

“ 품질과 성공 ”

MBSE가 왜 중요한가?

2010. 11. 30

박 영원
시스템공학과

www.ajou.ac.kr



아주대학교

Contents

1. MBSE 발전 배경

2. SE와 기존 접근 문제점

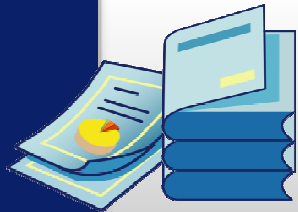
3. MBSE 개념과 종류

4. MBSE 혜택과 도전요소

5. 결론

MBSE의 발전 배경

전통적인 문서기반의 시스템엔지니어링의 문제점



- 대량의 문서 관리의 어려움, 설계 정보의 급격한 증가
- 언어의 다의성과 서로 다른 경험으로 인한 잘못된 의사소통
- 추적성 관리의 어려움

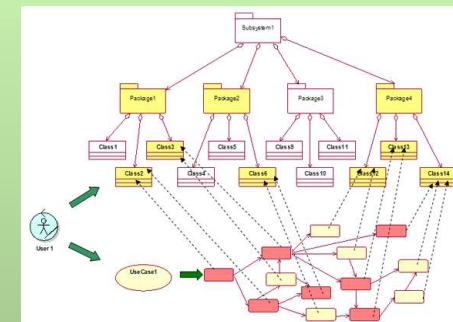


MBSE의 등장

◆ 미국방부의 전구고고도방공(THAAD) 등 사업에서 모델의 도입

- 소프트웨어 엔지니어링 개념
- 추상화
- 프로토타이핑
- 모델링 및 객체지향 설계 패러다임

- 추상화와 모델링을 통한 복잡 시스템의 분석
- 시스템 거동 및 하부수준 요구사항 개발을 위한 객체지향 접근의 적용



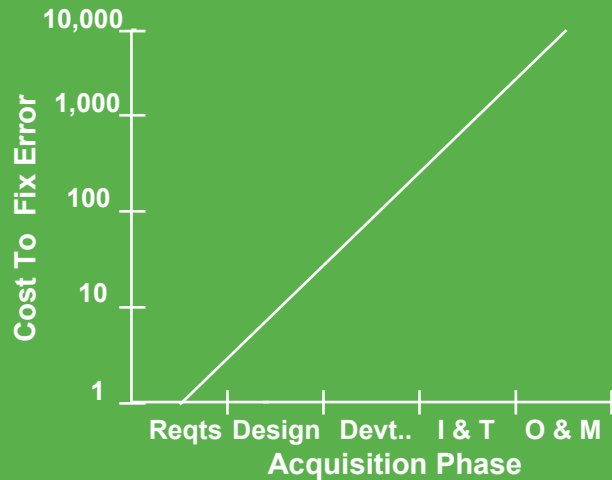
시스템공학의 주요 문제

- 시스템공학이 무엇을 하는지의 이해는 쉬움
- 시스템공학을 잘수행 하는 것은 어려움
- 복잡성의 관리가 가장 어려운 요소 중 하나
- 성숙된 시스템공학이 필요로 하는것은?
 - 좋은 시스템공학 공정
 - 문서화된 절차와 표준
 - 훈련된 인력:
“ 자동화된 도구가 시스템공학을 해주는 것은 아니고
훈련된 인력이 수행”

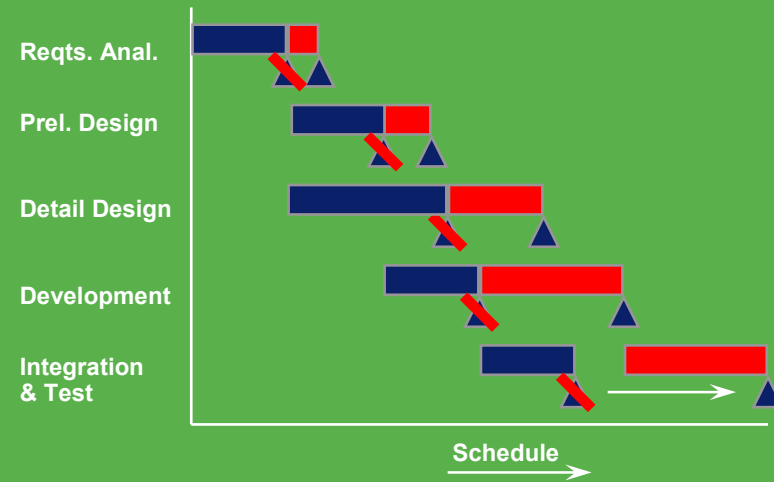
문서 기반의 **SE** 가 갖는 문제점

- 복잡성의 이해
- 요구사항 개발과 관리의 실패
- 소수의 경험있는 인력에 지나친 의존
- 반복 가능한 공정 부재
- 전체공정을 도와 줄 도구부재
 - 특수 부분에만 도구 적용 (e.g. 소프트웨어 개발[UML], 설계 [CAD/CAM], etc.)
- 정보의 공백

시스템공학을 강조하는 이유는? 잘못된 요구사항들의 영향 (잘못된 시스템공학)



