# FOR RISKS IN TEST SESSIONS USING HEURISTICS

TESTINGMIND

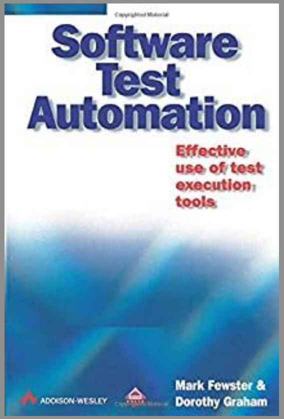
TEST AUTOMATION AND DIGITAL QA SUMMIT

#TAS21

USA Season 5
AUGUST 12 – 13, 19 – 20 & 26 – 27, 2021

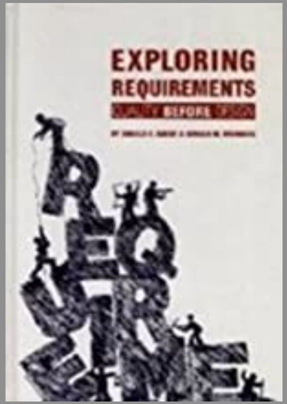
Paul Lyles, CSM
https://www.linkedin.com/in/softwarequalitychampion/

MERCURY INTERACTIVE

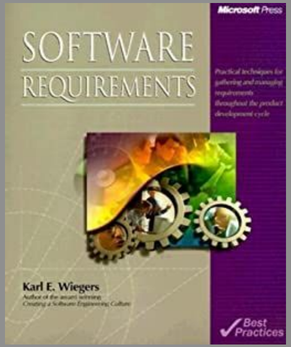

Software Test Automation

Effective use of test execution tools

Mark Fewster & Dorothy Graham


EXPLORING REQUIREMENTS

QUALITY BEFORE DESIGN


SOFTWARE REQUIREMENTS

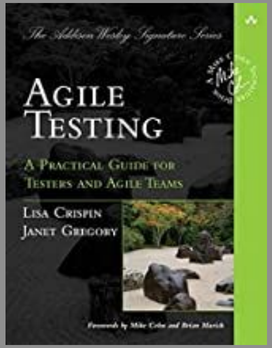
Karl E. Wiegers

QUALITY ASSURANCE INSTITUTE WORLDWIDE

CSTE

Certified Software Tester

Act
Plan
Study
Do


The Addison-Wesley Signature Series

AGILE TESTING

A PRACTICAL GUIDE FOR TESTERS AND AGILE TEAMS

LISA CRISPIN

JANET GREGORY

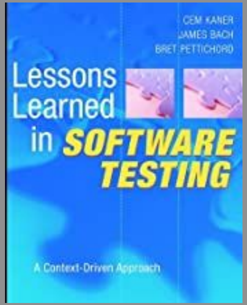
Agile LEADERSHIP NETWORK HOUSTON

Scrum Alliance CSM CERTIFIED

- Factory (metrics)
- Control (process)
  - TQM
  - IEEE
  - ISO
  - CMMI
- Agile Testing (story)
- Context-Driven (adapt)

AST ASSOCIATION FOR SOFTWARE TESTING

MINISTRY OF TESTING

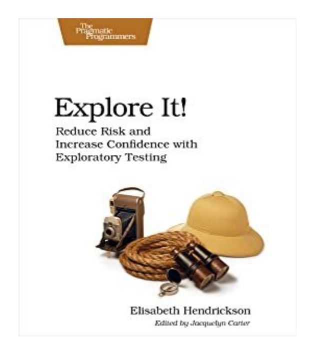

CEM KANER
JAMES BACH
BRET PETTICHORD

Lessons Learned in SOFTWARE TESTING

A Context-Driven Approach

RAPID SOFTWARE TESTING RST A CONTEXT-DRIVEN METHODOLOGY

BBST

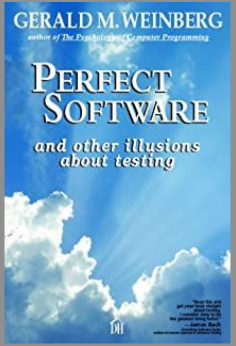

The Pragmatic Programmers

Explore It!

