The human lives in a safety software by Thinktools

THE HUMAN LIVES IN A SAFETY SOFTWARE BY THINKTOOLS

If there is no result despite your passion and will,
Change the tool now.

ThinkforBL

WHO WE ARE

We are Industrial-field focused software practical engineering expert.

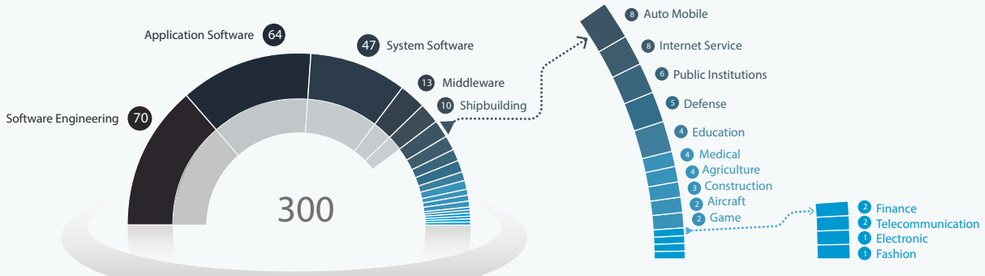
—

ThinkforBL is a specialist in practical engineering that delivers optimized solutions to practical problems in the software industry from scientific and technical perspectives. Instead of phenomenon, we focus on the cause and essence of problems, we present global standards in software engineering and develop required diverse tools.

OVER 300 REFERENCES IN 10 YEARS

—

We are proud of our know-how, methodology and technology, accumulated over 10 years of consulting experience with more than 300 software R & D companies of each domain.



< References by Industrial Area >

Thinktools IS

The human lives in a safety software by

The master does not choose a tool?
The one who makes a good choice of tools is the master.

Diverse supporting tools and areas that require a high level of engineering activity exist according to the nature of the software industry.
ThinkTools is especially designed for Safety Software, Mission Critical Area, CPS (Cyber-Physical Systems), and for governance of large projects.

CETA — Cause and Effect Test Auto Analyzer
Semi-Automated Tool supporting test design

DRONACE — Drone Advanced Certification Environment
Automatic Verification Tool in Virtual Environment of performance of software in drones

SMAC — Safety Mechanism Analysis in Code
Tool that Proposes Safety Mechanism Codes of Electric/Electronic Software

Project Management	17		Architecture . Design	4
Process	4		Marketability	20
Quality Management	72		Tool . Infrastructure	32
Testing	35		Internalization . Localization	12
Code Inspection	21		Organizational Cooperation	7
Requirement Engineering	12		Maintenance	13
Human Resource Design(HRD)	7			

< References by Software Engineering Area >

OUR COVERAGE AREA based on SWEBOK 3.0

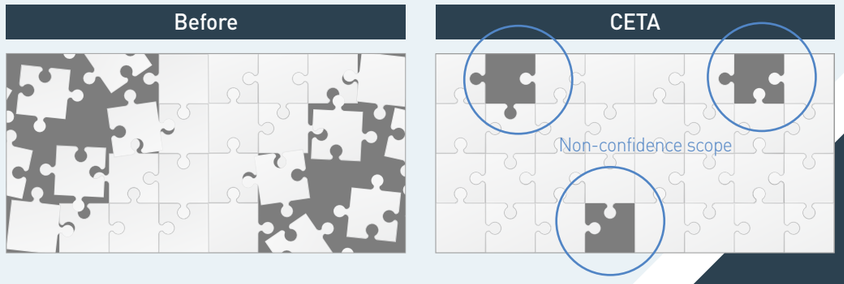
—

		T-Biz Engineering				SWEBOK 2.0										SWEBOK 3.0			
		ORGANIZATIONAL COOPERATION	INTERNALIZATION .LOCALIZATION	MARKETABILITY	HUMAN RESOURCE DESIGN	SOFTWARE REQUIREMENTS	SOFTWARE DESIGN	SOFTWARE CONSTRUCTION	SOFTWARE TESTING	SOFTWARE MAINTENANCE	SOFTWARE CONFIGURATION MANAGEMENT	SOFTWARE ENGINEERING MANAGEMENT	SOFTWARE ENGINEERING PROCESS	SOFTWARE ENGINEERING TOOLS AND METHODS	SOFTWARE QUALITY	SOFTWARE ENGINEERING PROFESSIONAL PRACTICE	SOFTWARE ENGINEERING ECONOMICS	COMPUTING FOUNDATIONS	MATHEMATICAL FOUNDATIONS	ENGINEERING FOUNDATIONS
		1	12	20	7	12	4	21	35	13		17	4	32	72					
Universal software	System		•			• (C)		•	• (C)						• (C)		•	•		
	Middleware		•			• (C)		•	• (C)						• (C)		•	•		
	Application			•		• (C)										•	•			
	Construction	•	•			• (C)										•	•		•	•
	Agriculture	•		•		• (C)								•			•			•
	Public Institutions	•				• (C)				•		•				•				
	Education	•	•	•		• (C)											•			
	Logistics			•		• (C)											•			•
	Internet Service	•	•	•		• (C)											•			
	Telecommunication						•	•	• (C)	•	•				• (C)		•			
Industrial software	Finance					• (C)	•	•	• (C)			•				•	•	•		
	Auto mobile	•	•	•		• (C)	•	• (S)	• (C)		•	•	•	•	• (S/C)	•	•	•	•	•
	Aircraft	•	•				•	•	• (D/C)		•	•	•	•	• (D/C)	•		•	•	•
	Shipbuilding	•				• (C)	•	•	• (C)	•	•	•	•	•	• (C)	•	•	•	•	•
	Medical	•				• (C)	•	•	• (C)						• (C)	•	•	•	•	•
	Defense	•	•	•			•	•	• (C)	•	•	•			• (C)	•		•	•	•
	Energy chemistry	•						•	• (C)				•	•	• (C)	•		•	•	•
	Nuclear Power	•					•	•	• (C)	•	•		•	•	• (C)	•		•	•	•
	Railway	•					•	•	• (C)	•	•		•	•	• (C)	•	•	•	•	•

THINKTOOLS ONE. CETA
WHAT IS CETA?