비용 초과



일정 지연

예,

- 사용자 인터페이스 불량
- 제한된 역량
- 취약한 기능성

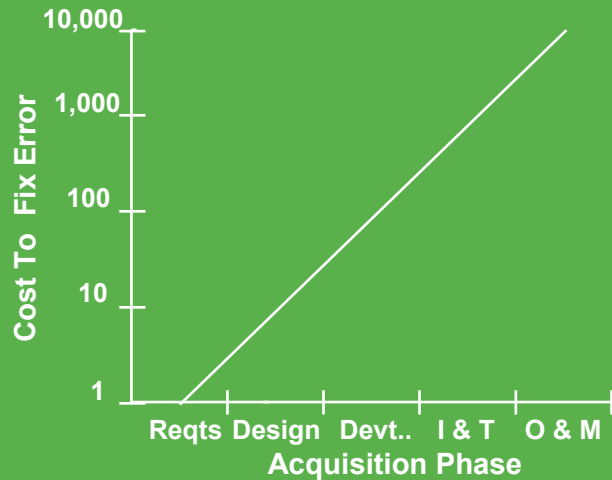
취약한 능력

미국 사례,

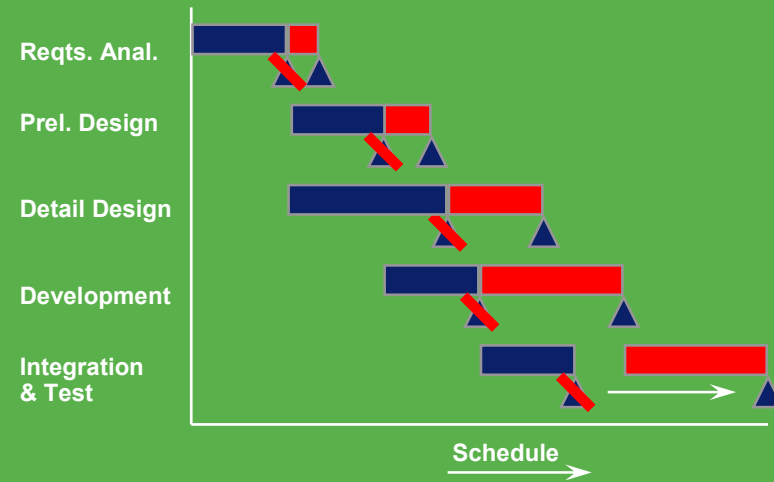
- **Sgt. York**, 자체 방어 못하는 국지 방공시스템
- 해군 스텔스 전투기

사업 취소

잘못된 요구사항들의 영향 (잘못된 시스템공학) 소프트웨어 개발에도 큰 문제



비용 증가



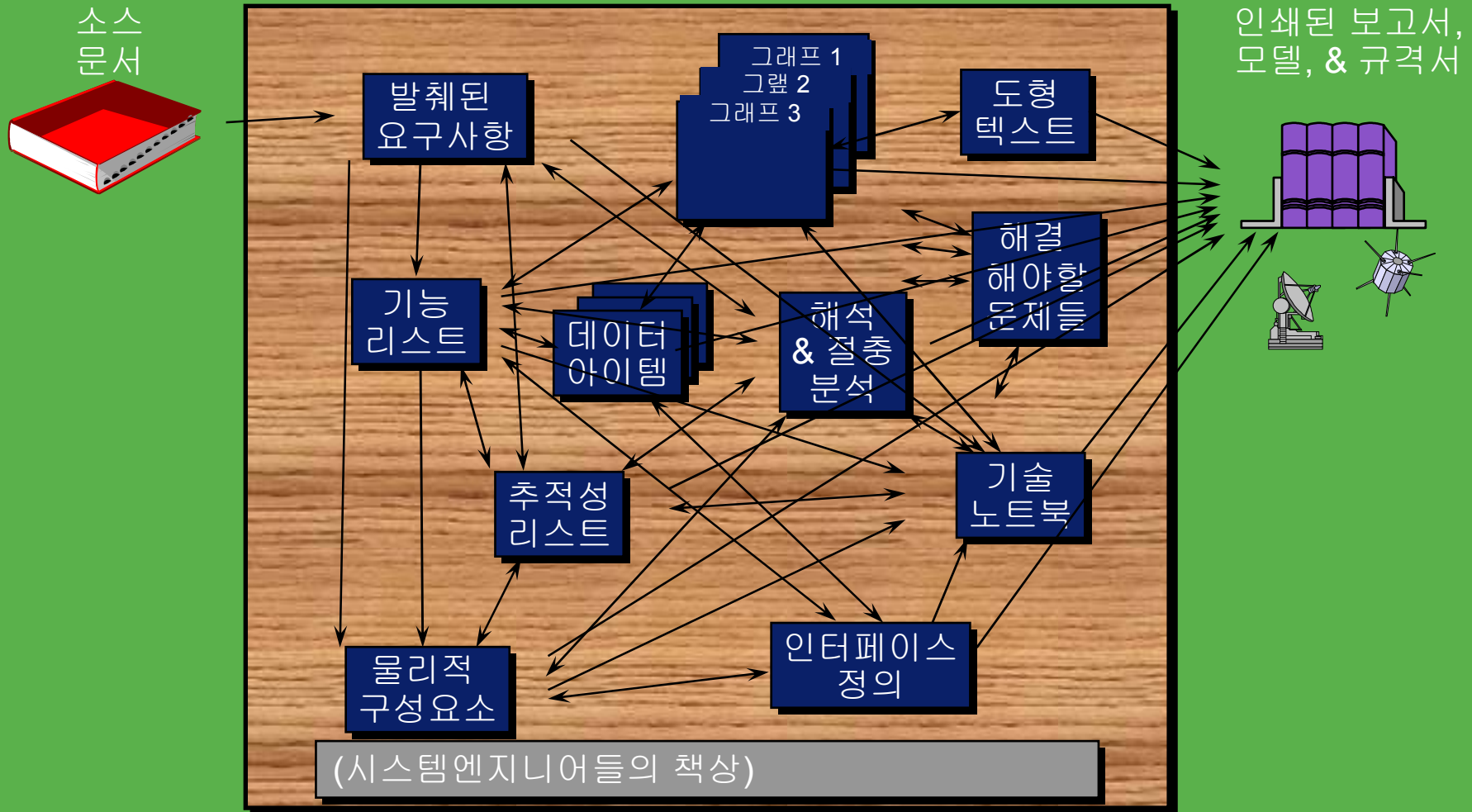
일정 지연

예), 소프트웨어 개발 (Standish Group Survey):

