



소프트웨어공학

강의노트

증명과 검증

❖ 학습안내

이번 시간의 학습내용과 학습목표를 확인해보세요.

■ 학습내용

- 증명과 검증
- V-model
- 중대한 시스템의 검증

■ 학습목표

- 증명과 검증을 설명할 수 있다.
- 증명과 검증을 활용하여 V-model을 설명할 수 있다.
- 중대한 시스템을 검증하는 테스트 방법에 대하여 설명할 수 있다.



❖ 학습내용

[1] 증명과 검증

1. 증명과 검증

◆ 증명과 검증의 정의

- 증명(Verification)
 - 증명 또는 확인으로 번역
 - 제품을 **올바르게 생성**하고 있는가?(Are we building the product right?)
 - 처음 설계한 명세목록과 맞는지 확인하는 것
 - 처음 **설계관점**에서 **개발이 정확하게 진행**되었는지 확인함

프로그램소스, 함수 등 개발 산출물을 조사하여
처음 설계대로 구현되었는지 확인하는 과정임

- **화이트 박스 테스트(White Box Test)** 시행
- 각 단계에서 **개발자의 시각**으로 검토함
- 즉 이전 단계에서 생성된 산출물이
현 단계의 산출물에 **정확히 반영**되었는지를 검토함

“ 요구사항 명세서의 내용이 최종산출물에 맞도록 구현되었는지
보는 것이 아니라, ”

요구사항 명세서의 내용은 분석, 설계에 반영되었는지

분석, 설계내용이 프로그램에 구현되었는지

- 검증(Validation)
 - **올바른 제품** 생성하고 있는가?(Are we building the right product?)
 - 처음 사용자가 요청한 **요구조건목록과 맞는지** 확인하는 것
 - 처음 **요건관점**에서 개발이 정확하게 진행되었는지 확인함
 - 즉 개발된 목적물을 실행하여 처음 **사용자가 요청한대로** 구현되었는지 확인하는
과정임

❖ 학습내용







[1] 증명과 검증

1. 증명과 검증(계속)

- ◆ 증명과 검증의 정의(계속)
 - 검증(Validation)(계속)
 - 블랙 박스 테스트(Black Box Test) 시행
 - 최종단계에서 **사용자의 시각**으로 검토함
 - 최초 **요구사항**이 최종 정보시스템에 잘 **반영**되었는지를 검토

요구사항 명세서의 내용이
최종산출물에 맞도록 구현되었는지 봄

◆ 증명과 검증의 목표

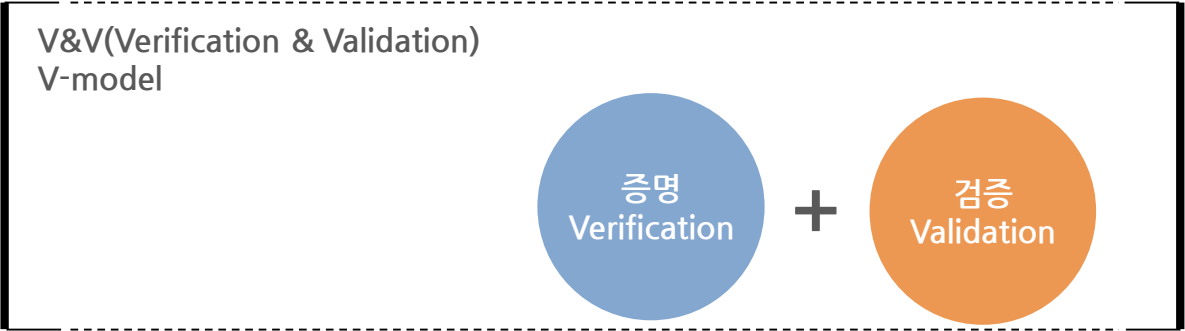
-  소프트웨어 시스템이 **목적에 적합함**을 보임
-  소프트웨어 시스템을 **신뢰**할 수 있음
-  소프트웨어를 **의도대로 사용**하는데 충분함
-  증명과 검증과정이 끝났다고 반드시 소프트웨어가 **결점이 전혀 없는 것은 아님**
-  구현과정을 끝낸 구현 목적물에 대한 증명과 검증을 통한 **품질 향상**
-  개발 생명주기 전 단계에 걸친 산출물의 **정합성 검토**를 통한 품질 향상

❖ 학습내용

[1] 증명과 검증

1. 증명과 검증(계속)

- ◆ 증명과 검증의 목표(계속)
 - 증명과 검증은 두 과정을 함께 사용하면 좋은 품질의 정보시스템이 구현가능하며 이 두 가지를 묶어서 V&V(Verification & Validation) 또는 V-model이라고 함



- ◆ 증명과 검증을 이해하는 시각

협이의 시각

- 단순히 증명과 검증을 테스트 기법으로 보는 시각
- 구현 단계 후 구현된 목적물에 대하여 증명테스트(화이트박스테스트 기법)와 검증테스트(블랙박스 테스트)를 시행하는 것
- 구현 목적물의 정합성을 증명하고 검증하는데 국한하여 봄