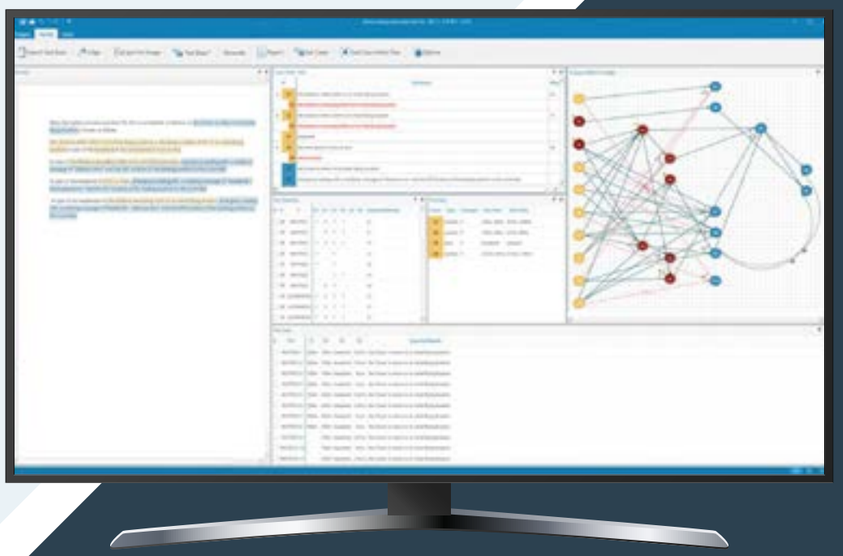
Finally, the tool that tests exceptional circumstances in all cases is released.

Up to now, we have accepted the wrong way, thinking that testing all cases is impossible. CETA is a functional level technical design tool that (semi) automatically derives all conditions of human error. From now on, you can run a test 'as much as you need', not 'as much as you can think of.'
When it comes to safety and protecting life, 99% is not enough.



< Tester Experience Based Design>

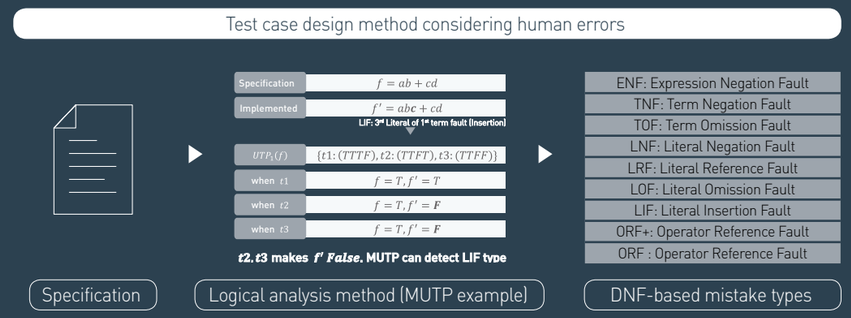
< Technology Based Test Design>



WHY
SHOULD YOU USE CETA?

The reason for the defects and accidents despite the test was because of the improper and inaccurate test.

CETA performs scientifically, technically, and objectively the test design, which used to be conducted subjectively by relying on the tester's experience. CETA automatically derives human errors of all cases in specific situations. For efficient testing, CETA suggests the test priority that is appropriate for the functional characteristics among tens of thousands of automatically drawn test cases.

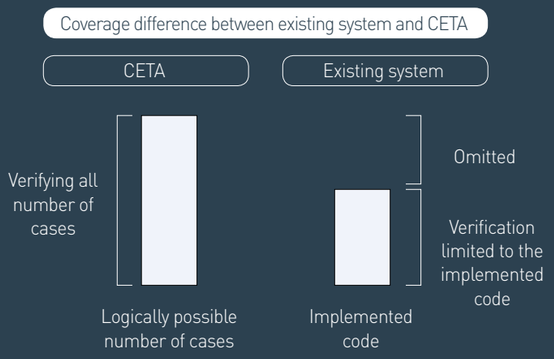


As the physical impact of Software on industry is growing, the completeness of tests is also extremely important.

In previous industrial fields, testing was sometimes considered a cumbersome and inefficient process.
Today, however, is the time when Software controls all industries, where Software's failure paralyzes industries and brings human casualties.
CETA provides the most realistic and systematic support to firmly and surely protect the safety of colleagues and families that are irreplaceable.

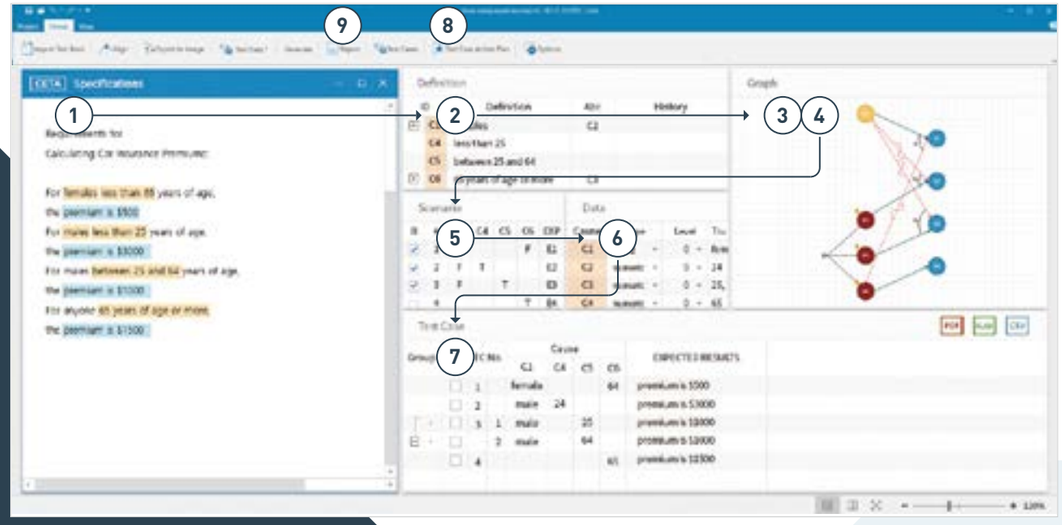
Don't be fooled by the coverage of existing tests.

Existing code-based test coverage does not mean quality coverage. It only means that the implemented code functions properly, failing to check whether the safety mechanisms that should be implemented are in fact implemented.
The lack of information about how much a particular function has been tested against the number of logically possible cases can disrupt the decision making in the field.