- 일정과 예산에 맞춰 개발된 사업은 16%에 불과
- 61%의 정해진 규격만이 개발
- 평균 비용초과 89%
- 31% 사업은 완성전에 중단

취약한 능력, 사업 중단

시스템엔지니어의 고민: 복잡성과 동기화



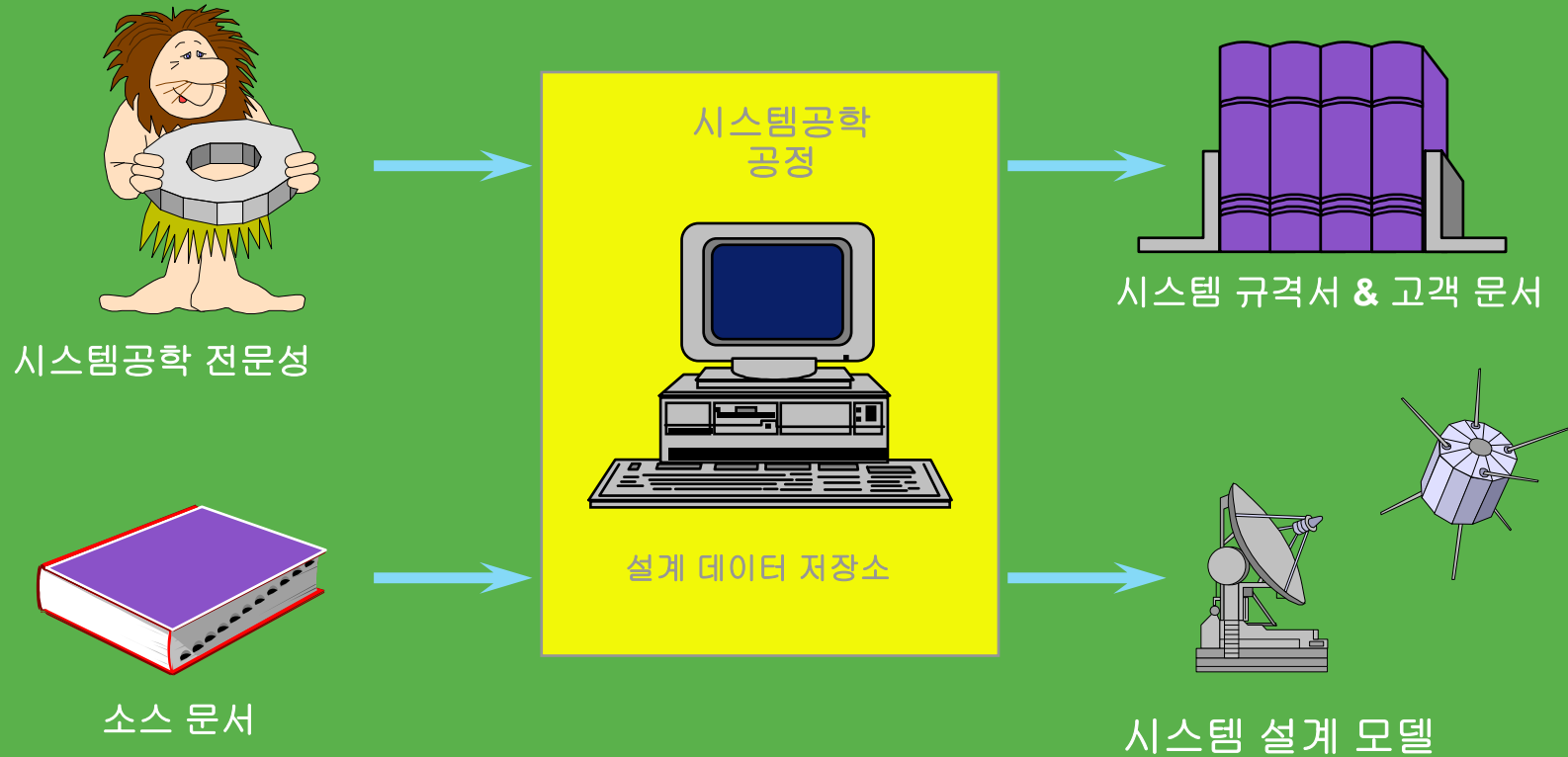
하나를 바꾸면 다른것들에 영향.

시스템공학이 중요한 이유는 ?

- 시스템공학
 - 전체 개발비의 **10%**의 노력이지만
 - 결과를 가지고 개발비용의 **80%** 이상에 대한 의사결정 수행
- 전체 시스템 설계 해결책
 - 만약 어떤 일부가 타당하지 않다면, 전체 시스템이 타당하지 않음
 - **i.e.**, “대강” 만족하는 시스템을 개발 할수 없음

시스템공학은 시스템전체 개발을
포함하는 활동

시스템공학의 공정: 입력과 출력물

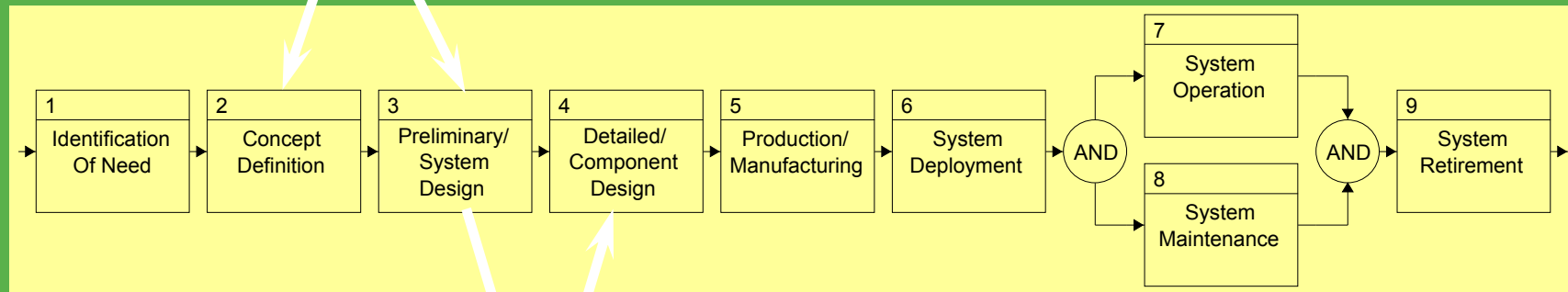


시스템 규격과 설계에 하향식, 중간진입 그리고 역공학(상향식) 접근 필요

제품 수명주기 - 어느부분을 지원하나?