광의의 시각

- 증명과 검증을 광의의 시각으로 해석하는 것이 일반적임
- 정보시스템 개발 생명주기(SDLC) 전 과정에 걸쳐서 벌어지는 품질관리(QM) 차원에서 증명과 검증을 이해함
- 즉, 검토 목적물이 단순히 구현된 목적물 뿐만 아니라 관련 명세서 등 단계별 산출물에 대하여 증명과 검증을 수행함

❖ 학습내용

[1] 증명과 검증

2. 테스트

- ◆ 테스트의 정의
 - 소프트웨어 구현만큼 소프트웨어 테스트 과정도 중요
 - 테스트를 통하여 **구축된 시스템이 검증됨**
 - 테스트는 프로젝트의 구현단계가 끝난 후 다음 절차에 맞도록 수행하는 것이 일반적임
- ◆ 테스트 과정
 - 이 모든 테스트 과정들은 문서화되고 기록 관리하여야 함

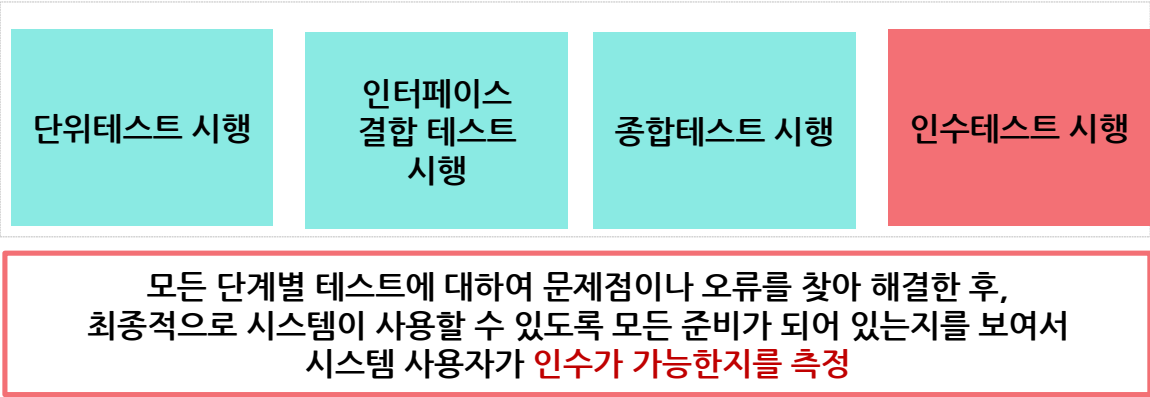


❖ 학습내용

[1] 증명과 검증

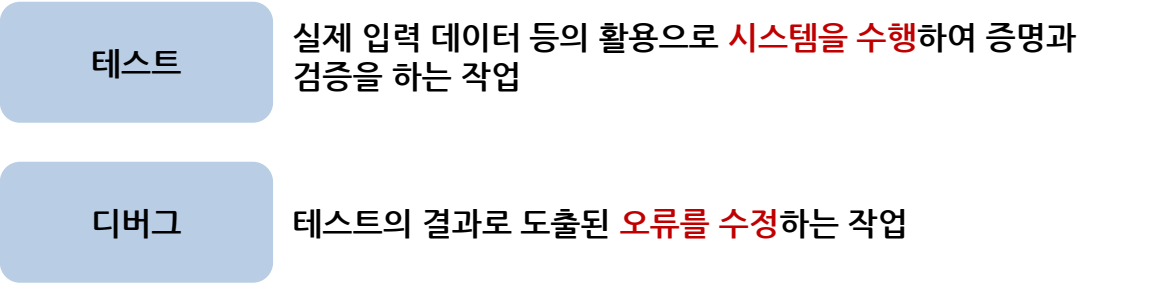
2. 테스트(계속)

- ◆ 테스트 과정(계속)
 - 이 모든 테스트 과정들은 문서화되고 기록 관리하여야 함(계속)



3. 디버그

- ◆ 디버그 정의
 - 증명과 검증은 명세 검토, 테스트 등으로 수행



- 오류 수정**, 컴퓨터 프로그램의 잘못을 찾아내고 고치는 작업
- 일단 작성된 프로그램들이 정확한지, 잘못 작성된 부분이 없는지를 조사하는 과정
- 프로그램을 만든 후 원하는 결과 나오지 않거나 문제가 확인되면, 프로그램의 코드를 처음부터 **한 줄씩 실행하면서** 무엇이 잘못되었는지, 오류 나는 부분은 어디인지 **일일이 찾아 고치는 것**

