SECOND-GUESSING IN TRACING TASKS CONSIDERED HARMFUL?

20 MARCH 2018

BHUSHAN CHITRE , *JANE HUFFMAN HAYES* UNIVERSITY OF KENTUCKY

ALEXANDER DEKHTYAR, CALIFORNIA POLYTECHNIC STATE UNIVERSITY

Utrecht, The Netherlands, 19 – 22 March
!REFSQ
24th Intl. Working Conference on Requirements Engineering: Foundation for Software Quality
2018
Utrecht University

OUTLINE

BACKGROUND AND MOTIVATION

PROPOSED APPROACH

PRELIMINARY STUDY

CONCLUSIONS

FUTURE WORK

BACKGROUND

Requirements tracing – "ability to describe and follow life of requirement in both forward and backward directions"*

Trace matrix - collection of trace links, "specified association between pair of artifacts, one comprising source and one comprising target."+

Tracing between artifacts:

- Requirements to design
- Test cases to requirements
- Code to requirements

*Gotel, O. C. Z. and Finkelstein A. C. W., An analysis of the requirements traceability problem, Proceedings of the 1st International Conference on Requirements Engineering (ICRE '94), IEEE Computer Society Press, Colorado Springs, Colorado, USA, pp. 94-101, April 18-22 1994.

+Gotel, O., Cleland-Huang, J., Huffman Hayes, J., Zisman, A., Egyed, A., Grünbacher, P., Dekhtyar, A., Antoniol, G., Maletic, J. and Mäder, P. Traceability fundamentals. Chapter 1 in Cleland-Huang, J., Gotel, O. and Zisman, A. (Eds.) Software and systems traceability, Springer, 2012, pp.3–22.

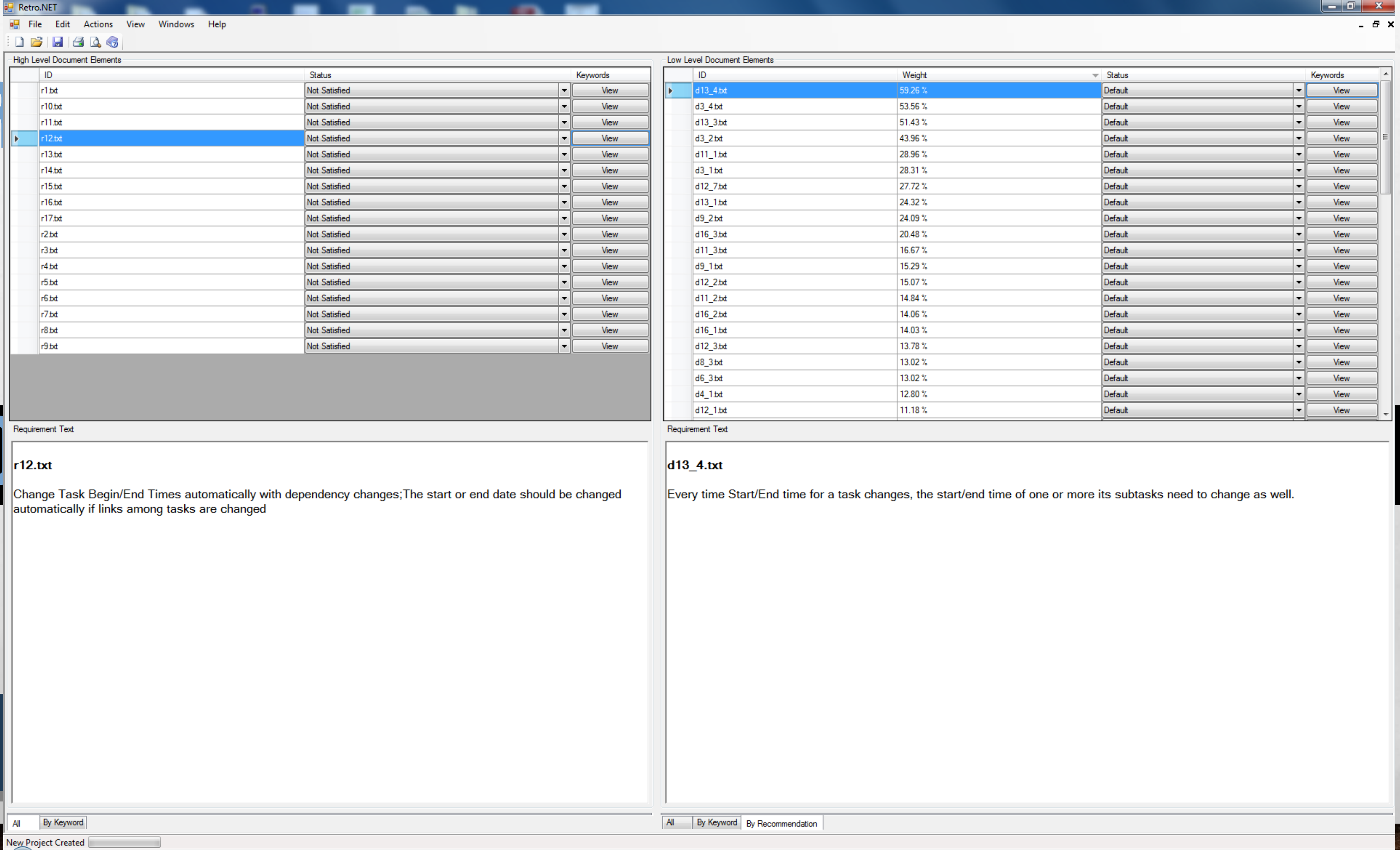
PROBLEM

- Automated methods/tools for candidate trace matrix (TM)
  - Information retrieval based and other techniques
  - Not 100 % accurate
  - Often retrieve unrelated items (false links)

SOLUTION

- Candidate TM verified by ***human analysts***

But **certain analyst behaviors ---> decreased accuracy**



High Level Document Elements

ID	Status	Keywords
r1.txt	Not Satisfied	View
r10.txt	Not Satisfied	View
r11.txt	Not Satisfied	View
r12.txt	Not Satisfied	View
r13.txt	Not Satisfied	View
r14.txt	Not Satisfied	View
r15.txt	Not Satisfied	View
r16.txt	Not Satisfied	View
r17.txt	Not Satisfied	View
r2.txt	Not Satisfied	View
r3.txt	Not Satisfied	View
r4.txt	Not Satisfied	View
r5.txt	Not Satisfied	View
r6.txt	Not Satisfied	View
r7.txt	Not Satisfied	View
r8.txt	Not Satisfied	View
r9.txt	Not Satisfied	View

Low Level Document Elements

ID	Weight	Status	Keywords
d13_4.txt	59.26 %	Default	View
d3_4.txt	53.56 %	Default	View
d13_3.txt	51.43 %	Default	View
d3_2.txt	43.96 %	Default	View
d11_1.txt	28.96 %	Default	View
d3_1.txt	28.31 %	Default	View
d12_7.txt	27.72 %	Default	View
d13_1.txt	24.32 %	Default	View
d9_2.txt	24.09 %	Default	View
d16_3.txt	20.48 %	Default	View
d11_3.txt	16.67 %	Default	View
d9_1.txt	15.29 %	Default	View
d12_2.txt	15.07 %	Default	View
d11_2.txt	14.84 %	Default	View
d16_2.txt	14.06 %	Default	View
d16_1.txt	14.03 %	Default	View
d12_3.txt	13.78 %	Default	View
d8_3.txt	13.02 %	Default	View
d6_3.txt	13.02 %	Default	View
d4_1.txt	12.80 %	Default	View
d12_1.txt	11.18 %	Default	View

Requirement Text

r12.txt

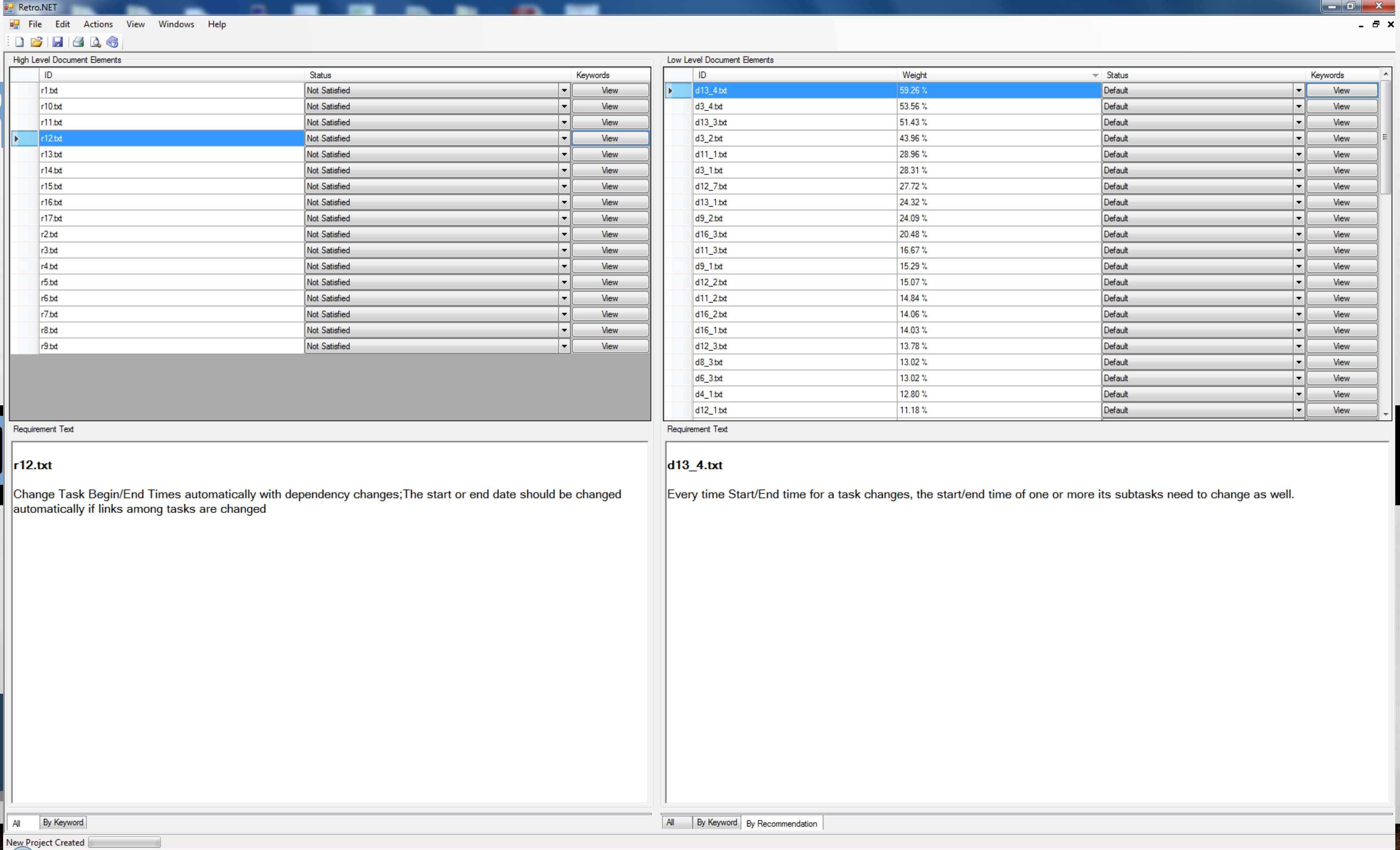
Change Task Begin/End Times automatically with dependency changes;The start or end date should be changed automatically if links among tasks are changed

Requirement Text

d13_4.txt

Every time Start/End time for a task changes, the start/end time of one or more its subtasks need to change as well.

New Project Created



High Level Document Elements

ID	Status	Keywords
r1.txt	Not Satisfied	View
r10.txt	Not Satisfied	View
r11.txt	Not Satisfied	View
r12.txt	Not Satisfied	View
r13.txt	Not Satisfied	View
r14.txt	Not Satisfied	View
r15.txt	Not Satisfied	View
r16.txt	Not Satisfied	View
r17.txt	Not Satisfied	View
r2.txt	Not Satisfied	View
r3.txt	Not Satisfied	View
r4.txt	Not Satisfied	View
r5.txt	Not Satisfied	View
r6.txt	Not Satisfied	View
r7.txt	Not Satisfied	View
r8.txt	Not Satisfied	View
r9.txt	Not Satisfied	View

Low Level Document Elements

ID	Weight	Status	Keywords
d13_4.txt	59.26 %	Default	View
d3_4.txt	53.56 %	Default	View
d13_3.txt	51.43 %	Default	View
d3_2.txt	43.96 %	Default	View
d11_1.txt	28.96 %	Default	View
d3_1.txt	28.31 %	Default	View
d12_7.txt	27.72 %	Default	View
d13_1.txt	24.32 %	Default	View
d9_2.txt	24.09 %	Default	View
d16_3.txt	20.48 %	Default	View
d11_3.txt	16.67 %	Default	View
d9_1.txt	15.29 %	Default	View
d12_2.txt	15.07 %	Default	View
d11_2.txt	14.84 %	Default	View
d16_2.txt	14.06 %	Default	View
d16_1.txt	14.03 %	Default	View
d12_3.txt	13.78 %	Default	View
d8_3.txt	13.02 %	Default	View
d6_3.txt	13.02 %	Default	View
d4_1.txt	12.80 %	Default	View
d12_1.txt	11.18 %	Default	View

Requirement Text

r12.txt

Change Task Begin/End Times automatically with dependency changes;The start or end date should be changed automatically if links among tasks are changed

Requirement Text

d13_4.txt

Every time Start/End time for a task changes, the start/end time of one or more its subtasks need to change as well.