THINKTOOLS ONE. CETA
DETAILS FROM CETA

Test Design Process using CETA



1. Cause & Effect deriving
2. Cause & Effect abbreviation
3. Cause & Effect graph preparation
4. Cause & Effect graph inspection
5. Confirm test scenarios and select base scenarios
6. Test data input
7. Test case output
8. Test case prioritization
9. Test case design report generation

01 **Publication & Presentation at IEEE sponsored Test Conference (ICST – International Conference Software Testing)**
2017.03.17 IEEE InSTA 2017 in Tokyo
2019.04.22 IEEE InSTA 2019 in Xian

02 **Presentation at Asia's Software Quality Management Conference (ASQN – Asia Software Quality Network)**
2018.06.29 ASQN 2018 in Beijing
2019.09.09 ASQN 2019 in Tokyo1

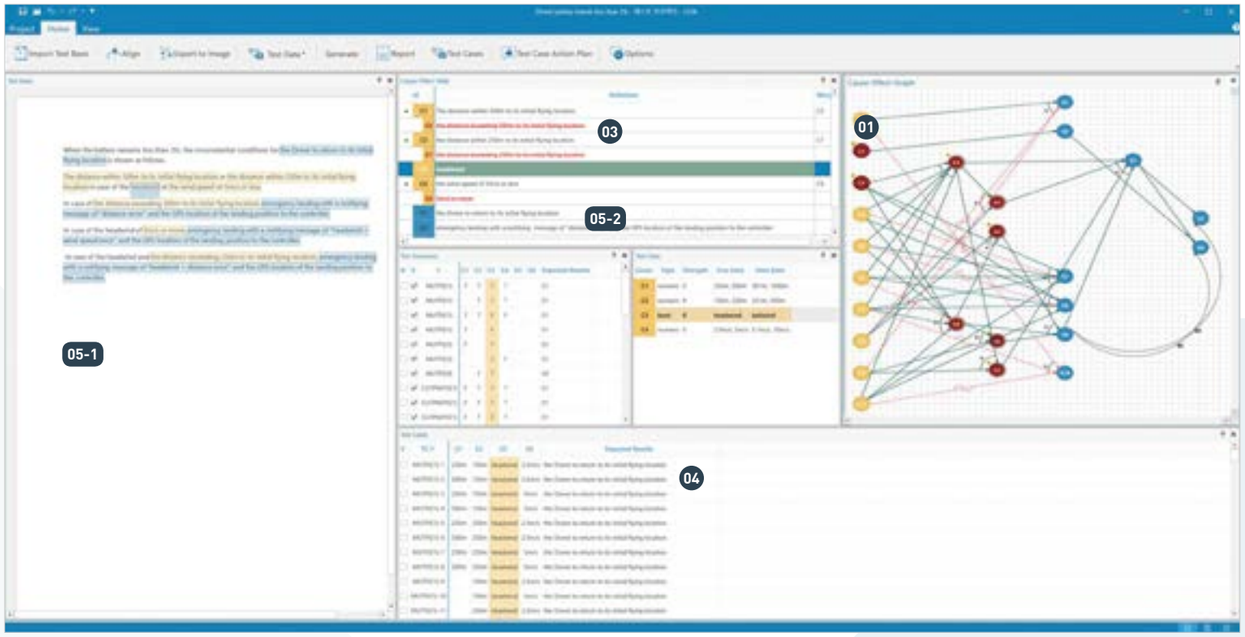
03 **Established functional safety-based software verification standard of the Korea Information and Communication Technology Association (TTA)**
Test Design Method based on Specification for Software Functional Safety Verification (TTAK.KO-11.0251)
Test Coverage Measurement Method for software Functional Safety Verification (TTAK.KO-11.0250)

What coverage should I look at in my current situation?

< CHARACTERISTIC OF FUNCTION >	< TEST COVERAGE >	
When confirming cause combination defect of result logic type	UTPC (Unique True Point Coverage)	Viewpoint : CODE
When confirming cause reflection defect of result logic type	CUTPNFP (Unique True Point And Near Faise Point Pair Coverage)	
When only the basic operation of the function checking is needed	Base Scenario Coverage	Viewpoint :SPEC
When checking existence of exception handling implementation	Invailid Scenario Coverage	
When the input data is large and has diverse logical effects	Cause Coverage	
When producing complex and varied results	Effect Coverage	
When data validation is important	Cause Test Data Coverage	
When causal factors have complex logical relationships	Valid Scenario Coverage	
	Complex Logical Relation Coverage	

THINKTOOLS ONE. CETA
DETAILS FROM CETA

Functions & Effects



01
VISUALIZATION OF LOGICAL SENTENCE BASED ON SPECIFICATIONS (Requirements, Functions)

It's a big challenge to describe how software functions operate based on human abstract language. Because software always means logical conditions and logical status, by describing in a way that is easy to express logically, prompt understanding can be achieved and mutual communication errors can be reduced.

02
ERROR SUGGESTION REGARDING SUSPECT SPECIFICATIONS

Human-written requirements have many errors such as omission, overlap, ambiguity, and conflict. But, in a technical written sentence, it's difficult to find logical errors. CETA automatically identifies contents of the specification statement suspected as having a logical error and presents.

03
SUGGESTION REGARDING DUPLICATED ITEMS IN THE SPECIFICATION

Despite being the same contents, there are repetitive items in the requirements specification statement. By automatically identifying such, CETA suggests duplicate eliminations to prevent mistakes.

04
AUTOMATIC SUGGESTION FOR ALL POSSIBLE CASES

Depending on the type of logical mistake of human, it derives the number of all possible cases automatically and creates as a test case.

05
EXTREMELY CONVENIENT UI BASED ON MOUSE CLICK AND DRAG/DROP

You don't need to be burdened by inputting. Starting with loading specifications, we have reinforced the convenience enough to finish any design with just a few mouse clicks and dragging

05-1 Cause/Effect input by dragging

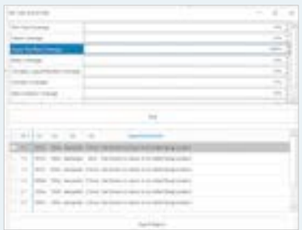
05-2 Abbreviation by dragging

06
INSTANT VISUALIZATION OF ALL ITEMS BY TRACEABILITY

Don't be afraid of complex screens like a net. CETA visualizes by highlighting so you can track and view just selected items out of dozens of items in a complex statement

07
AUTOMATIC SUGGESTION OF EFFICIENT TEST PRIORITIZATION TO ACHIEVE THE BEST TEST COVERAGE

Performing all of the vast test cases is neither practical nor effective.
By classifying importance and types of functions into 9 perspectives, CETA automatically derives tests for the goal coverage achievement and prioritizes. Rather than securing additional time, it is more meaningful to use the secured time well.