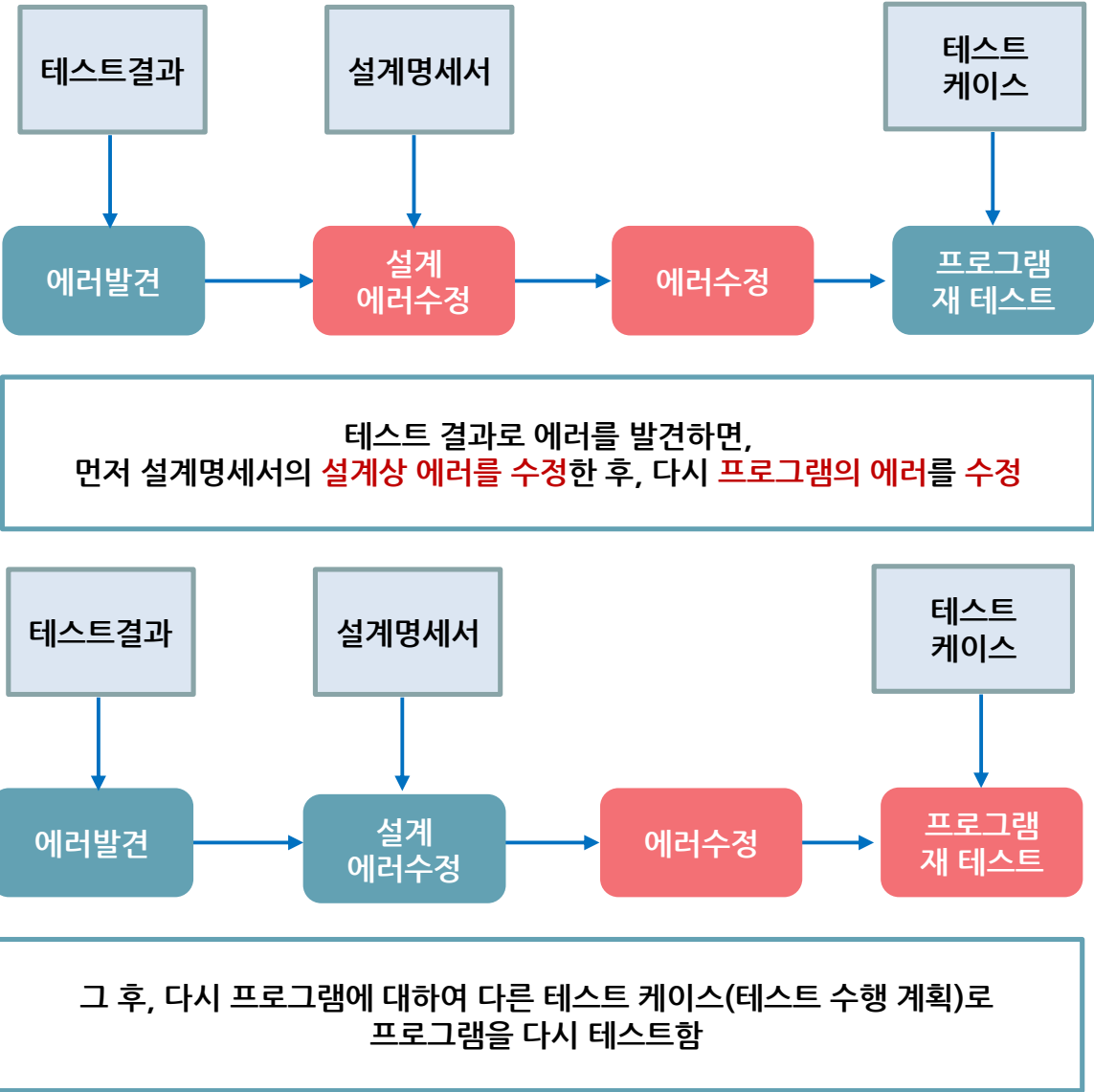
디버깅 과정은 프로젝트의 테스트 단계에서 묶어서 실행한다는 개념보다 구현단계에서 구현과 단위테스트를 통하여 **지속적으로 버그(Bug, 오류)를 고치고 수정하는 행위를 의미함**

❖ 학습내용

[1] 증명과 검증

3. 디버그(계속)

