

The Needs Analysis of Software Safety Education Program for Common Competency Area

Ji-Woon Kang* and Sung-Ryong Do**

Abstract

As the era of the 4th Industrial Revolution enters, the importance of software safety is increasing, but related systematic educational curriculum and trained professional engineers are insufficient. The purpose of this research is to propose the high priority elements for the software safety education program through needs analysis. For this purpose, 74 candidate elements of software safety education program were derived through contents analysis of literature and nominal group technique (NGT) process with five software safety professionals from various industries in South Korea. Targeting potential education participants including industrial workers and students, an on-line survey was conducted to measure the current and required level of each element. Using descriptive statistics, t-test, Borich needs assessment and Locus for focus model, 16 high priority elements were derived for software safety education program. Based on the results, suggestions were made to develop a more effective education program for software safety education.

Keywords

Borich Needs Assessment, Education Program, Locus for Focus Model, Needs Analysis, Software Safety

1. Introduction

In 2014, the Korean government declared the “Strategies to realize a Software-Oriented Society,” stressing that software should be at the center of innovation, growth, and value creation [1]. In particular, the development of embedded software source technology was declared important in industries such as aviation, automobiles, and railways. The Ministry of Public Administration and Security has accelerated the implementation of the master plan for safety innovation established in 2015 and has been carrying out field-oriented tasks such as realizing life-friendly safety policies, establishing preventive and initial response plans, and expanding disaster safety infrastructure [2]. On the other hand, the central government, local governments, and public institutions are performing detailed tasks and checking them periodically with the master plan for safety innovation.

As software safety-related systems and embedded software grow in scale and complexity, assuring safety is becoming more and more of an important issue [3-5]. Problems arising from software are not mere defects or malfunctions of services and products, but are becoming critical issues that can lead to enormous economic losses to the country, organization, and society and even further, loss of lives [6-9].

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Manuscript received December 31, 2020; first revision May 17, 2021; accepted August 1, 2021.

Corresponding Author: Sung-Ryong Do (imdosungryong@gmail.com)

* Telecommunications Technology Association, Seongnam, Korea (jiun.kang@gmail.com)

** Dept. of Convergence Software, Kyungmin University, Uijeongbu, Korea (imdosungryong@gmail.com)

Given the recent cases of sudden, unintended acceleration crashes of cars and plane crashes, it is essential to ensure the functional safety of software in a safety-essential system. In order to ensure software safety in these safety-critical systems, it is necessary to improve the capabilities of software engineers.

However, there are some problems. First, the majority of software engineers are not mindful or fully aware of software safety and in many cases, lack expertise in various software engineering safety technologies. This is because of the characteristics of software, especially its invisibility and complexity, which means it does not have a physical form and is difficult to verify the final result during the development process [10,11]. Second, there is a shortage of personnel trained in software quality. Since it is the result of intellectual labor of human resources, the quality of the result depends on the expertise of that human resource, and therefore a more systematic and professional approach is required. However, according to [12] the software industry's human resource supply and demand status, the number of domestic IT quality engineers is quite small (2.6%), and the training for incumbent employees related to software quality and testing is only 1.7% [13].

Therefore, it is necessary to systematically develop software safety education programs. This research suggests high priority elements for software safety education programs in the common competency area based on needs analysis, in order to enhance software engineers' safety awareness and expertise.

2. Literature Review

2.1 Prior Research on Software Safety

The analysis targets are as follows. Chin et al. [14] developed a software safety-related education curriculum proposal by investigating the roles and skills necessary for developing safety essential software and investigating the expectations of the end-users. In addition, the authors [15] designed a software safety professional qualification system consistent with international safety standards by investigating the required software safety activities from various perspectives, including development planning, design measures, test strategies, product management strategies, etc., based on IEC 61508, ISO 26262, DO-178B/C and EN 50128. Telecommunications Technology Association developed Software Safety Inspection Guide to improve the reality that national institutions have difficulty in diagnosing software safety, with the aim of ensuring national safety and enhancing safety [16]. Universities such as Southern California and Johns Hopkins offer educational curriculums for each safety activity such as safety requirements, safety analysis, safety design, and safety testing [14].