주요 시스템공학 활동은
이곳에 집중

시스템공학 활동은 수명주기 전반에
응용되나 주로 강조 해야하는 부분들은
이곳에 집중

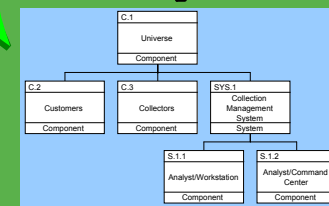
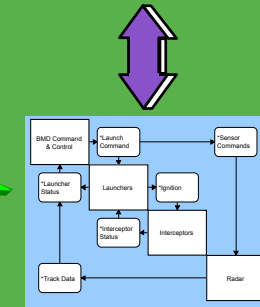
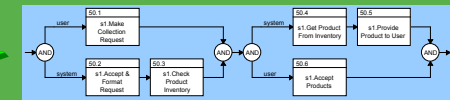


시스템 수준설계의 출력물들이 하부시스템설계나 구성요소 설계의 시작점이 됨 (e.g., 소프트웨어, 센서, 통제 & 제어, 무기체계 등)

시스템을 완전히 규격화 하려면 ?

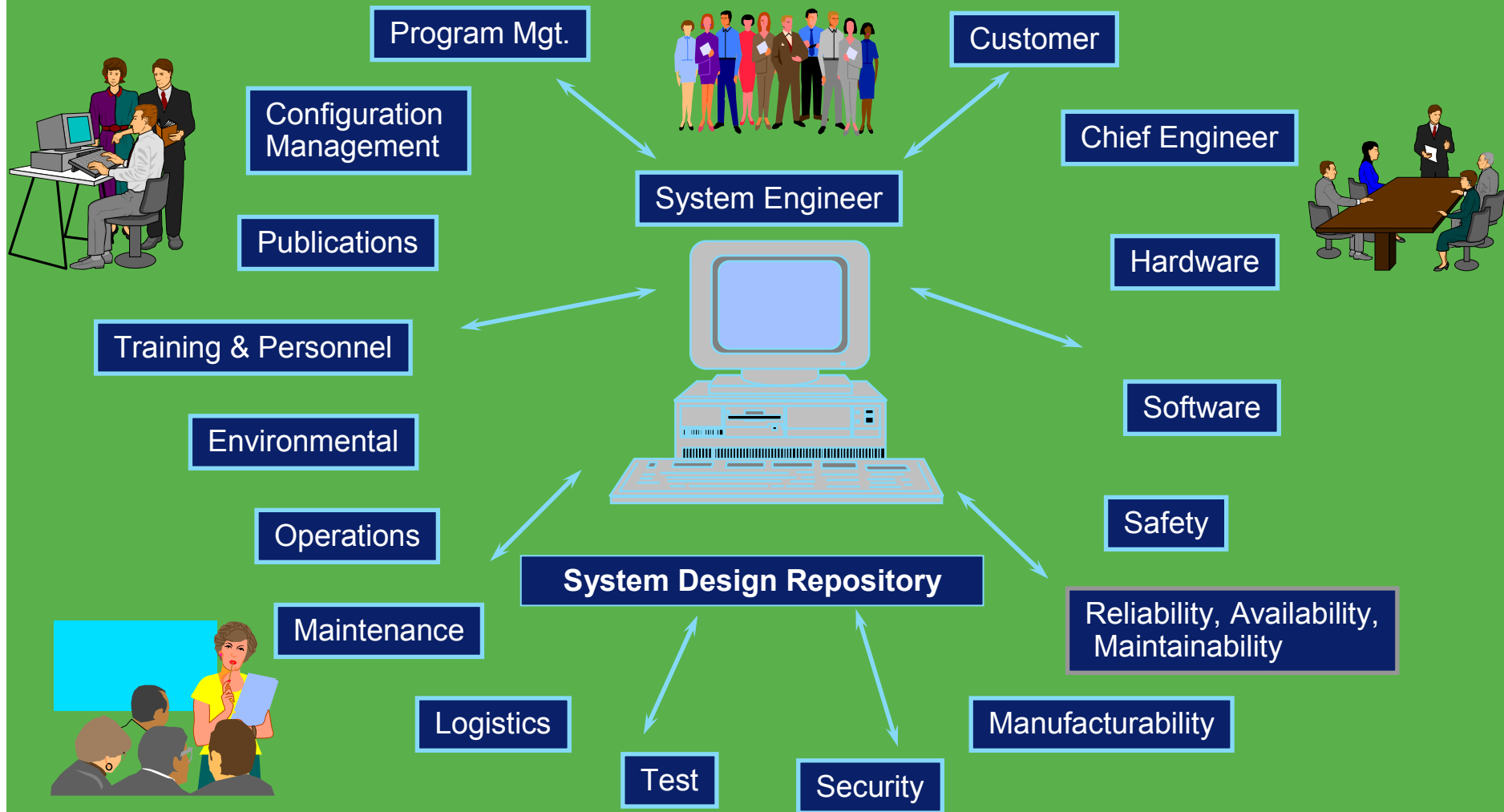
다음은 명확히 해주어야:

- 제어 (기능적 거동) 모델
- 데이터 (입력/출력) 모델
- 물리적 아키텍처 (구성요소) 모델
- 성능/자원
 - 상기 모델들의 부분 또는 조합들로 표현



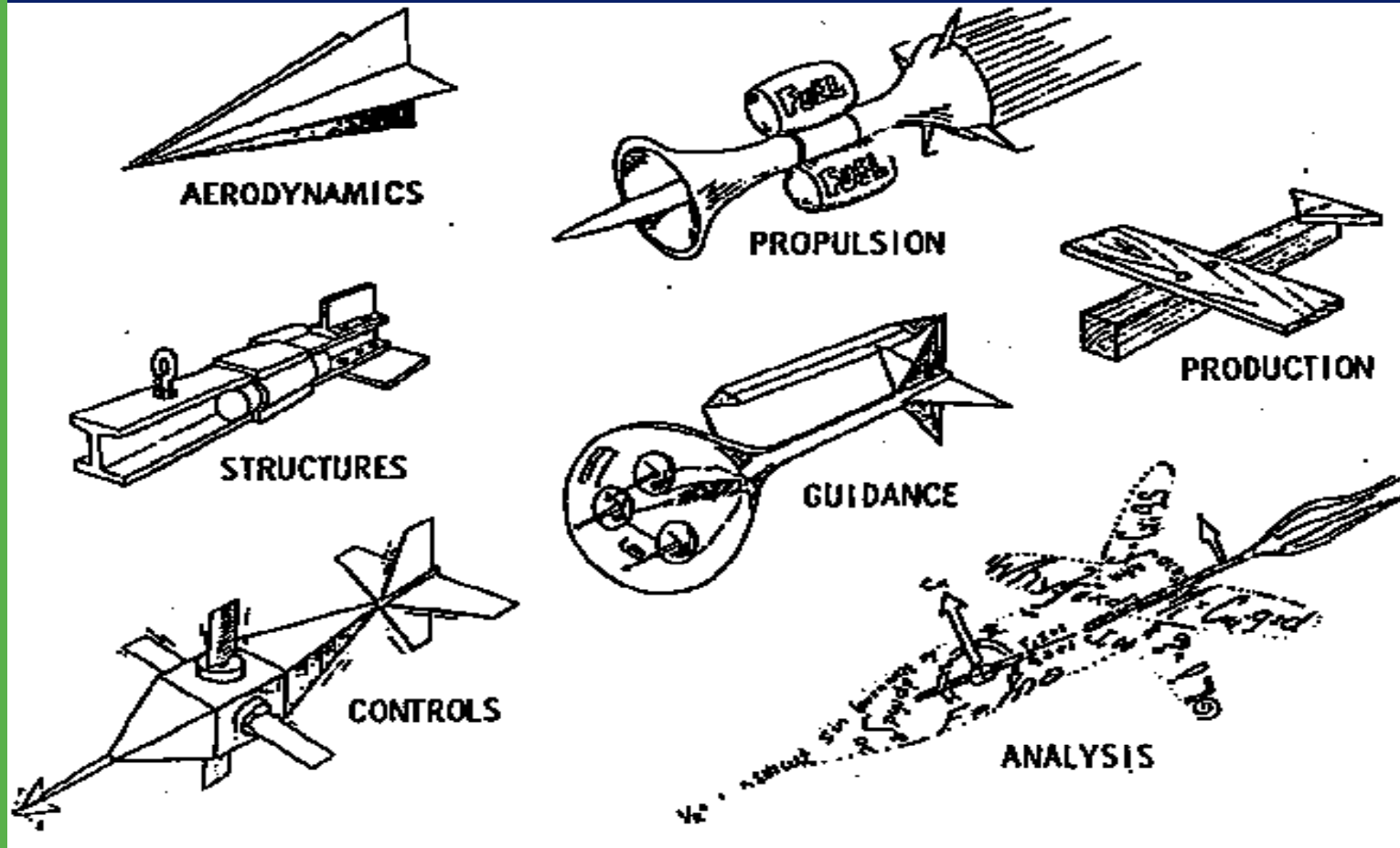
동시공학 환경에서 통일적, 일관성과 추적성을 가지고 의사소통이 가능하도록 해야 함

동시공학은 시험, 검토, 문서화 전에 즉시 설계 가능하도록 함



왜 시스템공학이 필요한가?

각 전문가들의 관점에서 보는 이상적 비행기 그림



많은 의사 결정이 요구되는 공정

	Examples of Decisions in Systems Engineering
Conceptual Design	<ul style="list-style-type: none"> • Should a conceptual design effort be undertaken? • Which system concept (usually a mixture of technologies) should be the basis of the design? • Which technology for a given subsystem should be chosen? • What existing hardware and software can be used? • Is the envisioned concept technically feasible, based on cost, schedule and performance requirements? • Should additional research be conducted before a decision is made?
Preliminary Design	<ul style="list-style-type: none"> • Should a preliminary design effort be undertaken? • Which specific physical architecture should be chosen from several alternatives? • To which physical resource should a particular function be allocated? • Should a prototype be developed? If so, to what level of reality? • How should validation and acceptance testing be structured?
Full-Scale Design	<ul style="list-style-type: none"> • Should a full-scale design effort be undertaken? • Which configuration items should be bought instead of manufactured? • Which detailed design should be chosen for a specific component given that one or more performance requirements are critical?
Testing	<ul style="list-style-type: none"> • What issues should be tested? • What equipment, people, facilities should be used to test each issue? • How much testing should be devoted to each issue? • What adaptive (fallback testing in case of a failure) testing should be planned for each issue?
Product Refinement	<ul style="list-style-type: none"> • Should a product improvement be introduced at this time? • Which technology or technologies should be the basis of the product improvement? • What redesign is best to meet some clearly defined deficiency in the system? • How should the refinement of existing systems be implemented given safety,

MBSE의 개념

the formalized application of modeling to support system requirements, design, analysis, verification and validation, beginning in the conceptual design phase and continuing throughout development and later life cycle phases.
(crisp, INCOSE, 2006)

■ Model의 활용

설계 주기 단축을 위한 엔지니어의 시간과 노력을 감소.
정보의 일관성과 완전성 검토를 통한 설계 오류 감소.
향후 유지보수, 제품 업그레이드 또는 제품 교체에
필요한 개발시의 엔지니어링 결과 보존.
모호하지 않은 시스템 기술.
엔지니어의 수작업 감소를 통한 정확성 향상.

