MINDMAP & TEXT SUMMARY ON
TESTINGEDUCATION.ORG - BUG ADVOCACY
COURSE BY: CEM KANER & JAMES BACH

Bug Advocacy

RAHUL PARWAL & KHUSHBOO RAJPUROHIT

FOREWORD BY JAMES MARCUS BACH

This ebook is dedicated to all the passionate learners out there who believe in the power and compounding effects of learning.

Acknowledgement

This book is published based on the material on Bug Advocacy available at testingeducation.org which is under the restrictions of the Creative Commons Attribution - Share Alike License.

We would like to explicitly acknowledge the authors and copyright holders, i.e. Dr. Cem Kaner and James Marcus Bach for the remarkable work that they have done and made publicly available for study, reference, and self-learning.

We first came to know about the term, **Bug Advocacy** after reading the well-acclaimed testing book, Lessons Learned in Software Testing by Dr. Cem Kaner, James Marcus Bach, and Bret Pettichord. Bug Advocacy seemed like the ultimate sword for testing professionals to use in their day-to-day work. We started with applying basic bug advocacy skills to our day-to-day work. Our hunger to master this skill began to increase within a short time after we began to see the positive effects of this practice. The crash course on bug reporting from Pradeep Soundararajan and Santhosh Tuppad helped us too.

Eventually, we started with the Testing Education Course Material on Bug Advocacy by Dr. Cem Kaner and James Marcus Bach, available here: Bug Advocacy (testingeducation.org). Although this course looks very small to start with as the duration is only around 2.4 Hours (141 mins). However, we both had to spend around a month's time to consume, grasp, and process the plethora of knowledge that is shared in this course.

We used to have weekly sync-up calls to brainstorm on various areas covered in this course. We realized that each chapter is filled with so much information and would require a lot of notes, processing, & challenge to our existing thought process on testing.

Finally, we were able to conclude this course and pen down our reflections via mindmap & textual notes.

This summary e-book is an organized and refined output of our reflection from this course and our current understanding of testing.

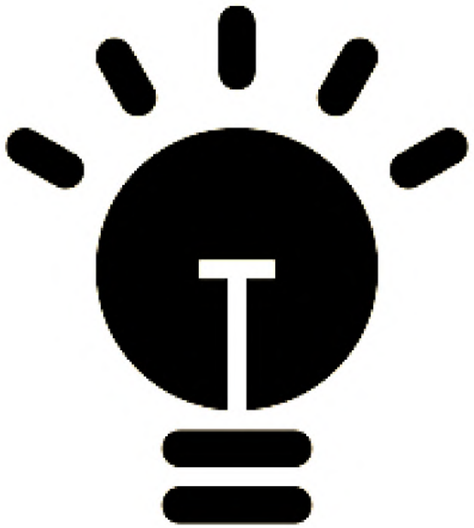
We are thankful to each of our reviewers for helping us with their feedback on mind maps/summaries. This e-book is useful for anyone who wants to understand, revise, study, or learn about Bug Advocacy and its foundational concepts.

Happy Reading! Happy Learning!

NOTE: We strongly recommend people interested in professional training on Bug Advocacy to check out professional versions of these courses offered by the Association for Software Testing and Altom's BBST Training Courses. For self-paced learners like us, who would like to study this topic in-depth, we recommend the learning material available at TestingEducation.org.

RAHUL PARWAL
STUDENT OF SOFTWARE TESTING
TESTING TITBITS

KHUSHBOO RAJPUROHIT
STUDENT OF SOFTWARE TESTING
LINKEDIN



Foreword

I must disclose this: I don't believe the job of a tester is to improve the product.

Improving the product is a whole different skill set than testing. Of course, you can be a developer as well as a tester. You can do lots of things in addition to being a tester. But if we restrict ourselves to talking about testing, THAT is about discovering the truth about the status of the product. I cannot, as a TESTER, take responsibility for the quality of the product, since, as a TESTER, I have absolutely no control over it.

So, it might appear that I disagree with Cem Kaner when he says that the best tester is the one who gets bugs fixed. But, not really. Not in any big way. This is because the work of a tester is worthless if you cannot convey what you have discovered to your clients in a meaningful and compelling way.

While I don't control or assure quality, I feed the minds of the people who do. I act in my clients' best interests to make clear to them the business risks that they face if they were to release the product under test. This aspect of testing is a bit of a gray area, which does reach into business analysis, development, and design. At least a little bit. That's why this class and this booklet can help you be a more effective and relevant tester. Testing is a social process. We engage in it as people helping people.

