On the Current State of Academic Software Testing Education in Sweden

Ayodele A. Barrett
Mälardalen University, Sweden
ayodele.barrett@mdu.se

Eduard Paul Enoiu
Mälardalen University, Sweden
eduard.paul.enoiu@mdu.se

Wasif Afzal
Mälardalen University, Sweden
wasif.afzal@mdu.se

***Abstract*—Well-trained software development personnel, in the art and science of software testing, will effectively and efficiently develop quality software products with potentially fewer, less-critical defects. Thus software testing education is considered to be an important part of curricula for a university degree in Computer Science or Information Systems. The objective of this paper is to determine how much dedicated knowledge in the field of software testing is taught within Swedish universities. To achieve this objective, a systematic search of syllabi for software testing-related courses was done. From 25 Swedish universities offering Computer Science (or related) degrees, 14 currently offer dedicated courses in software testing. Some findings include: 32% of the individual courses were offered at the undergraduate level; 28% of the universities offer courses for specialised testing training; and, for the vast majority of the universities, dedicated software testing courses account for about 5% of the total degree credits. While some universities fare better than others, the overall state of academic software testing education in Sweden is limited but promising.**

***Index Terms*—software testing, software testing education, Sweden**

I. Introduction

Software testing (ST), verification and validation are some of the most important approaches to ensuring high-quality software [9]. Existent research suggests that activities undertaken to verify and validate software during development often account for a substantial part of the project's budgetary and time allocations [3]. Indeed, Hynninen *et al.* [4] state that testing and quality assurance are the most expensive set of activities during the development life-cycle of software. Utting and Legeard [5] suggest a range of between 30 and 60% of the overall development effort used by software testing, while Wong [9] puts the estimate at more than 60%.

Despite such aforementioned studies documenting that a sizeable percentage of development costs is spent on software quality assurance, research papers suggest that there is an over-emphasis on teaching software development to the detriment of teaching ST in university curricula. In 2005, Chan *et al.* [10] put the figure at as low as 28% of university-trained software testers surveyed who had received formal software testing training whilst at university. A decade later, Lemos *et al.* [7] note that it is not uncommon for Computer Science graduates to complete their studies without knowing how to test their code. Jesus *et. al.* [15] concur in their observation that ST education, at the undergraduate level, regularly is overlooked in favour of software design and implementation activities, resulting in a noticeable dearth of practised software testers.

One reason cited for the lack of well-trained software testers, or indeed knowledgeable IT professionals in general, is the minimal proportion of computing curricula dedicated to software testing. This suggests that students, and subsequently members of software development teams, are ill-equipped proactively to prevent defect propagation in earlier phases of software development or efficiently detect defects in later phases of a project [1], [13]. Thus syllabus inadequacy with regard to software testing forms the basis of this paper.

Specifically, the paper focuses on providing information on software testing education at Swedish universities. In addition, a comparison between the course offerings from these universities and a vendor-neutral organisation offering various certifications in ST is undertaken.

The aims of this paper thus are:

1) to outline the current state of software testing education at Swedish universities;
2) to identify areas of curriculum commonality between Swedish university testing courses and a commercial software testing training entity.

The paper is organised as follows: In Section II, the situational context is set for the paper. There is also discussion on prior, similar studies on the state of software testing education in other regions. There is a brief description, in Section III, of the methodology utilised in this paper. In Section IV, various syllabi for testing courses are examined. In Section V, comparison is made between academic course offerings and those from a commercial training entity. Section VI contains a discussion of the results. Threats to validity for the paper are presented in Section VII with the paper's conclusion presented in Section VIII.

II. Background and Related Work

A 2010 survey of randomly selected universities in Canada and the United States found that it was not uncommon for computer science degree programs to omit completely, courses dedicated to teaching software testing (ST) from their curricula. With emphasis on testing tools and relevant Systems Under Test (SUT) used to demonstrate course work, the result showed that two of the top five Canadian universities omitted standalone testing courses, while the number in the United States was seven out of 10 universities [16].

Chan *et al.* [10] conducted a Hong Kong-based study of companies with independent test teams. It was found that most of the employees had not received formal university training in ST. Additionally, the survey revealed that while IT employees were given additional training, the provision of training support for employees was not consistent across the different industries surveyed.

The state of undergraduate ST education in Brazil was analysed [12]. The study compared recommendations by the Brazilian Computer Society (SBC) to course curricula of 25 Brazilian universities. Additionally, the curricula of 21 international universities were compared to recommendations from the Association for Computing Machinery (ACM). These universities were from the US (13), the UK (3), Switzerland (2), China (1), the Netherlands (1) and Singapore (1). For both the Brazilian and international sets of universities, it was found that too few lectures were allocated for teaching testing courses. Thus, ST practices were inadequately covered.

A survey of South African IT practitioners was conducted to highlight the importance of ST skills for graduates [21]. The survey was undertaken to justify the need for creating specialised tertiary qualifications for students wishing to pursue a career in ST and quality assurance in South Africa.

Melo *et al.* [11] conducted a survey to provide a worldwide perspective, albeit with the exclusion of Africa, on how lecturers covered ST. The issues identified included course content, teaching approaches, educational resources utilised and how students were examined. Among their findings were similarities in topics covered, with functional testing being the most popular and that the traditional, classroom-based approach to teaching was most commonly used,.