◆ 디버깅 절차

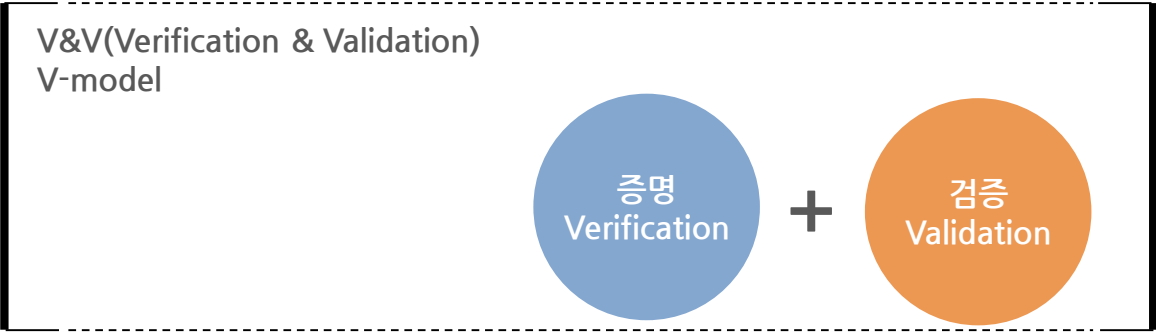


❖ 학습내용

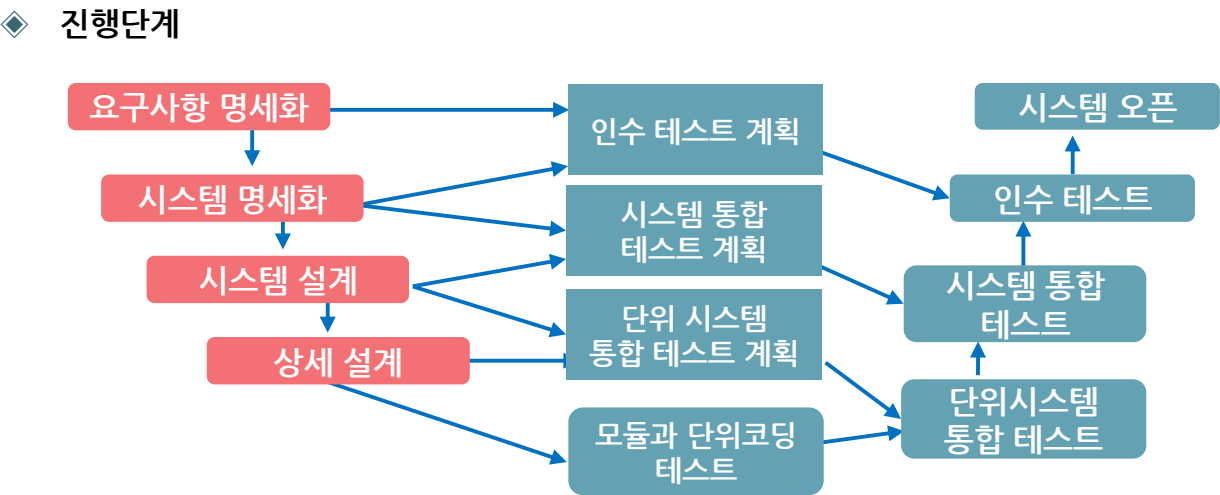
[2] V-model

1. V-model

- ◆ V-model의 정의
 - (복습)증명과 검증은 두 과정을 함께 사용하면 좋은 품질의 정보시스템이 구현가능하며 이 두 가지를 묶어서 **V&V(Verification & Validation)** 또는 **V-model**이라고 함



2. V-model의 진행단계



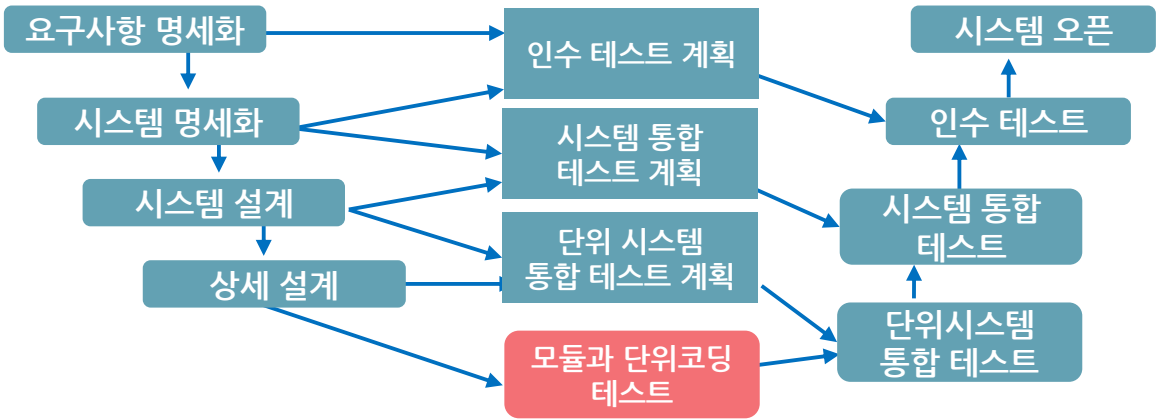
- 프로젝트에서의 진행절차의 일부
- 요구사항과 시스템에 대하여 명세서를 작성한 후, 이를 가지고 **시스템 설계**를 함
- 시스템 설계 후 **상세 설계** 작업이 이루어짐

