프트웨어 구조 기술서 (Software Architecture Description) 소프트웨어 인터페이스 설계 기술서[상위수준] (Software Interface Design Description) 데이터베이스 설계 기술서[상위수준] (Database Design Description) 사용자 문서화 기술서[초기] (User Documentation Description) 테스트 또는 확인 계획서[소프트웨어 통합] (Test or Validation Plan) |
| 소프트웨어 상세 설계 | <ul style="list-style-type: none"> 소프트웨어 설계 기술서 (Software Design Description) 소프트웨어 인터페이스 설계 기술서[상세수준] (Software Interface Design Description) 데이터베이스 설계 기술서[상세수준] (Database Design Description) 사용자 문서화 기술서[갱신] (User Documentation Description) 테스트 또는 확인 계획서[소프트웨어 단위] (Test or Validation Plan) 테스트 또는 확인 계획서[소프트웨어 통합-갱신] (Test or Validation Plan) |
| 소프트웨어 구현 및 시험 | <ul style="list-style-type: none"> 소프트웨어 테스트 또는 확인 결과 보고서[단위 및 데이터베이스] (Test or Validation Results Report) 사용자 문서화 기술서[소프트웨어 갱신] (User Documentation Description) 테스트 또는 확인 계획서[소프트웨어 통합-갱신] (Test or Validation Plan) |
| 소프트웨어 통합 | <ul style="list-style-type: none"> 소프트웨어 통합 계획서 (Software Integration Plan) 테스트 또는 확인 결과 보고서[소프트웨어 통합] (Test or Validation Results Report) 사용자 문서화 기술서[갱신] (User Documentation Description) 테스트 또는 확인 절차[소프트웨어 자격] (Test or Validation Procedures) |
| 소프트웨어 자격 시험 | <ul style="list-style-type: none"> 테스트 또는 확인 결과 보고서[소프트웨어 자격] (Test or Validation Results Report) 사용자 문서화 기술서[갱신] (User Documentation Description) 소프트웨어 통합 감리 보고서[소프트웨어 기능 및 물리적 형상 감리] (Software Integration Audit Report) |
| 체계 통합 | <ul style="list-style-type: none"> 테스트 또는 확인 결과 보고서[체계 통합] (Test or Validation Results Report) 테스트 또는 확인 절차[체계 자격] (Test or Validation Procedures) |
| 체계 자격 시험 | <ul style="list-style-type: none"> 테스트 또는 확인 결과 보고서[체계 자격] (Test or Validation Results Report) 소프트웨어 통합 감리 보고서[체계 기능 및 물리적 형상 감리] (Software Integration Audit Report) |
| 소프트웨어 설치 | <ul style="list-style-type: none"> 소프트웨어 설치 계획 (Software Installation Plan) |
| 소프트웨어 인수지원 | <ul style="list-style-type: none"> 테스트 또는 확인 결과 보고서[소프트웨어 인수] (Test or Validation Results Report) |

4.3 진화적 개발전략에 대한 산출물 관리체계

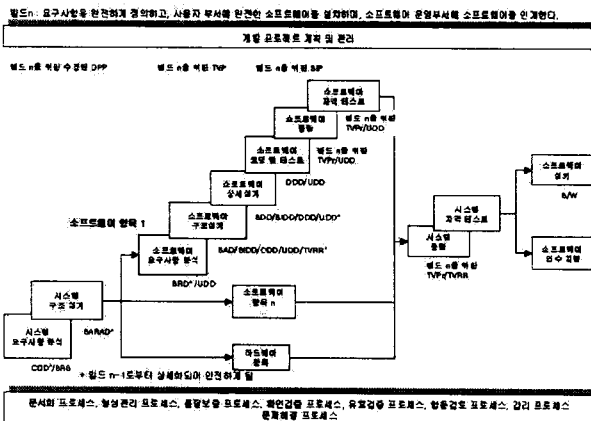
진화적 개발전략을 적용할 때도 점증적 개발전략과 같이 다수의 빌드에 따라 산출물을 관리하며, 해당 빌드에 대한 산출물 관리는 총괄적 개발전략을 참고한다.

