

Traceability

-James D. Palmer

컴퓨터공학부

200711433

배수진

Contents

- Introduction
- System life cycle for traceability management
- Need for traceability
- Problems and issues concerning traceability
- Contemporary traceability practices
- An ideal process for traceability
- Return on investment for traceability
- Current traceability tools
- Common tool characteristics
- Commercial case tools for traceability

1. Introduction

- Traceability란?

추적가능성

: 어떤 추상화 레벨의 모델 또는 코드가 보다 높은 추상화 레벨의 특정한 다른 모델로부터 기인하였음을 알고 인식할 수 있는 능력.

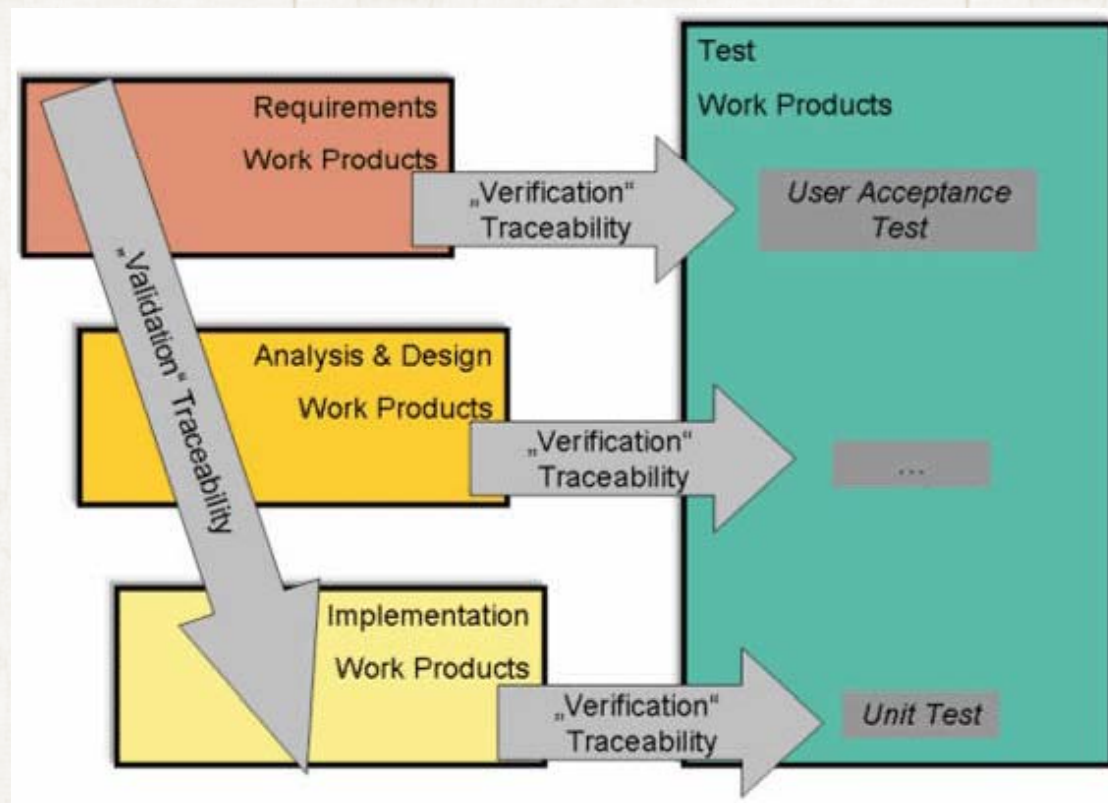
- 성공적인 시스템일 수록 사용자의 요구사항을 최대한으로 만족시킨다.

2. System life cycle for traceability management

- System life cycle ?
 - 시스템을 개발하는 단계들
 1. Requirements Analysis
 2. Design
 3. Build
 4. Testing
 5. Operations and Maintenance

3. Need for traceability

- 중요한 두 가지 목표는?
 - verification 과 validation



4. Problems and issues concerning traceability

- Trace 과정에서 맞닥뜨리게 되는 문제들은 프로젝트 자체의 복잡성으로부터 기인.
- 크고 복잡한 시스템일 수록 어려움이 많을 수 있음.
- 그러므로, 복잡한 System일 수록 공통요소에 초점을 맞추어야 한다.

5. Contemporary traceability practices

- 전형적인 trace의 output은 traceability matrix로 나타냄.
- Traceability matrix는 high level requirements들을 각각 연결하고 또 모든 다른 requirements들과 연결시킨다.