Reduce Risk and Increase Confidence with Exploratory Testing

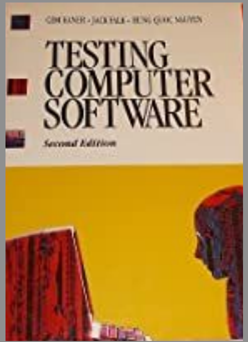
Elisabeth Hendrickson

Edited by Jacquelyn Carter


GERALD M. WEINBERG

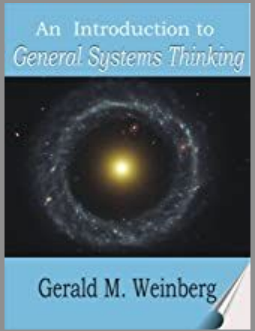
PERFECT SOFTWARE

and other illusions about testing


CEM KANER · JACK FALK · HUNG QUOC NGUYEN

TESTING COMPUTER SOFTWARE

Second Edition


An Introduction to General Systems Thinking

Gerald M. Weinberg

TACIT & EXPLICIT KNOWLEDGE

HARRY COLLINS

# CONTEXT-DRIVEN SCHOOL

of software testing

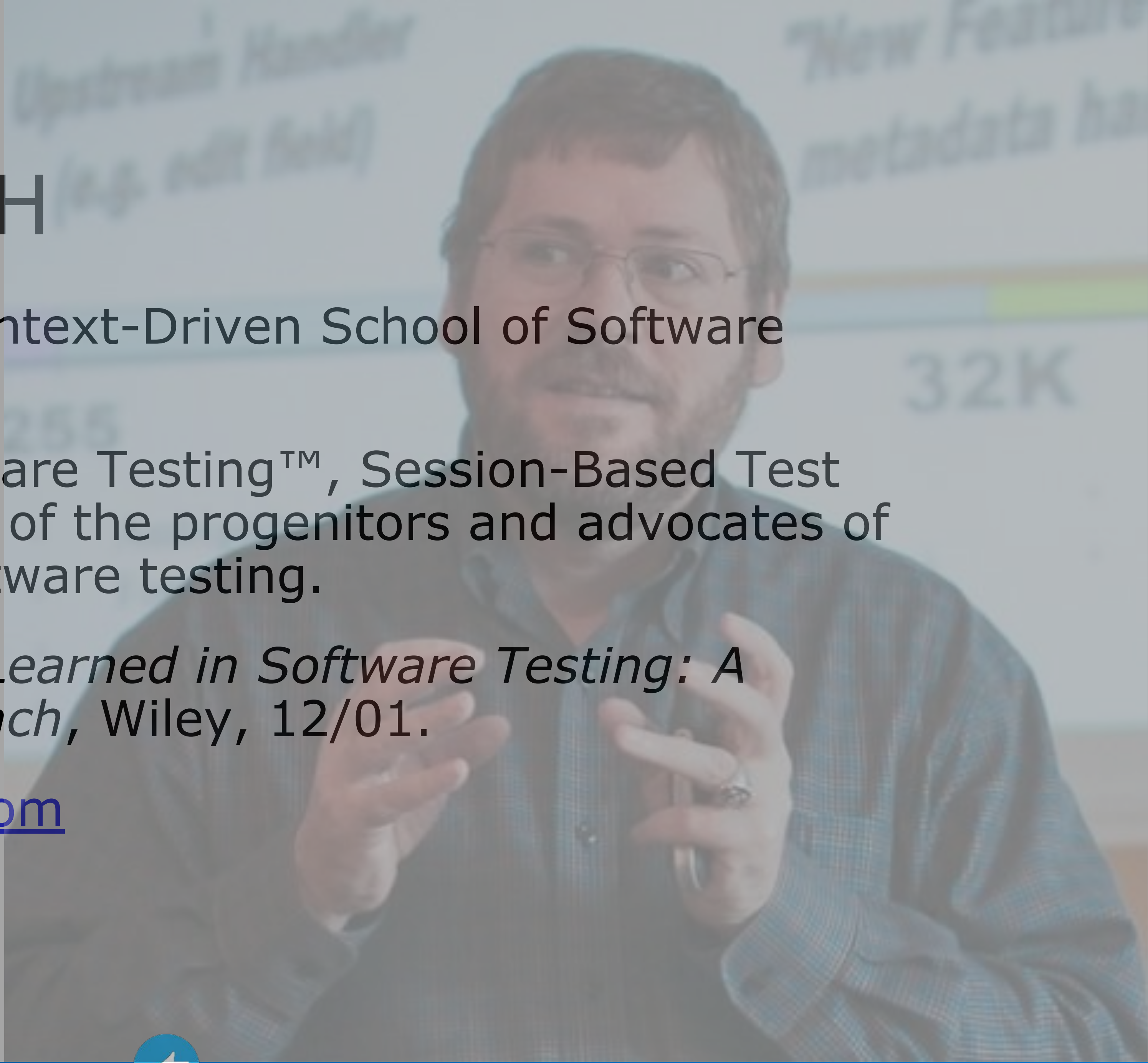
# CONTEXT-DRIVEN PRINCIPLES

1. Value of any practice depends on its context
2. Good practices in context (no best practices)
3. People, working together (most important part of context)
4. Projects unfold over time (often unpredictably)
5. Product is a solution (if unsolved, then non-working product)
6. Good software testing is a challenging intellectual process
7. Judgment and skill are required to effectively test products
    - exercised cooperatively
    - throughout the entire project
    - enables doing the right things at the right times