As such, an educational curriculum for software safety, a knowledge system for qualification systems, and safety guidelines at industrial sites have been proposed in accordance with international requirements. However, in the case of results dealing with the system and scope of knowledge, no specific derivation process is presented, and there is a tendency to focus on opinions among experts. First of all, the software safety elements derived from the above research results were defined at a higher level.

Therefore, results should be derived through more objective procedures, and the approach for this process needs to be more systematic. In this research, we collected and reviewed software safety-related researches from various sources including research institutes, associations, and international or industry standards to list potential educational elements with high priority.

2.2 Educational Needs Analysis

When developing an education program, the process of design, development, implementation, evaluation, and improvement is performed based on an analysis of educational needs. The program development process represented by the instructional systems design (ISD) model also shows the importance of analysis. For example, which content is extracted from the analysis, the first step in the ADDIE (analysis, design, development, implementation, evaluation) model, determines the process of the training program and the success of the evaluation. According to Sork [17], educational needs come from the difference between the current and desirable levels of ability, including knowledge, skills, and attitudes that can change through education or learning opportunities.

The analysis of educational needs is important, and the needs can be identified through either quantitative analysis or qualitatively through literature and by methods such as surveys. A survey is the most commonly used method for identifying educational needs [18]. According to the research trends in needs analysis studies identified by Cho [19], studies on needs analysis show that there is not much effort being made in sorting priorities. In addition, a more systematic method is required because the required analysis is often a numerical comparison, such as a presentation of the average or proportion of items.

By reviewing the various methods of demand analysis that have been used as individual methods in various studies, Cho [20] suggested a packaged process of deriving education priorities using a t-test, the Borich needs assessment, and the Locus for Focus model. The process of applying the three different analysis methods can be described in five major steps.

- Step 1: t-test to see the difference between “what is” and “what should be.”
- Step 2: Borich needs assessment formula to set priority.
- Step 3: Coordinate plane results presented through the Locus for Focus model.
- Step 4: Identify the number of items in the HH segment (high importance/high discrepancy) of the Locus for Focus model and determine how many Borich needs are ranked with high priority.
- Step 5: Borich needs determine the top and bottom groups through identification of items that are redundant between the top priority items in the formula and the HH segment items in the Locus for Focus models.

By using the three different methods simultaneously, it is possible to supplement the deficiencies of each method and provide comprehensive information for better decision making.

3. Research Method

3.1 Research Subject and Data Collection

An online survey was distributed to workers and students who either have work experience in the software or safety related industry or have participated in trainings or seminars held by the Telecommunications Technology Association in South Korea up to August 2020. The online survey was conducted using Google-Forms for 2 weeks from September 17–18, 2020, and emails to encourage cooperation were sent out twice during the survey period. Of the total 2,978 questionnaires distributed, 281 copies (9.4% response rate) were collected, and finally, 259 copies (8.7% effective response rate) were used for the analysis, excluding 22 copies that were answered unfaithfully. The demographic information of the research subjects is described under Table 1.

Table 1. Demographic information of research subjects

	Category	Number of subjects	Percentile
Gender	Male	221	85.3
	Female	38	14.7
	Sum	259	100
Age	<20s	37	14.3
	30s	119	31.6
	40s	221	39.4
	50s	251	11.6
	>60s	8	3.1
	Sum	259	100
Education	Bachelor (2-year)	6	2.3
	Bachelor (4-year)	108	41.7
	Master	87	33.6
	Doctor	58	22.4
	Sum	259	100
Job title	Assistant	37	14.3
	Associate	39	15.0
	Research engineer	50	19.3
	Senior	66	25.5
	Principal	67	25.9
	Sum	259	100
Working period (yr)	No answer	11	4.3
	<1	24	9.3
	1–3	58	13.1
	3–5	86	10.8
	5–10	121	13.5
	10–15	174	20.5
	15–20	209	15.5
	>20	39	15.0
	Sum	259	100