❖ 학습내용

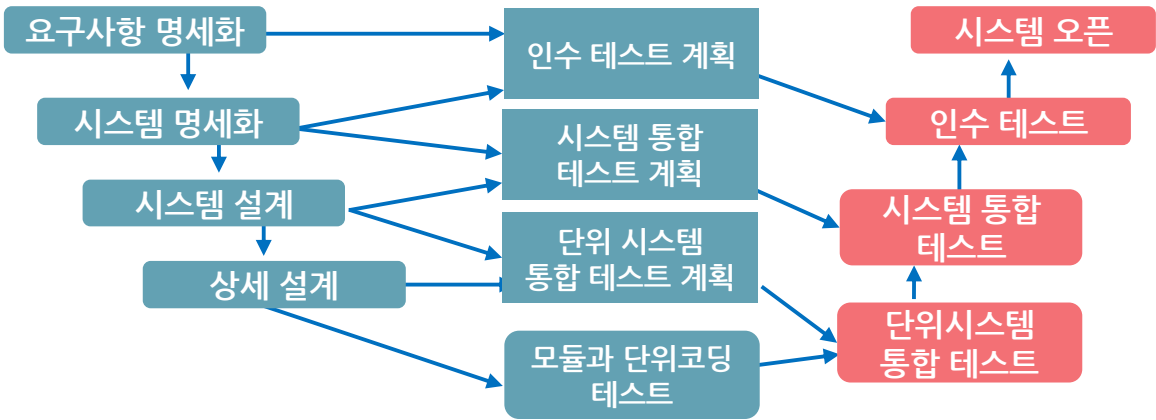
[2] V-model

2. V-model의 진행단계(계속)

◆ 진행단계(계속)



- 설계 후 개발 단계에서 개발단위마다 **모듈과 단위 코딩**에 대하여 테스트함



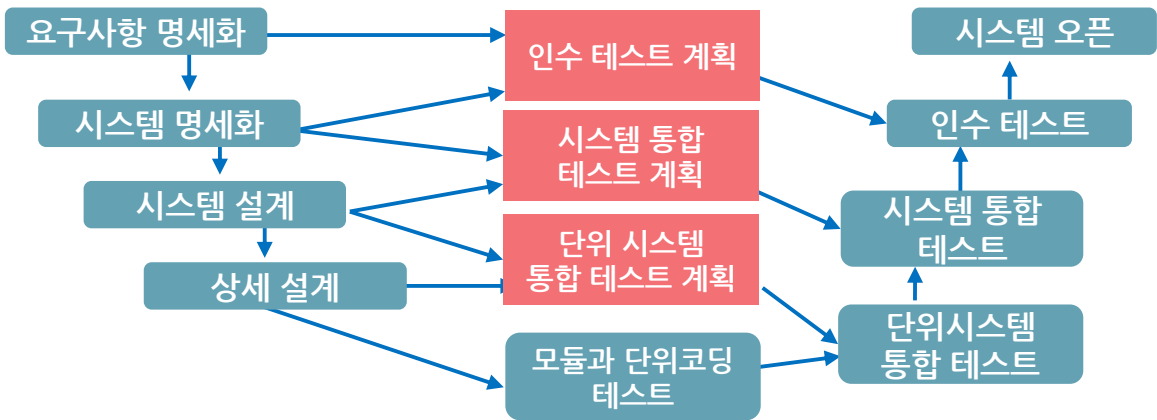
- 테스트 단계에서는 각각의 부분 시스템에 대하여 **통합테스트**를 한 후, 시스템에 대하여 통합 테스트를 함
- 최종 인수자가 인수테스트 수행 후 **시스템을 오픈**하게 됨

❖ 학습내용

[2] V-model

2. V-model의 진행단계(계속)

◆ 진행단계(계속)



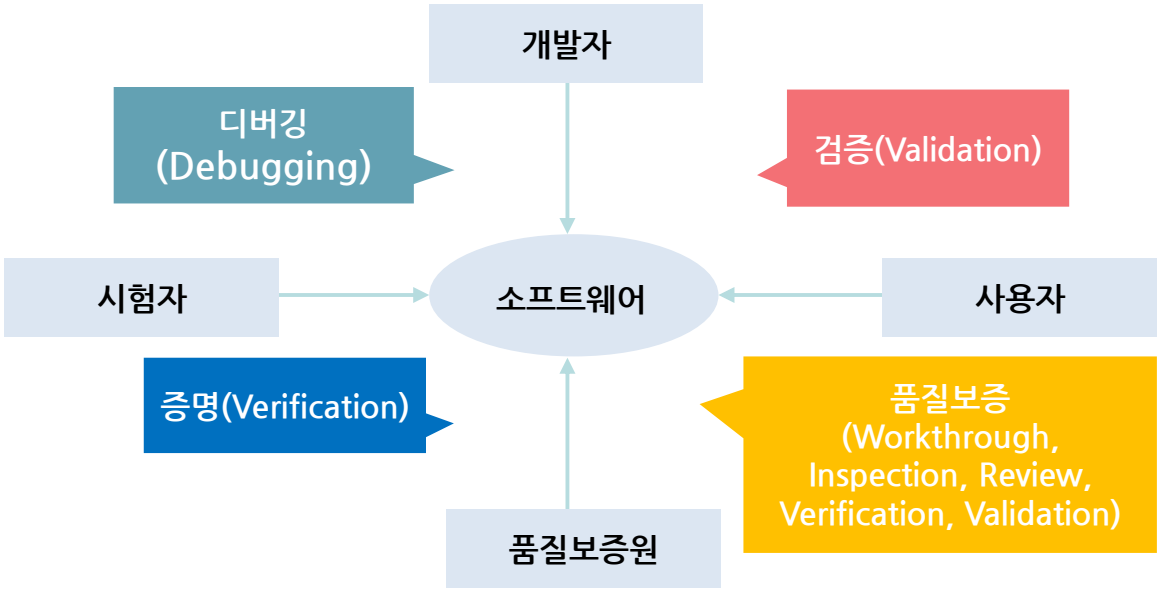
- 여기서 중요한 사항은 단위시스템 통합테스트를 위한 계획은 시스템 설계단계 끝부분과 상세설계 처음부분에 수행함
- 마찬가지로 시스템 통합테스트와 인수테스트도 이전 시스템명세화 단계와 시스템 설계단계에서 계획되어야 함