- ◆ 전통적인 시스템엔지니어링에서 MBSE로의 전환은 방법론(PMT)의 발전에 따른 시스템 엔지니어링 패러다임의 전환.

Process

support

Method

support

Tool

*enable
PMT*

Environment

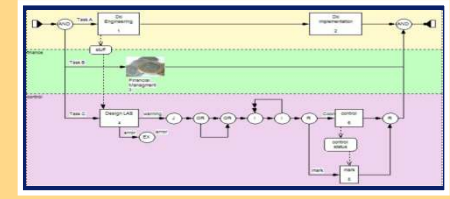
SE 지원 다이어그램의 개념, 규칙, 예시

● INCOSE Handbook v2.0의 모델링방법과 활동

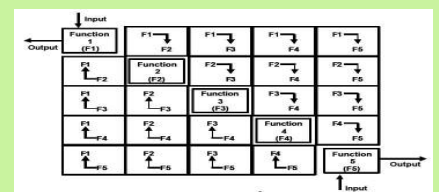
모델링 방법	요구사항 정의 프로세스				해결방안 정의 프로세스		
	OC	FR/PR	FA/A	RT	AD	SAS	PC
Context Diagram	●		●			●	●
Functional Block Diagram	●	●	●	●			
Functional Flow Diagram	●	●	●	●		●	
Time-line charts	●	●					
N2 charts	●		●			●	●
Hierarchy Diagram		●				●	
Quality Function Deployment		●			●		
IDEF0			●				
Data Flow Diagram		●	●	●			
Control Flow Diagram		●	●	●			
Behavior Diagram			●	●			
State Transition Diagram			●				
Entity Relationship Diagram			●				
functional Thread Analysis							
Modeling and Simulation			●				
Timing Analysis Results			●				
Real-Time Structure Analysis			●				
Requirements Allocation			●				
Object-Oriented System Modeling			●				
Requirement Traceability Matrix							
Data Dictionaries			●				
Simulation Results			●				
Functional and Concurrent Thread Analysis			●				

- OC : 운영개념 정립
- FR : 기능요구사항 도출
- PR : 성능요구사항 도출
- FA/A : 기능분석 및 할당
- RT : 요구사항 추적
- AD : 대안 정의 및 정제
- SAS : 시스템 아키텍처 합성
- PA : 물리적 아키텍처 정의

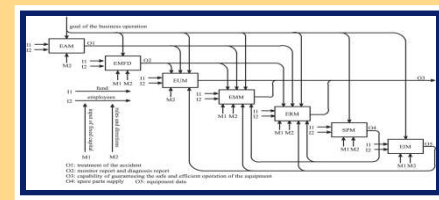
EFFBD



N2 chart

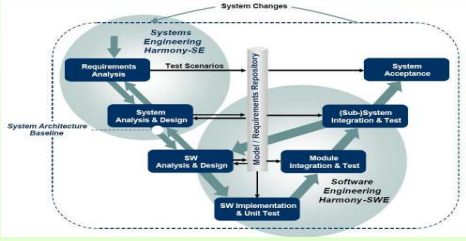
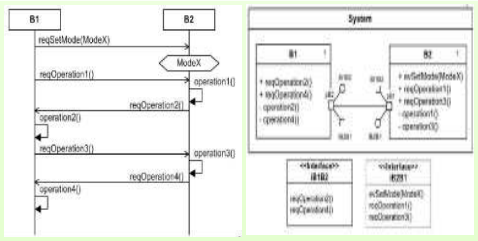
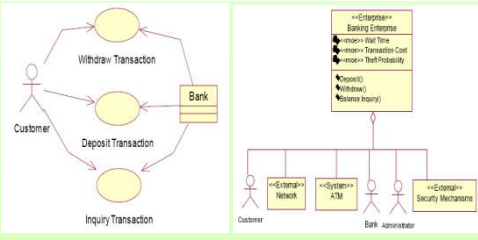
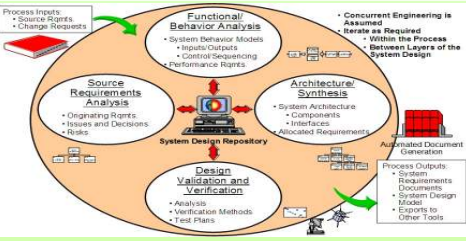
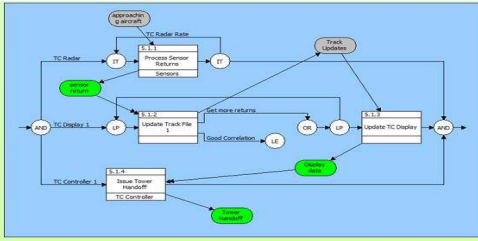
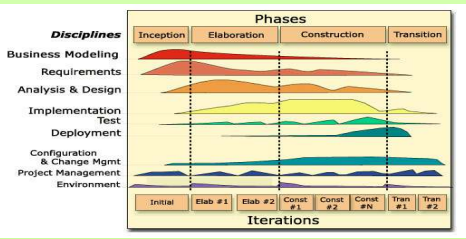
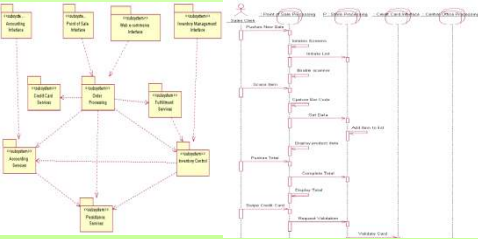