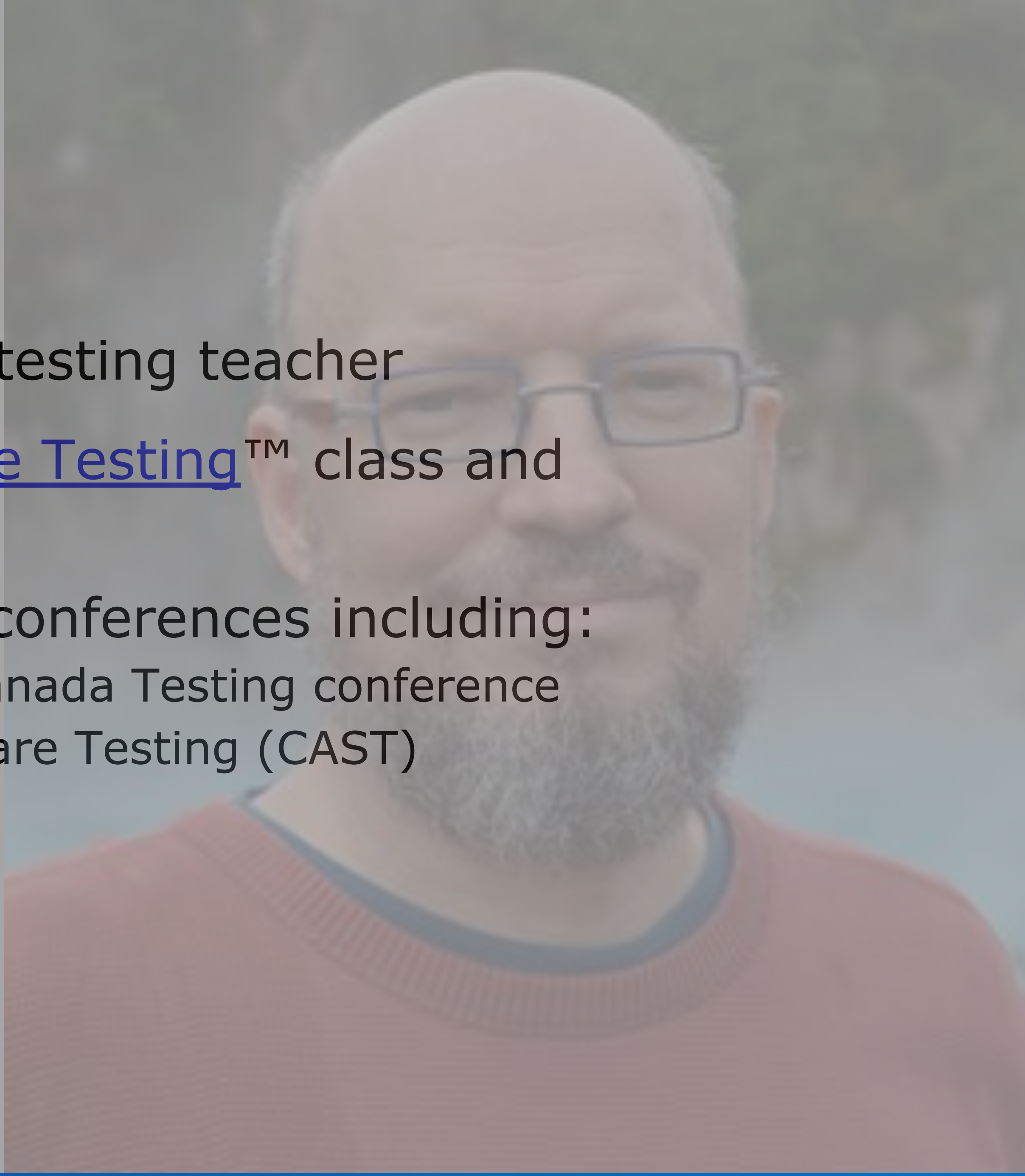
https://context-driven-testing.com

# JAMES M. BACH

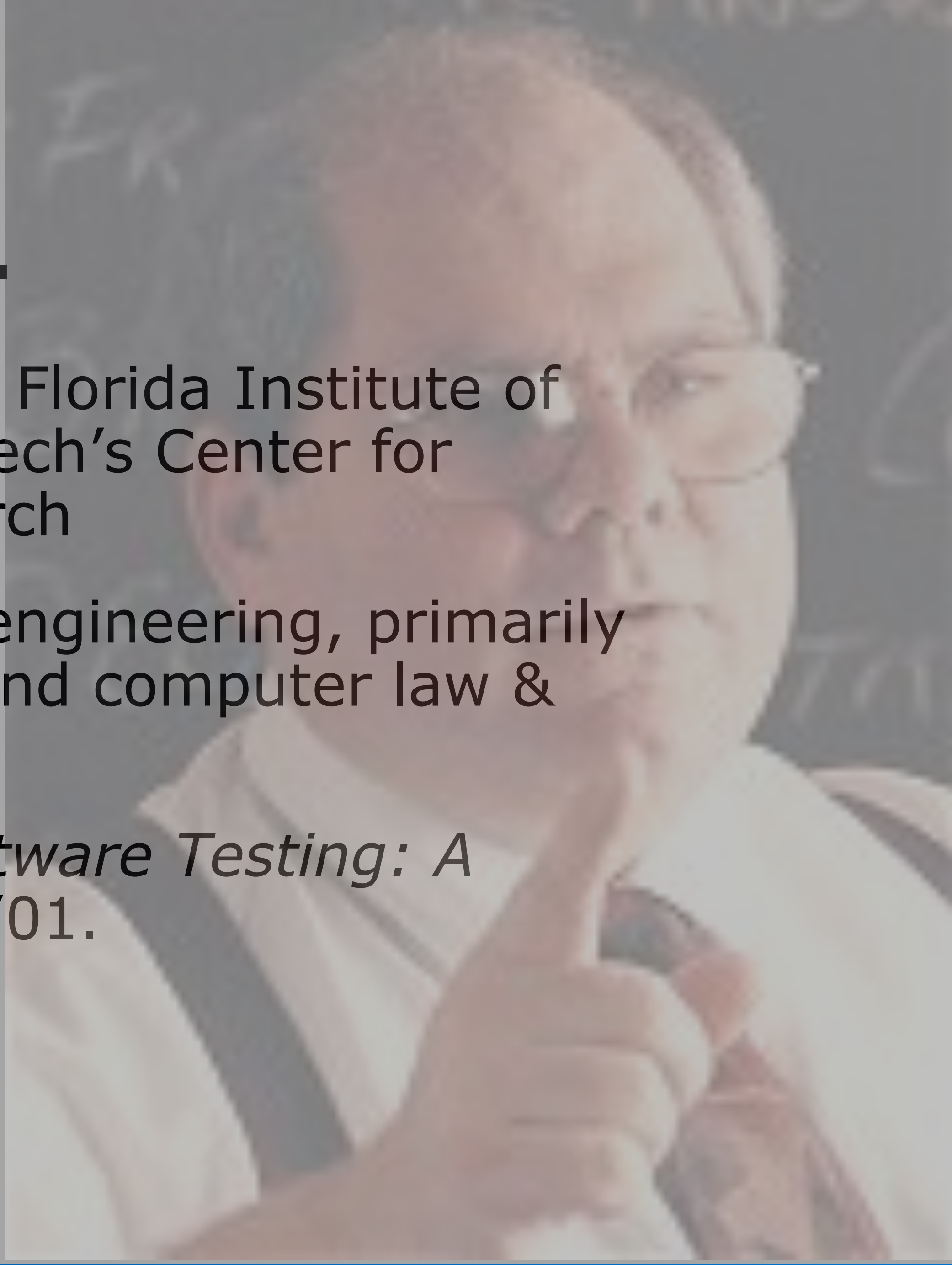
- Founding member: Context-Driven School of Software Testing
- Creator of Rapid Software Testing™, Session-Based Test Management, and one of the progenitors and advocates of skilled exploratory software testing.
- Co-author of *Lessons Learned in Software Testing: A Context-Driven Approach*, Wiley, 12/01.
- [http://www.satisfice.com](http://www.satisfice.com)



# MICHAEL BOLTON



- Consulting software tester and testing teacher
- Co-author of the Rapid Software Testing™ class and methodology
- Conference speaker at various conferences including:
  - STAR East, STAR West and STAR Canada Testing conference
  - Conference for Association of Software Testing (CAST)
  - TestBash
  - EuroStar
  - Quality Jam
- http://www.developsense.com



# CEM KANER, J.D, PH.D.

- Professor of Software Engineering at Florida Institute of Technology and Director of Florida Tech's Center for Software Testing Education & Research
- Teacher and researcher in software engineering, primarily software testing, software metrics, and computer law & ethics.
- Co-author of *Lessons Learned in Software Testing: A Context-Driven Approach*, Wiley, 12/01.
- http://kaner.com

# QUALITY

value

to some person

who matters

Gerald M. Weinberg

James Marcus Bach

# THREATS

product quality

on-time delivery

# EXPOSE VALUE THREATS

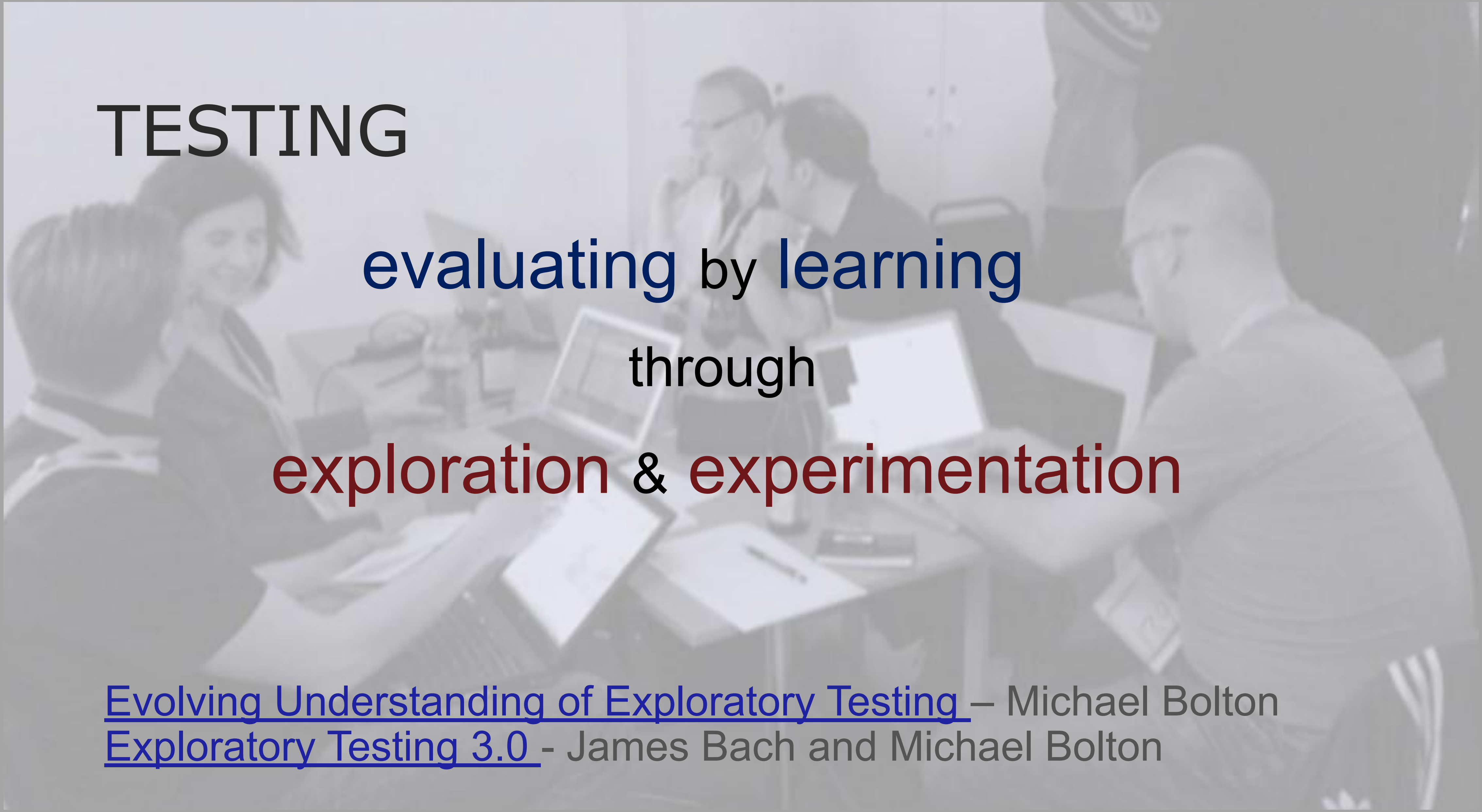
identify

prioritize

explore

James Bach on Risk-Based Testing

Risk-based Testing with James Bach



# TESTING

evaluating by learning

through

exploration & experimentation

Evolving Understanding of Exploratory Testing – Michael Bolton
Exploratory Testing 3.0 - James Bach and Michael Bolton