❖ 학습내용

[2] V-model

3. 유사개념과의 관계

◆ V&V, 테스트, 품질보증의 관계



[3] 중대한 시스템의 검증

1. 신뢰성 검증

◆ 중대한 시스템 검증

- 기업이나 조직의 핵심업무를 수행하는 중대한 시스템(Legacy System)의 경우 보다 **엄격한 테스트**를 필요로 함
- 이때 **신뢰성검증, 안정성 보증, 보안성 평가** 과정을 집중 수행함



- 정보 시스템이 주어진 조건 하에서 의도하는 기간에 **요구된 기능을 적정하게 수행할 확률**
- 1+2를 수행하였을 경우 3이 나올 것이라고 시스템을 신뢰함

❖ 학습내용

[3] 중대한 시스템의 검증

1. 신뢰성 검증(계속)

◆ 중대한 시스템 검증(계속)



- 소프트웨어 위험 요소 제거를 통해 소프트웨어 오류로 인한 **시스템의 사고를 예방**하는 것



- 데이터나 프로그램을 **권한이 없는 이용자가 사용할 수 없도록** 하는 것

◆ 신뢰성 검증 정의

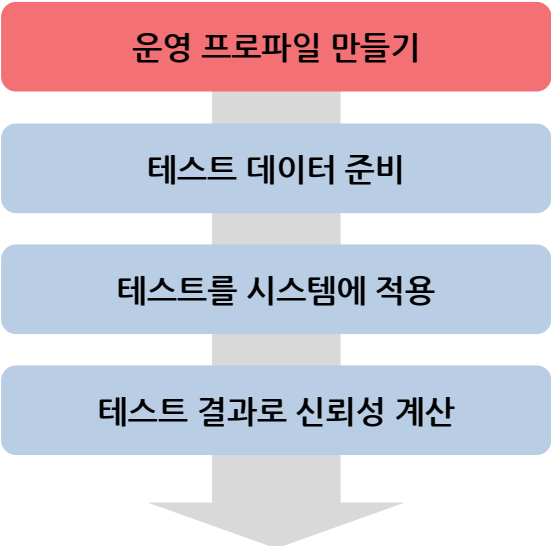
- 정보 시스템이 주어진 조건 하에서 의도하는 기간에 요구된 기능을 적정하게 수행할 확률인 신뢰성을 검증하기 위하여 **신뢰성 테스트**를 수행함
- 신뢰성 테스트는 모든 기능에 대해 시스템이나 시스템 구성 요소(Component) 또는 소프트웨어 프로그램이 다운되지 않고 **안정적으로 수행**되는지를 확인하는 시험
- 신뢰성 테스트를 위하여 가상으로 진행하는 시험환경 및 실제운영을 가정한 **테스트 데이터의 선택**이 중요
- 신뢰성 테스트의 결과를 가지고 **신뢰성을 계산**
- 신뢰성 계산 결과를 가지고 시스템이 신뢰성이 있는지 평가

❖ 학습내용

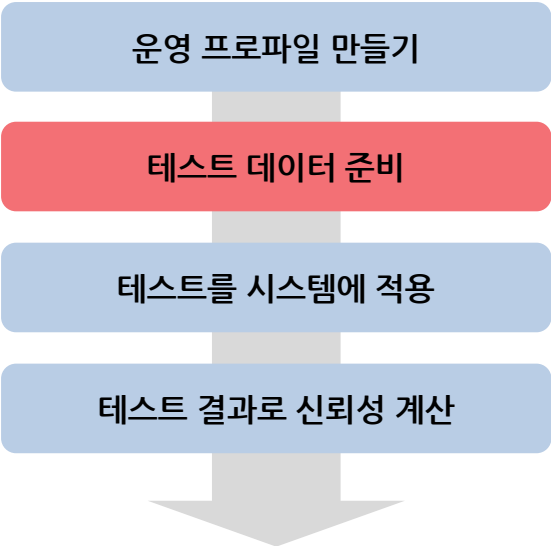
[3] 중대한 시스템의 검증

1. 신뢰성 검증(계속)