마찬가지로 진화적 개발전략에 대하여 MIL-STD-498에서 제시한 방법론에 따라서 수명주기 단계와 단계별 산출물을 표현하면 다음의 <그림 9>과 <그림 10>과 같다. 점증적 개발전략에서의 마찬가지로 국방 분야의 자동화정보체계 개발 수명주기 단계에

이것을 <표3>과 같이 적용할 수 있다. [1]



<그림 9> 진화적 개발전략의 산출물 관리체계 (빌드 1)



<그림 10> 진화적 개발전략의 산출물 관리체계 (빌드 n)

<표 3> 수명주기 산출물 관리체계 적용 예 (국방 자동화정보체계, 진화적 개발전략)

| 개발수명주기단계 | 산출물 (무서) | 빌드 1 | 빌드 2 | ... | 빌드 n-1 | 빌드 n |
|------------------|---|-----------|---------------|-----|--------|------|
| 체계요구사항 분석 및 체계설계 | 운동개념기술서 체계규격서 체계설계서 인터페이스요구사항명세서 | 초기 또는 부분적 | 수정 및 개선(상세화) | | | 완전함 |
| 소프트웨어 요구사항 분석 | 소프트웨어요구사항명세서 인터페이스요구사항명세서 | 초기 또는 부분적 | 수정 및 개선(상세화) | | | 완전함 |
| 소프트웨어 설계 | 소프트웨어설계명세서 인터페이스설계명세서 데이터베이스설계명세서 | 초기 또는 부분적 | 수정 및 개선(상세화) | | | 완전함 |
| 소프트웨어 구현 및 시험 | 개발명세서 단위소프트웨어시험결과 | | 없음 | | | 있음 |
| 체계통합 및 시험 | 개발명세서 체계통합시험결과 체계통합시험결과보고서 | | 없음 | | | 있음 |
| 시험평가 | 개발명세서 기술시험평가계획서 기술시험평가세부계획서 기술시험평가결과보고서 운용시험평가계획서 운용시험평가세부계획서 운용시험평가결과보고서 | | 없음 | | | 있음 |
| 체계 결승 및 인계 인수 | 개발명세서 실행가능한 SW, 문서파일 사용지대뉴얼, 운영 매뉴얼 소프트웨어 버전기술서 | | 해당 빌드에 대한 산출물 | | | |
| | 세부 진행절차 | 초기 | 수정 및 개선 | | | 완전함 |

5. 결론 및 향후 연구 방향

본 논문에서는 다양한 정보시스템 개발전략에 대하여 간략하게 설명하고, 점증적·진화적 개발전략에 대한 일반 지침을 제시하였고, 개발전략에 따른 개발 수명주기 단계 산출물을 효율적으로 관리하는 체계를 제안하였다.

국방 분야와 같은 대규모 정보시스템 구축 사업에서는 체계적인 사업 관리가 무엇보다 중요하다. 운용개념의 연구부터 시스템 획득 방안 결정 및 획득, 운영 및 유지보수에 이르기까지 체계적인 관리가 필요하다. 특히, 정보시스템 개발전략에 따라 효과적으로 사업관리를 수행하는 것이 필요하다.

본 논문에서 제시한 개발전략에 대한 적용 지침 및 개발 산출물 관리체계는 대규모 사업을 추진하는 사업관리자에게 도움을 줄 수 있을 것이다. 향후 연구 방향으로는 실무에 구체적으로 적용할 수 있는 세부 지침의 개발 및 개발전략에 따른 테일러링 방안 등이 있다.

[참고문헌]

- [1] 효율적인 정보체계 사업추진을 위한 관리 절차 연구, 한국국방연구원, 2002
- [2] 국방획득관리규정, 국방부, 2002. 1. 10
- [3] 자동화정보체계 획득 및 관리 세부지침, 국방부, 2002. 7. 1
- [4] 소프트웨어 수명주기 프로세스에 의한 소프트웨어 개발 및 문서화 관리론, 조완수, 법영사, 1998
- [5] MIL-STD-498, "Software Development and Documentation", December 1994
- [6] IEEE/EIA 12207.0, "Software Life Cycle Processes", March 1998
- [7] IEEE/EIA 12207.1, "Software Life Cycle Processes - Life Cycle Data", April 1998
- [8] IEEE/EIA 12207.2, "Software Life Cycle Processes - Implementation Considerations", April 1998