Suspected risk

Unknown unknown risk

Known high risk

Shallow testing

Known low risk

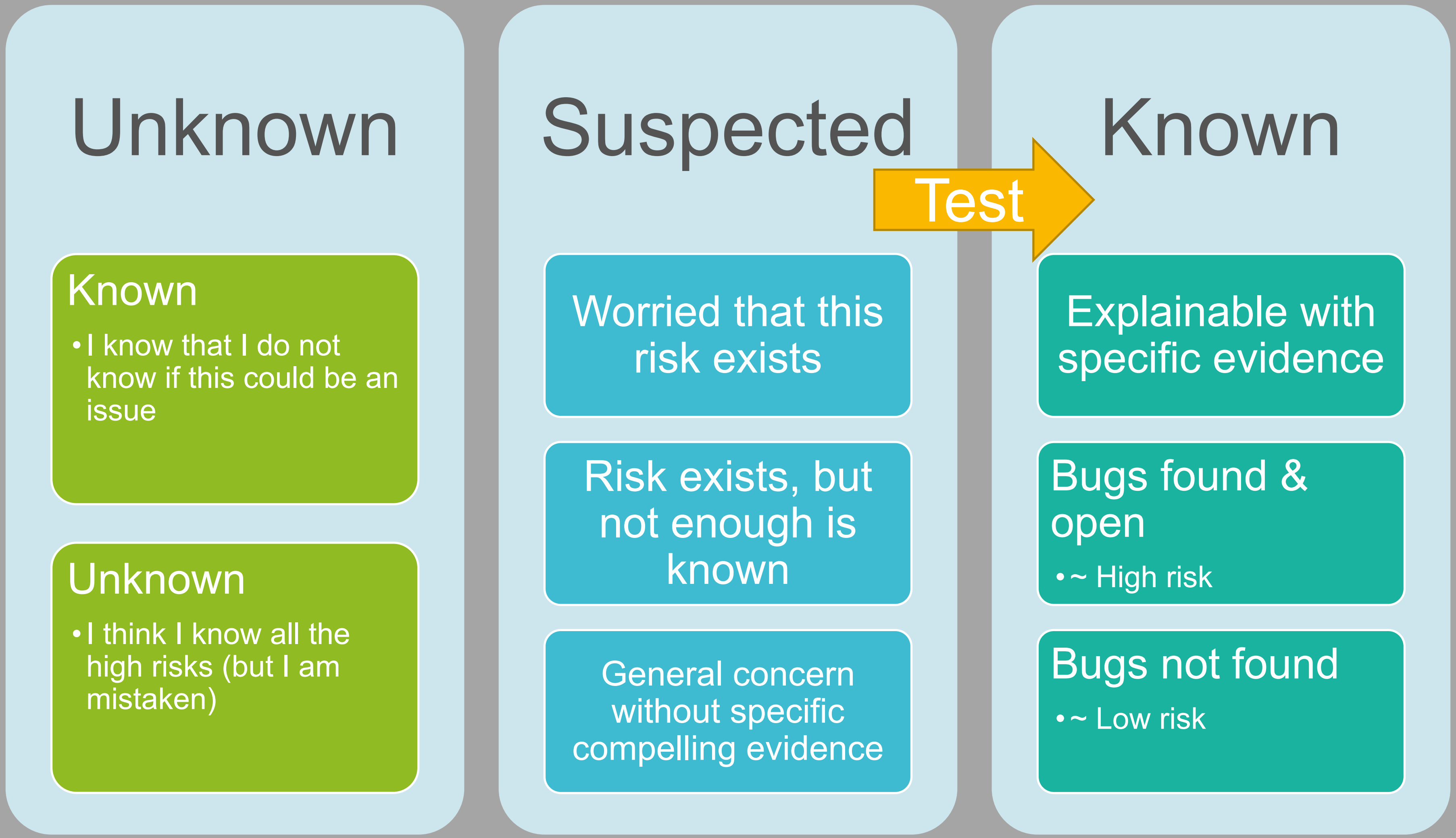
Deep testing

Suspected risk

Unknown risk

# RISK AWARENESS

James Bach on Risk-Based Testing



## Unknown

**Known**

- I know that I do not know if this could be an issue

**Unknown**

- I think I know all the high risks (but I am mistaken)