Traceability Example

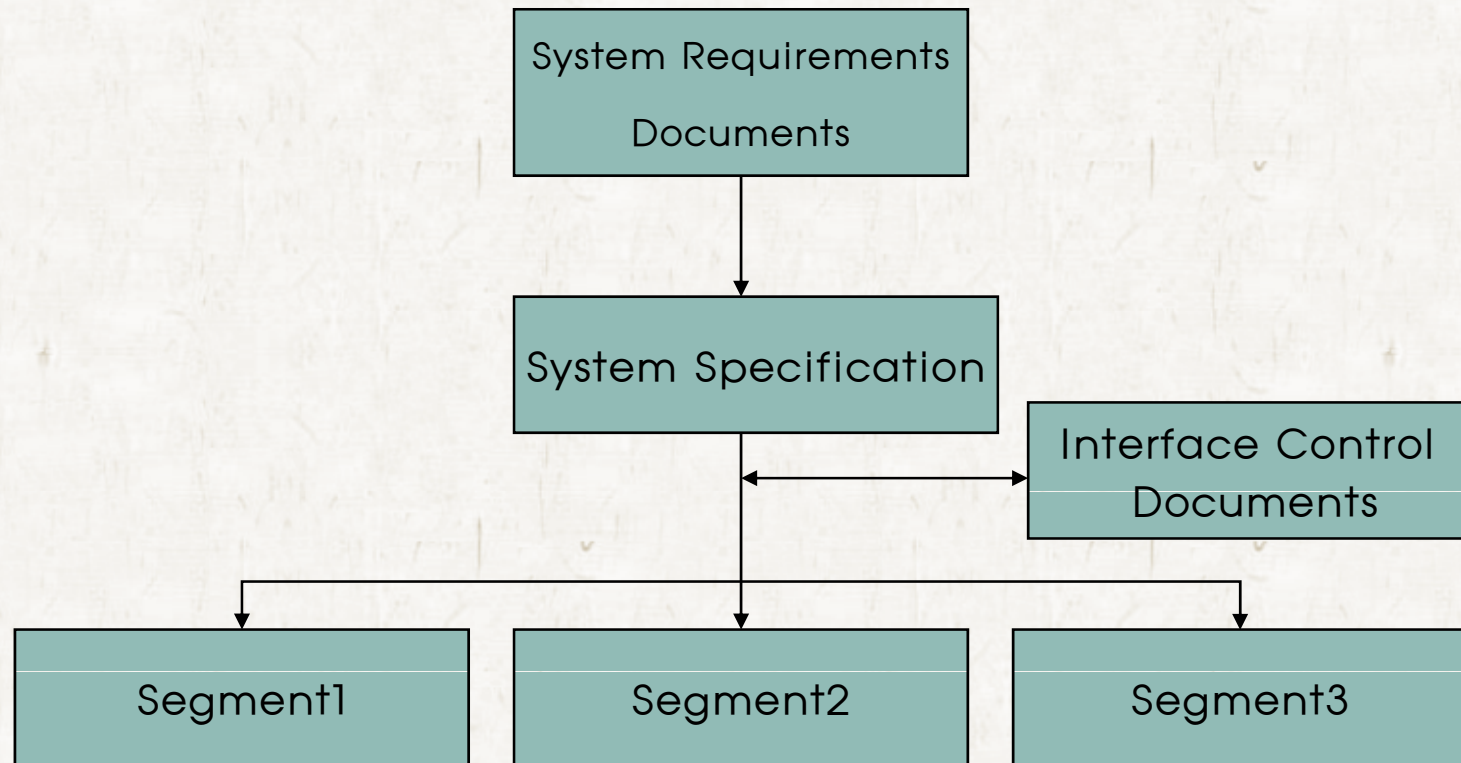
Traceability Matrix

| SRD | SS | Seg 1 | Seg 2 | Seg 3 | ICD |
|---------|---------|-----------|-----------|-----------|-----------|
| 3.1.2.1 | 3.3.4.5 | 3.2.2.5.6 | 3.5.3.2 | | 3.1.4.6.7 |
| | 3.3.4.6 | 3.2.2.5.7 | | | 3.1.4.6.8 |
| | | 3.4.5.6.2 | | | 3.1.4.6.9 |
| 3.4.3.1 | 3.6.7.2 | 3.5.2.5.1 | 3.7.4.3.1 | 3.6.4.5.2 | 3.3.2.4.5 |
| | 3.8.4.2 | | 3.7.4.3.2 | | 3.3.2.4.7 |