08
VISUALIZATION FOR DIFFERENT CHANGES (Next Version)

No need to start from the beginning, every time a change occurs. CETA compares the changes and displays only the impacted items, allowing efficient management and the analysis of logical impact of the change.

CETA is the world's only semi-automated tool supporting test design

Registrations

- Method and Apparatus For Auto Generation Of Test Case (Patent 10-1554424)
- Method and Apparatus For generating test case, computer readable recording medium (Patent 10-1826618)
- Display panel with picture design(30-0905834)
- Test-based quantification measurement methods and devices for safety verification of field software codes (Patent 10-1899778)
- Method and Apparatus For Software Analysis (Patent 10-1706098)
- Method and Apparatus For Software Hazard Analysis (Patent 10-1734418)
- Method and Apparatus For Generating Test Case To Support Localization Of Software (Patent 10-1588027)

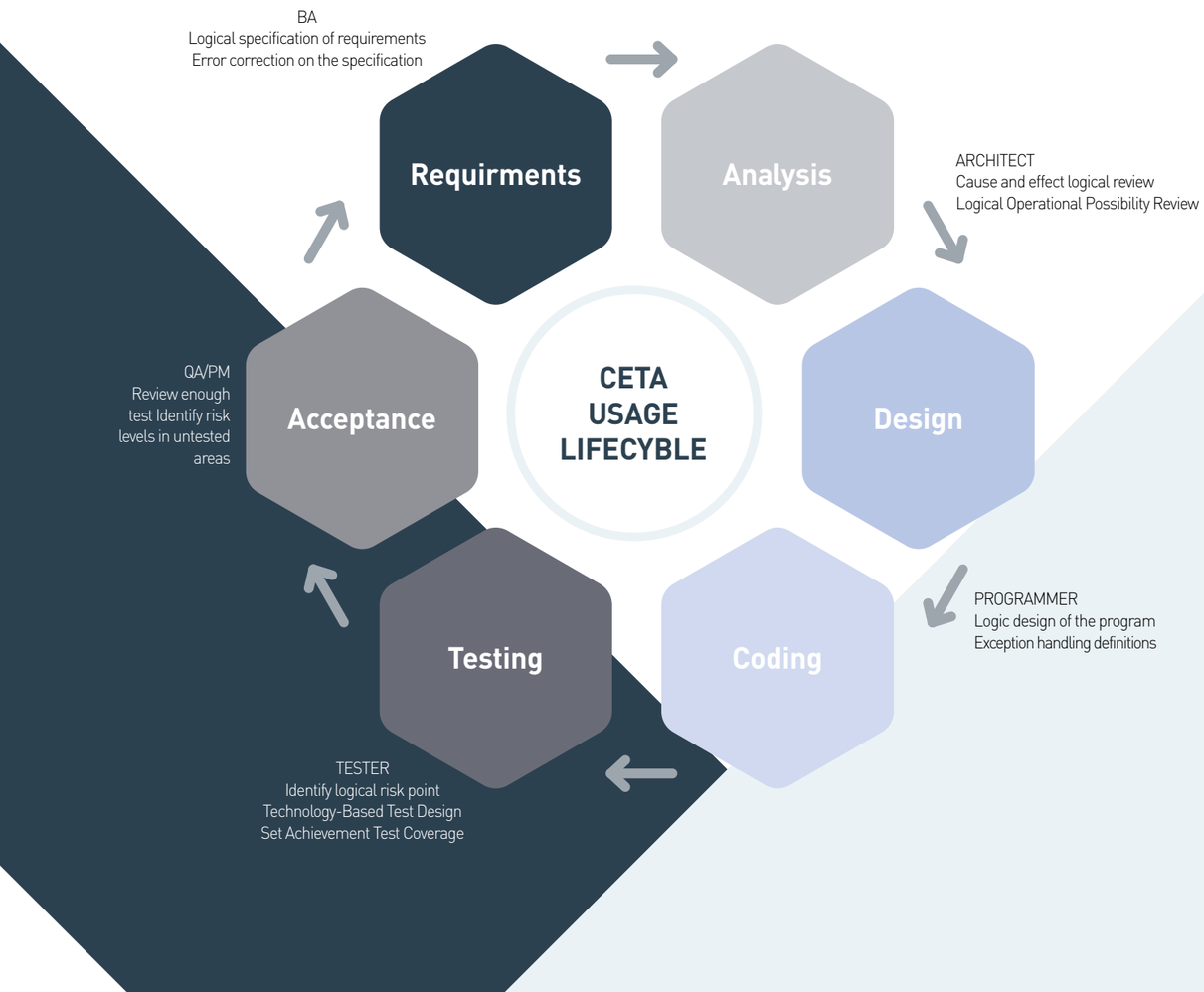
PCT

- Visualization method for tracking analysis of testcase design information, testcase generating apparatus, and computer-readable recording medium (PCT/KR2017/013459)

THINKTOOLS ONE. CETA
DETAILS FROM CETA

How to use CETA

- Visual expression of logical and dynamic flows for critical or complex functions in the system is possible, and easily spots errors in the specification document.
- By easily discovering areas for exception treatment required areas and logical risk areas to enable test plan development.
- The target verification level is set by considering the importance of the function and the difficulty of implementation, and can be used as an objective basis for delivery, acquisition, and release.
- When a change occurs, it suggests the validity of a selective test based on logical analysis of impact, without retesting.
- On the SDLC of the process of software development and change, it enables high-level logic expression communication among stakeholders.



▶ Is CETA a test (performance) automation tool?
This is a tool for design (semi) automation tool, not a test performance. Thus, the performance should be done manually. That is, as if the calculation solves after just putting in the coefficients in the quadratic equation, CETA also automatically designs the rest after the user inputs logical parts. The logical part of 'what is wanted to be created' by the user should be determined directly. then, the rest is performed by CETA.