New Project Created

PROBLEM

- Automated methods/tools for candidate trace matrix (TM)
    - Information retrieval based and other techniques
    - Not 100 % accurate
    - Often retrieve unrelated items (false links)

SOLUTION

- Candidate TM verified by ***human analysts***

But certain analyst behaviors ---> decreased accuracy

MOTIVATION

Prior work [1, 2] shows these lead to errors of judgement

- Long time to decide
- Revisiting a link (backtracking)

Could be tied to human decision making systems – System 1 (S1) – fast, instinctive thinking and System 2 (S2) – slow, deliberate, logical thinking – above behaviors belong to S2

[1] J. Hayes, A. Dekhtyar, and S. Sundaram, "Advancing candidate link generation for requirements tracing: The study of methods," *IEEE transactions on Software Engineering,*, Vol. 32, no. 1, pp. 4-19, Jan. 2006.

[2] Wei-Keat Kong and Jane Huffman Hayes, "Proximity-based traceability: An empirical validation using ranked retrieval and set-based measures". Published in the Proceedings of Empirical Research in Requirements Engineering workshop (EMPIRE2011), an RE 2011 workshop.

PROPOSED APPROACH/RESEARCH QUESTIONS

RQ1: Analyst behaviors that reliably lead to making errors, and where fall on Kahneman's thinking system dichotomy (S1, S2)? (Phase 1 – discover)

RQ2: What enhancements for automated tracing tools can be designed to curb unwanted behaviors? (Phase 2 – enhance)

RQ3: Improvement in accuracy of final TM constructed by analysts using enhanced software? (Phase 3 – evaluate)

DISCOVERY OF ANALYST BEHAVIORS

- Replicate experiment of Kong et al. (RETRO-LOGGING) – more data
- Classify data per Kahneman dichotomy
- Is TM analysis performed best within System 1 decision-making?

DEVELOPMENT OF SOFTWARE ENHANCEMENTS

- For each behavior discovered, design feature(s) to enhance RETRO.NET
  - Warnings
  - Prohibitions
  - Restructuring

STUDY OF THE IMPACT

- Second replication of Kong et al. but use experimental and control groups
  - Do software enhancements actually curb behaviors?
  - Is decrease in unwanted behaviors accompanied by decrease in number of errors analysts make?

PRELIMINARY STUDY

Unwanted behavior/Software enhancements

- <u>Long time to decide</u>   analyst more than average time on link decision, prompt with warning
- <u>Backtracking</u>   analyst re-visit previous link decision then prompt with warning

Fourteen subjects in two groups

- RETRO.NET control (non-enhanced) – five participants finished
- RETRO.NET experimental (enhanced) – nine participants finished

“Changestyle” – 32 reqts to 17 tests

RESULTS

Measured precision, recall, f2 - measure, lag of final TM and time it took to complete task (minutes) – experimental better on most measures *not* time

Group	Aggregation	Prec.	Recall	*F2*	Lag	Time	Delta (TP)	Delta (FP)
RETRO	actual	0.063	1	0.251	1.1	NA	N/A	N/A
Control	Mean	0.083	0.776	0.262	2.552	75	1.6	53
	Median	0.068	0.971	0.254	1.96	60	0	9
Experimental	Mean	0.156	0.961	0.329	1.85	82	1.222	118.7
	Median	0.069	0.971	0.283	1.765	86	1	59.5

DISCUSSION/CONCLUSIONS

- Basic prompts might avert analysts from undesired behaviors – at expense of time
- Identified items for future study:
  - Collect number of times prompts appear
  - Collect amount of time analyst takes when dismissing, reacting to prompt
  - Track action taken by analyst after prompt
  - Track number of false positives (etc.) added and removed
  - Potentially track each individual true positive link displayed by RETRO.NET to learn its final disposition

FUTURE WORK

- Phase 1: Discover analyst behavior
- Phase 2: Enhance software to curtail/validate curtailment of unwanted behavior
- Phase 3
  - Undertake wider scope similar study
  - Collect richer data from larger groups
  - Undertake statistical analysis

ACKNOWLEDGMENT

- We thank participants from software engineering classes who participated in study
- We thank NASA and NSF as prior grants funded the development of RETRO.NET
- We thank Jody Larsen, the developer of RETRO.NET
- We thank NSF for partially funding this work under grants CCF-1511117 and CNS-1642134

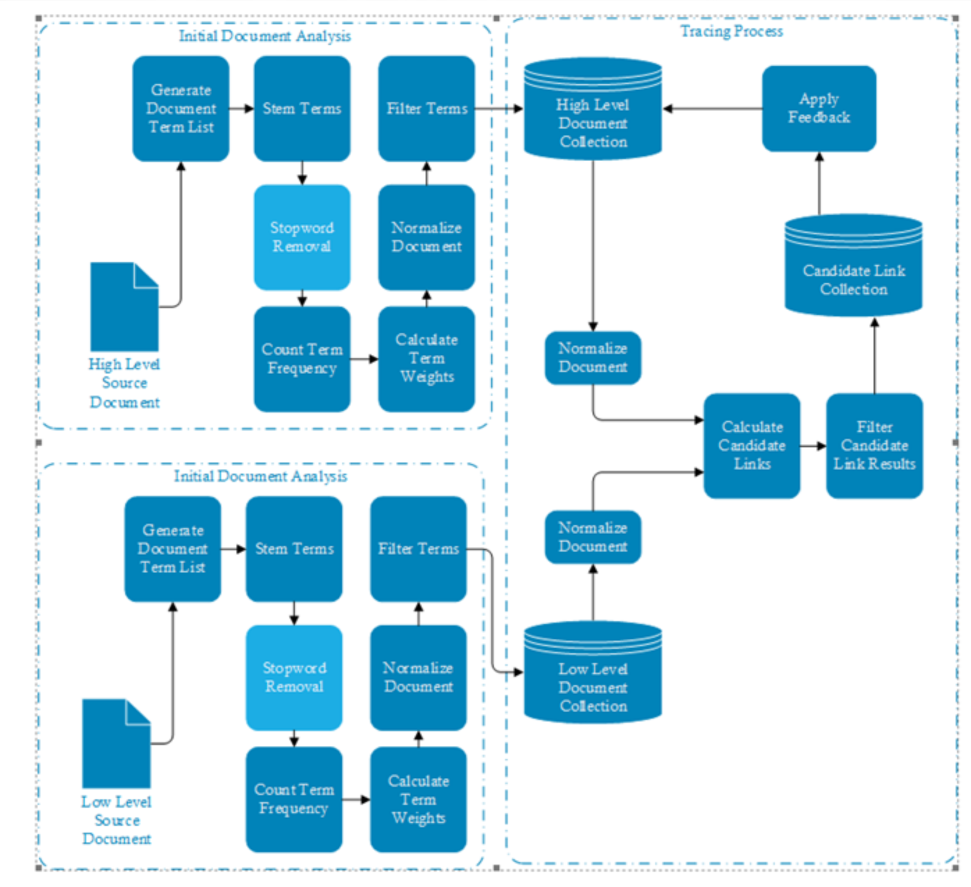
REFERENCES

1. David Cuddeback, Alex Dekhtyar, Jane Huffman Hayes. Automated Requirements Traceability: The Study of Human Analysts. Proceedings of IEEE International Conference on requirements Engineering (RE), September 2010, Sydney, Australia, 231-240.
2. Alex Dekhtyar, Olga Dekhtyar, Jeff Holden, Jane Huffman Hayes, David Cuddeback, Wei-Keat Kong. On Human Analyst Performance in Assisted Requirements Tracing:
Statistical Analysis. In the Proceedings of IEEE International Conference on Requirements Engineering (RE) 2011, Trento, Italy.
3. Jane Huang, Orlena Gotel, and Andrea Zisman. 2014. Software and Systems Traceability. Springer Publishing Company, Incorporated.
4. Markus Borg, Per Runeson, and Anders Ardö. 2014. Recovering from a decade: a systematic mapping of information retrieval approaches to software traceability. Empirical Softw. Engg. 19, 6 (December 2014), 1565-1616.
5. Jane Huffman Hayes, Alex Dekhtyar, Senthil Sundaram, Ashlee Holbrook, Sravanthi Vadlamudi, Alain April, REquirements TRacing On target (RETRO): Improving Software Maintenance through Traceability Recovery. Innovations in Systems and Software Engineering: A NASA Journal (ISSE) 3(3): 193-202 (2007).
6. Wei-Keat Kong, Jane Hayes, Alex Dekhtyar, Jeff Holden, (2011), How Do We Trace Requirements? An Initial Study of Analyst Behavior in Trace Validation Tasks, in Proceedings, 4th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE'2011), May 2011.
7. J. Hayes, A. Dekhtyar, and S. Sundaram, "Advancing candidate link generation for requirements tracing: the study of methods," IEEE Transactions on Software Engineering., vol. 32, no. 1, pp. 4-19, Jan. 2006.
8. D. Kahneman, Thinking, Fast and Slow. New York, NY, USA: Farrar, Straus, 2011.

THANK YOU!

QUESTIONS?

HOW PTIONAL IF NEEDED)

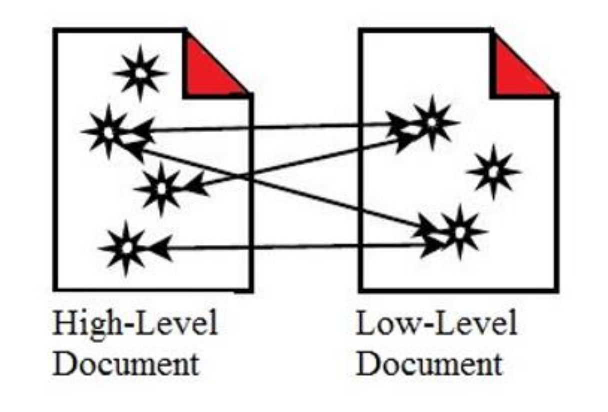

Initial Document Analysis
Generate Document Term List
Stem Terms
Filter Terms
Stopword Removal
Normalize Document
Count Term Frequency
Calculate Term Weights
High Level Source Document
Tracing Process
High Level Document Collection
Apply Feedback
Candidate Link Collection
Normalize Document
Calculate Candidate Links
Filter Candidate Link Results
Normalize Document
Low Level Document Collection
Initial Document Analysis
Generate Document Term List
Stem Terms
Filter Terms
Stopword Removal
Normalize Document
Count Term Frequency
Calculate Term Weights
Low Level Source Document

Analysis and Tracing Process

Credit: Jody Larsen, “High Performance automated traceability.”

INTRODUCTION:

- SAFETY CRITICAL SOFTWARE SYSTEMS – IMPORTANCE OF REQUIREMENTS
  - HIGH-LEVEL DOCUMENT
  - LOW-LEVEL DOCUMENTS
- AUTOMATED METHODS GENERATE CANDIDATE TMS USING INFORMATION RETRIEVAL METHODS



DEPENDENT AND INDEPENDENT VARIABLES

- The independent variables: different version of RETRO.NET "control" and "experimental."

- The dependent variables: precision, recall, f2-measure, lag and time to perform the experiment.

- Controlled variable: Answer set RTM of "ChangeStyle" dataset and "Retro.NET" tool.

IR MEASURES DEFINITIONS

$$Precision = \frac{|\#\ of\ Correct\ Links\ Returned|}{|\#\ of\ Returned\ Links|}$$

$$Recall = \frac{|\#\ of\ Correct\ Links\ Returned|}{|Total\ \#\ of\ Correct\ \ Links|}$$

***f* – measure:** is the harmonic mean of recall

The ***f*$_2$- measure**, i.e., *f* -measure for a = 2.