◆ 신뢰성 검증 수행 절차



- 신뢰성 검증을 위해 4단계를 수행
 - 최초 동일한 유형의 기존 시스템을 연구하여 **운영 프로파일**을 만드는 것부터 시작
 - 운영 프로파일은 **시스템 입력의 유형**과 정상적인 사용 시 **이러한 입력이 발생할 확률을 먼저 파악**



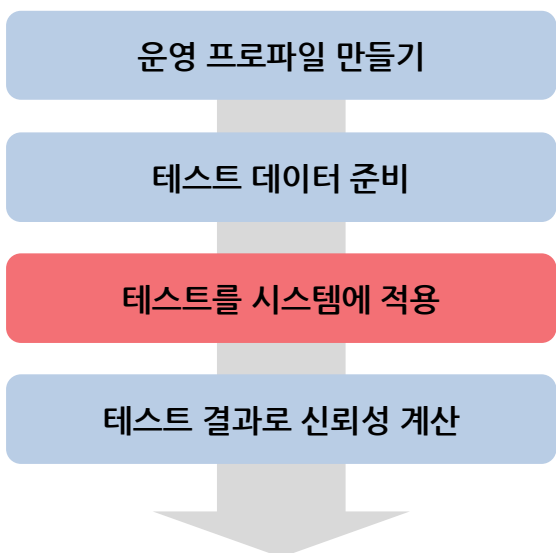
- 운영 프로파일을 실험해 볼 수 있는 **테스트 데이터**를 준비
- 이러한 데이터는 **수작업이나, 데이터 생성 프로그램** 등을 이용

❖ 학습내용

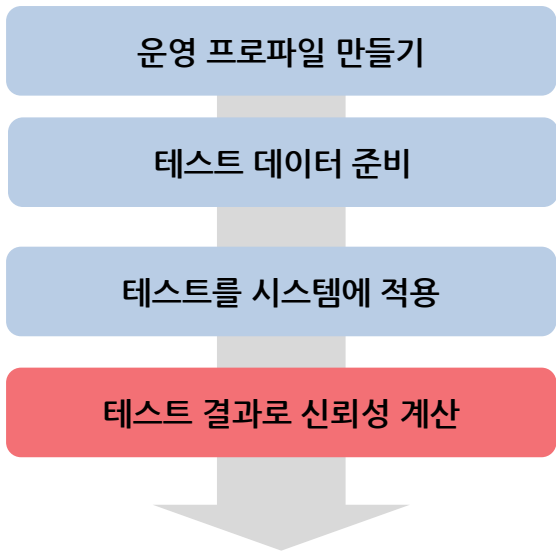
[3] 중대한 시스템의 검증

1. 신뢰성 검증(계속)

- ◆ 신뢰성 검증 수행 절차(계속)
 - 신뢰성 검증을 위해 4단계를 수행(계속)



- 이 데이터 및 발생한 고장의 수와 유형을 세어 시스템을 테스트 및 고장시간 등도 기록하여 체크



- 통계적으로 유의한 횟수의 고장이 관측된 후 소프트웨어 신뢰성을 계산하고 적절한 신뢰성 척도 값을 산출

❖ 학습내용

[3] 중대한 시스템의 검증

2. 안전성 보증

- ◆ 소프트웨어 안전성 보증
 - 소프트웨어 위험 요소 제거를 통해 소프트웨어 오류로 인한 **시스템의 사고를 예방**하는 것
 - 소프트웨어의 규모가 커지고 복잡해지면서 소프트웨어의 기능적 실패(Failure)를 만들어 내는 **위험(Hazard)요소들을 분석하기가 힘들**
 - 금융, 자동차, 철도, 항공, 전력, 국방, 의료, 교육 등 대부분 분야에서 SW 의존도가 높아짐에 따라, SW 오류로 인한 **사고의 피해 범위와 규모가 확대**됨
 - 이런 기능적 실패는 곧 큰 사고로 이어지게 되기 때문에 이러한 소프트웨어의 기능적 실패를 제거하고 안전성과 신뢰성을 높이기 위해 **소프트웨어의 안전성**에 대한 관심이 높아짐
 - 이러한 소프트웨어 안전성 확보를 위하여 **프로세스, 도구 활용, 안전성 평가 모델, 관련 인력 확보** 등의 문제해결이 필요함