▶ Who should use CETA? Can I, the ignorant of program language, draw logical symbols?
Anyone can. If possible, it is advisable that a tester should do it. When the planner focuses on how Software works and what values can be created, while the tester thinks about when errors occur, it will be more efficient. Anyone can use it but it will be used more usefully in the hands of a person who worries about negative possibility of functions.

▶ Would it be better to generate the statement CETA from scratch?
You don't go directly to MRI if you have a cold, right? It's possible to do apply it to all functions, but it's not efficient. We recommend selective use in critical, complex or dangerous areas.

▶ When can I do it all when I don't have time?
Because there is no time, efficient testing is required. You cannot stop studying for the exam just because you don't have enough time. CETA is the accurate fortune-telling teacher who verifies the most critical elements and suggests solutions within a short period of time. If you are not going to give up on the safety, effectively prepare with CETA.

▶ We've been using the existing code coverage, what's the difference?
By analogy, code coverage is like taking a practice exam having the code that we implemented as a test subject scope. On the contrary, CETA takes all possible number of cases for the results that a code function can produce as the scope of the exam. If you take the practice exam only to the extent that you have studied, the score will be good immediately. However, it will fail to prepare for the actual test or reality. CETA will check and confirm even the parts that I have not studied. Making 100% immediate coverage is not important. What matters and the actual purpose of testing is to make sure the product doesn't have any problems after it's actually released.

▶ Is the method reliable?
Existing tests relied on sense without an equation to solve. In the end, some geniuses could give answers without gaining universal reliability. CETA is the quadratic formula. Without any excellent sense or experience on the part of the tester, only with the training of applying the formula carefully, you can verify objective results with CETA that is highly reliable.

▶ I can't believe that a not very famous company has developed such a great technology.
We too cannot believe why many companies do not make such attempts or research. The reality that too many companies rely on testers without architecture or programming fluency, especially on their experience and sense, for their important tests is hard to believe.
A method created by James Meyer, the master of testing field, in 1974, CETA had been stagnant in development for a while. Then, recently, methods linked to DNF have been researched overseas. Then, in the process of improving the method to enable the use in the field by making it a bit simpler, we devised the quantitative strategy to drive efficiency to pursue efficiency, rather than perfection. And CETA is developed as an automated tool of that. We think that it is the difference between ignoring the faced problems to be convenient right now or figuring out technical methods to solve the issues.

▶ What is the future direction of CETA?
We believe that there's more to see. CETA is an integral calculator built for those who find it difficult to survey each of the entire area. If companies do not calculate the area because they don't want to study integral and stick to the visual estimation process as they do now, serious problems can occur. We look forward to seeing more companies use a CETA-like approach to create a more stable and efficient industry environment before experiencing serious problems caused by testing issues. Recently, for Vietnamese workforce that is relatively inexpensive and learn quickly, we are teaching CETA methodology. We think it is good to expect the visualization of another result based on the cooperation with them.

THINKTOOLS TWO.DRONACE
WHAT IS DRONACE?

Will the software installed on your drone run safely in any circumstances?

Unlike traditional drones, which are directly controlled by people, industrial drones that are being developed today move based on artificial intelligence.
For instance, if dropping a life tube to an accident site 2km off the coast based on remote control, a second accident may be caused by difficulties such as video reception, radio interference, or delay in remote transmission and reception.
In the industrial field, autonomous drones that judge and perform missions themselves are appearing. It is difficult to concretely verify the safety of Software of drones, which has to perform this complex task, with the existing test method at the level of diagnosing the sensor and HW failure rate.

DROANCE is the only tool that checks the safety of Software installed on drones by virtually configuring the physical testing environment.

01 **Publication & Presentation at IEEE sponsored Test Conference (ICST – International Conference Software Testing)**
2017.03.17 IEEE InSTA 2017 in Tokyo
2019.04.22 IEEE InSTA 2019 in Xian

02 **Presentation at Asia's Software Quality Management Conference (ASQN – Asia Software Quality Network)**
2018.06.29 ASQN 2018 in Beijing
2019.09.09 ASQN 2019 in Tokyo

03 **2019.05.24 The Korean Association of SMART Policing participation**

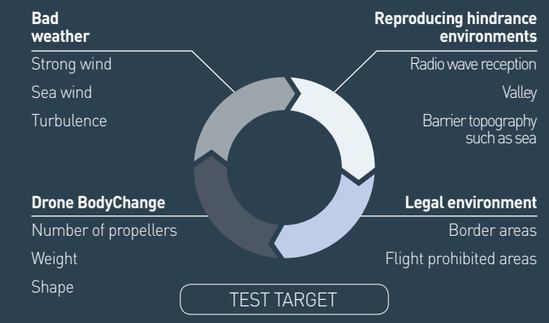
Registration
- Method and Apparatus For Recording Flight Data Of Unmanned Aerial Vehicle (Patent 10-1703236)
- Method and Apparatus For Testing Autonomic Flight Control Of Unmanned Aerial Vehicle Patent 10-1769281)

WHY SHOULD YOU USE DRONACE?

Test environments can neither be artificially created nor be just waited.

The autonomous mission performance Software installed on drones must cope with all types of bad weather and disability situations, in addition to unexpected situations. However, it is impossible to create a physical environment like the actual climate situation to check whether the measures are safe. On the other hand, you also cannot afford to wait until there is a strong wind, or reproduce the sea breeze or turbulence in terms of time limitation.
If the drone avoids the borders safely is verified during the actual flight, it can also create the risk of military problems.

- Safety of the Flight Controller (Safety of flight control): Does it control the posture safely despite strong winds?
- Accuracy of Autonomous Flight (Accuracy of Autonomous Flight): Does it reach its destination correctly in turbulence?
- Compliance with Safety Law (compliance with safety laws): Can it avoid when it enters a flight prohibited area?
- Reliability of Mission Execution: Does it completely and ultimately perform the given mission?



Flight Controller

Sensor (GPS, Temp etc.)