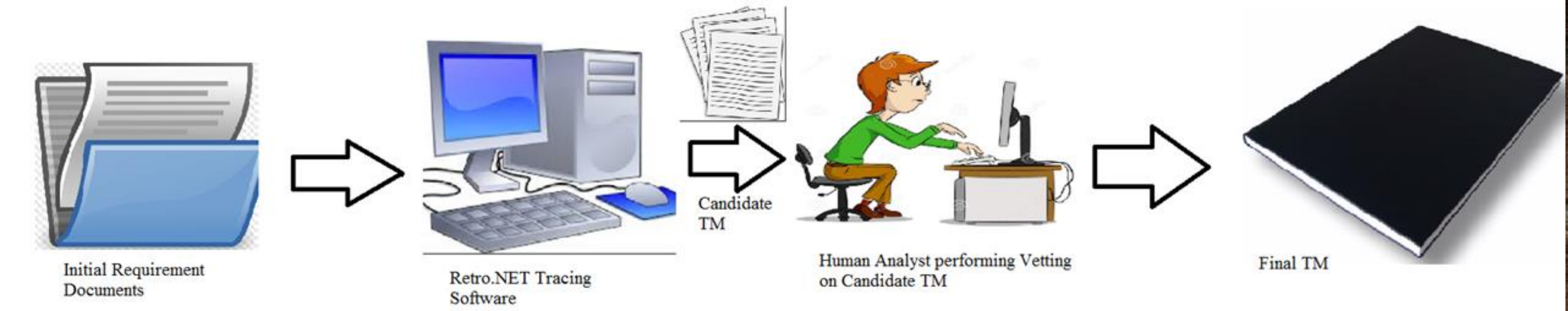
$$f_a = \frac{1 + a^2}{\frac{a^2}{recall} + \frac{1}{precision}}$$

Lag: Lag is a measure of the separation between true and false links. For a requirement q, (q, d) for true link. lag(q, d), the lag of an individual link (q, d), is the number of false links that have higher relevance scores than (q, d).

$$lag = \frac{\sum_{(a,d)\in\tau} lag(q,d)}{|\tau|}$$

HOW TRACING WORKS?

Tracing Task


Initial Requirement Documents
Retro.NET Tracing Software
Candidate TM
Human Analyst performing Vetting on Candidate TM
Final TM

THREATS TO VALIDITY

Internal validity:

- Tracing tool
- Human error,
- Hypothesis guessing,
- Personal bias in constructing of the answer set

Construct validity: There were minimal threats to construct validity as standard IR measures (precision, recall, f2 and etc.)

External validity: Experimental dataset

Conclusion validity: statistical analysis

Reliability validity: The study process is defined and easily repeatable.

HUMAN ANALYST RECRUITMENT

- We recruited Upper division software engineering computer science students.
- They signed the Informed consent and filled pre-study survey as a form of agreement to participate in our study.
- Held demo/training session to let users get familiar with tool and tracing process.
- Then they worked with testing dataset called “Moonlander” on their own time out the class with provided instructions.

RESULTS AND ANALYSIS

Total of 14 subjects participated in a preliminary study conducted in Spring 2017 at University of Kentucky.

We collected:

- Pre- and post-study survey
- Time logs (time to perform tracing)
- Final TM results (XML)

Out of 14 results

- 5 analysts were in control group (worked on non-enhanced RETRO.NET)
- 9 analysts were in experimental group (worked on enhanced RETRO.NET)

PROPOSED APPROACH/RESEARCH QUESTIONS

We propose three-step experimental study to:

1) Determine if there really are behaviors that lead to errors of judgement for analysts
2) Enhance the requirements tracing software to curtail such behaviors, and
3) Determine if curtailing such behaviors results in increased accuracy

THE STUDY

- Both groups used “changestyle” dataset - 32 requirements traced to 17 system tests
- Collected:
  - Pre- and post-study survey
  - Time logs (time to perform tracing)
  - Final TM results (XML)

Second-Guessing in Tracing Tasks Considered Harmful?

Bhushan Chitre , *Jane Huffman Hayes* University of Kentucky

Alexander Dekhtyar, California Polytechnic State University

Utrecht, The Netherlands, 19 – 22 March

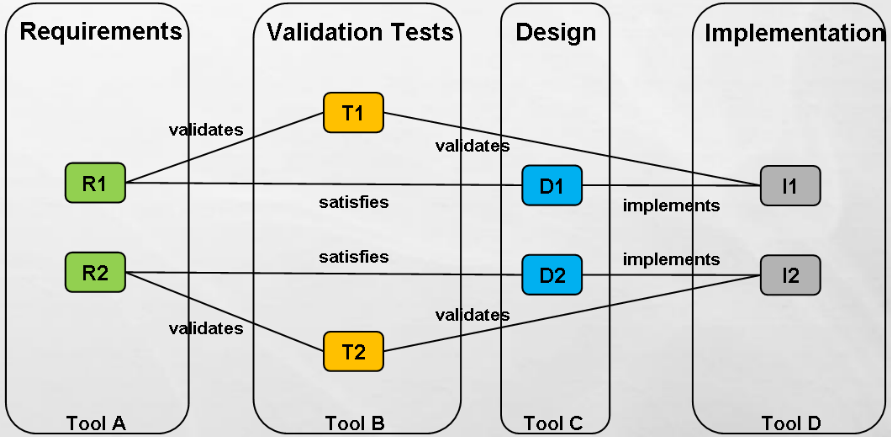
REFSQ 2018

24th Intl. Working Conference on Requirements Engineering: Foundation for Software Quality

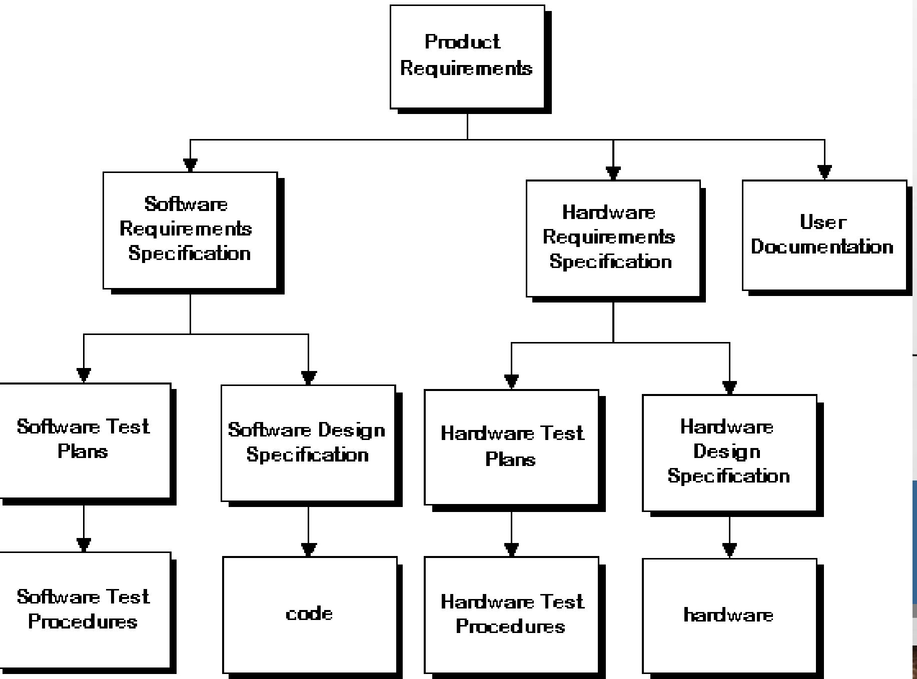
Utrecht University

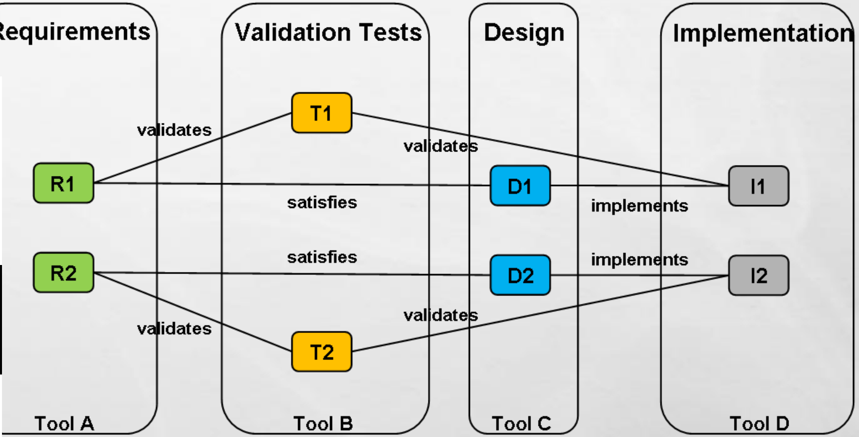
SOFTWARE ENGINEERING ARTIFACTS

- Guide and inform development
- Support verification and validation
- Relate to each other



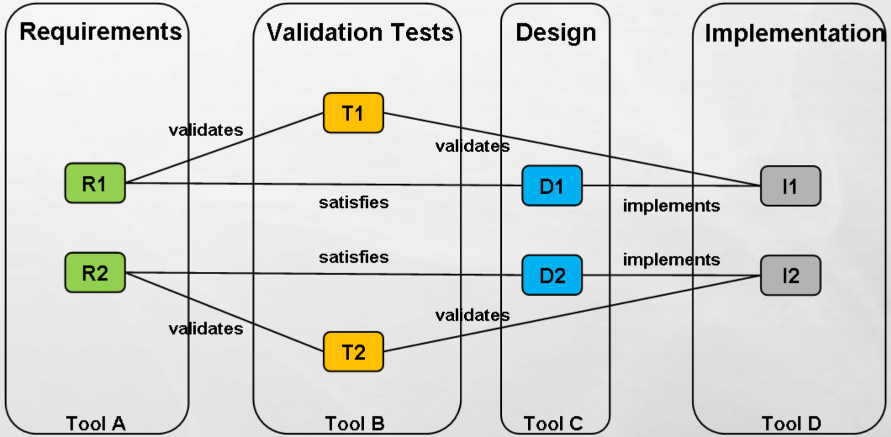
SOFTWARE ENGINEERING ARTIFACTS





SOFTWARE ENGINEERING ARTIFACTS

- **Guide and inform development**
- **Support verification and validation**
- **Relate to each other**



Req 1: When roll hold mode becomes the active mode, the roll hold reference shall be set to the actual roll attitude of the aircraft, except under the following conditions:

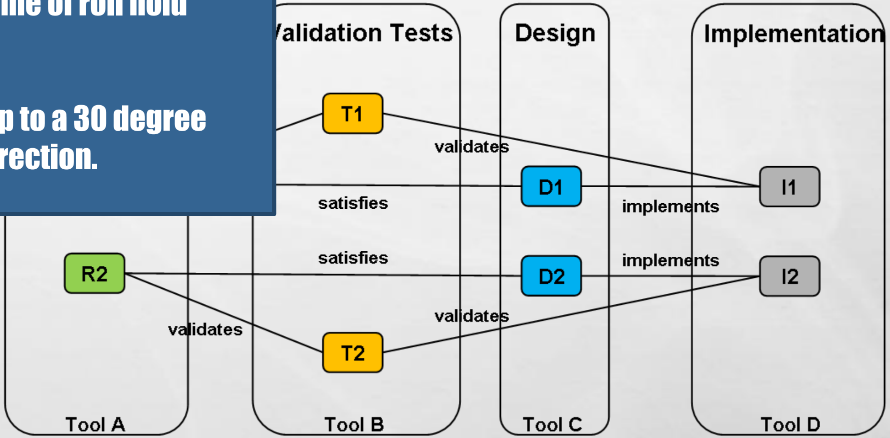
The roll hold reference shall be set to zero if the actual roll angle is less than 6 degrees in either direction, at the time of roll hold engagement.

The roll hold reference shall be set to 30 degrees in the same direction as the actual roll angle if the actual roll angle is greater than 30 degrees at the time of roll hold engagement.

The roll reference shall be set to the cockpit turn knob command, up to a 30 degree limit, if the turn knob is commanding 3 degrees or more in either direction.

