

Enhancing Software Testing Education

Sensemaking in testmodelling with TestCompass



I AM SORRY DAVE

MOVIEWALLPAPERS.NET



I AM AFRAID YOU DID NOT LEARN HOW TO TEST

imgflip.com

WHO AM I?

- → Niels Doorn
- → PhD Student @ Open Universiteit
 - → Studying students' sensemaking to improve teaching and learning strategies of software testing
 - → Supervisors: Tanja Vos, Beatriz Marín, Erik Barendsen, and Migchiel van Diggelen
- → Teamleader / lecturer-researcher @ NHL Stenden University of Applied Sciences



0

MY TOPIC: STUDYING STUDENTS' SENSEMAKING TO IMPROVE TEACHING AND LEARNING STRATEGIES OF SOFTWARE TESTING

- → My aim to design a teaching-learning strategy that supports students to learn software testing which:
- works in different educational contexts
- is integrable in existing courses
- can be taught from the start of the program

0

STUDYING STUDENTS' SENSEMAKING TO IMPROVE TEACHING AND LEARNING STRATEGIES OF SOFTWARE TESTING

- → Theme 1: The intended learning outcomes for software Testing
- → Theme 2: The sensemaking of students and test experts
- → Theme 3: Introducing testing early in introductory programming courses using Test Informed Learning with Examples

0

STUDYING STUDENTS' SENSEMAKING TO IMPROVE TEACHING AND LEARNING STRATEGIES OF SOFTWARE TESTING

- → Theme 1: The intended learning outcomes for software Testing
- Theme 2: The sensemaking of students and test experts
- → Theme 3: Introducing testing early in introductory programming courses using Test Informed Learning with Examples



SOFTWARE TESTING AND EDUCATION

Enhancing Software Testing Education



No more

Just enough tests for a passing grade

O[?]RLY

Niels Doorn

WHY DO WE NEED TO IMPROVE EDUCATION?



SOFTWARE TESTING IN EDUCATION: A CURRENT OVERVIEW

Treated as a secondary concern in computer science education

Lead engineers at Google call it the biggest gap between education and industry

Students' inability to adequately test software



TESTING IS INTRODUCED LATE! JUST LOOK AT THE BOOKS



TESTING IS INTRODUCED LATE! JUST LOOK AT THE BOOKS



- Three books give examples of test cases
- Three books contain a definition of testing
- Seven books introduce assert, of which two in appendix











COGNITIVE DEMANDS ON STUDENTS

Multitasking: Generating test cases, understanding code, and predicting outcomes

Analytical Thinking: for example Identifying edge cases & potential failures

Education focus is on solving problems, not on testing

CHALLENGES IN TEACHING SOFTWARE TESTING

Curriculum constraints

Over-reliance on theory vs. practical application

Complexity of software testing as an intellectual activity

THE ROLE OF SENSEMAKING IN SOFTWARE TESTING

Sensemaking is a dynamic process of building or revising an explanation in order to ``figure something out'' -- to ascertain the mechanism underlying a phenomenon in order to resolve a gap or inconsistency in one's understanding

(Odden and Russ 2019)





EXPLORATORY STUDY ON SENSEMAKING

EXPLORATORY STUDY ON SENSEMAKING

Pilot study: Identification of four different sensemaking approaches

Follow-up study with 50 students from a Spanish university

Methodology: Modeling test cases using Test Compass



Towards understanding students' sensemaking of test case design

Niels Doorn^{a,*}, Tanja E.J. Vos^{a,b}, Beatriz Marín^b
^a Open Universiteit, Valkenburgerweg 177, Serlen, The Netherlands
^b Universiteit Politeciaca de Valmeta, Camiro de Vera, s/n. 46022, Valencia, Spain