## Suspected

Worried that this risk exists

Risk exists, but not enough is known

General concern without specific compelling evidence

Test

## Known

Explainable with specific evidence

**Bugs found & open**

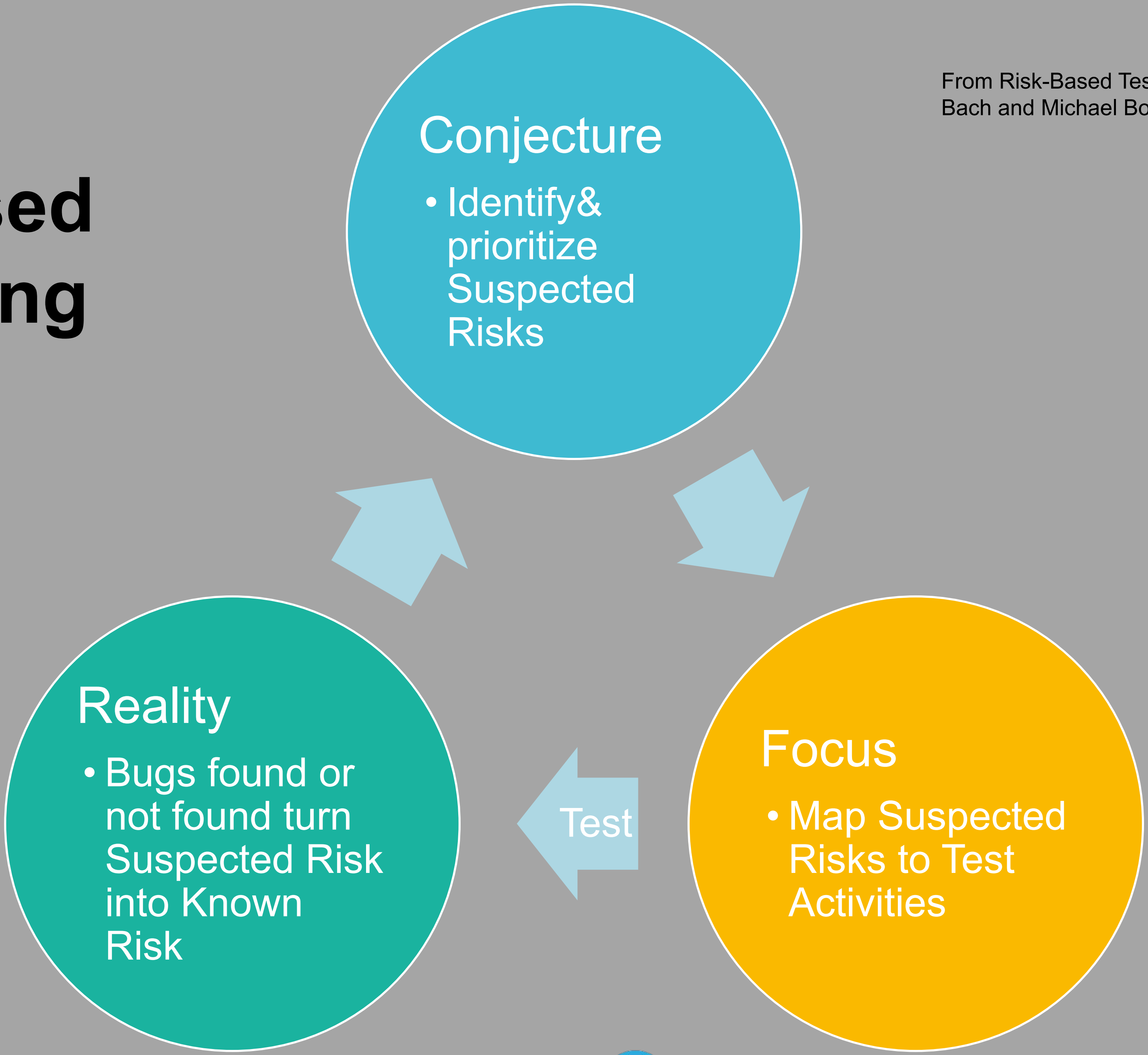
- ~ High risk

**Bugs not found**

- ~ Low risk

# Risk –Based Testing

From Risk-Based Testing by James Bach and Michael Bolton



Conjecture

- Identify& prioritize Suspected Risks

Focus

- Map Suspected Risks to Test Activities

Reality

- Bugs found or not found turn Suspected Risk into Known Risk

# WHEN NOT MUCH IS KNOWN

James Bach on Risk-Based Testing
Risk-based Testing with James Bach

- Vague sense of risk
- Learn all you can about the product
- Survey testing
- Interview people
- Get context and background
- Use heuristics for general systems

# HEURISTIC

fallible method
guideline
serving to discover

**Oracle** – principle or mechanism used to recognize a problem

Heuristics for Understanding Heuristics – Michael Bolton

# GENERAL SYSTEMS HEURISTICS

James Bach on Risk-Based Testing

- **Cardinality** – Can there be 0, 1, or more than one object?
- **Boundaries** – Is there one limit? More? Are they consistent?
- **Extrapolation** – If this far, can we go farther?
- **Interpolation** – What is between two things in separate places?
- **Intersections** – Do components collide? Can one contaminate?
- **Surface integrity** – Does behavior change as input changes?
- **Symmetry/Asymmetry** – If a behavior exists for A, does a corresponding behavior exist for B?
- **Pattern Completion** – Is a pattern apparent that has not yet been completed, or is obscured?
- **Negation** – Whatever is there might disappear or reverse.

# ORACLE HEURISTICS

- Consistency Oracles: FEW HICCUPPS
- Exploratory Skills and Dynamics
- Heuristic Test Strategy Model
- Touring Heuristic: FCC CUTS VIDS
- General Functionality and Stability Test Procedure
- Regression Testing Heuristic: RCRCRC
- Test Heuristics Cheat Sheet

# CONSISTENCY ORACLES
# FEW HICCUPPS

- **F**amiliarity
- **E**xplainability
- **W**orld

- **H**istory
- **I**mage
- **C**omparable Products
- **C**laims
- **U**ser Desires
- **P**urpose
- **P**roduct
- **S**tatutes and Standards

FEW HICCUPPS – Michael Bolton

# TEST STRATEGY (CHOOSING TESTS) HEURISTICS

## PROJECT ENVIRONMENT

- Mission
- Information
- Developer relations
- Test Team
- Equipment & Tools
- Schedule
- Test Items
- Deliverables

## PRODUCT ELEMENTS SFDIPOT

- **S**tructure
- **F**unctions
- **D**ata
- **I**nterfaces
- **P**latform
- **O**perations
- **T**ime

Heuristic Test Strategy Model – James Bach

# TEST STRATEGY HEURISTICS (CHOOSING TESTS)

## QUALITY CRITERIA CATEGORIES

- Capability
- Reliability
- Usability & Accessibility
- Charisma
- Security
- Scalability
- Compatibility
- Performance

## TEST TECHNIQUES FDSFSCURA

- **F**unction Testing
- **D**omain Testing
- **S**tress Testing
- **F**low Testing
- **S**cenario Testing
- **C**laims Testing
- **U**ser Testing
- **R**isk Testing

Heuristic Test Strategy Model – James Bach

FDSFSCURA - Michael Bolton

# QUICK TESTS

- Product Tours
- Happy Path
- Interruptions
- Undermining
- Adjustments
- Dog Piling
- Continuous Use
- Feature Interactions
- Click on Help
- Input Constraint Attack

- Click Frenzy
- Shoe Test
- Blink Test
- Error Message Hangover
- Resource Starvation
- Multiple Instances
- Crazy Configs
- Cheap Tools

What Exploratory Testing is Not – Part 4 – Quick Tests
Exploratory Testing 09-2009 – Michael Bolton

# QUICK TESTS: PRODUCT TOURS

## FCC CUTS VIDS

- **F**eature Tour
- **C**omplexity Tour
- **C**laims Tour
- **C**onfiguration Tour
- **U**ser Tour
- **T**estability Tour
- **S**cenario Tour
- **V**ariability Tour
- **I**nteroperability Tour
- **D**ata Tour

Of Testing Tours and Dashboards

Touring Heuristic

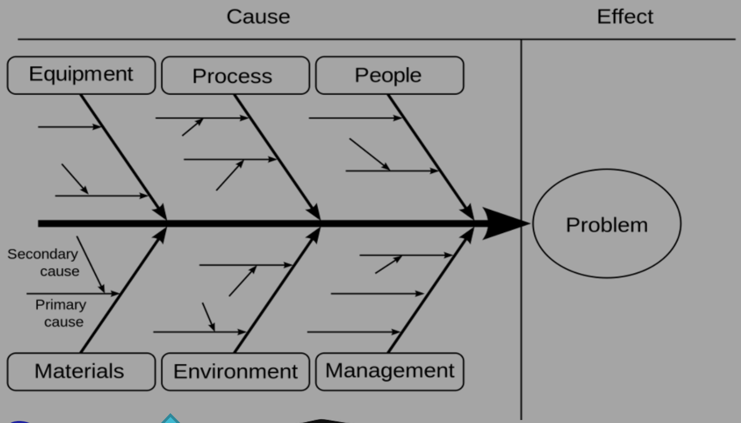
## OTHER QUICK TESTS

- File Tour
- Menus and Windows Tour
- Keyboard and Mouse Tour
- Documentation Tour
- Sample Data Tour
- Variability Tour
- Complexity Tour
- Continuous Use / All-Nighter Tour
- Money Tour (like User Tour)