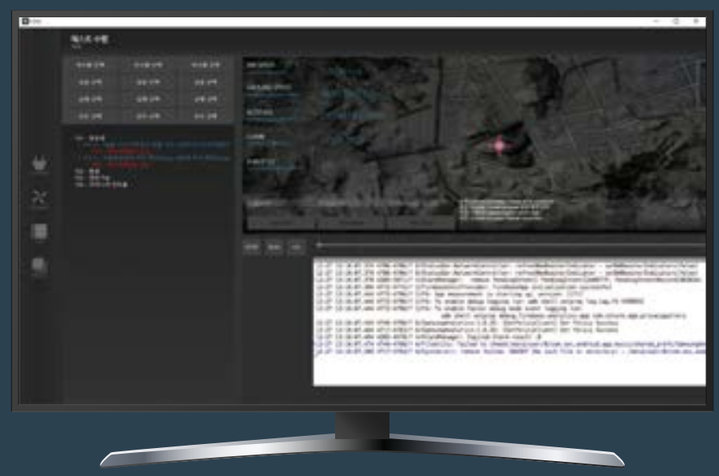
Software

Actuator (Motor etc.)

Virtual Sensor

Value (Serial)

Drone flight Command (Serial)



Drone installed Software testing screen

THINKTOOLS TWO.DRONACE
DETAILS FROM DRONACE

Functions & Effects

The following are 7 requirements defined for automated testing of CPS software modules.

01 Hooking
- Inject sensor data (GPS, Acceleration, Angular speed, Magnetic field, Battery) of external environment to be tested between the software and the sensor
- Even if the drone is located indoors, the data injected into the sensor is utilized to make it virtualize the test site as an external environment.

02 Sniffing
- The installed software extorts signals sent to the actuator (motor) for autonomous flight or mission fulfillment
- No movement because no signal is actually transmitted to the motor, but the drone Software is made to judge as if moving normally.

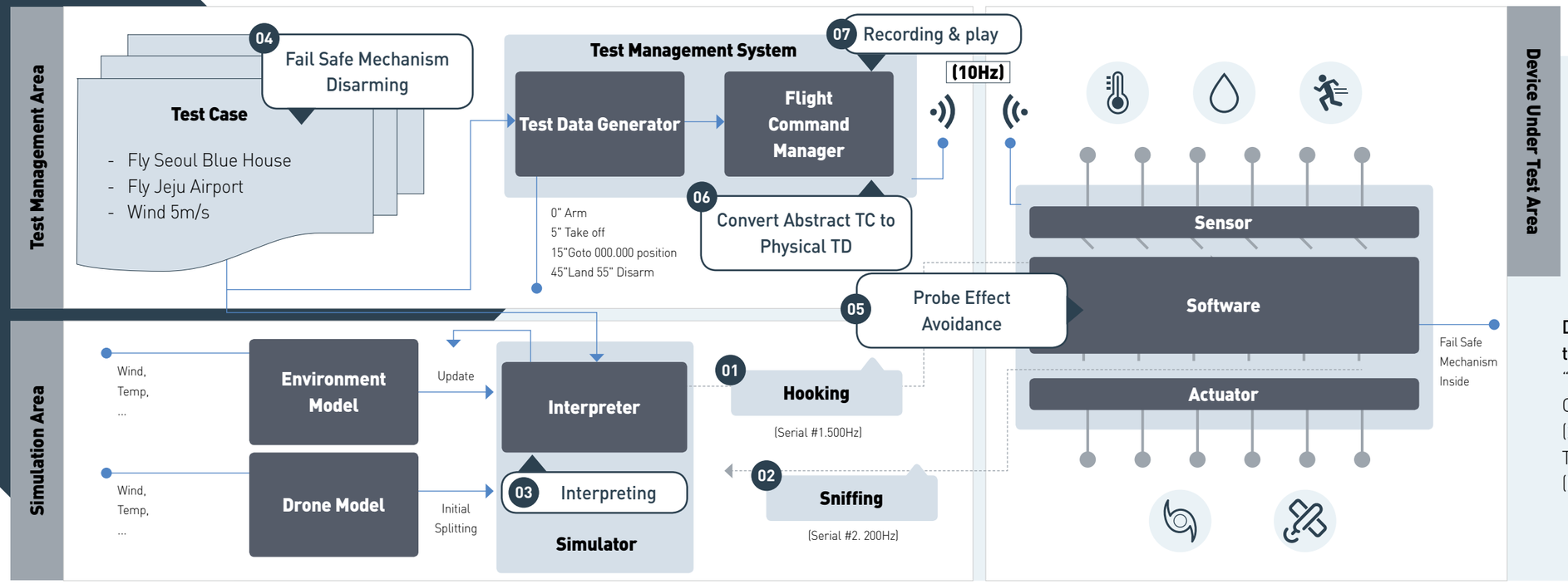
03 Interpret of actuator control raw value
- By analyzing the signal sent to actuator (motor) according to the body type and number of motors, interpret which behavior was ordered to the aircraft
- Reproduce and inject sensor data corresponding to the interpreted movement so that it can determine that the drone is actually moving

04 Disarm fail safe mechanism
- Disabling the built-in error monitoring function from detecting sensor data artificially injected into the drone
- Injecting all virtual data that occur from previous test environment normally to the next test environment or forced rebooting normally.

05 Avoid Probe Effect
- Considering the performance-sensitive test environment features, avoid the impact of virtual test environments on actual aircraft performance.
- Separately configure by dividing between the sensor data virtually injected and the transmission and reception process of sensor data

06 Convert the abstract test scenario to physical level virtual data
- Modeling algorithms that converts the abstract language of test scenarios into virtual sensor data at the actual physical level
- Generating all interpolation data in time units to perform the corresponding test scenario (500 Hz)

07 Support Test Recording and Replay
- Support test recording play as an auxiliary function for test result checking and debugging
- Same as the airplane's blackbox, configuring to reproduce the then-current flight situation