IDEF0



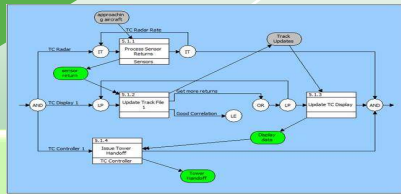
- 대부분의 표준은 시스템 분석 및 설계의 방법(methods)를 제한하지 않음.
- 다양한 모델링 방법이 고려됨.
- 거동, 데이터흐름, 기능의 대표적인 다이어그램에 대한 개념, 규칙 및 예시 제공

MBSE의 종류

MBSE 방법론	프로세스	모델링 언어	특징
<p>IBM Telelogic harmony-SE</p>			<ul style="list-style-type: none"> • Telelogic Rhapsody 를 통해 지원 • Vee model에 기반 • UML을 사용하는 객체지향 접근법을 활용
<p>INCOSE Object-oriented Systems Engineering Method (OOSEM)</p>	<p>Major SE Development Activities</p> <ul style="list-style-type: none"> Analyze Needs <ul style="list-style-type: none"> • Causal analysis • Mission use cases/scenarios • Enterprise model Define System Requirements <ul style="list-style-type: none"> • System use cases/scenarios • Elaborated context • Req'ts diagram Optimize & Evaluate Alternatives <ul style="list-style-type: none"> • Parametric Diag • Trade study Validate & Verify System <ul style="list-style-type: none"> • Test system • Test cases <p>Common Subactivities</p> <ul style="list-style-type: none"> Define Logical Architecture <ul style="list-style-type: none"> • Logical decomposition • Logical scenarios • Logical subsystems Synthesize Allocated Architecture <ul style="list-style-type: none"> • Node diagram • HW, SW, Data arch • System deployment 		<ul style="list-style-type: none"> • 아향식 접근법 • SysML을 사용하는 객체지향 접근법 • 도구(Tool)에 독립적인 방법론
<p>Vitech Model-Based System Engineering (MBSE) Methodology</p>			<ul style="list-style-type: none"> • Vitech CORE를 통해 지원 • SDL를 통한 ERA DB 정의 • Onion model을 중심 • EFFBD, N2, IDEFO와 시뮬레이션
<p>IBM Rational Unified Process for Systems Engineering (RUP SE)</p>			<ul style="list-style-type: none"> • 모델기반 시스템 개발을 위한 RUP의 확장된 형태 • 반복적이고 점진적인 나선형 모델로 생명주기 표현 • UML을 사용하는 객체지향 접근법

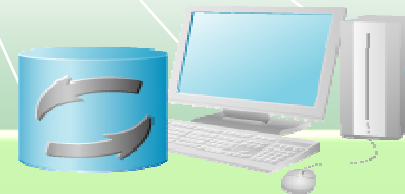
MBSE의 이점

시스템 엔지니어링을
더욱 효과적이고 효율적으로 수행할 수 있게 지원.



모델의 활용

- 이해관계자들 사이의 의사소통을 원활하게 함.
- 시스템의 전반적인 구조를 가시적으로 정의.

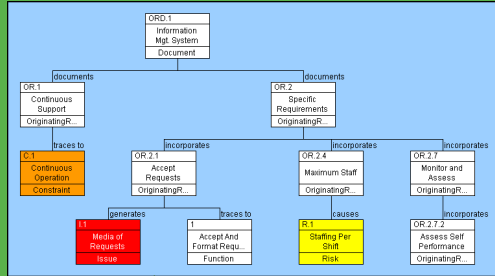


전산지원도구의 활용

- 방대한 설계 정보의 효율적인 관리.
- 모델링 및 시뮬레이션을 통한 프로토타입 최소화.
- 이해관계자의 요구사항에서부터 검증 요구사항에 이르기까지의 효율적인 추적성 확보.

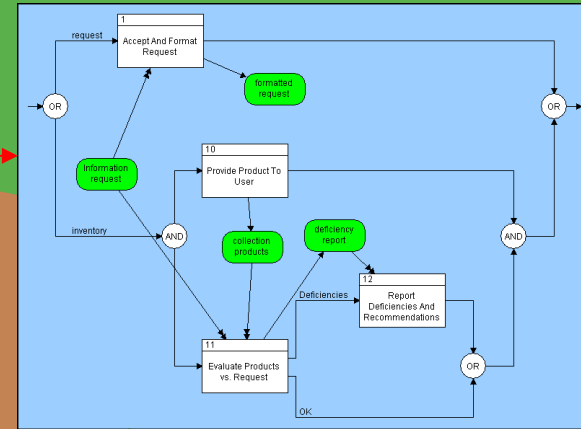
통합 수명주기 지식관리를 통한 일관성

원천 요구사항 영역



원천 요구사항과 거동 추적

거동 영역

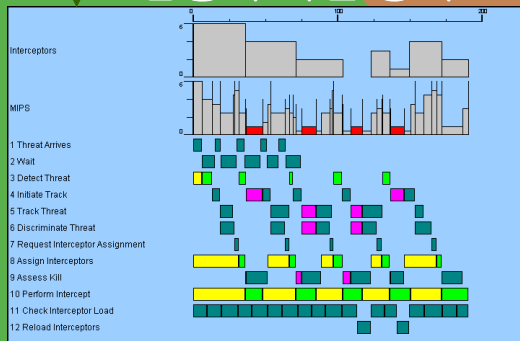


거동의 물리적 요소 할당

시스템 설계
자료 저장소

verified by

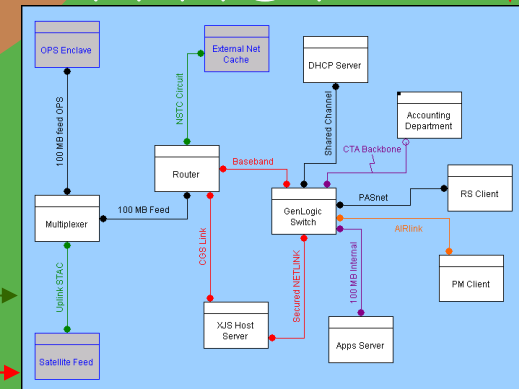
검증과 확인 영역



verified by

verified by

아키텍처 영역



원천요구사항과 물리적
구성요소 추적

시스템공학 활동 시간표 - (양파의 각 층에 대하여)