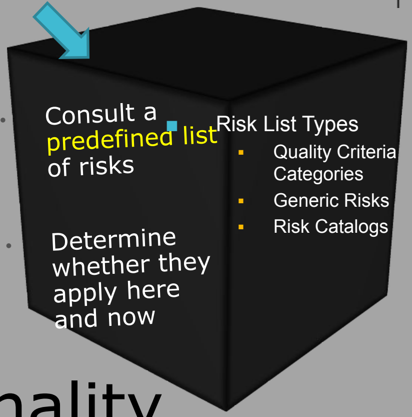
# WHEN MUCH IS KNOWN

[James Bach on Risk-Based Testing](#)
[Risk-based Testing with James Bach](#)

- Analyze problems
- Brainstorm risks



- Black-box analysis
  - Use [quality criteria categories](#)
  - Generic risk list
  - Risk catalogs

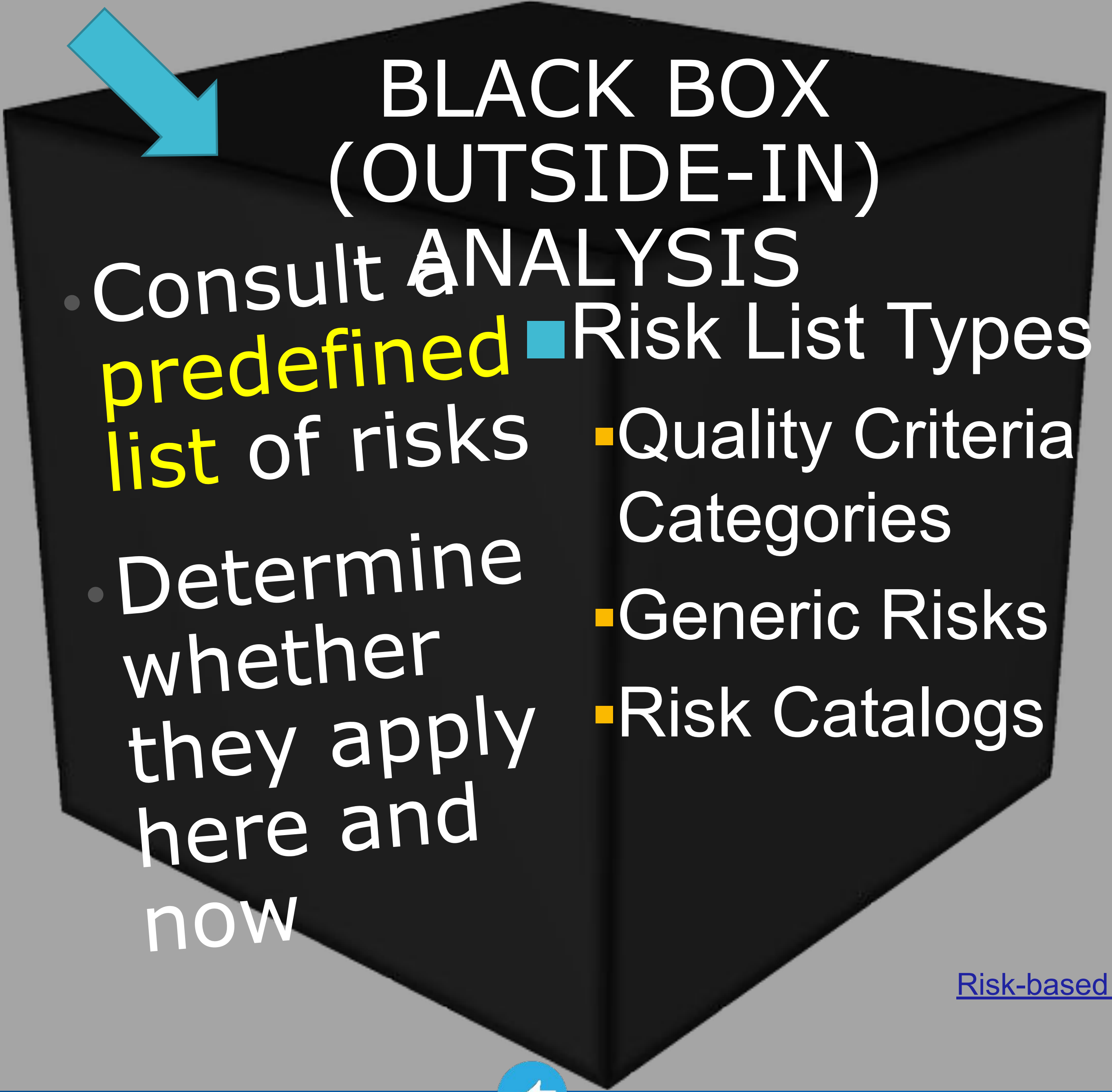


Consult a predefined list of risks

Determine whether they apply here and now

Risk List Types
- Quality Criteria Categories
- Generic Risks
- Risk Catalogs

- White-box analysis
  - Use knowledge of code / functionality
  - 4-part risk story



# BLACK BOX (OUTSIDE-IN) ANALYSIS

- Consult a predefined list of risks
- Determine whether they apply here and now

- Risk List Types
  - Quality Criteria Categories
  - Generic Risks
  - Risk Catalogs

Risk-based Testing by James Bach

# QUALITY CRITERIA CATEGORIES

- Capability
- Reliability
  - Robustness
  - Error handling
  - Data Integrity
  - Safety
- Usability
  - Learnability
  - Operability
  - Accessibility

- Charisma
  - Aesthetics
  - Uniqueness
  - Necessity
  - Usefulness
  - Entrancement
  - Image
- Security
  - Authentication
  - Authorization
  - Privacy
  - Security holes

Heuristic Test Strategy Model by James Bach

# QUALITY CRITERIA CATEGORIES

- Scalability
- Compatibility
  - Application Compatibility
  - Operating System Compatibility
  - Hardware Compatibility
  - Backward Compatibility
  - Resource Usage
- Performance

- Installability
  - System requirements
  - Configuration
  - Uninstallation
  - Upgrades/patches
  - Administration
- Development
  - Supportability
  - Testability
  - Maintainability
  - Portability
  - Localizability