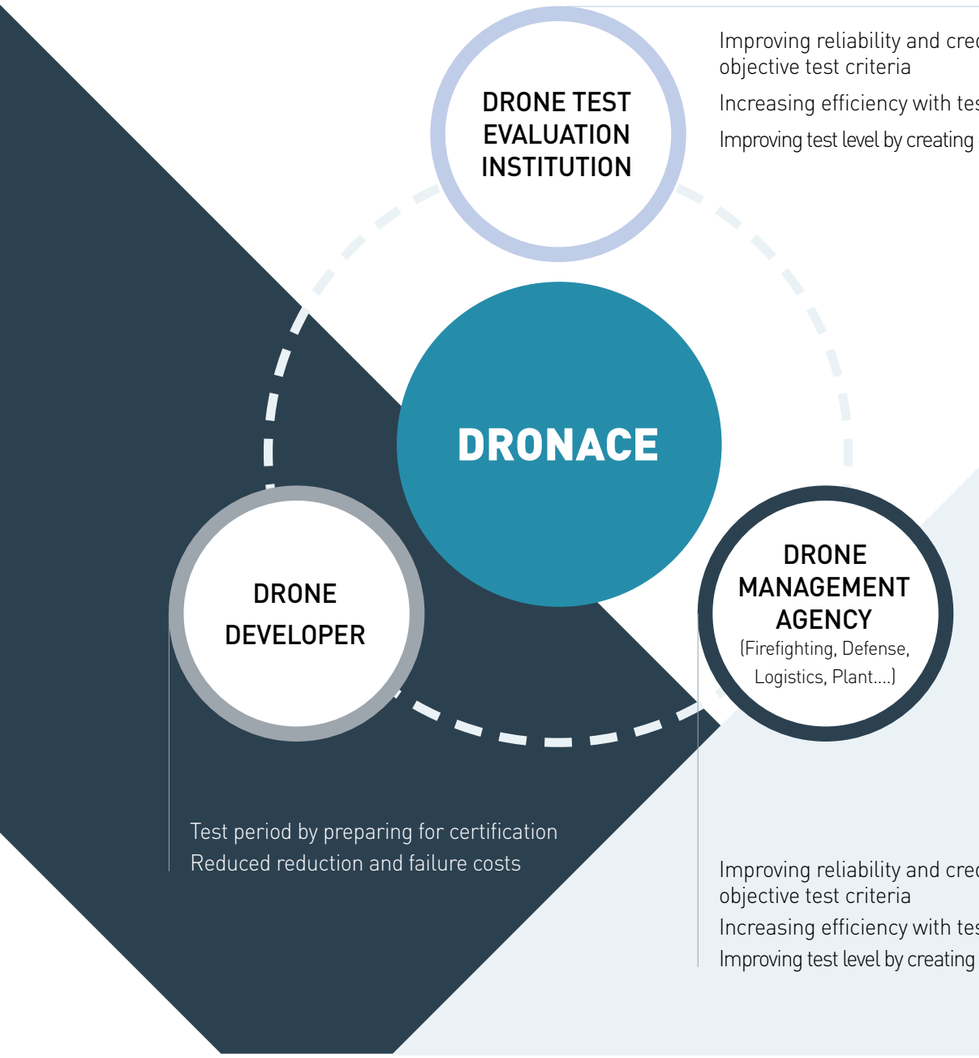
Defined 7 requirements for automated testing of the CPS Software module

"Suggestion of Testing Method for Industrial Level Cyber-Physical System in Complex Environment." (2019), IEEE International Conference on Software Testing, Verification and Validation Workshop (ICST), 2019, pp 148-152

THINKTOOLS TWO.DRONACE
DETAILS FROM DRONACE

How to use DRONACE

Creating safer drone world with less time and effort investment



Improving reliability and credibility by presenting objective test criteria
Increasing efficiency with test automation
Improving test level by creating diverse test conditions

Test period by preparing for certification
Reduced reduction and failure costs

Improving reliability and credibility by presenting objective test criteria
Increasing efficiency with test automation
Improving test level by creating diverse test conditions

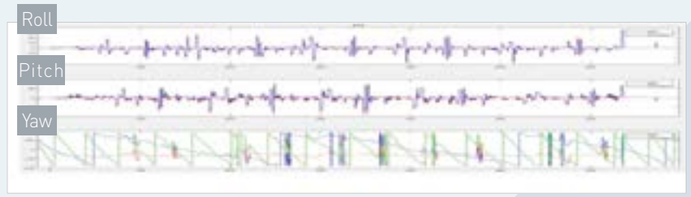
Implementing simulation technology for testing specialized drone models per domain of diverse industries

- Flight simulation after remodeling the drone is tested in more detail by analyzing the specifications of each newly developed drone's industrial domain characteristics
- By adjusting numerous variable factors such as drone length, weight, and attached motor characteristics, is achieved flight simulation with up to 98% accuracy for the drone to be tested

Safety Mechanism Structure - Safe ISO 26262 Deliverable D3.6b

Drone Flight Simulator reflect Weather

Accuracy comparison graph of simulation and actual flight



THINKTOOLS THREE. SMAC
WHAT IS SMAC?

Artificial intelligence makes your Electric/Electronic code as safety code.

Safety code is code that reflects the safety mechanism and that can identify, avoid, and recover from diverse risk circumstances.

- Identification of risk
- Avoidance of risk
- Recovery in case of risk

Instead of code safety securing consulting that safety experts had to do manually, artificial intelligence technology completes your code securely and safely.
SMAC's artificial intelligence is more objective, professional, and safe than the subjective consulting of individuals.

Registrations
- Method and Apparatus For Analyzing Safety Of Software (Patent 10-1734872)
- Method and Apparatus For Analyzing Safety Of Automotive Software (Patent 10-1675986)

PCT
- Method and Apparatus For Analyzing Safety Of Automotive Software (PCT/KR2017/006036)
- Method and Apparatus For Analyzing Safety Of Software (PCT/KR2016/010738)

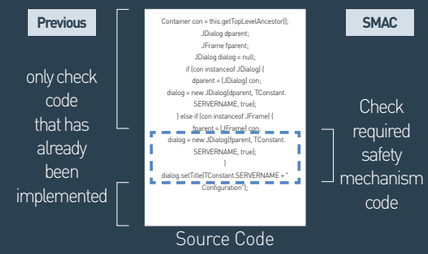
Code Analysis Result Screen

Action Plan Screen

WHY
SHOULD YOU USE SMAC?

In the Mission Critical Area, mistakes are not allowed.

Existing code checking tools fail to guarantee safety. These tools only check code that has already been implemented, while failing to verify the implementation of the critically required safety mechanism code. Even if expert consulting is conducted to solve this issue, reliance on individual consultants' subjective experience and capability is unavoidable. SMAC is an artificial intelligence that learned the contents of Safety Mechanism Structure of Safe ISO 26262 Deliverable D3.6b, the international standard in the field of electric/electronic and that analyzes the developed code to discover places in need of safety mechanisms to provide guidance.