A 2017 online survey by Scott *et al.* [28] was conducted to investigate the state of software development practices in Estonia and Sweden. 13 Swedish companies responded, of which 92% were large to very large companies (above 250 employees). The focus of the paper was primarily on software development practices. Nevertheless, six of the 36 questions were related to testing and quality assurance. Respondents could choose answers from various options, of which one was "we always use it." Some findings were: the rate of companies that always conducted *code review* was 38%, *security testing* was 8% and *end-to-end (system) testing* was at 15%.

A. Context - About Sweden

Sweden has been described as the Silicon Valley of Europe, a nod to its embrace of technology, willingness to innovate and availability of quality infrastructure [30]. Sweden's high adoption rate of technology has been credited, in part, to a government ICT policy from 2000 (An Information Society for All) seeking to, amongst other things, put a computer in every home and develop competence in ICTs[1]. Hence, Sweden is the birthplace of several well-known and global technology-based companies in areas such as telecommunication (Ericsson[2]), entertainment (Spotify[3]) and finance (Zettle[4]).

About three decades ago, the software testing (ST) scene in Sweden was abecedarian. It was a time when opportunities, for example training in the field, were limited. Accordingly, the Swedish Association for Software Testing[5] (SAST) was founded in 1995 and helped foster a community of software testers in Sweden. Having grown to over 3,000 members, seminars are held quarterly each year for members to meet and share ideas. There is also a developed relationship with academia. Local and international leaders from both industry and academia are often invited as speakers.

Academically, Sweden is home to a total of 39 universities, of which 14 are one-faculty, speciality universities such as the Karolinska Institute[6], focused exclusively on medical and health sciences. There are a further 25, generalised universities all with departments dedicated to teaching courses relating to Computer Science, Information Systems or similar[7].

III. Research Method

The method adopted is an iterative process drawing on Ellis' model of information gathering and extraction [17] and a follow-up study on its applicability to the World Wide Web [18]. In brief, information-seeking activities are grouped into six categories *viz.*:

1) Start - Identify sources of interest;
2) Chain - Follow up on relevant references found in given material;
3) Browse - Engage in a semi-directed search of contents or headings;
4) Differentiate - Use any differences between sources to filter material;
5) Monitor - Receive updates on any developments from selected sources;
6) Extract - Work through a source to systematically identify interesting material.

A. Implementation

To identify relevant information needed to study the state of software testing (ST) education in Sweden, the inquiry began with two websites: *antagning.se* (for Swedish speakers) and *universityadmission.se* (for English speakers). Both are official websites for anyone wishing to study at a Swedish university and are managed by the University and Higher Education Council (UHR) of Sweden[8]. Students are provided with pertinent information such as eligibility criteria and important registration dates for different universities. Students can also search for information about specific universities or

[1]https://www.regeringen.se/rattsliga-dokument/proposition/2000/03/prop.-1999200086/ (accessed: January 2023)

[2]https://www.ericsson.com/ (accessed December 2022)
[3]https://www.spotify.com/ (accessed December 2022)
[4]https://www.zettle.com/ (accessed December 2022)
[5]https://www.sast.se/about.jsp (accessed: January 2023)
[6]https://ki.se/en (accessed: October 2022)
[7]https://studyinsweden.se/universities/ (accessed: October 2022)
[8]https://www.uhr.se/en/start/ (accessed December 2022)

courses[9,10]. Identification of these two websites corresponds to Step 1 (Start) of Ellis' model.

To use either of the sites one is required to choose, via a drop-down menu, from a list of university terms (Spring, Summer or Autumn). After which, one must enter a keyword in an adjacent text field. In the event of a positive search, course names and corresponding universities are displayed. Clicking on a course name will result in the user being re-directed, externally, to the course web page on the university's website. These actions correspond to Steps 2 (Chain) and 3 (Browse) of Ellis' model. The following keywords were used in this search (Swedish equivalents are in italics):

- Software testing (*Mjukvarutestning*)
- Software quality assurance (*Kvalitetssäkring av programvara*)
- Quality assurance (*Kvalitetssäkring*)
- Verification (*Verifiering*)
- Validation (*Validering*)

Discrepancies were identified in search results between the two websites. As one example, searching universityadmissions.se with parameters [*Autumn 2022*, *software testing*] yielded the result of one university and one of its courses *(MDU: Model-based testing in practice)*. A corresponding search of antagning.se produced two different universities, each with one course *(Uppsala University: Mjukvarutestning)* and *(Dalarna University: IT-säkerhet och mjukvarutestning)*. This corresponds to Step 4 (Differentiate) of Ellis' model.

Due to differences in search results between the two websites, the decision thus was made to conduct a manual search of the websites of universities offering technology courses. Nine universities were found not to provide information regarding dedicated ST courses. The list of universities is shown in Table I. Subsequently, the websites for universities offering ST courses were further analysed, the results of which are discussed in Section IV. These actions correspond to Step 6 (Extract) of Ellis' model. Step 5 (Monitor) was not relevant to this research.