0. 필요와 시스템 개념 정의

1. 원천요구사항 수집

2. 시스템 경계 정의

3. 원천 아키텍처 제약사항 정의

4. 시스템 사용처 정의

5. 통합 시스템 거동 도출

6. 구성요소 계층 도출

7. 거동을 구성요소에 할당

8. 내부 인터페이스 정의

9. 설계 선정

10. 효과성과 타당성해석 정의

11. 자원, 오차 검출, 그리고 회복거동 정의

12. 시험 요구사항과 시험계획 개발

13. 문서와 규격서 생성

- 각 활동은 시스템공학 팀의 무게중심 이동 표현
- 동시공학 가정

일정



MBSE 사례연구

해외 사례 조사

- A Methodology for Software Product Line Development in Navy Weapons System Acquisition : An Approach to Open Architecture
 - Naval postgraduate school, (2008~2009)
 - “Application of MBSE methods to development of combat system architectures”
 - MBSE methodology development (Martin Process, SysML, CORE)
 - Requirement analysis ~ Verification & Validation

국내 사례 조사

- 미래도시철도기술개발사업,
 - “도시형 자기부상열차 요구사항 데이터베이스 구축과제”
 - 아주대 시스템공학과 (2006~2008)
 - 요구사항 개발, 시스템 아키텍처, SE 관리
 - 요구사항 분석 ~ 합성

시스템 모델링 언어 표준 분석

목표

- MBSE 활동의 수행 방안으로 활용 가능한 다이어그램 식별
- 다양한 모델링 언어 분석

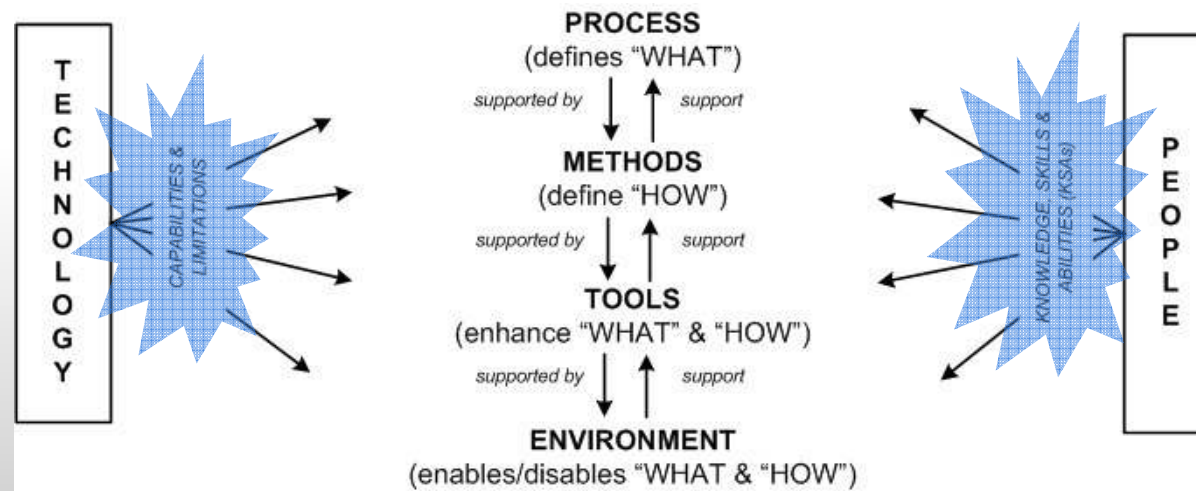
주요 고려 사항

- SysML, IDEF, FFBD, N2, ERD, PBD, ect.
- 도구가 지원하는 다이어그램 분석
- 정의, 규칙, 산출물 적용 방안

수행 절차

- 시스템엔지니어링 활동별 적용 가능한 다이어그램
- 시스템엔지니어링에 대표적으로 활용되는 다이어그램들의 개념, 규칙, 예시
- SysML이 제공하는 다이어그램들에 대한 개념, 규칙, 예시
- 전산지원도구에서 지원하는 다이어그램들의 개념, 예시, 규칙

MBSE 성공적 구현 위한 혁신



- 새로운 기술(method & Tool) 도입 위한 혁신
 - 지원 도구(enabling tools) 투자
 - 문서 기반 방식과의 호환을 고려한 변화
- 인력 훈련과 조직 공정 혁신
 - 인력 재교육 또는 신규 인력 투입
 - 기존 조직 방법론 및 문화의 혁신

결론

- 전문공학 설계에서 CAD/CAE/CASE 도구가 생산성 혁명 유발 -> SE도 정보화/지능화/자동화
- 시스템 설계 지식의 통합적 관리 - 저장, 공유, 변경, 활용, 동기화, 도식화, 의사소통, 문서 자동화 용이
- SE 전 공정간 산출물들의 정확성, 일관성 및 추적성 확보
- SE 활동의 효율성 및 효과성과 산출물 재사용성 증대

“ Add your slogan in here ”

Thank You !

www.ajou.ac.kr



아주대학교