3.2 Research Process and Tools

The purpose of this study was to suggest high priority elements for the common software safety education program. To this end, a needs analysis proposed by Cho et al. [21] was performed to derive candidate elements and grasp priority. First of all, the existing literature on software safety was reviewed, and the elements of a common education program in the software safety field were derived by applying nominal group technique (NGT). Based on the results from the literature review and NGT, an online survey was conducted to identify the gap between “what is” and “what should be” in order to derive the priorities for education needs. For the priority analysis, the t-test, Borich needs assessment, and Locus for Focus model were employed to identify the final priorities of the education elements. The process in this research is shown in Fig. 1.

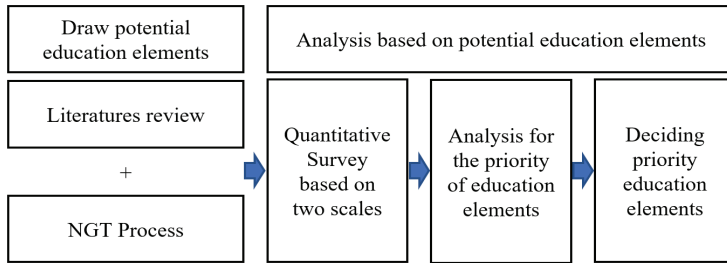


Fig. 1. Research process.

3.3 Validity of Survey Components

In order to secure the validity of the survey elements of this research, the degree of agreement was confirmed as to whether 74 potential educational components derived through the NGT and literature are suitable as the contents of the common educational program in the field of software safety. A questionnaire was conducted on 14 researchers in the QA/QC, test, and certification field of the software safety-related job in South Korea. According to Lawshe [22], the minimum content validity ratio (CVR) value of the components for 14 participants is 0.51. In the calculated result, there was one component (coding standard, CERT/CWE) that did not meet this the criteria. Although one element was not recognized for its validity, it was included in the needs analysis survey in reflection of several expert opinions that it is necessary for the overall composition. The potential 74 elements to be used in further surveys are shown in Table 2.

Table 2. Potential education components result

Detailed subject	Components
[a] SW overview	[1] definition and features of SW [2] concept of SW safety [3] concept of FS [4] concept of SWFS [5] importance of SWFS [6] method of SWFS
[b] SWFS standard	[1] necessity of SWFS standards [2] relationship between SWFS standards [3] establishment of SWFS standard process [4] IEC 61508 [5] ISO 26262 [6] IEC 62278(EN 50126), IEC 62279(EN 50128) [7] IEC 62304 [8] DO-178C [9] MIL-STD-882E [10] ISO 13482
[c] SWFS life cycle	[1] SWFS life cycle [2] SWFS life cycle in system level [3] SWFS life cycle in SW level [4] relationship between system and SWFS life cycle [5] gate activity for SWFS

Table 2. Continued

Detailed subject	Components
[d] Hazard analysis and risk assessment	[1] requirements of FS [2] overview of HA and RA [3] concept of safety integrity [4] concept of safety integrity level [5] malfunction identification [6] hazard identification [7] definition of hazardous event [8] SIL rating determination [9] definition of safety goal
Hazard Analysis process	
Risk Assessment process	
[e] FS analysis technique	[1] concept of FS analysis technique [2] PHL [3] PHA [4] HAZOP [5] FMEA [6] FTA [7] ETA [8] STPA
[f] SW Safety Requirements Specification	[1] FTTI and Fault [2] requirements description [3] SW architecture types and dependency [4] SW architecture design principle [5] module size and complexity [6] module design principle [7] safety requirements and mechanism [8] safety mechanism design technique
SW Architecture Design	
Detailed SW Design	
SWFS mechanism	
[g] Languages and supporting tool	[1] language consideration [2] tool qualification [3] static analysis [4] MISRA [5] CERT/CWE [6] cyclic complexity and source code size
Coding standard and static analysis	
[h] SWFS validation and testing	[1] V Model [2] unit / integration / system / acceptance test [3] equivalent partitioning [4] boundary value analysis [5] back to back [6] fault injection [7] statement [8] branch [9] MC/DC [10] review type [11] inspection
[i] SWFS management	[1] safety culture [2] SWFS plan [3] progress management [4] requirement management [5] configuration management [6] necessity of audit [7] audit process
SWFS supporting process	
SWFS audit	
[j] SWFS legislation and trends	[1] domestic legislation [2] international legislation [3] government [4] trends