B. Target Courses

The search was restricted to courses dedicated solely to teaching ST in their entirety. As a result, a number of courses were eliminated from consideration based on analysis of course contents and the table of contents of any prescribed textbook or, unavailability of course information. The list of eliminated courses is shown in Table II.

IV. Results - University Test Education

As described in Section III, software testing (ST) courses and associated universities were identified first by searching the official Swedish universities' admissions websites. Subsequently, department websites were searched also. As shown in Table I, of the 25 universities offering technology-related degrees, 16 were found to offer courses dedicated to ST. This number subsequently was reduced to 14. The University of Skövde and Karlstad University were omitted (see Table II). As such it is determined that just over half, 56%, of all the universities listed in Table I offer courses dedicated solely to teaching ST. This figure is not dissimilar to the 60% finding for Canadian universities in the 2010 study by Garousi and Mathur [16].

TABLE I
Software Testing (ST) education at Swedish Universities.

Offer ST Courses	**Do not Offer ST Courses**
Blenkinge Institute of Technology (BTH)	Borås, University of
Chalmers University of Technology	Gävle, University of
Dalarna University	Kristianstad University
Gothenburg, University of	Malmö University
Halmstad University	Mid Sweden University
Jönköping University	Södertön University
Karlstad University	Umeå University
KTH Royal Institute of Technology	University West
Linköping University	Örebo University
Linnæus University	
Luleå Technical University	
Lund University	
Mälardalen University (MDU)	
Skövde, University of	
Stockholm University	
Uppsala University	

The rest of this section contains the frame of reference used in determining the state of testing education in Sweden *viz.* course content (Section IV-A), level of study (Section IV-B), delivery mode (Section IV-C), course plurality (IV-D) and student assessment (Section IV-E).

A. Course Content

To identify what aspects of software testing (ST) are most commonly taught at Swedish universities, the courses listed in Table III are divided into two categories *viz.* specialised courses and non-specialised courses. Referring to line numbers from the aforementioned table, courses from Blenkinge (lines 2, 3, 4), Halmstad (line 8), Karlstad (line 11), Linnæus (line 15) and MDU (lines 19, 20) are deemed to be specialised courses based on process (e.g. agile testing, model based testing), testing type (penetration testing, user acceptance testing) or system type (embedded systems, web and mobile applications). The areas of commonalities are analysed within the 16, non-specialised courses.

Reviewing contents of the respective syllabi, non-specialised courses were shown to cover topics relating to some or all of the following five categories: (i) test levels (ii) test types (iii) static testing (iv) automation and/or tools (v) test management.

It was observed that there were differences in the terminology used. As an example, for the test level category, Linköping's course[11] (Software Testing) used the terms "...testing at the unit, *module* and system level", while Stockholm's

[9]https://www.antagning.se (accessed: October 2022)

[10]https://www.universityadmissions.se (accessed: October 2022)

[11]https://studieinfo.liu.se/pdf/en/kursplan/TDDD04/ht-2022 (accessed: January 2023)

TABLE II
COURSES ELIMINATED FROM CONSIDERATION

University	Course Name	Reason
Blenkinge	Software Quality Management	Emphasis is placed on a strategic approach to quality management
Dalarna	System Maintenance and Testing of IT Systems	The emphasis is on technical IT service management
Karlstad	Quality in Data Driven Systems	No information has been provided regarding the course content and material
	Automation Technologies in Software Development	Course covers automation of all software engineering activities
KTH	Automated Software Testing and DevOps	Course spans development and operations
Skövde	Software Testing	The course is currently unavailable

TABLE III
SWEDISH UNIVERSITIES' SOFTWARE TESTING COURSES

	University	Course Name	Level[a]	Credit	Delivery	Valid From
1.	Blenkinge	Software Testing	M	7.5	Campus	2017
2.		Adaptive Lean Software Testing	M	7.5	Distance	2020
3.		Quality Assurance of Security Aware Applications	M	6	Distance	2022
4.		PEN testing and Ethical hacking	M	7.5	Distance	2023
5.	Chalmers	Quality Assurance and Testing*	M	7.5	Campus	2021
6.	Gothenburg	Quality Assurance and Testing*	M	7.5	Campus	2021
7.		Software Quality and Testing	U	7.5	Campus	2023
8.	Halmstad	Testing and Verification of Embedded Systems	M	7.5	Campus	2019
9.	Jönköping	Software Product Quality Assurance	M	6	Campus	2016
10.	Karlstad	Software Testing Foundations	U	7.5	Distance	2020
11.		User Tests, Prototyping and Evaluation	U	7.5	Campus	2010
12.	KTH	Software Reliability	M	7.5	Campus	2020
13.		Software Reliability	U	9	Campus	2019
14.	Linköping	Software Testing	M	6	Campus	2022
15.	Linnæus	Testing for the Web and Mobile Applications	M	3.5	Distance	–
16.		Software Testing	U	7.5	Campus	2020
17.	Luleå	Testing IT Systems	U	7.5	Distance	2019
18.	Lund	Software Testing	M	7.5	Campus	2022
19.	MDU	Automated Test Generation	M	2.5	Distance	2019
20.		Model Based Testing in Practise	M	2.5	Distance	2018
21.		Quality Assurance - The Applied Science of Software Testing	M	7.5	Distance	2018
22.		Software Verification and Validation	M	7.5	Campus	2016
23.	Stockholm	Introduction to Testing of IT Systems	U	7.5	Campus	2019
24.	Uppsala	Software Testing	M	5	Campus	2019
25.		Test Methodology	U	5	Campus	2010