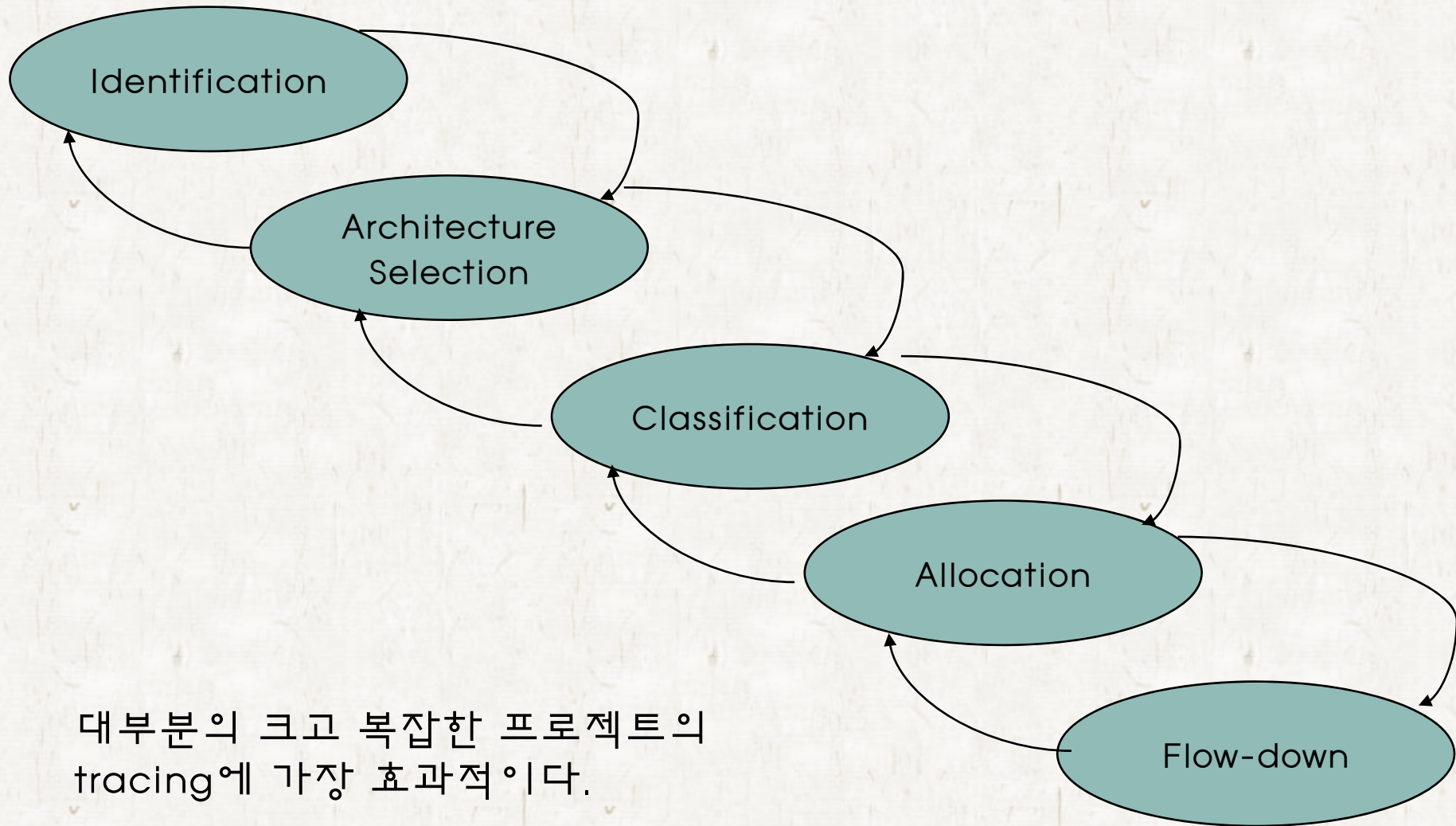
Traceability Example

- SRD – System Requirements Document
 - High level requirements
 - 사용자에게 의해 만들어짐
- SS – System Specification
 - More detailed requirements
 - 개발자의 해석
- Segments
 - SS를 보다 상세한 부분들로 나눔

Traceability Example



6. An ideal process for traceability



대부분의 크고 복잡한 프로젝트의 tracing에 가장 효과적이다.

7. Return on investment for traceability

- Traceability 에 투자한 것에 대한 성과를 정확히 측정하기란 사실상 불가능.
- 측정하기 위해서는? 추정!!
 - 다양한 초안들을 준비.
 - 프로세스 개발의 다양한 면에서의 실패에 따른 결과를 simulation
 - 이러한 failure들을 피하는 것에 대한 이익을 추정해 보아야 함.
 - 위험 요소들은 잠재적인 이익을 산출할 때 고려해야 하며, 프로젝트가 사용자의 요구를 충족시키지 못할 경우도 포함.

8. Current traceability tools

- 전형적으로 syntactic language components 들이 이용된다.
 - Hypertext linking
 - Unique identifiers
 - Syntactical similarity coefficients
 - Combinations of these
- 문제점은?
 - 단어가 어떠한 의미로 쓰였는지, 문맥상의 의미가 어떠한지를 파악하지 못함.
- 현재는 traceability를 위한 tool의 표준이 정해져 있지 않다.

9. Common tool characteristics

- 이해가 잘 되어야 한다.
- 개발자가 이용할 개발 환경의 특성과 match 되어야 한다.
- 다양한 활동과 서비스에 대해 유연성과 가용성을 지녀야 한다.
- 완전히 자동화 되어서는 안 된다.
 - Human decision은 classification과 system architecture design에 필수적이기 때문!!

10. Commercial case tools for traceability

- Single discipline
 - Cadre TeamWork for Real-Time Structured Analysis (CADRE)
- Multiple discipline
 - Requirements Traceability Manager (RTM)
 - SLATE
 - DOORS

-END-

감사합니다,