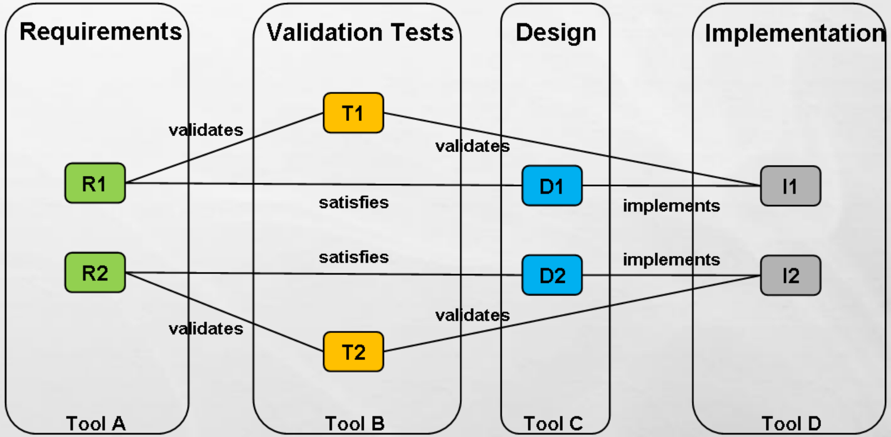
ARTIFACTS

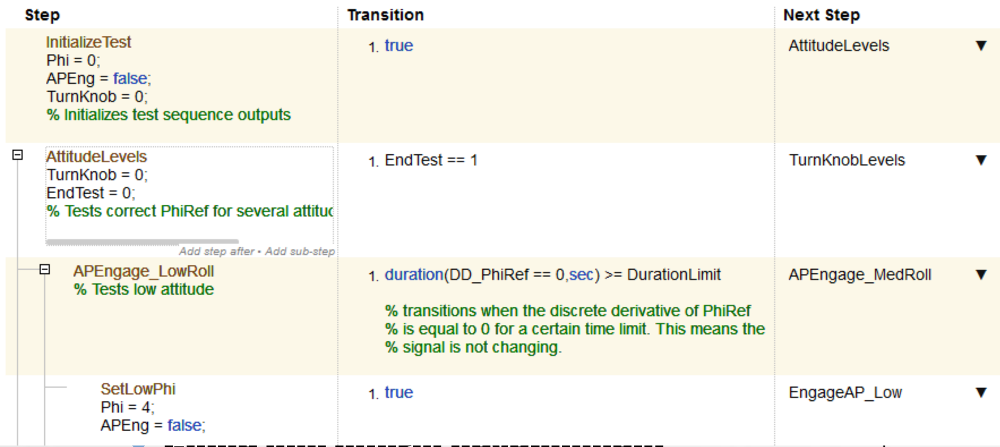
- Support verification and validation
- Relate to each other



SOFTWARE ENGINEERING ARTIFACTS

- Guide and inform development
- Support verification and validation
- Relate to each other





Step	Transition	Next Step

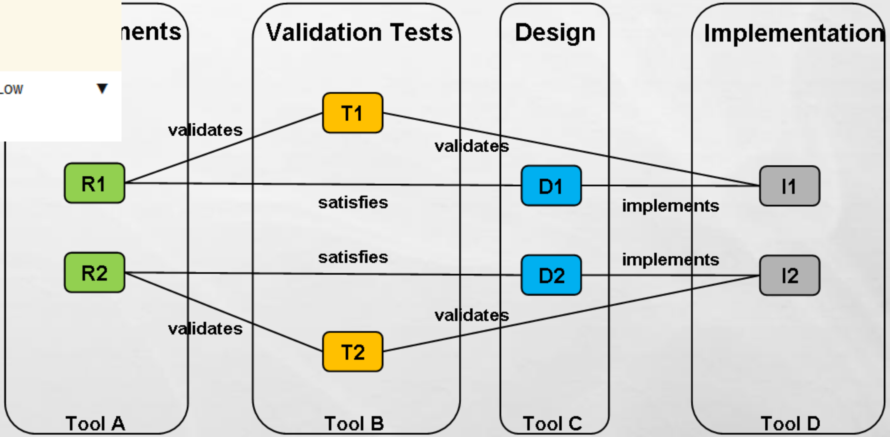
InitializeTest
Phi = 0;
APEng = false;
TurnKnob = 0;
% Initializes test sequence outputs
	1. true	AttitudeLevels ▼

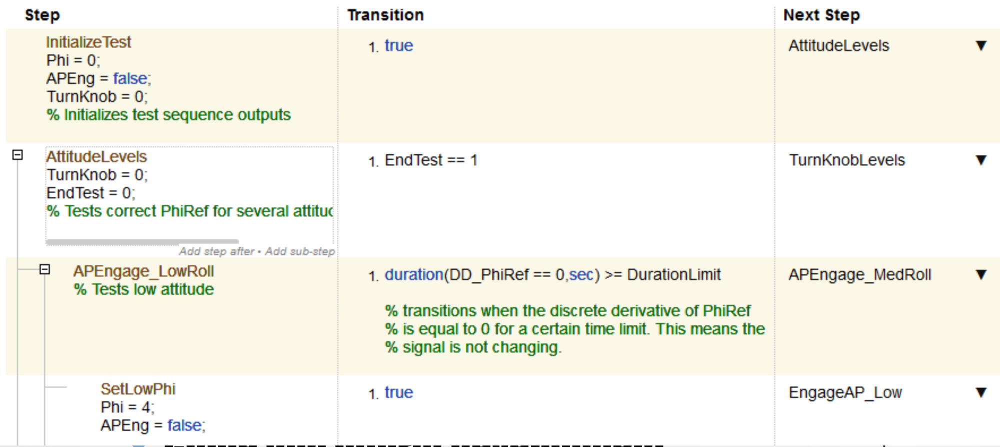
⊟ AttitudeLevels TurnKnob = 0; EndTest = 0; % Tests correct PhiRef for several attituc Add step after • Add sub-step	1. EndTest == 1	TurnKnobLevels ▼
⊟ APEngage_LowRoll % Tests low attitude	1. duration(DD_PhiRef == 0,sec) >= DurationLimit % transitions when the discrete derivative of PhiRef % is equal to 0 for a certain time limit. This means the % signal is not changing.	APEngage_MedRoll ▼

SetLowPhi
Phi = 4;
APEng = false;
	1. true	EngageAP_Low ▼

ING ARTIFACTS

- Guide and inform development
- Support verification and validation
- Relate to each other





Step	Transition	Next Step

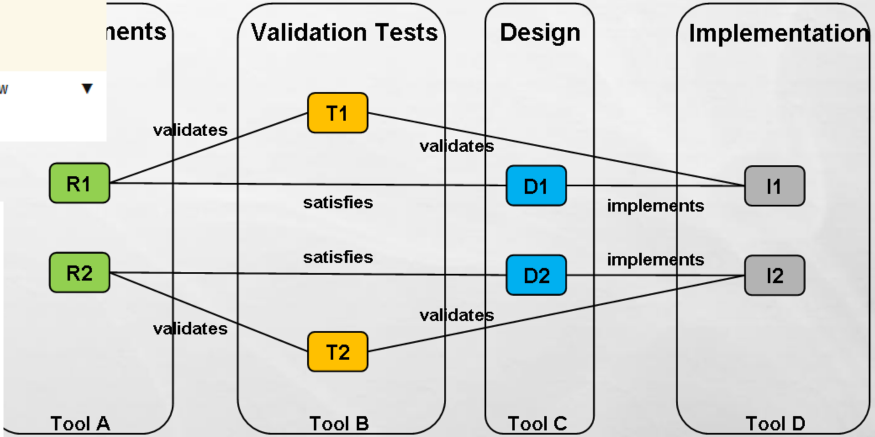
InitializeTest
Phi = 0;
APEng = false;
TurnKnob = 0;
% Initializes test sequence outputs
	1. true	AttitudeLevels ▼

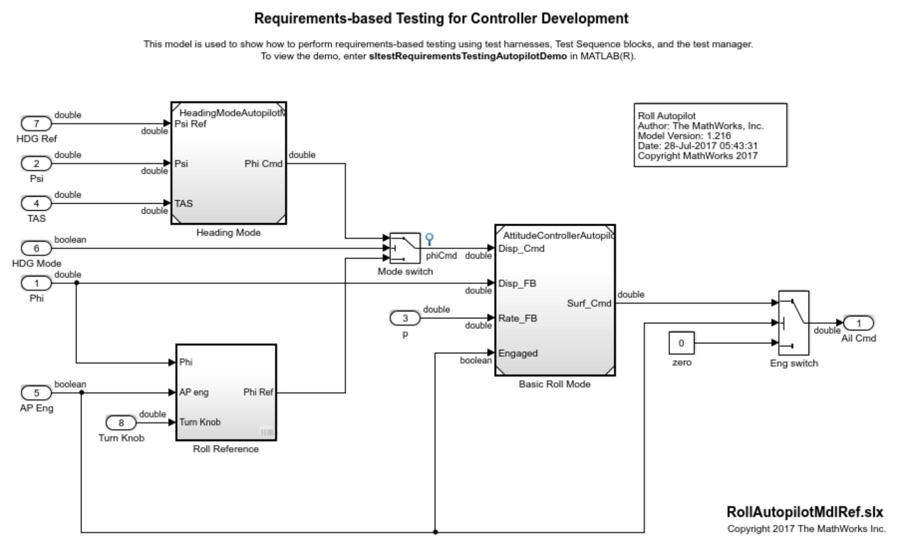
⊟ AttitudeLevels TurnKnob = 0; EndTest = 0; % Tests correct PhiRef for several attituc Add step after • Add sub-step	1. EndTest == 1	TurnKnobLevels ▼
⊟ APEngage_LowRoll % Tests low attitude	1. duration(DD_PhiRef == 0,sec) >= DurationLimit % transitions when the discrete derivative of PhiRef % is equal to 0 for a certain time limit. This means the % signal is not changing.	APEngage_MedRoll ▼

SetLowPhi
Phi = 4;
APEng = false;
	1. true	EngageAP_Low ▼

ING ARTIFACTS

- Guide and inform development





Requirements-based Testing for Controller Development

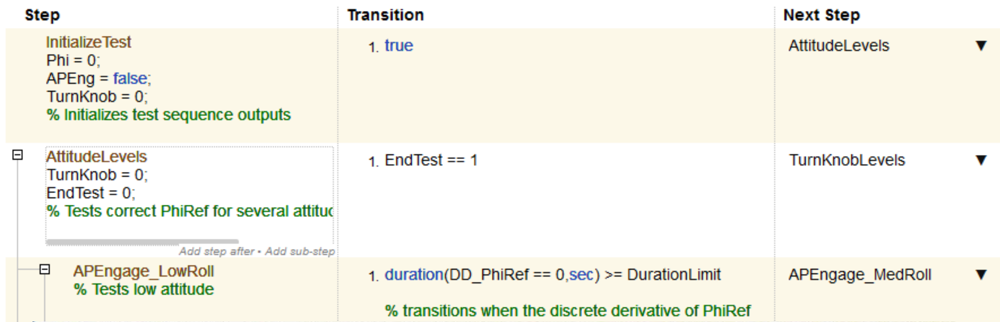
This model is used to show how to perform requirements-based testing using test harnesses, Test Sequence blocks, and the test manager.
To view the demo, enter **sltestRequirementsTestingAutopilotDemo** in MATLAB(R).

Roll Autopilot
Author: The MathWorks, Inc.
Model Version: 1.216
Date: 28-Jul-2017 05:43:31
Copyright MathWorks 2017

RollAutopilotMdlRef.slx

Copyright 2017 The MathWorks Inc.

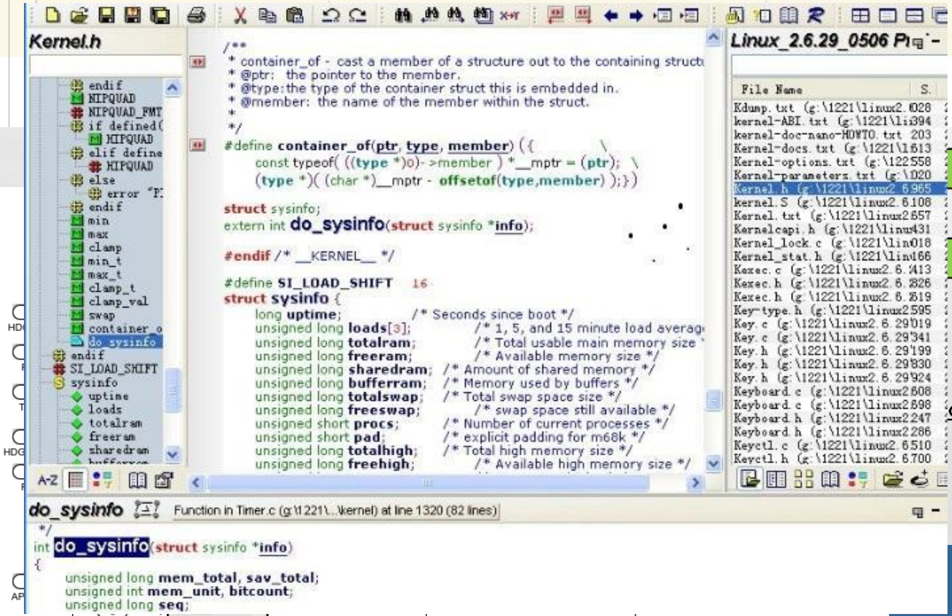
ING ARTIFACTS



Step	Transition	Next Step

InitializeTest
Phi = 0;
APEng = false;
TurnKnob = 0;
% Initializes test sequence outputs
	1. true	AttitudeLevels ▼

AttitudeLevels TurnKnob = 0; EndTest = 0; % Tests correct PhiRef for several attituc Add step after • Add sub-step	1. EndTest == 1	TurnKnobLevels ▼
APEngage_LowRoll % Tests low attitude	1. duration(DD_PhiRef == 0,sec) >= DurationLimit % transitions when the discrete derivative of PhiRef	APEngage_MedRoll ▼