[a]Level: M = Master's; U = Undergraduate *Joint course

course[12] (Introduction to Testing of IT Systems) used the terms "...testing at unit, *integration* and system level." Additionally, there were differences in the breadth of areas of coverage. For example, Karlstad's course[13] (Software Testing Foundations) referred broadly to "different levels of testing", Linköping's course (Software Testing) explicitly referenced three test levels "unit, module and system", while Linnæus' course[14] (Software Testing) explicitly referenced four test levels "unit, integration, system and acceptance".

Table IV contains a very high overview of each course's contents[15]. As shown in the table all of the courses cover, to a lesser or greater degree, the area of test levels, test types, the use of tools and/or automation, and aspects of managing the test process such as test planning or generating test documentation. An interesting observation is that just two of the courses explicitly referenced static testing (ST). At face value, this appears to confirm observations by Shepard *et al.* who are of the opinion that "*[h]ighly effective practices such as software inspection are hardly taught at all*" [1, p. 103].

[12]https://sisu.it.su.se/pdf_creator/20801/54291 (accessed: January 2023)

[13]https://www3.kau.se/kursplaner/en/DVGB17_20201_en.pdf (accessed: January 2023)

[14]https://kursplan.lnu.se/kursplaner/syllabus-1DV609-1.pdf (accessed: January 2023)

[15]Course names have been abbreviated to accommodate for table width

B. University Level

There have been calls to introduce software testing (ST) education as early as possible at the undergraduate level [1], [9]. Testing is a broad subject requiring knowledge of many concepts, techniques and tools [2] and too little time spent on undergraduate testing courses may result in testing professionals who are not adequately trained [12]. As argued by Dale, "*We cannot teach the students to write correct and easily tested code without introducing them to the theory and practice of software testing [...] Testing must begin in CS1 and be reinforced in each succeeding course.*" [6, p. 364]. Referring to data from Table III (page 4), this section will contain an analysis of ST courses that are offered at either the undergraduate or the postgraduate level of Swedish universities.

TABLE IV
COURSE CONTENT: BASIC SOFTWARE TESTING
TL: TEST LEVELS, TT: TEST TYPES, ST: STATIC TESTING, A/T: USE OF TOOLS/AUTOMATION, TM: TEST MANAGEMENT

University	Course	TL	TT	ST	A/T	TM
BTH	Software Testing	x	x		x	x
Chalmers	QA & Testing	x	x		x	x
Gothenburg	QA & Testing	x	x		x	x
	Qual. & Testing	x	x		x	x
Jönköping	Product QA	x	x		x	x
Karlstad	Foundations	x	x		x	x
KTH	S/w Reliability	x	x		x	x
	S/w Reliability	x	x		x	x
Linköping	Software Testing	x	x		x	x
Linnæus	Software Testing	x	x		x	x
Luleå	Software Testing					
Lund	Software Testing	x	x	x	x	x
MDU	Sc. of Testing	x	x		x	x
	Software V&V	x	x		x	x
Stockholm	Testing IT Sys.	x	x	x	x	x
Uppsala	Software Testing	x	x		x	x
	Test Methodology	x	x		x	x

There are seven universities which offer ST courses only at the Master's level. These are BTH, Chalmers, Halmstad, Jönköping, Linköping, Lund and MDU universities. This means that half, or 50%, of all identified universities, make available testing courses only at the advanced level. Furthermore, of these eight, six of these universities offer only a single course at the Master's level. BTH and MDU are the only universities which offer a plurality of ST courses at the Master's level. Halmstad University, however, serves as a special case. Its sole course is marked as an elective course. Thus, there is a possibility that registered students need never study a dedicated ST course.

Three universities only offer ST courses at the undergraduate level. Karlstad offers two ST courses at the undergraduate level with Luleå and Stockholm each offering a single, undergraduate course. The final four universities, Gothenburg, KTH, Linnæus and Uppsala, offer courses both at the undergraduate and Master's levels.

C. Course Delivery Mode

Swedish universities provide education either via distance learning or, by requiring a physical presence on campus. Distance education, a mode of education that offers an alternative to classroom-based education, provides students the flexibility to study without being physically present at a place of learning. There is an increase in the number of universities moving towards digitising their courses [26] accelerated, undoubtedly, by the recent, global pandemic [33].

Advantages of distance learning, relative to classroom-based learning, include widening access to individuals who would otherwise not be able to study due to constraints such as time or distance. A major disadvantage often cited, however, is the lack of social interaction and a feeling of isolation among students. A Sweden-based, longitudinal study on teaching software testing (ST) via distance, web-based learning identified the feeling of distance between students and teachers as one of several challenges faced by students [26]. Nevertheless, Luleå University states that while its testing course is conducted at a distance over the Internet, the lessons are teacher-led and students are grouped together for assignments, such that "*[b]y working in groups for a large part of the course, the student learns to collaborate and communicate*[16]..." Incidentally, Luleå is the only university that does not offer a campus-based version of a ST course. That is, its sole testing course is offered via distance.