3.4 Data Analysis

Analysis of the data collected through the survey was conducted as follows. First, a paired t-test was conducted to see if there is any significant difference between “what is” and “what should be.” Next, after applying Borich needs formula and the Locus for Focus model, the prioritization process proposed by Cho [20] was applied, and as a result, the education components with the highest priority for common software education programs in the software safety field were derived. First, the formula of Borich needs assessment, which was used as a method to determine priority, is as follows.

$$Borich\ needs = \frac{\sum(RCL - PCL) \times \overline{RCL}}{N} \quad (1)$$

where RCL is required competence level, PCL is present competence level, and N is total number of cases.

The Locus for Focus model used in this study is a method of allocating the elements into quadrants with the level of necessity as the horizontal axis and the degree of discrepancy between “what is” and “what should be” as the vertical axis, showing priority as a result. Among the four quadrants, the first quadrant has a high degree of necessity and high discrepancy, making it the highest priority area for educational components.

4. Results

4.1 Result of t-test, Borich Needs Assessment

As shown in Table 3, there was a statistically significant difference between “what is” and “what should be” for the common education program elements in software safety areas ($p < 0.001$). As a result of Borich needs value assessment, it was confirmed that the rest of the elements except for the “definition of SW(a1)” were the highest, in the order of a2, a6, a3, a4, a5 followed by the malfunction identification criteria (d5), ISO 13482 standard (b10), safety integrity level (SIL) rating determination (d8), and hazard identification (d6).

Next, the result of analyzing priority using The Locus for Focus model is shown in Fig. 2. The average level of “what should be” perceived by the survey respondents, who are potential participants of the education, was 3.90 and the level of discrepancy was 1.08. These two averages were divided into axes, and out of the four quadrants, 21 elements were included in the first quadrant with the highest priority.

Lastly, the components with the highest priority in Borich were compared with the 21 elements located in the first quadrant in the Locus for Focus model for redundancy, as shown in Table 4. A total of 16 elements were derived as top priority items, which are in detail, the concept of software and safety and functional safety (a2, 3, 4, 5, 6), establishment of a standard process (b3), ISO 26262 standard (b5), safety integrity concept (d3), malfunction identification (d5), hazard identification (d6), hazardous event definition (d7), architecture types and dependencies (f3), architecture design principles (f4), safety mechanism design techniques (f8), fault injection (h6), and functional safety plan establishment (i2).