◆ 안전성 보증 검토사항

신뢰성

테스트에 의하여 측정된 지표임

안전성

측정된 신뢰성과 함께 **시스템, 개발환경, 개발프로세스와의 복합적인 관계를 검토**하여 시스템이 안전한지를 보증하는 것

- 시스템이 안전한가를 보증하기 위하여 다음과 같은 항목들을 기본적으로 검토함

- 1

정확하게 **의도된 함수인**지를 검토
- 2

유지보수가 가능하고, **이해가 가능한 구조인**가를 검토
- 3

알고리즘과 데이터 구조의 설계가 **명세화된 동작과 일치하는**가를 검증하기 위하여 검토

❖ 학습내용

[3] 중대한 시스템의 검증

2. 안전성 보증(계속)

◆ 안전성 보증 검토사항(계속)

- 시스템이 안전한가를 보증하기 위하여 다음과 같은 항목들을 기본적으로 검토함(계속)

4 코드와 알고리즘 및 데이터 구조 설계와의 **일관성**을 검토

5 시스템 시험사례의 **적합성**을 검토

3. 보안성 평가

◆ 보안성 개념

- 중대한 시스템의 대부분은 **네트워크**를 통해 상호 시스템 자원간 **연결**되어 있음
- 이에 따라 보안허점이 있는 경우 중대한 시스템의 대한 공격에 대하여 해당 시스템 뿐만 아니라 연결된 다른 시스템에 대하여도 중요사항들이 **유출**될 수 있음
- 당연히 중대한 시스템은 **보안(정보보호)**을 고려한 시스템 검증 필요
- 보안위협은 시스템과 **데이터의 기밀성, 무결성, 가용성에 대한 위협**을 의미하며 이에 대한 통제가 잘 되어 있는가를 소프트웨어 및 시스템의 검증 과정에서 실행하여야 함
- 이러한 위협에 대하여 시스템 보안을 향상시키기 위하여 통제해야 함

공격이 성공할 수 없도록 보장함

공격을 탐지하고 격퇴할 의도를 가짐

문제로부터 복구를 지원할 수 있도록 함

❖ 학습내용

[3] 중대한 시스템의 검증

3. 보안성 평가(계속)

◆ 보안위협 유형

- 보안위협: 시스템과 데이터의 기밀성, 무결성, 가용성에 대한 위협

기밀성의 위협(Confidentiality)

기밀성이 보장되지 않으면 해당 정보에 대하여 접근 권한이 없는 사람이나 프로그램 등을 이용하여 정보가 누출될 수 있음

무결성의 위협(Integrity)

무결성이 보장되지 않으면 소프트웨어나 데이터를 손상시키거나 훼손할 수 있음

가용성의 위협(Availability)

가용성이 보장되지 않으면 소프트웨어나 데이터에 접근하여 정상적인 일을 수행하여야 하는 권한이 있는 사용자도 접근을 제한당할 수 있음

❖ 학습내용

[3] 중대한 시스템의 검증

3. 보안성 평가(계속)

◆ 보안위협 점검

- 보안성 점검을 위한 4가지의 접근법

1 경험기반검증

- 시스템 검증 팀은 알려진 공격유형에 따라 **시스템 취약점을 분석**
- 이 경우 대부분 **도구기반검증방법과 병행**하여 실행
- 알려진 보안문제의 **체크리스트**를 생성할 수도 있으며, 모든 시스템 문서를 참고하고, 오류와 누락 등이 점검되기 때문의 테스트활동과 병행하여 수행되기도 함

2 도구기반검증

- 패스워드 검사기, 스캐닝 등의 **보안점검도구**를 이용하여 취약점을 분석
- 이 도구에는 이미 취약점에 대한 경험기반에 의한 패턴들이 내장되어 있음

3 타이거팀

- 어떤 팀이 구성되고 이 팀은 시스템 보안을 깨도록 목표를 줌
⇒ 이 팀을 **타이거 팀**이라 함
- 만일 팀 구성원들이 이전에 다른 시스템을 깨트린 경험이 있다면 이 방법은 매우 **효과적**일 수 있음

4 정형증명

- 시스템 구성자가 이미 예상되는 취약점들을 보강하여 작성한 **정형 보안 명세서**에 따라 보안성이 증명될 수 있음
- 하지만 이 경우는 많이 사용되지는 않음

❖ 핵심정리

1. 증명과 검증

- 증명은 요구사항 명세서의 내용은 분석 및 설계에 반영되었는지, 분석 및 설계내용이 프로그램에 구현되었는지 단계별로 확인하는 작업
- 검증은 개발된 목적물을 실행하여 처음 사용자가 요청한대로 구현되었는지 확인하는 과정
- 증명과 검증은 명세 검토, 테스트 등으로 수행

2. V-model

- 증명과 검증은 두 과정을 함께 사용하면 좋은 품질의 정보시스템이 구현가능하며 이 두 가지를 묶어서 V&V(Verification & Validation) 또는 V-model이라고 함

3. 중대한 시스템의 검증

- 기업이나 조직의 핵심업무를 수행하는 중대한 시스템(Legacy System)의 경우 보다 엄격한 테스트를 필요로 함
- 이를 위하여 신뢰성검증, 안정성 보증, 보안성 평가 과정을 집중 수행함