Table III (page 4) contains information on specific delivery modes for each testing course. Karlstad and Linnæus both offer one course for distance and campus-based learning. BTU and MDU, both with the largest number of ST course offerings, offer a similar number of distance courses, three apiece. Both universities offer also one campus-based course. Nine universities, the majority, offer their testing courses as campus-based. These are Chalmers, Gotenburg, Halmstad, Jönköping, KTH, Linköping, Lund, Stockholm and Uppsala.

D. Course Plurality

It has been argued that testing and quality assurance is broad and there should be several courses to accommodate this [1]. Wong [9, p. 2] concurs, stating "*[s]oftware testing is an extremely broad subject, and even a dedicated one-semester course cannot adequately cover all the important concepts and techniques with an appropriate level of detail*." In this section, the plurality of subject areas, specialisation and coverage of the identified testing courses will be considered.

For the analysis in this section, the two courses from KTH will be considered the same course. This is because the Master's level course is presented as being different to the graduate-level version only by adding a 1.5 credit project to the graduate version[17]. Nine of the 14 universities offer only a single course in software testing (ST). Furthermore, all of these courses are what have been categorised earlier, in Section IV-A, as basic testing courses. As shown in Table IV, there was considerable overlap in the contents of each of these courses. It is acknowledged, however, that there likely will be differences in the breadth and depth of each of these courses.

Only two universities provide students with the opportunity potentially to specialise in different types of testing. The two universities are also the only institutions with a plurality of ST courses. These are Blenkinge and Mälardalen universities. Offering a total of four ST courses, a student studying at Blenkinge could potentially specialise in the area of security testing. Mälardalen offers a similar number of ST courses in which a student could potentially specialise in. Karlstad and Linnæus, in providing two disparate testing courses apiece, offer additional possibilities for specialisation albeit limited. Thus, of the 14 universities offering courses in ST, four of these offer options for potential specialisation.

Finally, in terms of coverage, ST continues to occupy but a fraction of the curriculum. As an example, Stockholm

[16]https://www.ltu.se/edu/course/I00/I0015N/I0015N-Test-av-IT-system-1.81215?termin=V23&kursView=kursplan

[17]https://www.kth.se/student/kurser/kurs/FDD3459?l=en

University's course, Introduction to Testing of IT Systems, with 7.5 credits, is part of a Bachelor's degree made up of 180 credits. Thus, the proportion of the syllabus with a dedicated ST course is 4%. At the Master's level, the situation is dire also. As an example, Linköping University's course, with 6 credits, is part of a Master's program made up of 120 credits. This means that the dedicated ST course accounts for 5% of the syllabus.

E. Student Assessment

Effective assessment of students, by lecturers, is important to assessing student grasp of course curricula. Assessments can be either direct or indirect. Examples of the former include assignments, exams, presentations and projects [23]. Various research papers on software testing (ST) education differ in preferred ways of assessing student coursework. As examples, Aniche *et al.* [8] stated their preference for lab work, which the authors considered to be an important learning method. Lambers [27] favoured a project-based approach to assessment while Enoiu [26] documented the use of assignments.

There is no uniform approach to assessing students in each of the testing courses being considered. Table V provides an overview of the preferred method of assessing student progress. Assessment types are: Assignment (A), Project work (Pj), Presentation (Ps), Lab work (L) and Exam (E). Numbers in relevant cells signify what portion of the total credit is allocated to a given assessment type. The total number of credits will correspond to that which was given earlier in Table III (page 4).

With the exception of exams, the other assessment types like assignments, projects or lab sessions are offered as tasks undertaken either by individuals or by groups of students. The most common form of assessment is via assignments. The second-most popular form of assessment are written exams. The least-most popular form of assessment are oral presentations.

A number of courses credit allocations are entirely allocated to assignments. These are BTH's Adaptive Lean Software Testing course and three of MDU's courses: Automated Test Generation, Model-based testing and Quality Assurance - The Applied Science of Software Testing. The credits for Linköping's course, Software Testing, also are allocated for just one assessment type - lab work.

Three universities omit information relevant to this section. Karlstad University does not provide the actual breakdown of credits per assessment type for both its course offerings. Similarly, Uppsala University's course on Software Testing does not provide how many credits are allocated to assignments and the exam although it provides relevant information for the undergraduate course - Test Methodology. Linnæus' course on web and mobile testing, however, omits completely which assessment types are utilised.