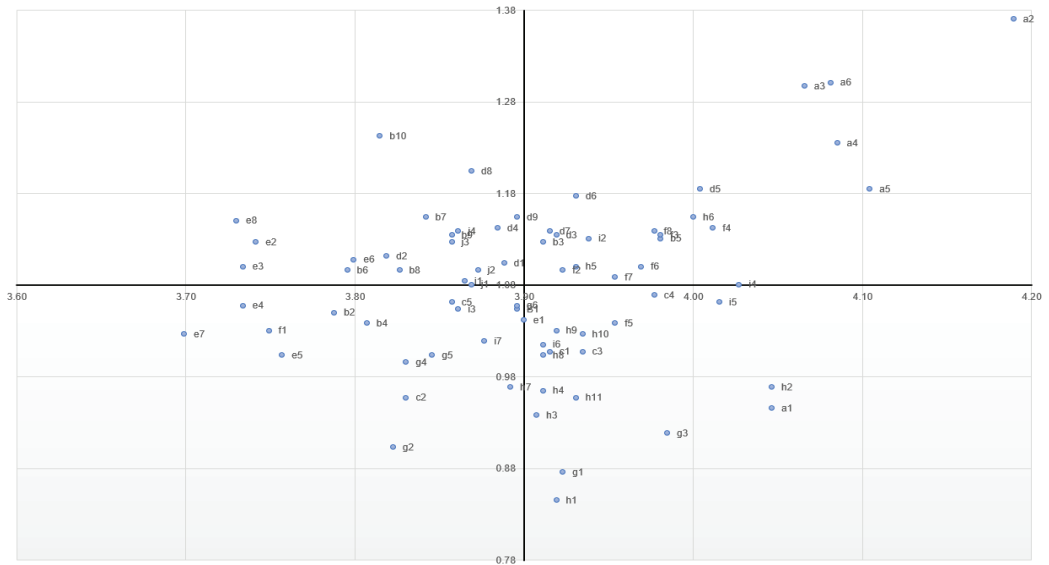


Fig. 2. Result of Locus for Focus model.

Table 3. Result of need analysis, t-test, Borish needs assessment

Items	What is		What should be		Discrepancy			Borich	Priority
	Mean	SD	Mean	SD	Mean	SD	t		
a1	3.10	0.87	4.05	0.87	0.95	1.11	13.77	3.83	62
a2	2.82	0.90	4.19	0.79	1.37	1.08	20.51	5.74	1
a3	2.77	0.92	4.07	0.81	1.30	1.06	19.69	5.27	3
a4	2.85	0.97	4.08	0.83	1.24	1.14	17.41	5.05	4
a5	2.92	0.94	4.10	0.88	1.19	1.18	16.21	4.86	5
a6	2.78	0.99	4.08	0.83	1.30	1.19	17.67	5.31	2
b1	2.84	0.95	3.90	0.86	1.05	1.03	16.42	4.11	44
b2	2.74	0.94	3.79	0.91	1.05	1.15	14.72	3.98	51
b3	2.78	0.90	3.91	0.86	1.13	1.05	17.33	4.41	21
b4	2.77	1.00	3.81	0.86	1.04	1.06	15.72	3.95	54
b5	2.85	1.02	3.98	0.98	1.13	1.14	16.01	4.50	14
b6	2.70	1.03	3.80	1.01	1.10	1.10	16.03	4.16	41
b7	2.69	1.04	3.84	1.03	1.15	1.16	16.05	4.44	20
b8	2.73	1.08	3.83	1.05	1.10	1.12	15.78	4.20	38
b9	2.72	1.09	3.86	1.07	1.14	1.14	16.05	4.38	23
b10	2.57	0.99	3.81	0.99	1.24	1.12	17.86	4.74	7
c1	2.91	0.92	3.92	0.84	1.01	1.09	14.89	3.95	57
c2	2.87	0.90	3.83	0.89	0.96	1.10	14.00	3.67	69
c3	2.93	0.90	3.93	0.86	1.01	1.02	15.85	3.96	53
c4	2.91	0.92	3.98	0.85	1.07	1.10	15.58	4.25	33
c5	2.80	0.96	3.86	0.84	1.06	1.03	16.55	4.10	46
d1	2.78	1.00	3.89	0.87	1.10	1.14	15.61	4.29	30
d2	2.71	0.99	3.82	0.89	1.11	1.11	16.13	4.25	35
d3	2.78	1.01	3.92	0.86	1.14	1.08	16.88	4.45	18
d4	2.74	1.01	3.88	0.83	1.14	1.07	17.12	4.44	19
d5	2.82	0.97	4.00	0.84	1.19	1.15	16.54	4.75	6
d6	2.75	0.98	3.93	0.85	1.18	1.15	16.42	4.63	9
d7	2.78	0.98	3.92	0.82	1.14	1.13	16.18	4.46	16
d8	2.66	0.99	3.87	0.83	1.20	1.02	19.09	4.66	8
d9	2.74	0.99	3.90	0.85	1.15	1.09	17.01	4.50	15