Heuristic Test Strategy Model by James Bach

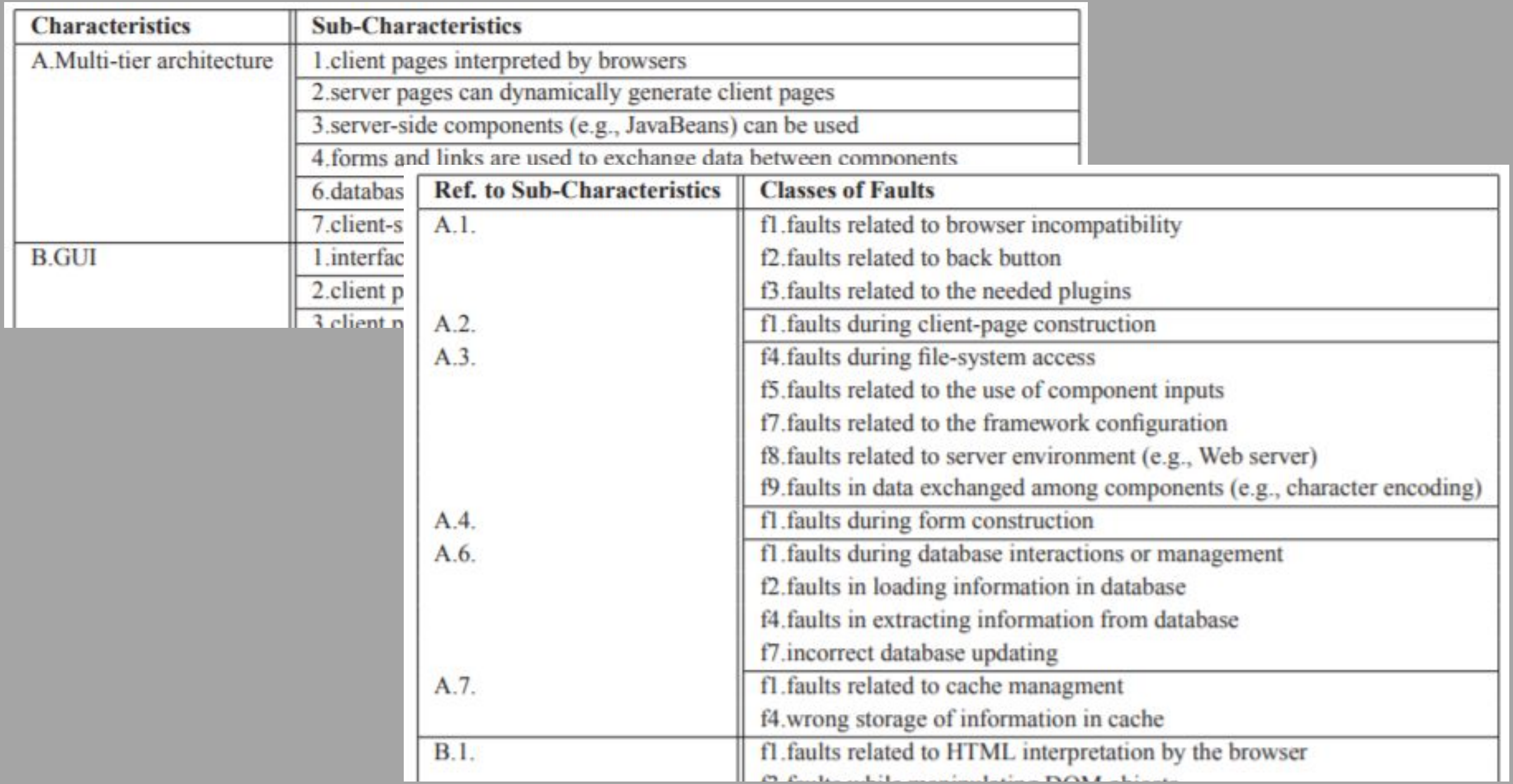
# GENERIC RISKS

- Complex
- New
- Changed
- Upstream dependency
- Downstream dependency
- Critical
- Precise
- Popular
- Strategic
- Third party
- Distributed
- Buggy
- Recent failure

Heuristic Test Strategy Model by James Bach

# RISK CATALOG

Marchetto, Alessandro & Ricca, Filippo & Tonella, Paolo. (2009). An Empirical Validation of a Web Fault Taxonomy and its Usage for Web Testing.. J. Web Eng.. 8. 316-345.



| Characteristics | Sub-Characteristics |
|---|---|
| A.Multi-tier architecture | 1.client pages interpreted by browsers |
| | 2.server pages can dynamically generate client pages |
| | 3.server-side components (e.g., JavaBeans) can be used |
| | 4.forms and links are used to exchange data between components |
| | 6.databas |
| | 7.client-s |
| B.GUI | 1.interfac |
| | 2.client p |
| | 3.client p |

| Ref. to Sub-Characteristics | Classes of Faults |
|---|---|
| A.1. | f1.faults related to browser incompatibility<br>f2.faults related to back button<br>f3.faults related to the needed plugins |
| A.2. | f1.faults during client-page construction |
| A.3. | f4.faults during file-system access<br>f5.faults related to the use of component inputs<br>f7.faults related to the framework configuration<br>f8.faults related to server environment (e.g., Web server)<br>f9.faults in data exchanged among components (e.g., character encoding) |
| A.4. | f1.faults during form construction |
| A.6. | f1.faults during database interactions or management<br>f2.faults in loading information in database<br>f4.faults in extracting information from database<br>f7.incorrect database updating |
| A.7. | f1.faults related to cache managment<br>f4.wrong storage of information in cache |
| B.1. | f1.faults related to HTML interpretation by the browser<br>f2.faults while manipulating DOM objects |

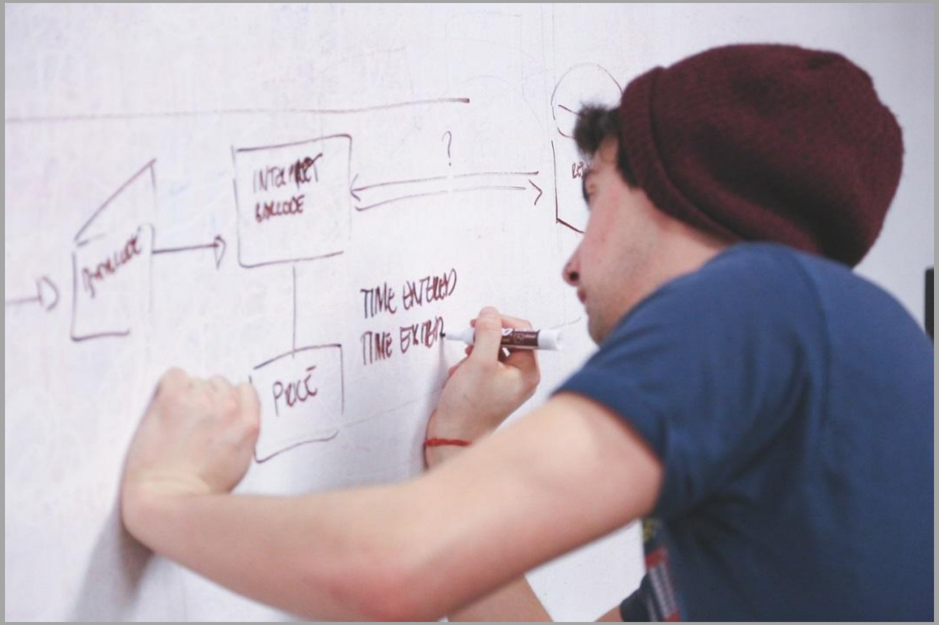
# USING RISK LISTS

1. Decide on component to analyze
2. Determine a meaningful scale of concern
3. Gather information about it
4. Visit each risk area on each list
   - determine its importance in the current situation
   - note impressions and reasons
5. Record any new risks
6. Record unknowns that impact your ability to analyze
   - mark for further research
7. Recheck distribution of concerns

Heuristic Test Strategy Model by James Bach

# WHITE BOX (INSIDE-OUT) ANALYSIS

- How does this feature work?
- What if this component fails?
- What are the
  - Vulnerabilities
  - Threats
  - Victims