TABLE V
COURSE ASSESSMENT

University	Course	A	Pj	Ps	L	E
BTH	Software Testing	2.5	5			
	Lean Testing	7.5				
	Security Testing	1	3	2		
	PEN Testing	1.5			6	
Chalmers	QA & Testing	3				4.5
Gothenburg	QA & Testing	3				4.5
	Qual. & Testing	3				4.5
Halmstad	Embedded Sys.	1.5	1.5			4.5
Jönköping	Product QA	2			2	2
Karlstad	Foundations	x			x	x
	User Testing			x	x	x
KTH	S/w Reliability-M				3	4.5
	S/w Reliability-U		1.5		3	4.5
Linköping	Software Testing				6	
Linnæus	Web Testing					
	Software Testing	3		0.5	3	1
Luleå	Software Testing	3				4.5
Lund	Software Testing		3			4.5
MDU	Automation	2.5				
	MBT	2.5				
	Sc. of Testing	7.5				
	Software V&V	2			1.5	4
Stockholm	Testing IT Sys.	3.5				4
Uppsala	Software Testing	x				x
	Test Methodology		2			3

V. INDUSTRY TEST TRAINING

Commercial training organisations provide opportunities, in addition to universities, for both learning and the dissemination of knowledge. While there are views that consider these organisations to be competitors of universities, there are opposing views believing these training companies can be complementary to universities. Indeed, as noted by Hitchcock [19], job advertisements oftentimes include a preference for industry certifications in addition to tertiary degrees, although Rob and Roy [29] quote research suggesting that having IT certifications is not a significant factor used in organisational hiring practices. Nevertheless, organisations are known to provide additional training for their employees [10].

The field of software testing (ST) is served by a number of commercial training establishments that offer both self-study and face-to-face training opportunities. Organisations such as the International Institute for Software Testing (IIST)[18], the Quality Assurance Institute (QAI)[19] and the International Software Qualification Board (ISTQB)[20] were established to provide vendor-neutral certification for individuals wishing to gain proficiency in ST. The latter is selected for discussion in this section being the only one that both is a non-profit organisation and, also provides information regarding how many certifications have been issued.

Founded two decades ago, ISTQB offers ST courses in over 130 countries through a number of accredited training providers and, under the auspices of regional or national member boards. The Swedish standard is overseen by the Swedish Software Testing Board (SSTB)[21].

[18]https://testinginstitute.com/about.php (accessed December 2022)

[19]ttp://www.qaiusa.com/software-certifications/software-testing-certifications/ (accessed December 2022)

[20]https://www.istqb.org/ (accessed November 2022)

[21]https://www.sstb.se/se (accessed December 2022)

The entry-level certificate, the Foundation Level Certification (CTFL), forms the cornerstone for specialising in three streams as depicted in Figure 1.

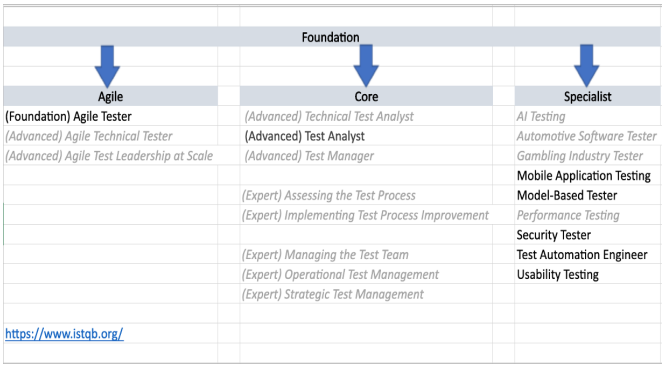

Fig. 1. University Courses vs. ISTQB

Identified in Table VI is a list of Swedish universities that provide standalone courses corresponding to *specialist* certifications offered by ISTQB. The list, however, does not include universities whose courses may include sections covering topics offered by ISTQB. As an example, Jönköping University is omitted because Agile testing is listed as a section of its ST course offering.

TABLE VI
SE UNIVERSITIES VS. ISTQB

ISTQB Course	Hours	University	Credits
Agile Testing (Advanced)	16h-	BTH	7.5
Mobile application testing	12h55´	Linnæus	3.5
Model-based testing	12h25´	MDU	2.5
Security testing	16h45´	BTH	6
Test automation	19h30´	MDU	2.5
Usability testing	14h05´	Karlstad	7.5

Whilst university courses generally offer greater academic depth, certifications are quicker to attain [19]. Table VI contains information which confirms this difference. It most be noted, however, that to register for ISTQB core, advanced and expert levels candidates are required to have a sufficient practical experience. The number of years of experience differs from course to course.

In Table VI, the number of hours scheduled for each ISTQB course is provided. For each university course the number of credits, based on the European Credit Transfer System (ECTS), is provided. The ECTS is an agreed-upon credit system used by European governments and universities for the easy comparison and transfer of study credits between countries and institutions of learning. In Sweden, one ECTS credit (*högskolepoäng*) is equivalent to 25–30 study hours [32]. The hours listed for the ISTQB courses are direct contact time with trainers, while ECTS includes both direct contact times and time spent in self-study.

Universities reviewed offer fewer opportunities for specialised ST disciplines. It must be noted, however, that there are two specific courses of ST offered at the universities but with no similar or corresponding ISTQB course:

- Penetration testing (BTH)
- Embedded systems testing (Halmstad)

VI. Discussion

With regards to course content, of the 25 courses analysed, 17 of the courses are generalised ST courses leaving only 8 courses for the possibility to specialise. Also, half of the universities reviewed do not offer ST at an undergraduate level, offering courses only at the Master's level. Additionally, nine of the 14 universities offer only campus-based lessons, leading to less flexibility for those disadvantaged by time or distance.