Kernel.h

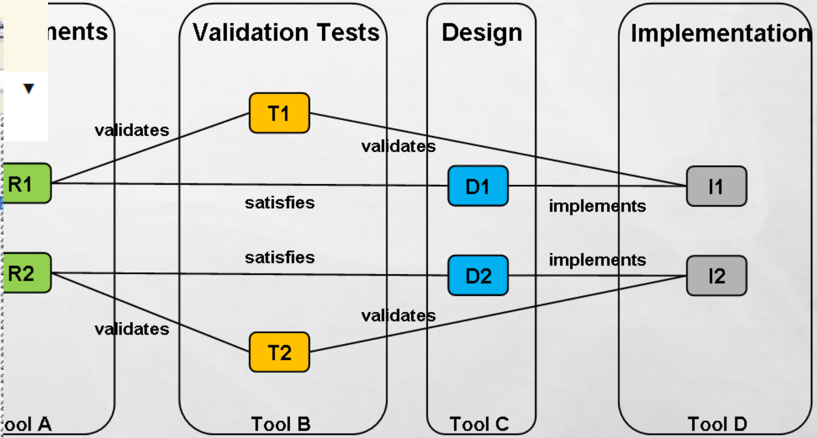
```
/**
 * container_of - cast a member of a structure out to the containing struct
 * @ptr:  the pointer to the member.
 * @type:the type of the container struct this is embedded in.
 * @member: the name of the member within the struct.
 *
 */
#define container_of(ptr, type, member) ({              \
        const typeof( ((type *)0)->member ) *__mptr = (ptr);  \
        (type *)( (char *)__mptr - offsetof(type,member) );})

struct sysinfo;
extern int do_sysinfo(struct sysinfo *info);

#endif /* __KERNEL__ */

#define SI_LOAD_SHIFT   16
struct sysinfo {
        long uptime;                    /* Seconds since boot */
        unsigned long loads[3];                 /* 1, 5, and 15 minute load averag
        unsigned long totalram;                 /* Total usable main memory size *
        unsigned long freeram;                  /* Available memory size */
        unsigned long sharedram;        /* Amount of shared memory */
        unsigned long bufferram;        /* Memory used by buffers */
        unsigned long totalswap;        /* Total swap space size */
        unsigned long freeswap;                 /* swap space still available */
        unsigned short procs;           /* Number of current processes */
        unsigned short pad;             /* explicit padding for m68k */
        unsigned long totalhigh;        /* Total high memory size */
        unsigned long freehigh;                 /* Available high memory size */
```

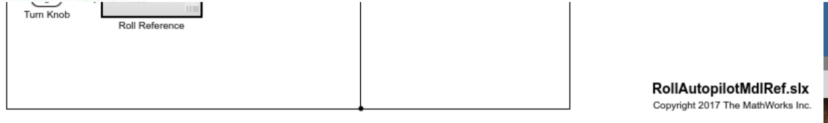
Linux_2.6.29_0506 Pr

```
File Name                                  S.
Kdump.txt (g:\1221\linux2.028
kernel-ABI.txt (g:\1221\li394
kernel-doc-nano-HOWTO.txt 203
Kernel-docs.txt (g:\1221\1613
Kernel-options.txt (g:\122558
Kernel-parameters.txt (g:\1020
Kernel.h (g:\1221\linux2.6.965
kernel.S (g:\1221\linux2.6.108
Kernel.txt (g:\1221\linux2.657
Kernelcapi.h (g:\1221\linu431
Kernel_lock.c (g:\1221\lin018
Kernel_stat.h (g:\1221\lin166
Kexec.c (g:\1221\linux2.6.413
Kexec.h (g:\1221\linux2.6.826
Kexec.h (g:\1221\linux2.6.619
Key-type.h (g:\1221\linux2.595
Key.c (g:\1221\linux2.6.29019
Key.c (g:\1221\linux2.6.29341
Key.h (g:\1221\linux2.6.29199
Key.h (g:\1221\linux2.6.29830
Key.h (g:\1221\linux2.6.29924
Keyboard.c (g:\1221\linux2.608
Keyboard.c (g:\1221\linux2.698
Keyboard.h (g:\1221\linux2.247
Keyboard.h (g:\1221\linux2.286
Keyctl.c (g:\1221\linux2.6.510
Keyctl.h (g:\1221\linux2.6.700
```

do_sysinfo   Function in Timer.c (g:\1221\...\kernel) at line 1320 (82 lines)

```
*/
int do_sysinfo(struct sysinfo *info)
{
        unsigned long mem_total, sav_total;
        unsigned int mem_unit, bitcount;
        unsigned long seq;
```





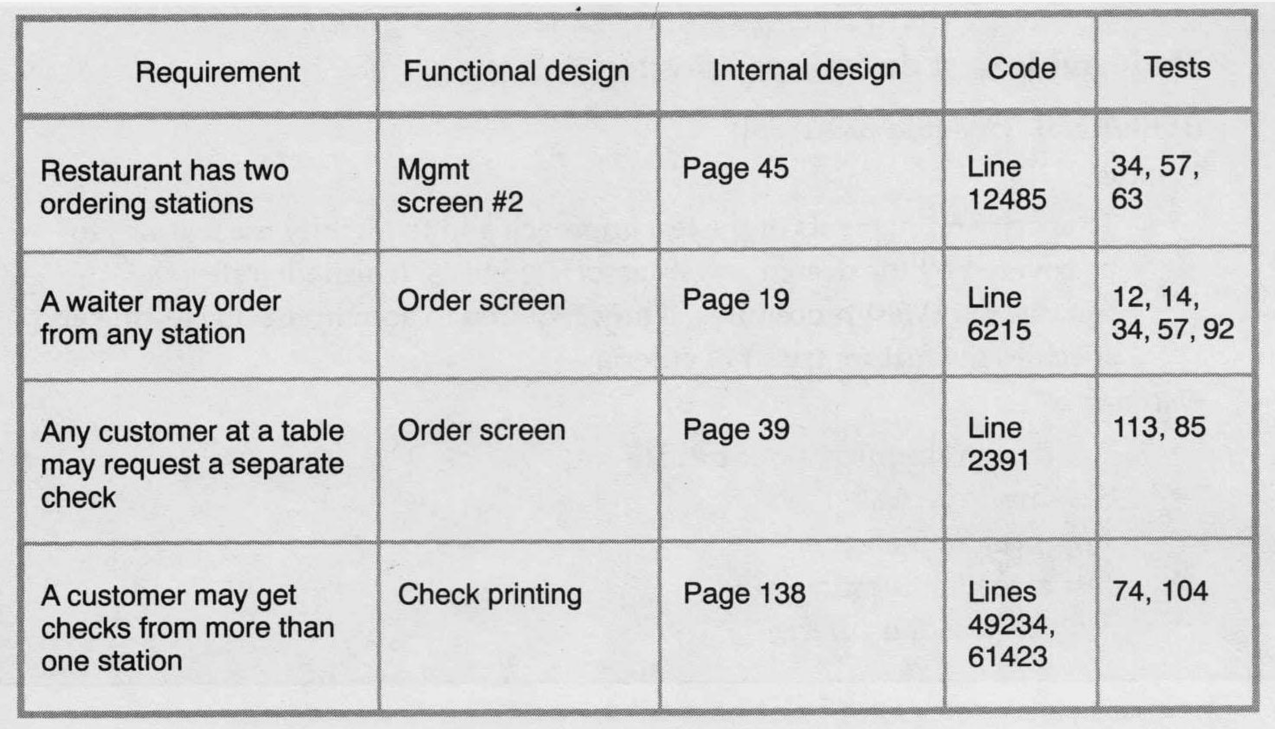
RollAutopilotMdlRef.slx

TRACE MATRIX

- Tracing to identify relationships
- Trace matrix supports
  - Change impact
  - Regression testing
  - Criticality assessment+

TRA

- Tracing
- Trace m
  - Cha
  - Reg
  - Crit



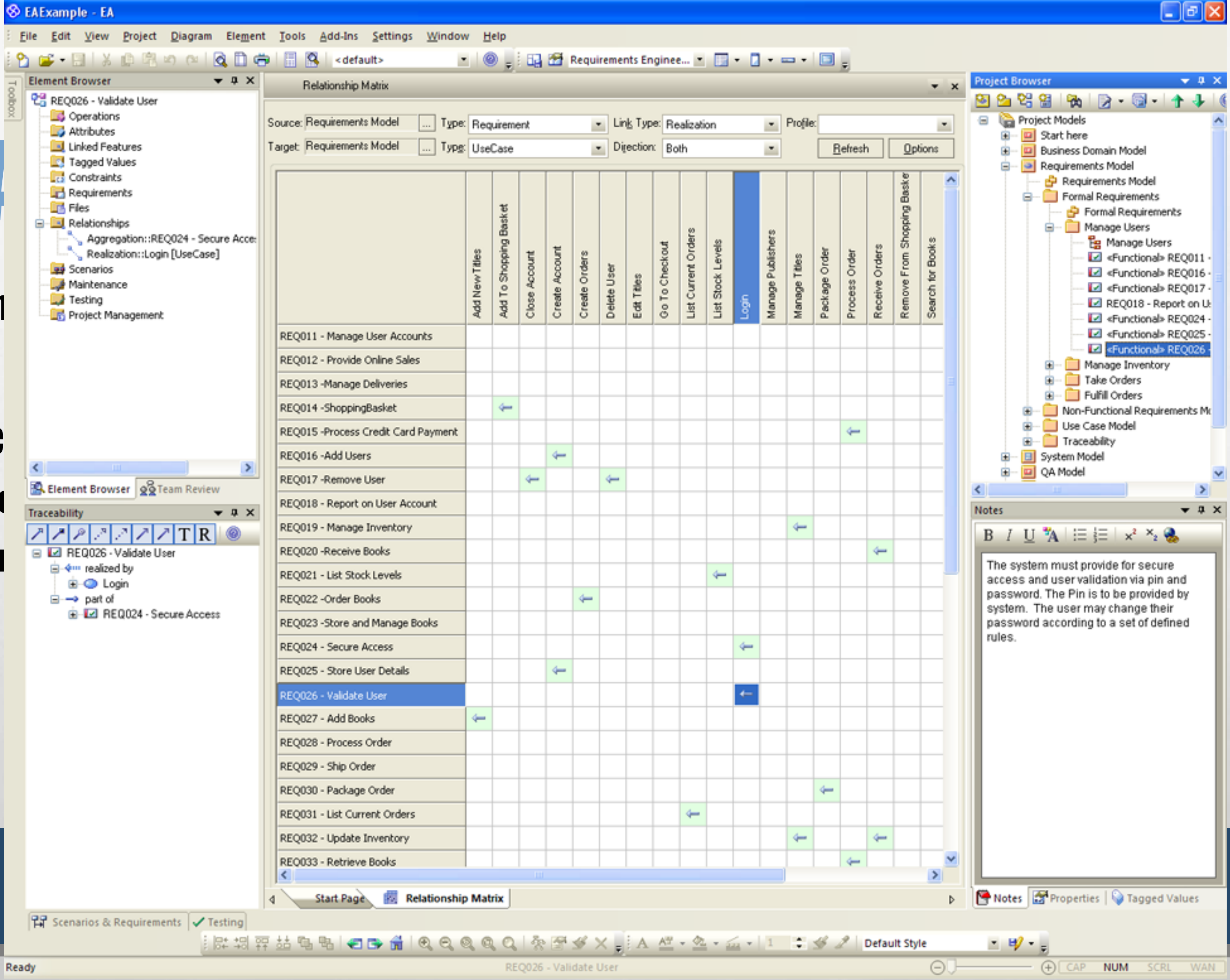
Requirement	Functional design	Internal design	Code	Tests
Restaurant has two ordering stations	Mgmt screen #2	Page 45	Line 12485	34, 57, 63
A waiter may order from any station	Order screen	Page 19	Line 6215	12, 14, 34, 57, 92
Any customer at a table may request a separate check	Order screen	Page 39	Line 2391	113, 85
A customer may get checks from more than one station	Check printing	Page 138	Lines 49234, 61423	74, 104

TRACE MATRIX

- Tracing to identify relationships
- Trace matrix supports
  - Change impact
  - Regression testing
  - Criticality assessment+

TRA

- Tracin
- Trace
  - C
  - R
  - Cr



Element Browser

- REQ026 - Validate User
  - Operations
  - Attributes
  - Linked Features
  - Tagged Values
  - Constraints
  - Requirements
  - Files
  - Relationships
    - Aggregation::REQ024 - Secure Acce:
    - Realization::Login [UseCase]
  - Scenarios
  - Maintenance
  - Testing
  - Project Management

Traceability

- REQ026 - Validate User
  - realized by
    - Login
  - part of
    - REQ024 - Secure Access

Relationship Matrix

Source: Requirements Model Type: Requirement Link Type: Realization Profile:

Target: Requirements Model Type: UseCase Direction: Both

	Add New Titles	Add To Shopping Basket	Close Account	Create Account	Create Orders	Delete User	Edit Titles	Go To Checkout	List Current Orders	List Stock Levels	Login	Manage Publishers	Manage Titles	Package Order	Process Order	Receive Orders	Remove From Shopping Baske	Search for Books
REQ011 - Manage User Accounts																		
REQ012 - Provide Online Sales																		
REQ013 -Manage Deliveries																		
REQ014 -ShoppingBasket		←																
REQ015 -Process Credit Card Payment															←			
REQ016 -Add Users				←														
REQ017 -Remove User			←			←												
REQ018 - Report on User Account																		
REQ019 - Manage Inventory													←					
REQ020 -Receive Books																←		
REQ021 - List Stock Levels										←								
REQ022 -Order Books					←													
REQ023 -Store and Manage Books																		
REQ024 - Secure Access											←							
REQ025 - Store User Details				←														
REQ026 - Validate User											←							
REQ027 - Add Books	←																	
REQ028 - Process Order																		
REQ029 - Ship Order																		
REQ030 - Package Order														←				
REQ031 - List Current Orders									←									
REQ032 - Update Inventory													←			←		
REQ033 - Retrieve Books															←			

Project Browser

- Project Models
  - Start here
  - Business Domain Model
  - Requirements Model
    - Requirements Model
    - Formal Requirements
      - Formal Requirements
      - Manage Users
        - Manage Users
        - «Functional» REQ011 -
        - «Functional» REQ016 -
        - «Functional» REQ017 -
        - REQ018 - Report on U
        - «Functional» REQ024 -
        - «Functional» REQ025 -
        - «Functional» REQ026 -
      - Manage Inventory
      - Take Orders
      - Fulfill Orders
    - Non-Functional Requirements M
    - Use Case Model
    - Traceability
  - System Model
  - QA Model

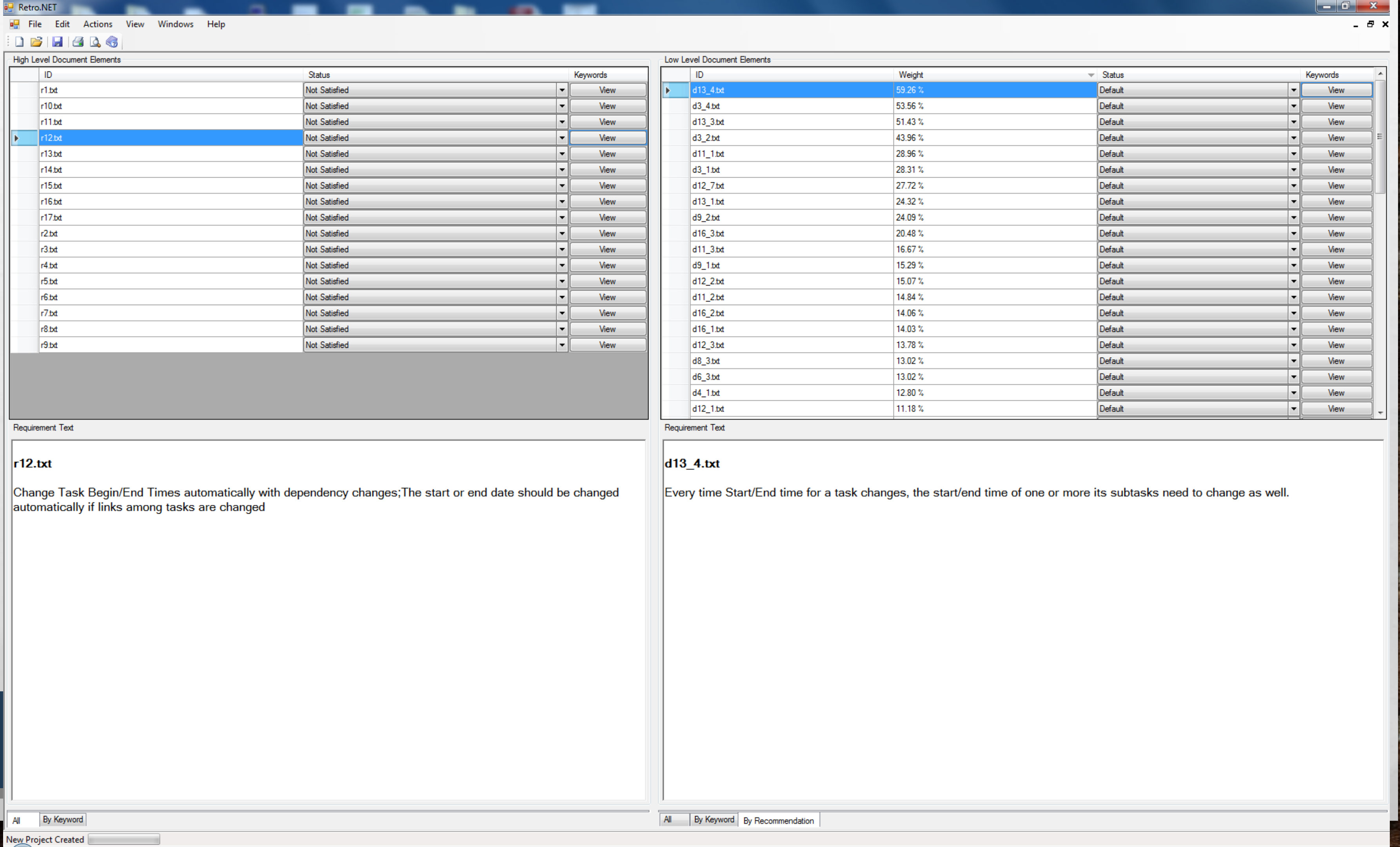
Notes

The system must provide for secure access and user validation via pin and password. The Pin is to be provided by system. The user may change their password according to a set of defined rules.

REQ026 - Validate User

TRACE MATRIX

- Tracing to identify relationships
- Trace matrix supports
    - Change impact
    - Regression testing
    - Criticality assessment+



High Level Document Elements

ID	Status	Keywords
r1.txt	Not Satisfied	View
r10.txt	Not Satisfied	View
r11.txt	Not Satisfied	View
r12.txt	Not Satisfied	View
r13.txt	Not Satisfied	View
r14.txt	Not Satisfied	View
r15.txt	Not Satisfied	View
r16.txt	Not Satisfied	View
r17.txt	Not Satisfied	View
r2.txt	Not Satisfied	View
r3.txt	Not Satisfied	View
r4.txt	Not Satisfied	View
r5.txt	Not Satisfied	View
r6.txt	Not Satisfied	View
r7.txt	Not Satisfied	View
r8.txt	Not Satisfied	View
r9.txt	Not Satisfied	View

Low Level Document Elements

ID	Weight	Status	Keywords
d13_4.txt	59.26 %	Default	View
d3_4.txt	53.56 %	Default	View
d13_3.txt	51.43 %	Default	View
d3_2.txt	43.96 %	Default	View
d11_1.txt	28.96 %	Default	View
d3_1.txt	28.31 %	Default	View
d12_7.txt	27.72 %	Default	View
d13_1.txt	24.32 %	Default	View
d9_2.txt	24.09 %	Default	View
d16_3.txt	20.48 %	Default	View
d11_3.txt	16.67 %	Default	View
d9_1.txt	15.29 %	Default	View
d12_2.txt	15.07 %	Default	View
d11_2.txt	14.84 %	Default	View
d16_2.txt	14.06 %	Default	View
d16_1.txt	14.03 %	Default	View
d12_3.txt	13.78 %	Default	View
d8_3.txt	13.02 %	Default	View
d6_3.txt	13.02 %	Default	View
d4_1.txt	12.80 %	Default	View
d12_1.txt	11.18 %	Default	View

Requirement Text

r12.txt

Change Task Begin/End Times automatically with dependency changes;The start or end date should be changed automatically if links among tasks are changed

Requirement Text

d13_4.txt

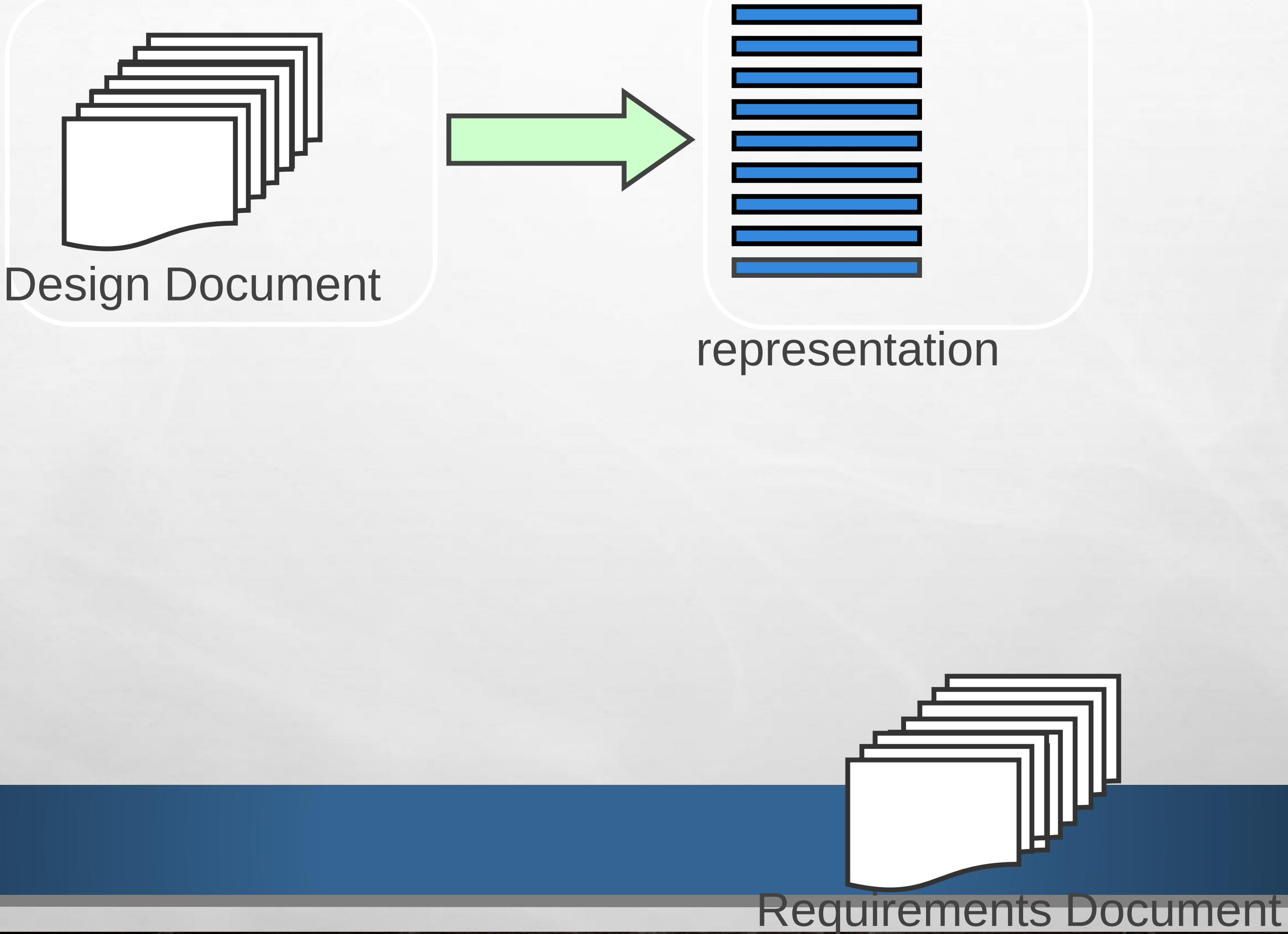
Every time Start/End time for a task changes, the start/end time of one or more its subtasks need to change as well.