Don't allow any automation or fancy technique to distract you from this basic truth.

Happy Reading!

James Bach

Creator of Rapid Software Testing Methodology

www.satisfice.com

Introduction

NOTE:

This e-book consists of the topics given below:

- Scope of Bug Reporting (what to report as bugs & what information to include).
- Bug Reporting as persuasive writing & the power of follow-up testing.
- Getting past reasons and excuses for deferring bugs.
- Learning from the Psychology of decision making & writing clear bug reports.

How to read mind maps:

- Start at 12 o'clock & go clockwise.
- Colors and icons have been added to the mind maps to give strength to the summary & make it easier to read
- Different colored lines have been used to separate different areas of the mind map.
- Symbols have been used to add extra strength to the association & they can have the meaning of their own (sometimes not).

Index

LECTURE 1

Basic Concepts – Bug Advocacy

Basic Concepts

Quality

- Conformance with requirements - Philip Crosby
- Quality is Fitness for Use! - Joseph Juran
  - Involves
    - Customer Satisfiers
      - Right Features
      - Adequate Instructions
    - Dissatisfiers
      - Unreliable
      - Hard to use
      - Too Slow
      - Incompatible
- Quality is value to some person - Jerry Weinberg
- Multidimensional Concept Includes following attributes:
  - Speed
  - Scalability
  - Localizability
  - Documentability
  - Trainability
  - Supportability
  - Reliability
  - Usability
  - Maintainability
  - Testability
  - Saleability
  - Functionality
- Different People, Different Notions of Quality
  - Managers
  - Client Service
  - Tech Writers
  - Programmers
  - Marketing
  - Testers

Bug Advocacy

- *Best tester ain't one who finds most bugs, but the one who gets the MOST bugs fixed.*

Bug Workflow

- Someone reports a bug
- Someone replicates the bug
- To Confirm
  - It is a real problem
  - The steps are correct
- Programmer looks into it and, Either
  - Involves Testers
    - Checks
      - Did the programmer fix the bug?
        - Just this bug or similar bugs too?
        - Reported symptoms or underlying problem?
      - Did the fix cause other side effects?
    - To Prioritize the unfixed Bugs
    - Reassign them to someone else
    - Fixes It
  - Involves Project Manager
    - Involves CCB or Triage Team
      - Makes "Final" Decision
      - Defers or Accepts Bugs
    - Decides NOT to fix them
    - Reject the Bug (as Feature)
    - Ask for more information
    - Waits for approval to fix it
    - Defer it/Move to backlog
    - Argues that it is not a bug

Cem Kaner, testingeducation.org

Bugs

- Anything that causes an unnecessary or unreasonable reduction of the quality of a software product.
- Bug is anything about the product that threatens it's value. - James Bach & Michael Bolton
- Other related terms
  - Fault — Mistake in the Code
  - Failure — Misbehavior caused by the fault
  - Error — Something wrong with the product
  - Critical Condition — Data value or environment condition or step to reveal the problem
  - Defect — Legal Implications - Used in Lawsuits
  - Symptom — Behavior that suggests underlying problem

Bug Reports

- Tester's primary work product
  - Seen by people outside testing group
  - People notice and remember it!
  - Decides quality of communication
- Tool to sell the programmer to spend their
  - Time
  - Energy