The proportion of ST in syllabi is dire. At the undergraduate level, out of a possible 180 course credits, the percentage ranges from as low as 4% (Stockholm University, a single 7.5 credit course) to a high of 8% (Karlstad University, two 7.5 credit courses). With the exception of BTH and MDU which both offer multiple courses, the ratio at the Master's level is just as in need of attention. Out of a possible 120 course credits, the percentage ranges potentially from 0% (Halmstad University's course is an elective) to a high of 6.25% (e.g. Chalmers, a single 7.5 credit course).

In comparing course offerings at universities to certifications offered by ISTQB, there is a wide gap in the diversity of areas of specialisation. While university courses offer greater breadth and depth in coverage of relevant topics, Linnæus and MDU, by offering specialised courses with fewer credits, demonstrate the possibility of offering courses that have the rigour typically associated with university education and the relative brevity associated with certifications.

VII. Threats to Validity

Some of the threats to the validity of this study are discussed in this section. To address *internal validity*, search strings were iterated on by conducting several initial searches. Aiming to minimise bias in the interpretation, analysis, and selection of the gathered sources, results were discussed and reviewed by multiple authors. When considering *construct validity* of the study, an already proposed method was employed. The information gathering and extraction model was based on the approach proposed by Ellis *et al.* [17]. In addition, the data was examined and checked several times to realise an agreement on the obtained results in this study. Regarding *external validity*, as the research is being conducted in the specific domain of testing and focused only on education in Sweden, more studies are needed to get a better view of academic verification and validation education in Europe and worldwide.

VIII. Conclusions

Just over two decades ago, Jones [2] lamented the fact that software testing (ST) often was treated as an afterthought in computer science curricula, that is, a curriculum postscript of sorts. The author further stated that "*educators must do a better job equipping students with skills and attitudes for dealing effectively with software quality concerns*" [2, p. 337].

Prior to that, an alarm was raised that "*traditional beliefs and practices pervading curricula ... namely, that software engineering was "additional material" to be covered, often in isolated upper-division courses, and not a natural component of all computer science courses*" [6, p. 364] continues to ring true with respect to ST in the Swedish context.

In this paper, a review of ST courses, as offered by Swedish universities, is presented. Having excluded single-faulty universities, the analysis covered 25 courses from 14 different universities. This paper laid bare, based on syllabi information garnered from the universities' websites, what offerings there are for students wishing to formally study or specialise in the field of ST within Sweden.

In future research, a study should go beyond the basic syllabi made available on departmental websites. Additionally, surveys with Sweden-based testing lecturers could help shed greater insight into how the dearth of software testing courses could be addressed.

ACKNOWLEDGMENT

We acknowledge the support of Mälardalen University through the Promote lifelong learning (FLL) grant and ModTest project. FLL is an MDU university-wide project aimed at developing and strengthening opportunities for MDU to offer education to those looking for further training, upskilling and reskilling. The authors also thank Carl-Anton Werner Axelsson for his valuable comments on this work.

REFERENCES

[1] Shepard, T., Lamb, M. and Kelly, D., "More Testing Should Be Taught," Communications of the ACM, vol. 44, no. 6, pp. 103–108, June 2001.

[2] Jones, E. L., "Integrating testing into the curriculum - Arsenic in small doses," SIGCSE Bulletin, vol. 33, no. 1, pp. 337–341, March 2001.

[3] Andersson, C. and Runeson, P., "Verification and Validation in Industry - A Quantitative Survey on the State of Practice," Proc. Symposium on Empirical Software Engineering, pp. 37–47, October 2002.

[4] Hynninen, T., Kasurinen, J., Knutas, A. and Taipale, O.,"Guidelines for software testing education objectives from industry practices with a constructive alignment approach," Proc. Conference on Innovation and Technology in Computer Science Education, pp. 278—283, July 2018.

[5] Utting, M. and Legeard, B., Practical Model-Based Testing - A Tools Approach, 1st Edition, 2006.

[6] McCauley, R., Dale, N., Hilburn, T., Mengel, S. and Murrill, B. W., "The Assimilation of Software Engineering into the Undergraduate Computer Science Curriculum," Proc. Technical Symposium on Computer Science Education, pp. 364–365, March 1995.

[7] Lemos, O. A. L., Ferrari, F. C., Silveira, F. F. and Garcia, A., "Experience Report: Can Software Testing Education Lead to More Reliable Code?," Symposium on Software Reliability Engineering, Gaithersbury, USA, 2–5 November 2015.

[8] Aniche, M., Hermans, F. and van Deursen, A., "Pragmatic Software Testing Education," Proc. Technical Symposium on Computer Science Education, pp. 414–420, March 2019.

[9] Wong, W. E., "Improving the State of Undergraduate Software Testing Education," ASEE Annual Conference & Exposition, San Antonio, USA, 10 - 13 June 2012.

[10] Chan, F. T., Tse, T. H., Tang, W. H. and Chen, T. Y., "Software Testing Education and Training in Hong Kong," International Conference on Quality Software, Melbourne, Australia, 19–20 September 2005.

[11] Melo, S. M., Moreira, V. X. S., Paschoal, L. N. and Souza, S. R. S., "Testing Education: A Survey on a Global Scale," Proc. Brazilian Symposium on Software Engineering, pp. 554–563, October 2020.