IR FOR TRACING

Design Document

Requirements Document

IR FOR TRACING

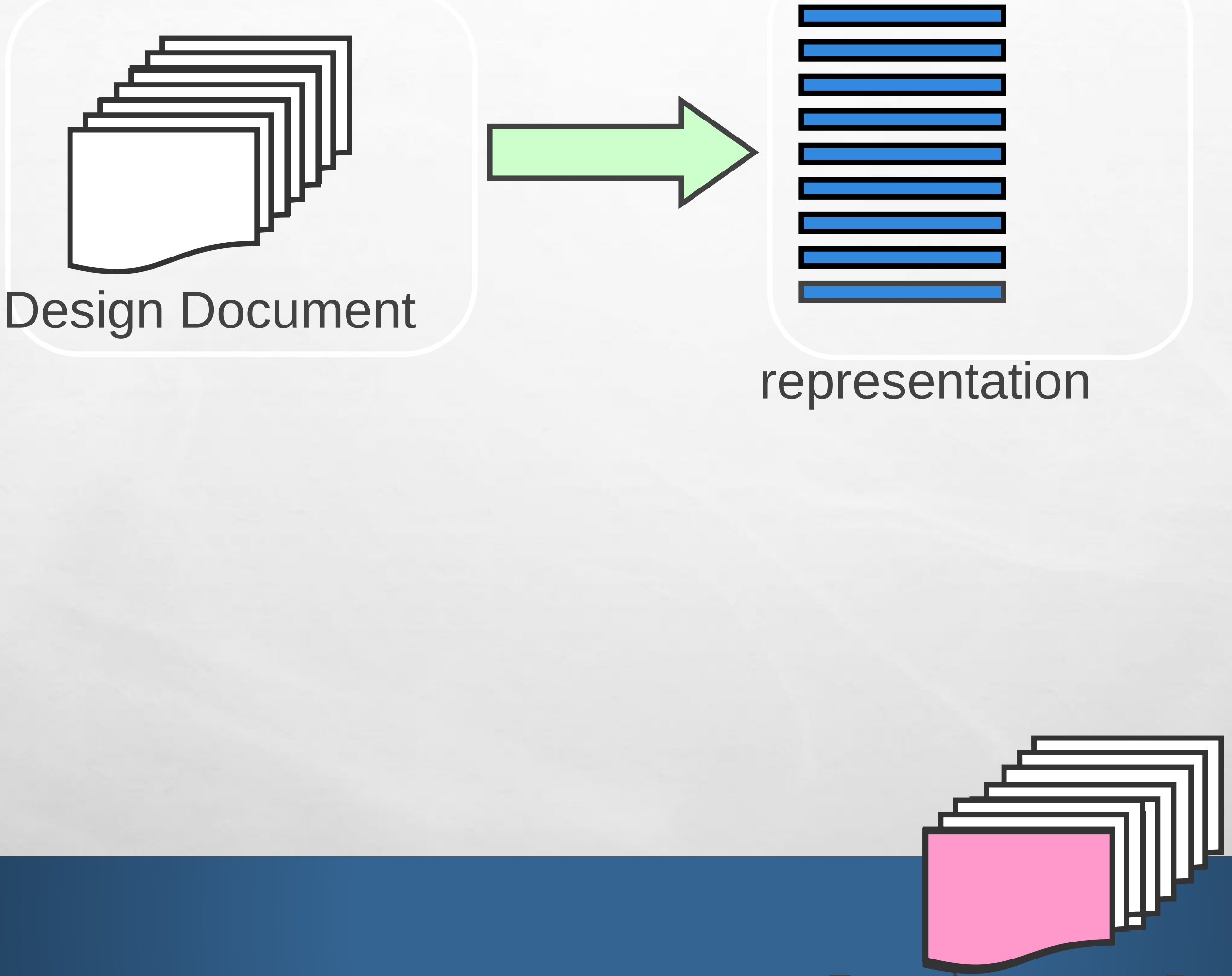


Design Document

representation

Requirements Document

IR FOR TRACING

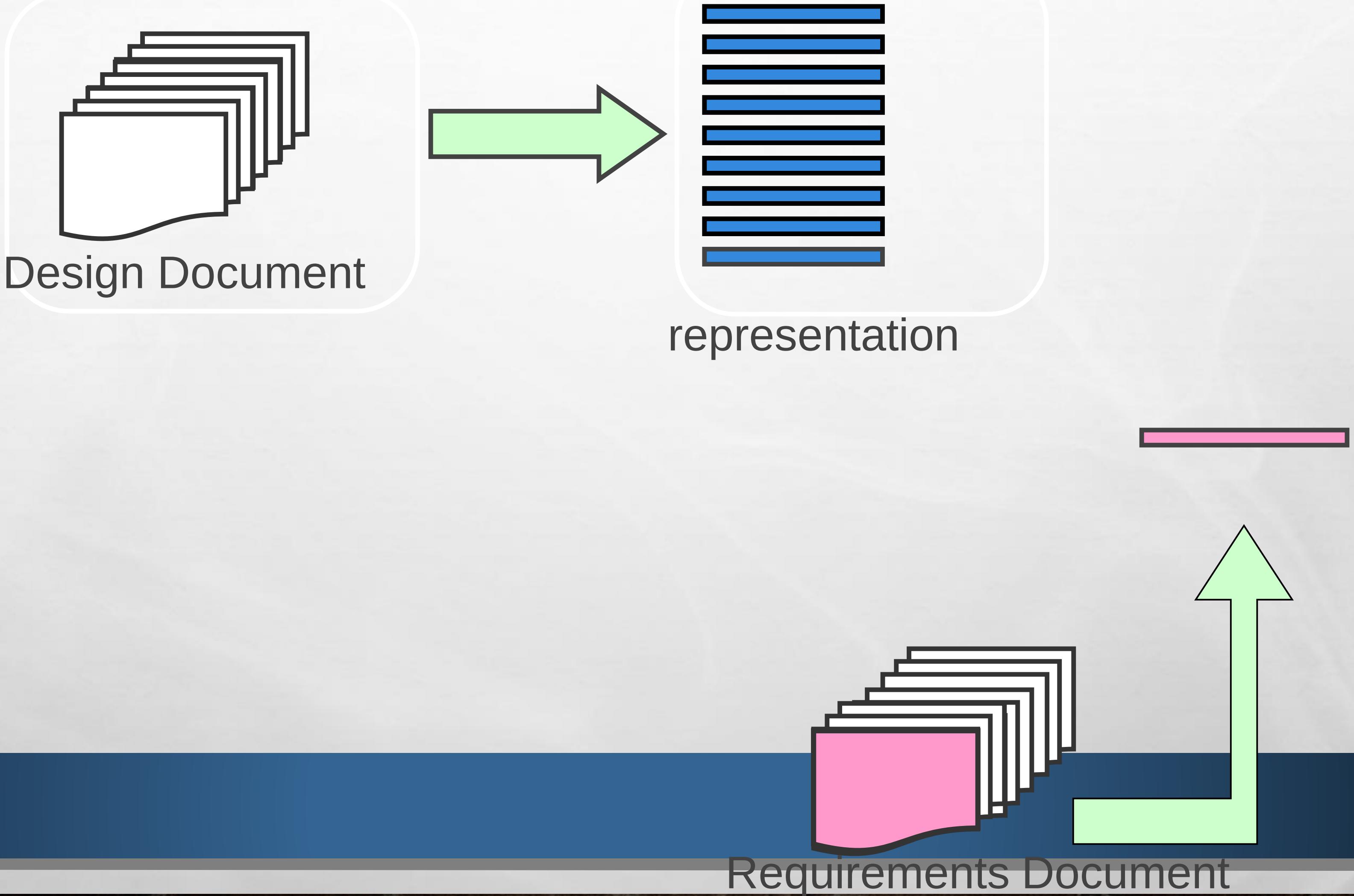


Design Document

representation

Requirements Document

IR FOR TRACING

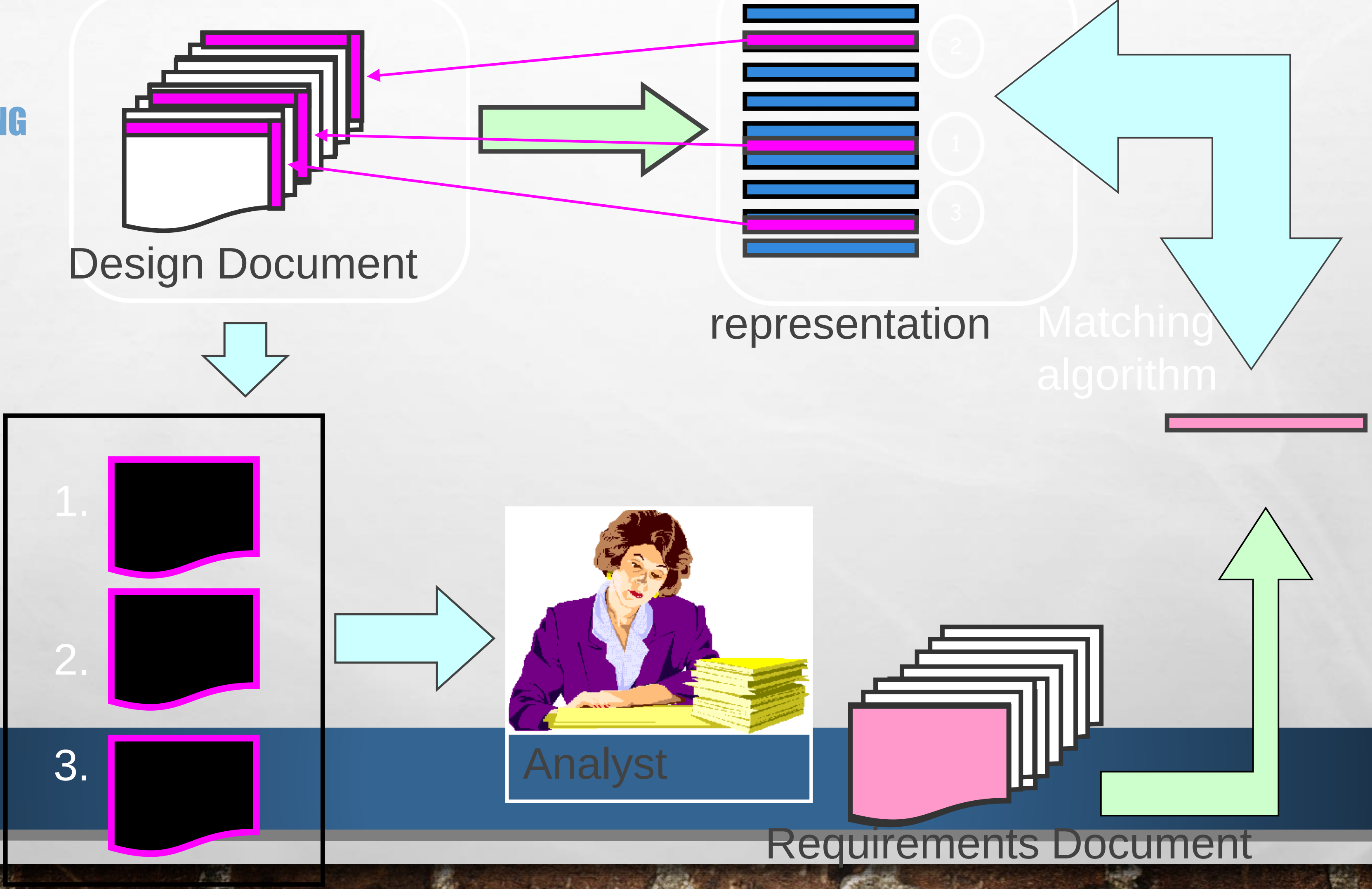


Design Document

representation

Requirements Document

IR FOR TRACING



Design Document

representation

Matching algorithm

2

1

3

1.

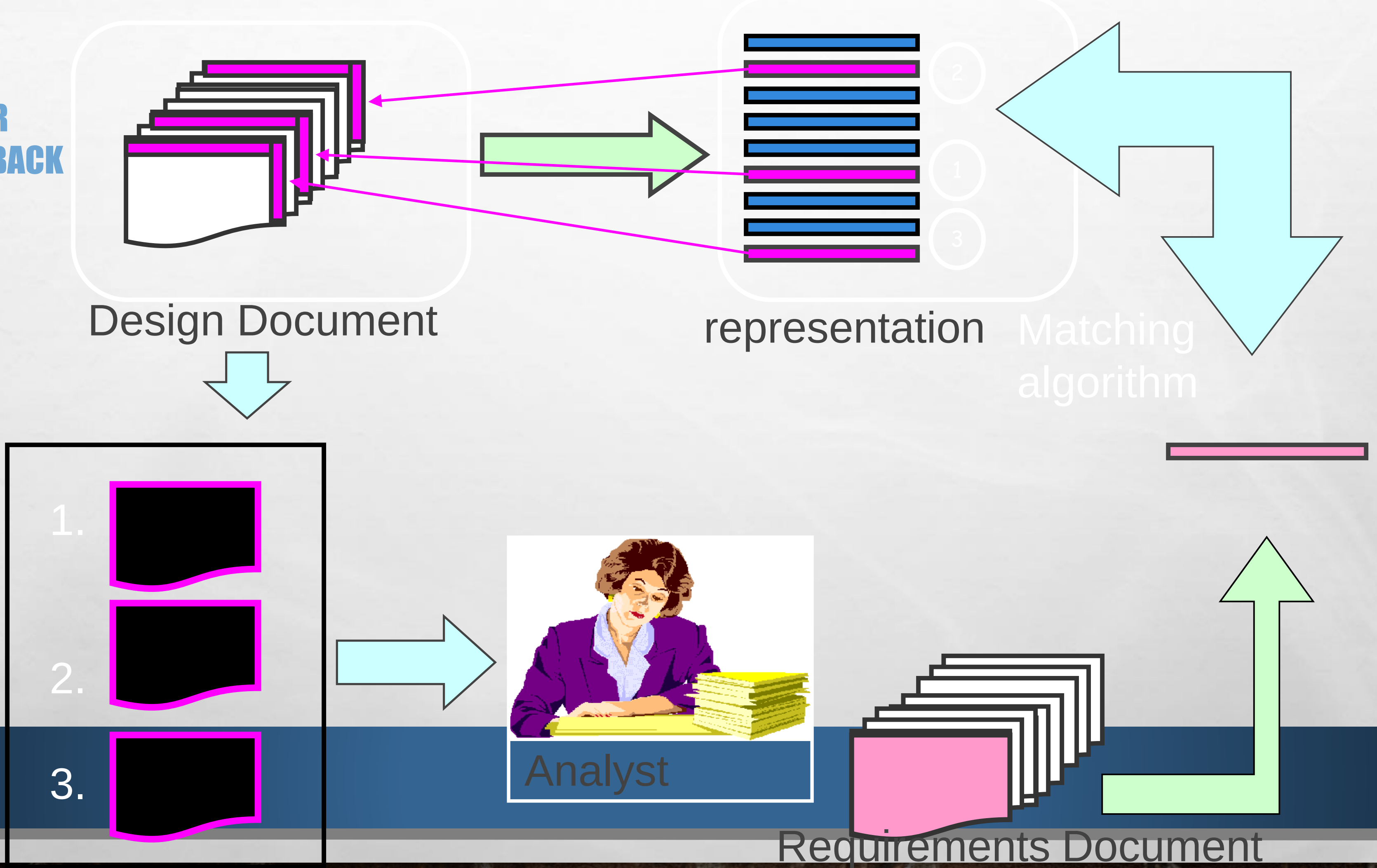
2.