Challenges in Bug Reporting

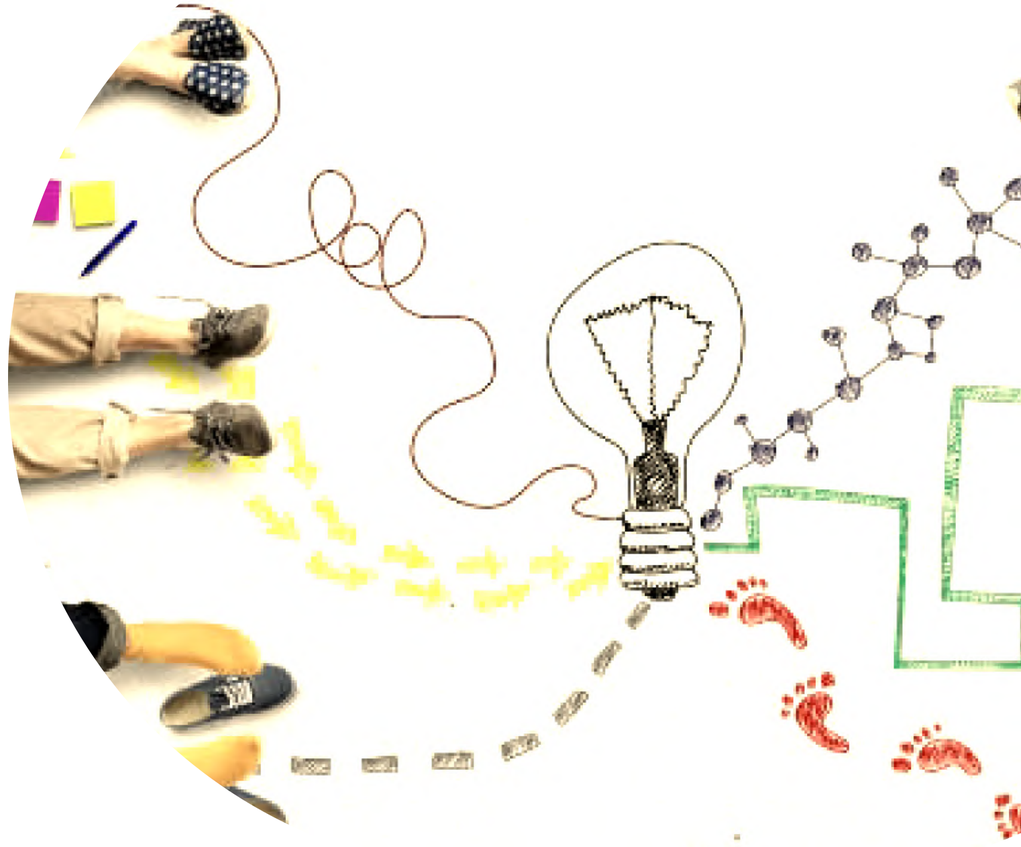
- How to balance time?
  - Conflict of Interests!
  - Improving our understanding and communication of our findings
  - V/S
  - Finding new bugs!
- Deal with conflicts of interest among stakeholders?
- Present problems to people under stress?
- Preserve our
  - Credibility
  - Integrity

THE SOLE PURPOSE OF TESTING

If you were a salesperson, what would be the purpose of testing for you?

The sole purpose of testing would be to discover information about the product under test so that the stakeholders can make well-informed decisions to retain the existing customers or increase new customers and sales.
Note that the above purpose has been written with the perception of a salesperson.

Think again and you will notice the purpose of testing varying from person to person in the team. Stakeholders for example will say that the purpose of testing is to make sure their requirements for the product are met. The marketing person will say that it is indeed to give the best possible user experience and also to retain the customers.



THE SOLE PURPOSE OF TESTING

Few thoughts on the Purpose of testing:

- Information we provide can get affected by our preference and level of skepticism.
- Did you notice how our role can make a difference in the growth of the company?
- We mostly make bug-related decisions under **uncertainty**.

"Most decisions in engineering and life are based on uncertainty. Most companies have an entrepreneurial mindset. They thrive on risk! What we as testers can do here is to minimize this uncertainty by exploring more information, which in turn helps them to make well-informed decisions."

1

Balancing time between communicating what we found to the team vs looking for more problems

For instance, a tester is exploring an MES (Manufacturing Execution Software), specifically exploring the functionality of a shop floor technician who is responsible for the execution of a manufacturing order. While collecting data of a part or tool, a tester notices an empty value in a dropdown. After investigating with the developer, a '#' in the field name caused the issue. That field name was used in the query API & hence an empty dropdown!

In this situation, it becomes difficult for the tester to discover more cases like this or try with other special characters. Should they use oracles and heuristics to find a bigger problem or to keep exploring the shop floor technician operation execution using the existing charters?

Feels like a tug-of-war!

"Testers must consider tradeoffs here. Evaluate risks and discuss with the decision-makers, where to invest time right now!"

2 Dealing with the conflicts among decision-makers or stakeholders.

The decision-makers might have different preferences and user personas in mind, and they might use that to provide clarification about the information (let's say, "bug") that the tester discovered.

Keeping that in mind, wouldn’t it be better if the tester asks questions, understand preferences to the core (try to understand what they really mean) and present the information such that the Product team gets what they want just by looking at the bug report? And if it turns out to be a feature, you can avoid reporting a false bug.

3 Presenting the problem to people under stress.

In stressful situations, if the developer or manager does not understand the bug report they are more likely to reject the bug.
If you have done your research well, then add your observation in a way that describes the possible risks ‘clearly’.