ARTICLE INFO	A B S T R A C T	
Keywords: Software Testing Sensemaking Software Engineering Computer science educational research Higher education	Context: Software testing is the most used technique for quality assurance in industry. However, in computer science education software testing is still retated as a second-class citizen and students are unable to test thris offware well enough. One reason for this is that teaching the subject of software testing is difficult as it is a complex intellectual activity for which students need to allecate multiple cognitive resources at the same time. A myraid of primary and accordary studies have tried to solve this problem in education, however still with very limited results.	
	Objective: Before we can design interventions to improve our pedagogical approaches, we need to gain more in-depth understanding and recognition of sensemaking as it is happening when students design test cases.	
	Method: An initial exploratory study identified four different sememaking approaches used by students while creating test models. In this paper we present a follow-up study with 50 students from a large university in Spain. The used methodology was based on the previous study with the improvements that originated from its evaluation. We asked the participants to create a test model based on a description of a test problem using a specialized web-based tool for modeling test cases. We measured how well these models if the test problem, the sensemaking process that students went through when creating the models, and the students' perception of the modeling test. The participants received an compassion for third refors, and we scheduled the experiment during a regular class. Appart from the created models and their metadata, we used a questionnaire to study their perspectives on the assignment. All the collected testual, graphical, and video data was analyzed using an iterative inductive analysis process to allow new information about the different sensemaking approaches to emerge.	
	Results: We gained better insights into the sensemaking processes of students while modeling test cases for a problem. The results enabled us to refine our previous findings, and we identified new sensemaking approaches.	
	Conclusions: Based on these results, we can further investigate ways to influence the sense- making process in education, the possible misconceptions that have a negative influence on it, and the desired mental model we want our students to have to desire net cases.	

1. Introduction

As the role of software in our society increases, its quality becomes more important. However, this quality is not always evident. The actual use of software is often different from the expected use, leading to failures that can highly impact the system in which

THE CASE WE USED

Testing the average age calculator

You work for a travel company. The sales department wants to know what the average age is of the people who booked their holidays with your company. One of the developers in your team has developed a program to calculate the average age for a hundred people at the time. The program can handle up to a hundred dates of births and calculates the average age in years. It gets its data from a remote server as a .txt file, where each line contains the name and the age.

Assignment: Design a test model in TestCompass to adequately test the application.





A

RESULTS FROM THE STUDY

Enhanced understanding of student sensemaking processes

Refinement of previous findings

Discovery of new sensemaking practices and approaches

DIFFERENT SENSEMAKING PRACTICES AND APPROACHES

Identified practice	Number of students
Lazy student practice	2
Clueless student practice	3
Convergent Tester approach	10
Developer approach	26
Divergent Tester approach	5

DIFFERENT SENSEMAKING PRACTICES AND APPROACHES

Identified practice	Number of students
Lazy student practice	2
Clueless student practice	3
Convergent Tester approach	10
Developer approach	26
Divergent Tester approach	5

DEVELOPER APPROACH



DIFFERENT SENSEMAKING PRACTICES AND APPROACHES

Identified practice	Number of students
Lazy student practice	2
Clueless student practice	3
Convergent Tester approach	10
Developer approach	26
Divergent Tester approach	5







NEXT STEPS

NEXT STEP IN MY RESEARCH: COMPARING TEST CASES, STUDENTS VS. EXPERTS

Now we now the approaches of students, we want to know the approaches of experts.

Hypothesis: experts rely heavy on heuristics and are more aware of biases and misconceptions

CREATE INNOVATIVE INSTRUCTIONAL DESIGNS

Serious games such as Testsphere with the mechanics of Fluxx

- Identification of test cases
- creating awareness of biases
- Identifying misconceptions



More holistic: we need to shift our teaching strategies

testing like an engineer solves a problem

diagnoses a patient

experimentation

focus on problem solving

θ

from the rational paradigm

> based on requirements and specifications

to the empirical paradigm

looking for the unknown unknowns



CONCLUDING THOUGHTS



₿

Read more about my research at _. research.nielsdoorn.nl





Questions