Table 3. Continued

Items	What is		What should be		Discrepancy			Borich	Priority
	Mean	SD	Mean	SD	Mean	SD	t		
e1	2.86	0.98	3.90	0.83	1.04	0.98	17.16	4.07	48
e2	2.61	0.98	3.74	0.83	1.13	0.98	18.55	4.22	36
e3	2.63	1.02	3.73	0.85	1.10	1.06	16.72	4.11	43
e4	2.68	1.02	3.73	0.86	1.06	1.11	15.38	3.95	56
e5	2.75	1.01	3.76	0.90	1.00	1.10	14.67	3.77	67
e6	2.69	0.99	3.80	0.86	1.11	1.03	17.34	4.21	37
e7	2.67	0.98	3.70	0.86	1.03	1.06	15.57	3.80	64
e8	2.58	1.00	3.73	0.86	1.15	1.05	17.68	4.29	31
f1	2.72	1.02	3.75	0.85	1.03	1.09	15.24	3.86	60
f2	2.83	0.95	3.92	0.86	1.10	1.06	16.57	4.30	29
f3	2.85	0.91	3.98	0.81	1.14	1.00	18.26	4.52	13
f4	2.87	0.91	4.01	0.85	1.14	1.01	18.12	4.58	11
f5	2.92	0.95	3.95	0.84	1.04	1.04	16.06	4.11	45
f6	2.87	0.91	3.97	0.82	1.10	1.01	17.59	4.37	24
f7	2.86	0.93	3.95	0.82	1.09	1.05	16.67	4.30	28
f8	2.84	0.95	3.98	0.86	1.14	1.05	17.38	4.53	12
g1	3.05	0.95	3.92	0.92	0.88	1.05	13.40	3.44	73
g2	2.92	0.95	3.82	0.88	0.90	1.10	13.25	3.45	72
g3	3.07	0.94	3.98	0.86	0.92	1.06	13.92	3.66	71
g4	2.83	0.98	3.83	0.86	1.00	1.13	14.20	3.82	63
g5	2.84	0.96	3.85	0.82	1.00	1.01	16.06	3.86	61
g6	2.84	0.93	3.90	0.85	1.06	1.07	15.84	4.12	42
h1	3.07	0.98	3.92	0.85	0.85	1.08	12.62	3.31	74
h2	3.08	0.92	4.05	0.87	0.97	1.08	14.42	3.92	59
h3	2.97	0.96	3.91	0.90	0.94	1.08	14.03	3.67	70
h4	2.95	0.97	3.91	0.87	0.97	1.09	14.20	3.78	65
h5	2.83	0.99	3.93	0.84	1.10	1.04	17.02	4.33	27
h6	2.85	0.95	4.00	0.83	1.15	1.03	18.10	4.62	10
h7	2.92	0.99	3.89	0.85	0.97	1.11	14.10	3.77	66
h8	2.91	0.96	3.91	0.87	1.00	1.07	15.06	3.93	58
h9	2.89	0.98	3.92	0.79	1.03	1.01	16.47	4.04	50
h10	2.91	0.96	3.93	0.88	1.03	1.12	14.82	4.04	49
h11	2.97	0.95	3.93	0.82	0.96	1.03	14.99	3.76	68
i1	2.78	0.98	3.86	0.85	1.08	1.16	14.99	4.19	39
i2	2.81	0.95	3.94	0.81	1.13	1.06	17.25	4.46	17
i3	2.81	0.94	3.86	0.85	1.05	1.03	16.54	4.07	47
i4	2.95	0.93	4.03	0.87	1.08	1.04	16.66	4.35	25
i5	2.95	0.93	4.02	0.86	1.06	1.05	16.32	4.26	32
i6	2.90	0.93	3.91	0.87	1.02	1.00	16.28	3.97	52
i7	2.86	0.98	3.88	0.88	1.02	1.05	15.61	3.95	55
j1	2.79	0.99	3.87	0.92	1.08	1.14	15.26	4.18	40
j2	2.78	1.03	3.87	0.88	1.10	1.13	15.68	4.25	34
j3	2.73	0.99	3.86	0.86	1.13	1.16	15.65	4.35	26
j4	2.72	1.01	3.86	0.85	1.14	1.14	16.04	4.40	22