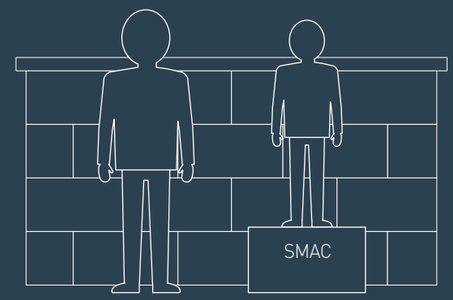


The lack of objective criteria in the delivery and inspection phase only leaves uncomfortable friction.

How can you trust if test results vary according to the grader?
How do you evaluate when the definition of good code differs by individual?
SMAC is learned to recognize the patterns that have been evaluated as good codes in the electric/electronic field and evaluates developed code.
Albeit not being the absolute standard, SMAC helps to present more universal standards.

You cannot wait advanced engineers forever who may never appear.

No matter how high you offer the salary, it takes time to improve the ability.
Especially in the field of electric/electronic, the word mistake must not be allowed. Even if fostering expert workforce by investing hundreds of millions in labor costs, the final results are not guaranteed. A beginner level engineer using SMAC can guarantee the safety at the expert level.

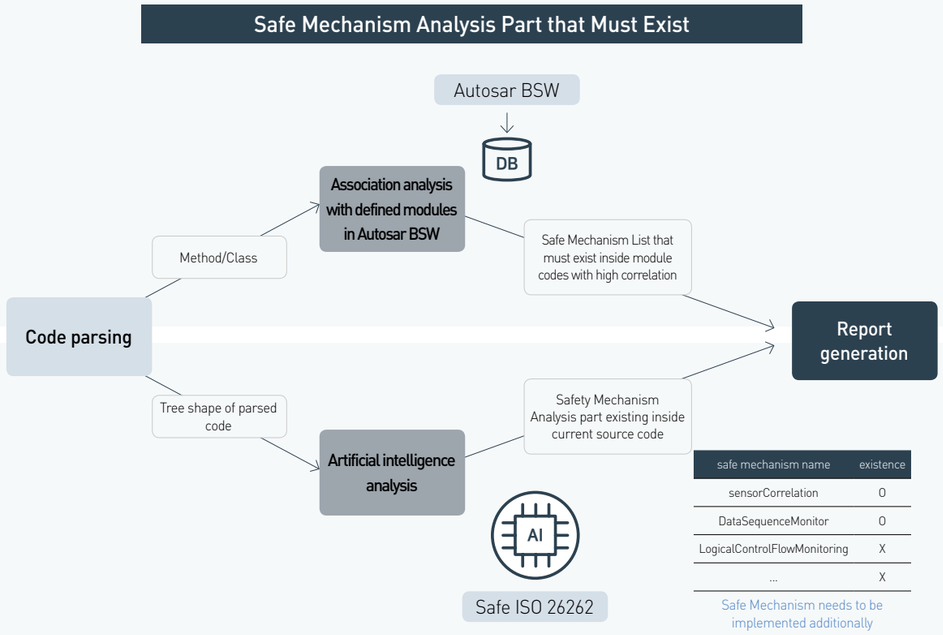


THINKTOOLS THREE. SMAC
DETAILS FROM SMAC

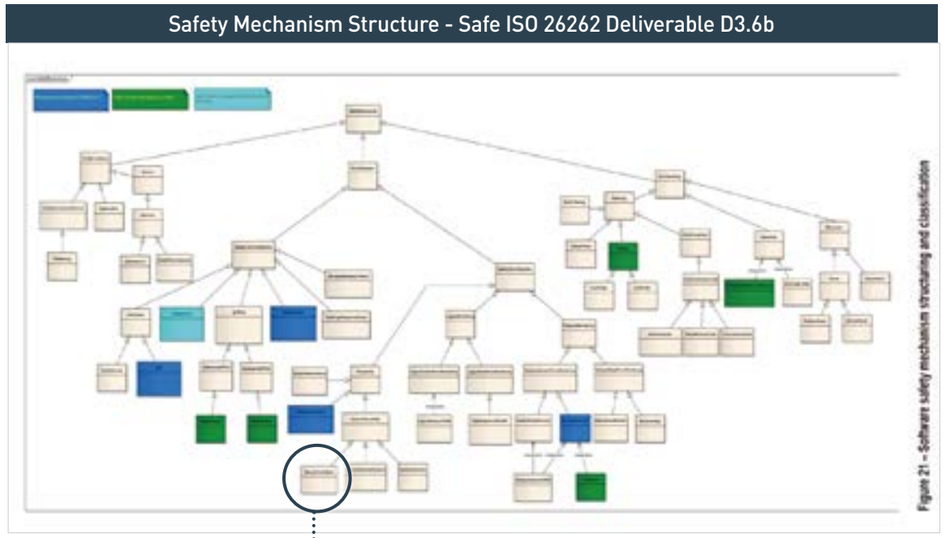
Functions & Effects

SMAC is the world's first artificial intelligence safety mechanism analysis tool.

SMAC analyzes the code inside electric/electronic Software to determine and notify at which location safety mechanism is required to become a safety code.
SMAC analyzes the need for additional Safety Mechanism for the method and components based on pre-learned artificial intelligence from source code
It analyzes and notifies which Safety Mechanism is required in the corresponding class or method location after receiving all inputs such as the divided code string and metric variable information of the source code.



The process of training the AI machine
to recognize diverse types of code as specific Safety Mechanism.



Safety Mechanism – SensorCorrelation

We love technology,
but not as much as being human.

Activating healthy hearts in all software functions
Allowing the wings of drones to fly with belief
Making ears and eyes of vehicles to protect human
For your better life

Think for a better life

ThinkforBL

SEOUL

LS BLDG., 6F, 15,
Nonhyeon-ro 87-gil,
Gangnam-gu, Seoul,
Korea

TEL
+82-2-562-6545
FAX
+82-2-562-6549

JEONJU

Deoksoo BLDG 5F,
20, Hongsan 1-gil,
Wansan-gu,
Jeonju-si,
Jeollabuk-do,
Korea

TEL
+82-63-246-6545
FAX
+82-63-246-6546

SINGAPORE

8 Shenton #04-01
AXA Tower,
Singapore
068811

VIETNAM

Y5 Office,
Hai Ba Trung Street,
Ben Nghe ward,
District 1,
Ho Chi Minh City,
Vietnam 700000

www.thinkforbl.com
E-mail contact@thinkforbl.com