[12] Valle, P. H. D., Barbosa, E. F. and Maldonado, J. C., "CS Curricula of the Most Relevant Universities in Brazil and Abroad: Perspective of Software Testing Education," Proc. Symposium on Computers in Education, pp. 62—68, November 2015.

[13] Carver, J. C. and Kraft, N. A., "Evaluating the Testing Ability of Senior-level Computer Science Students," Proc. Conference on Software Engineering Education and Training, pp. 169–178, May 2011.

[14] Clegg, B. S., Rojas, J. M. and Fraser, G., "Teaching Software Testing Concepts Using a Mutation Testing Game," Conference on Software Engineering, Buenos Aires, Argentina, 20–28 May 2017.

[15] de Jesus, G. M., Ferrari, F. C., Paschoal, L. N., de Souza, S. R. S. and Porto, D. P., "Is It Worth Using Gamification on Software Testing Education? An Extended Experience Report in the Context of Undergraduate Students," Journal of Software Engineering Research and Development, vol. 8, pp. 1–19, 2020.

[16] Garousi, V. and Mathur, A., "Current State of the Software Testing Education in North American Academia and Some Recommendations for the New Educators," Proc. Conference on Software Engineering Education and Training, pp. 89–96, March 2010.

[17] Ellis, D., Cox, D. and Hall, K., "A Comparison of the Information Seeking Patterns of Researchers in the Physical and Social Sciences," Journal of Documentation, vol. 49, no. 2, pp. 356–369, 1993.

[18] Choo, C. W., Detlor, B. and Turnbull, D., "Information seeking on the web: An integrated model of browsing and searching," First Monday, vol. 5, no. 2, 2000.

[19] Hitchcock, L., "Industry Certification and Academic Degrees: Complementary, or Poles Apart?," SIGMIS CPR Conference, St. Louis, USA, 19–21 April 2007.

[20] Deak, A., Stålhane, T. and Cruzes, D., "Factors Influencing the Choice of a Career in Software Testing among Norwegian Students," Conference on Software Engineering, Innsbruck, Austria, 11–13 February 2013.

[21] Feyt, L., Calitz, A. P. and Greyling, J. H., "An Investigation into the Need for Incorporating Software Testing and Quality Assurance into a Higher Education Curriculum," SACLA '11, Durban, South Africa, 6–8 July 2011.

[22] Martins, L., Brito, V., Feitosa, D., Rocha, L., Costa, H. and Machado, I., "From Blackboard to the Office: A Look Into How Practitioners Perceive Software Testing Education," Conference on Evaluation and Assessment in Software Engineering, Trondheim, Norway, 21–23 June 2021.

[23] Alelaiwi, A., "Indirect Assessment of Student Learning in a Software Testing Course," Life Science Journal, vol. 11, no. 6, pp. 575–580, 2014.

[24] Hilburn, T. B. and Towhidnejad, M., "Software Quality: A Curriculum Postscript?," Proc. Technical Symposium on Computer Science Education, pp. 167–171, May 2000.

[25] Scatalon, L. P., Carver, J. C., Garcia, R. E. and Barbosa, E. F., "Software Testing in Introductory Programming Courses: A Systematic Mapping Study," SIGCSE '19, Minneapolis, USA, 27 February–2 March 2019.

[26] Enoiu, E. P., "Teaching Software Testing to Industrial Practitioners Using Distance and Web-Based Learning," Proc. Workshop on Frontiers in Software Engineering Education, pp. 73—87, November 2019.

[27] Lambers, L., "How to Teach Software Testing? Experiences with a Sandwich Approach," Proc. Conference on Software Testing, Verification and Validation Workshops, pp. 425–428, October 2020.

[28] Scott, E., Pfahl, D., Hebig, R., Heldal, R. and Knauss, E., "Initial Results of the HELENA Survey Conducted in Estonia with Comparison to Results from Sweden and Worldwide,", Conference on Product-Focused Software Process Improvement, Innsbruck, Austria, 29 November–1 December 2017.

[29] Rob, M. A. and Roy, A., "The Value of IT Certification: Perspectives from Students and IT Personnel," Issues in Information System, vol. 14, issue 1, pp. 153–161, 2013.

[30] Fulton, C. and Mukherjee, S., "How Sweden became the Silicon Valley of Europe," https://www.reuters.com/business/finance/how-sweden-became-silicon-valley-europe-2021-08-11/ (accessed December 2022).

[31] Silvis-Cividjian, N., Went, M., Jansma, R., Bonev, V. and Apostolov, E., "Good Bug Hunting: Inspiring and Motivating Software Testing Novices," Proc. Conference on Innovation and Technology in Computer Science Education, pp. 171–177, June 2021.

[32] European Commission, Directorate-General for Education, Youth, Sport and Culture, *ECTS users' guide 2015*, Publications Office, 2017, https://data.europa.eu/doi/10.2766/87192 (accessed: January 2023).

[33] Lavonen, J. and Salmela-Aro, K., "Experiences of Moving Quickly to Distance Teaching and Learning at All Levels of Education in Finland," In: Reimers, F.M. (eds) Primary and Secondary Education During Covid-19. Springer, pp. 105–123, 2022.