Table 4. Top priority education elements for software safety education program

Detailed subject	Components	Priority methodology	
		Borich	Locus for Focus
[a] SW overview	[2] concept of SW safety	o	o
	[3] concept of FS	o	o
	[4] concept of SWFS	o	o
	[5] importance of SWFS	o	o
	[6] method of SWFS	o	o
	[3] establishment of SWFS standard process	o	o
[b] SWFS standard	[5] ISO 26262	o	o
[d] Hazard analysis and risk assessment Hazard analysis process	[3] concept of safety integrity	o	o
	[5] malfunction identification	o	o
	[6] hazard identification	o	o
	[7] definition of hazardous event	o	o
[f] SW Safety Requirements Specification SW architecture design	[2] requirements description		o
	[3] SW architecture types and dependency	o	o
	[4] SW architecture design principle	o	o
	[6] module design principle		o
	[7] safety requirements and mechanism		o
[h] SWFS validation and testing	[8] safety mechanism design technique	o	o
	[5] back-to-back		o
	[6] fault injection	o	o
[i] SWFS management SWFS supporting process	[2] SWFS plan	o	o
	[4] requirement management		o

5. Discussion and Conclusion

In this study, a needs analysis was conducted to grasp the priority of necessary educational elements for the development of a common education program in the field of software safety, and the discussion based on the research results is as follows.

First, as a result of this study, 16 top-priority educational elements for software safety education programs have been derived. If an educational program needs to be organized within limited time and resources, the derived educational elements need to be reflected first. However, this does not mean that the entire software safety education program should be organized only with the derived high-priority educational elements, and other various factors such as the purpose of the program or the main target industry should be considered.

Second, looking at the 16 derived educational elements, we found that the concepts and definitions of software safety and functional safety are important for common field education programs and thus have high priorities. This study was conducted to develop educational programs that reflect the common characteristics of various industrial sectors at the foundation level. With this consideration, we confirm that the overall concept and understanding need to be reflected in the program in an in-depth manner.

Third, this study is designed as a systematic process and is a practical case of looking at high priority educational elements in the field of software safety through quantitative and qualitative research results. This can serve as an important reference for developing education programs in different levels or

industries, and it can be applied in various fields dealing with knowledge, including the design of qualification systems and the establishment of knowledge.

Lastly, this work shows the result of a needs analysis to present top priority training elements before implementation of educational programs. The practical process from designing, implementing, operating, and evaluating the development of educational programs has not been presented in this study, which remains to be the limitation of the study. The mid-course results or post-training evaluation should also be analyzed and reflected for further improvement of the program, and through this process, the enhanced software safety education programs will surely lead to higher educational effectiveness.