3.

Analyst

Requirements Document

ENTER FEEDBACK



Design Document

representation

Matching algorithm

1.

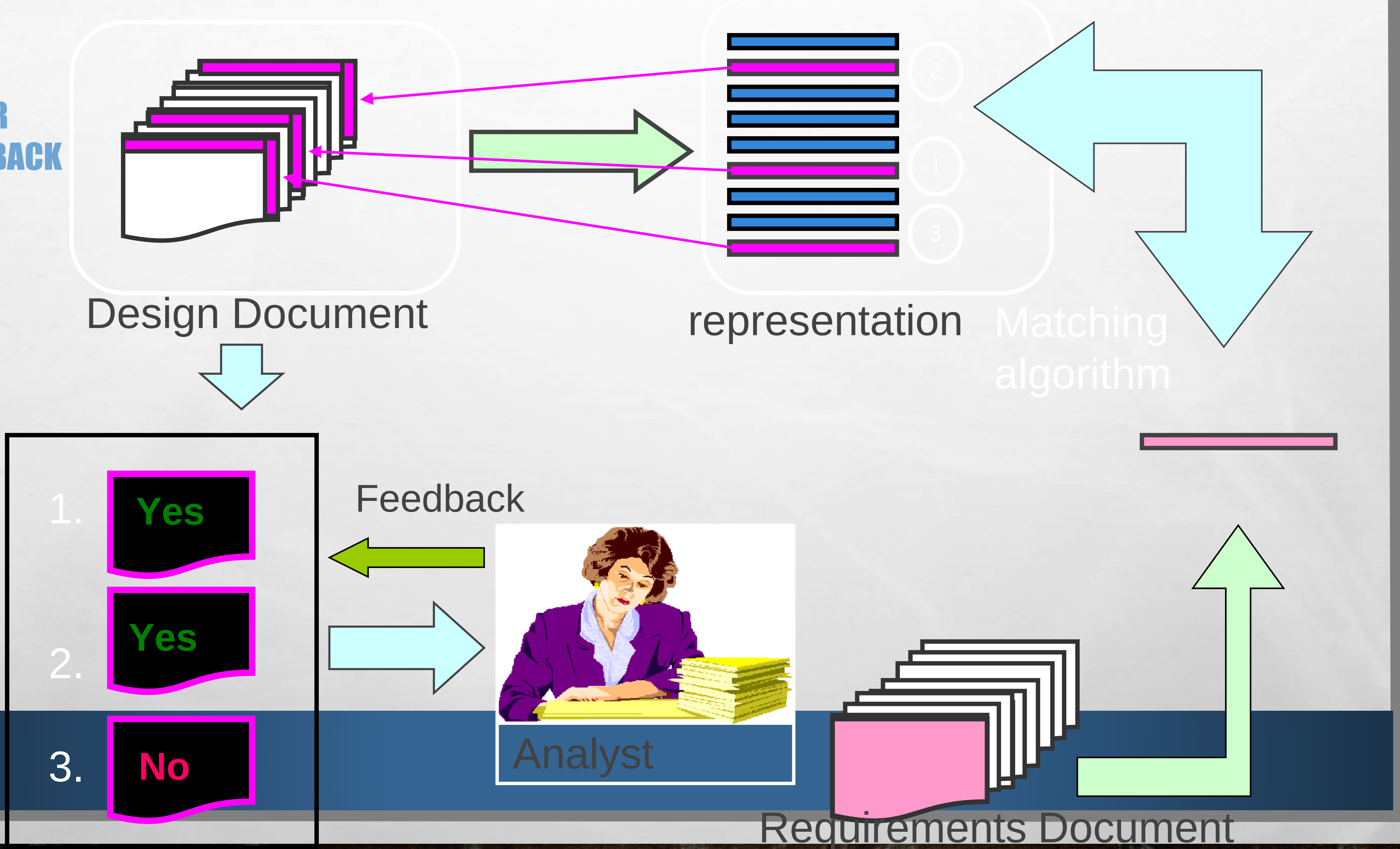
2.

3.

Analyst

Requirements Document

ENTER FEEDBACK



Design Document

2

1

3

representation

Matching algorithm

Feedback

1. Yes

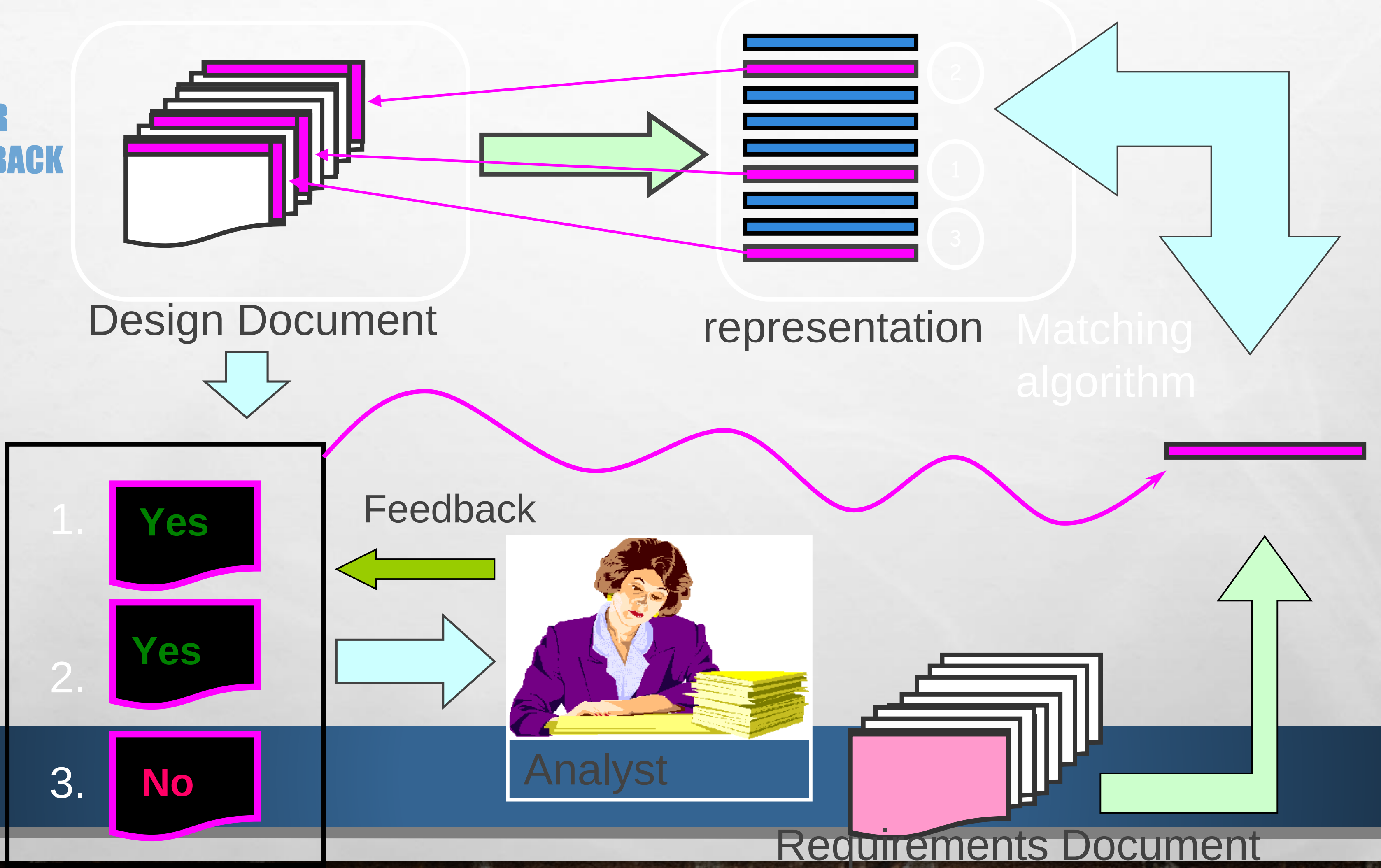
2. Yes

3. No

Analyst

Requirements Document

ENTER FEEDBACK



Design Document

representation

2

1

3

Matching algorithm

Feedback

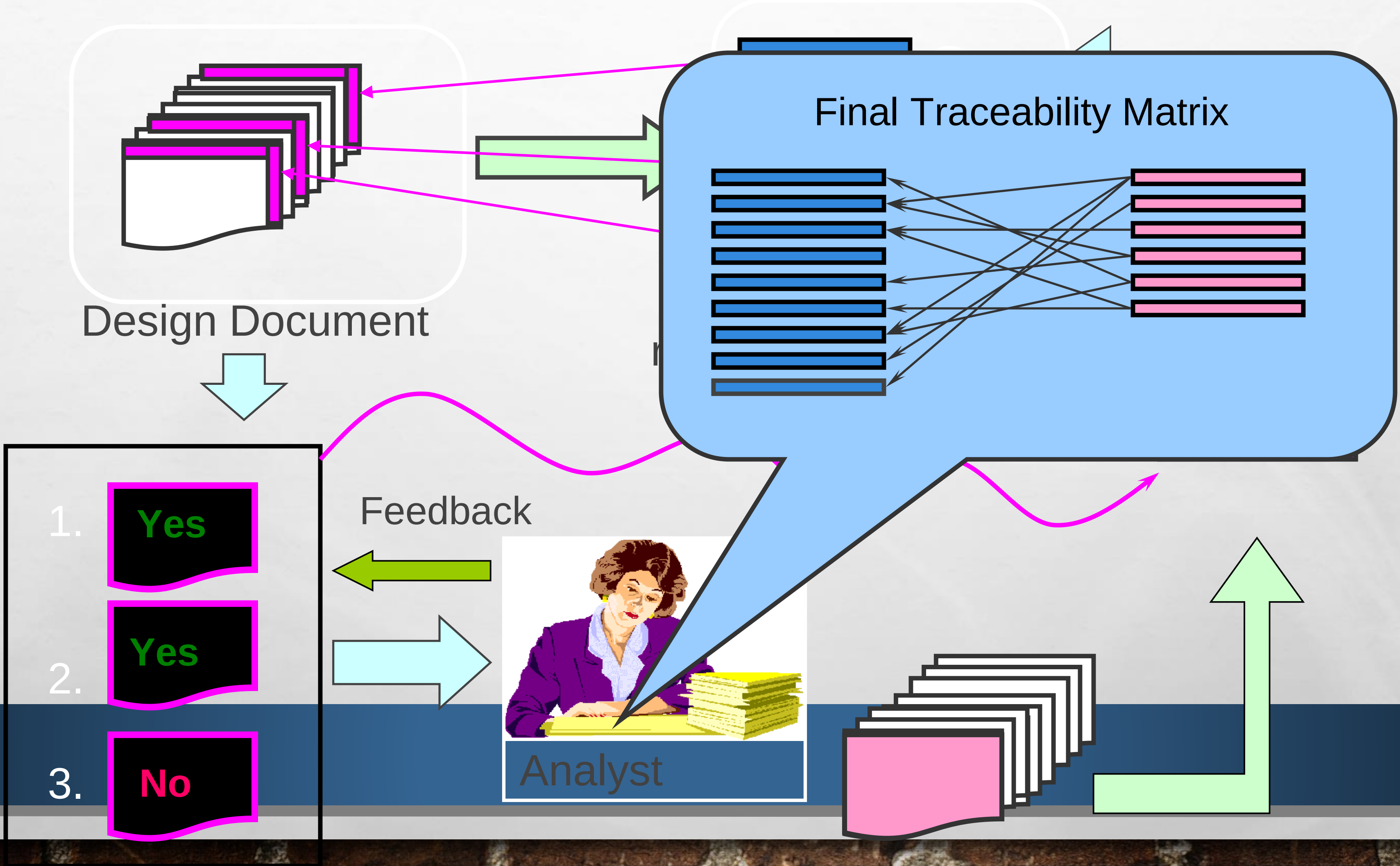
1. Yes

2. Yes

3. No

Analyst

Requirements Document



Final Traceability Matrix

Design Document

Feedback

1. Yes

2. Yes

3. No

Analyst