References

- [1] National Research Council for Economics, "Presentation of strategies to realize a software-oriented society," 2014 [Online]. Available: <https://eiec.kdi.re.kr/policy/materialView.do?num=134467>.
- [2] National Research Council for Economics, "Detailed report of safety innovation master plan," 2015 [Online]. Available: <https://www.korea.kr/news/pressReleaseView.do?newsId=156043103>.
- [3] D. Ferraris, C. Fernandez-Gago, and J. Lopez, "A model-driven approach to ensure trust in the IoT," *Human-centric Computing and Information Sciences*, vol. 10, article no. 50, 2020. <https://doi.org/10.1186/s13673-020-00257-3>
- [4] A. Soury, A. M. Rahmani, N. J. Navimipour, and R. Rezaei, "A symbolic model checking approach in formal verification of distributed systems," *Human-centric Computing and Information Sciences*, vol. 9, article no. 4, 2019. <https://doi.org/10.1186/s13673-019-0165-x>
- [5] M. M. Rad, A. M. Rahmani, A. Sahafi, and N. N. Qader, "Social Internet of Things: vision, challenges, and trends," *Human-centric Computing and Information Sciences*, vol. 10, article no. 52, 2020. <https://doi.org/10.1186/s13673-020-00254-6>
- [6] J. S. Park and J. H. Park, "Advanced in algorithms, security, and systems for ICT convergence," *Journal of Information Processing Systems*, vol. 16, no. 3, pp. 523-529, 2020.
- [7] L. Wang, C. Jin, and C. Xu, "An evaluative study of the operational safety of high-speed railway stations based on IEM-Fuzzy comprehensive assessment theory," *Journal of Information Processing Systems*, vol. 16, no. 5, pp. 1064-1073, 2020.
- [8] The Korea Economic Daily, "Boeing CEO 'software defect' 737 max crash, negligence admitted in a month," 2019 [Online]. Available: <https://www.hankyung.com/economy/article/201904055724i>.
- [9] Now News, "Tesla, another crash killed two... Is the autopilot a problem?" 2020 [Online]. Available: https://nownews.seoul.co.kr/news/newsView.php?id=20200102601003&wlog_tag3=naver.
- [10] E. M. Choi, *New Software Engineering*. Seoul, Korea: Jungiksa, 2014.
- [11] National IT Industry Promotion Agency, *Software Engineering White Book*. JinCheon, Korea: National IT Industry Promotion Agency, 2018.
- [12] Information Technology Industry Skills Council, *2019 Software Industry Labor Supply and Training Demand Survey Report*. Seoul, Korea: Korea Software Industry Association, 2020.
- [13] W. S. Park, M. S. Kim, O. M. Kwon, and Y. S. Nam, *2018 SW Industry Personnel Supply and Training Status Report*. Seoul, Korea: Korea Software Industry Association, 2018.
- [14] H. S. Chin, T. H. Park, S. Y. Min, E. K. Ji, J. B. Ryu, and J. H. Park, "A study on the development of educational curriculum to improve employee competency in software safety field," 2017 [Online]. Available: <https://spri.kr/posts/view/21816?code=research>.

- [15] H. S. Chin, T. H. Park, K. H. Park, H. R. Lim, M. K. Ki, and S. J. Lee, "A study on the qualification system of SW safety expert," 2017 [Online]. Available: <https://spri.kr/posts/view/21818?code=research>.
- [16] Telecommunications Technology Association, *Software Safety Inspection Guide*. Seongnam, Korea: Telecommunications Technology Association, 2017.
- [17] T. J. Sork, "Needs assessment," in *Fundamentals of Adult Education: Issues and Practices for Lifelong Learning*. Toronto, Canada: Thompson Educational Publishers, 2001, pp. 101-115.
- [18] K. Oh and J. I. Choi, *Education Program Development Methodology*. Seoul, Korea: Hakjisa, 2005.
- [19] D. Y. Cho, "The state of need analysis research for building programs for adults: 1990-2005," *Andragogy Today: Interdisciplinary Journal of Adult & Continuing Education*, vol. 9, no. 1, pp. 85-106, 2006.
- [20] D. Y. Cho, "Exploring how to set priority in need analysis with survey," *The Journal of Research in Education*, vol. 2009, no. 35, pp. 165-187, 2009.
- [21] D. Y. Cho, E. J. Jung, A. R. Joo, and H. Y. Shin, "The need analysis of outsourcing training programs for school principals," *Andragogy Today: Interdisciplinary Journal of Adult & Continuing Education*, vol. 15, no. 4, pp. 149-172, 2012.
- [22] C. H. Lawshe, "A quantitative approach to content validity," *Personnel Psychology*, vol. 28, no. 4, pp. 563-575, 1975.



Ji-Woon Kang <https://orcid.org/0000-0002-2805-388X>

He is a senior researcher in the human resource development department of Telecommunications Technology Association. He received a bachelor's degree from the school of computer engineering, Dongguk University in 2011. He is currently majoring in human resource development and adult continuing education in the department of education, Korea University as a doctoral candidate. His current research interests include engineering education and qualification.



Sung-Ryong Do <https://orcid.org/0000-0003-3394-8220>

He is an associate professor in the department of convergence software, Kyungmin University. He received M.S. and Ph.D. degrees in Computer Science from Sangmyung University, Korea, in 2010 and 2016, respectively. He worked at Hyundai Autron in Hyundai Motor Group from 2012 to 2017. His current research interests include software engineering, software process and software safety.