Heuristic Test Strategy Model by James Bach

# 4-PART RISK STORY

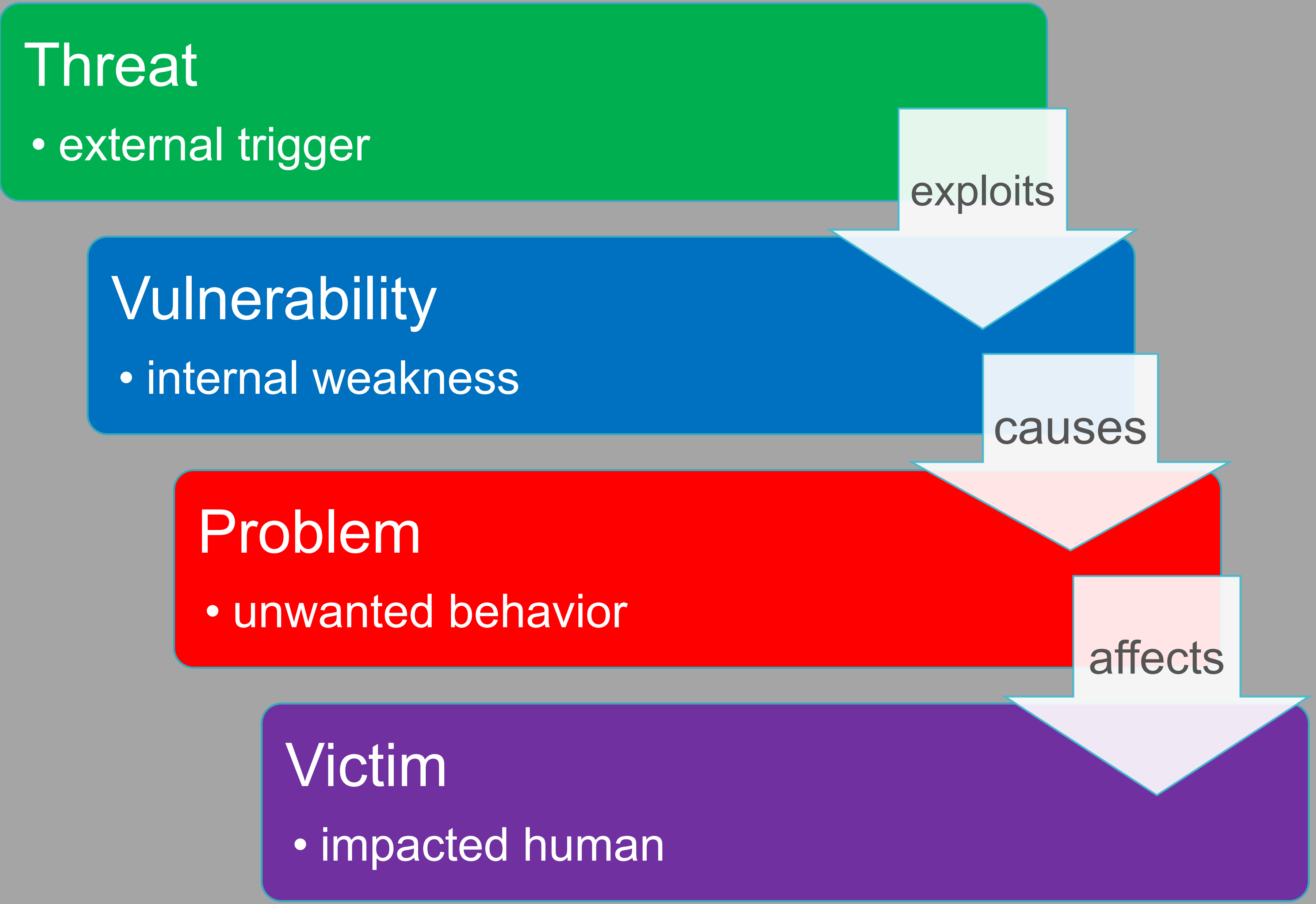
"Someone may be hurt or annoyed because of

something that might go wrong while operating the product,

due to some vulnerability in the product

that is exploited by some threat."

Risk-based Testing with James Bach - Satisfice.com



Threat

- external trigger

exploits

Vulnerability

- internal weakness

causes

Problem

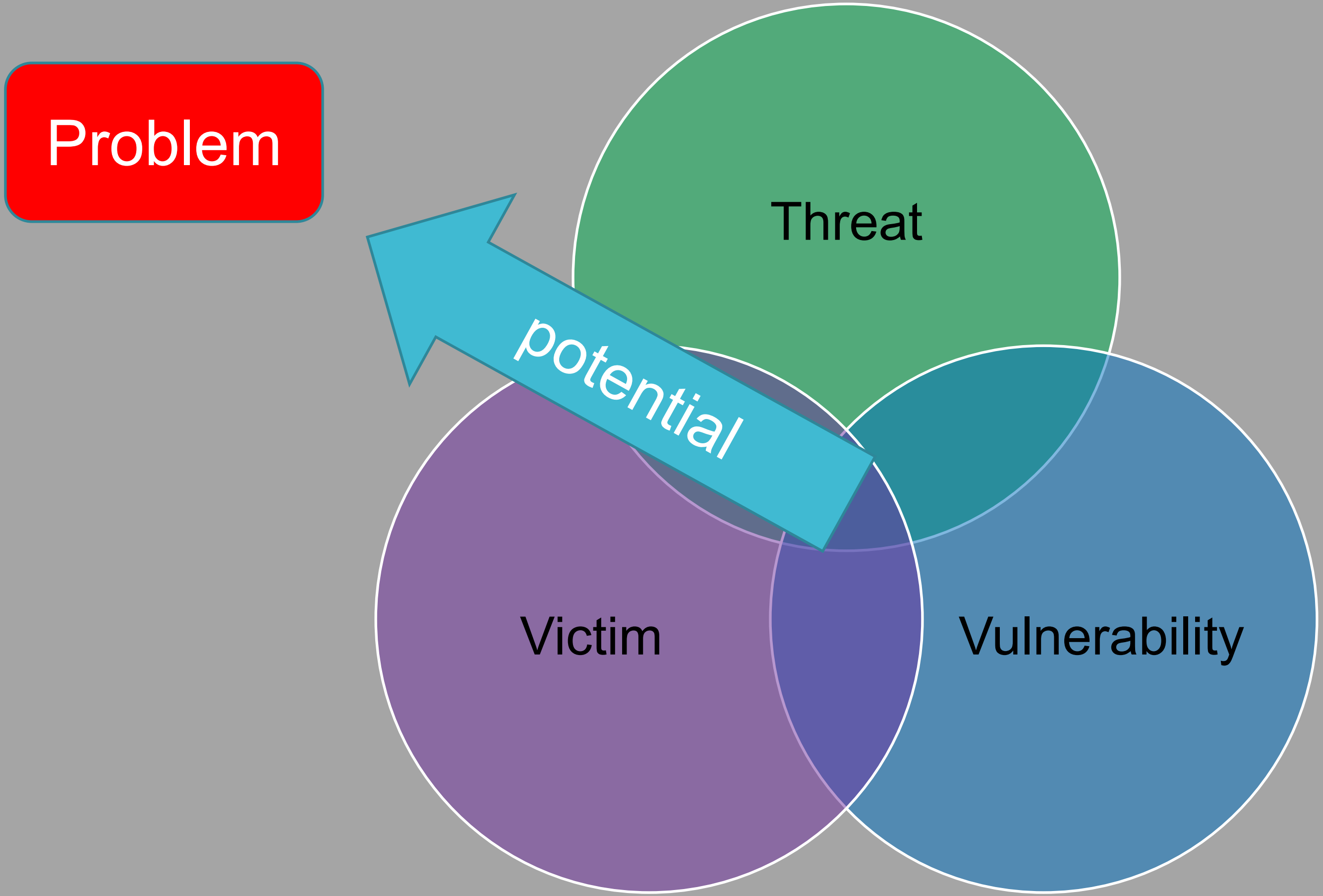
- unwanted behavior

affects

Victim

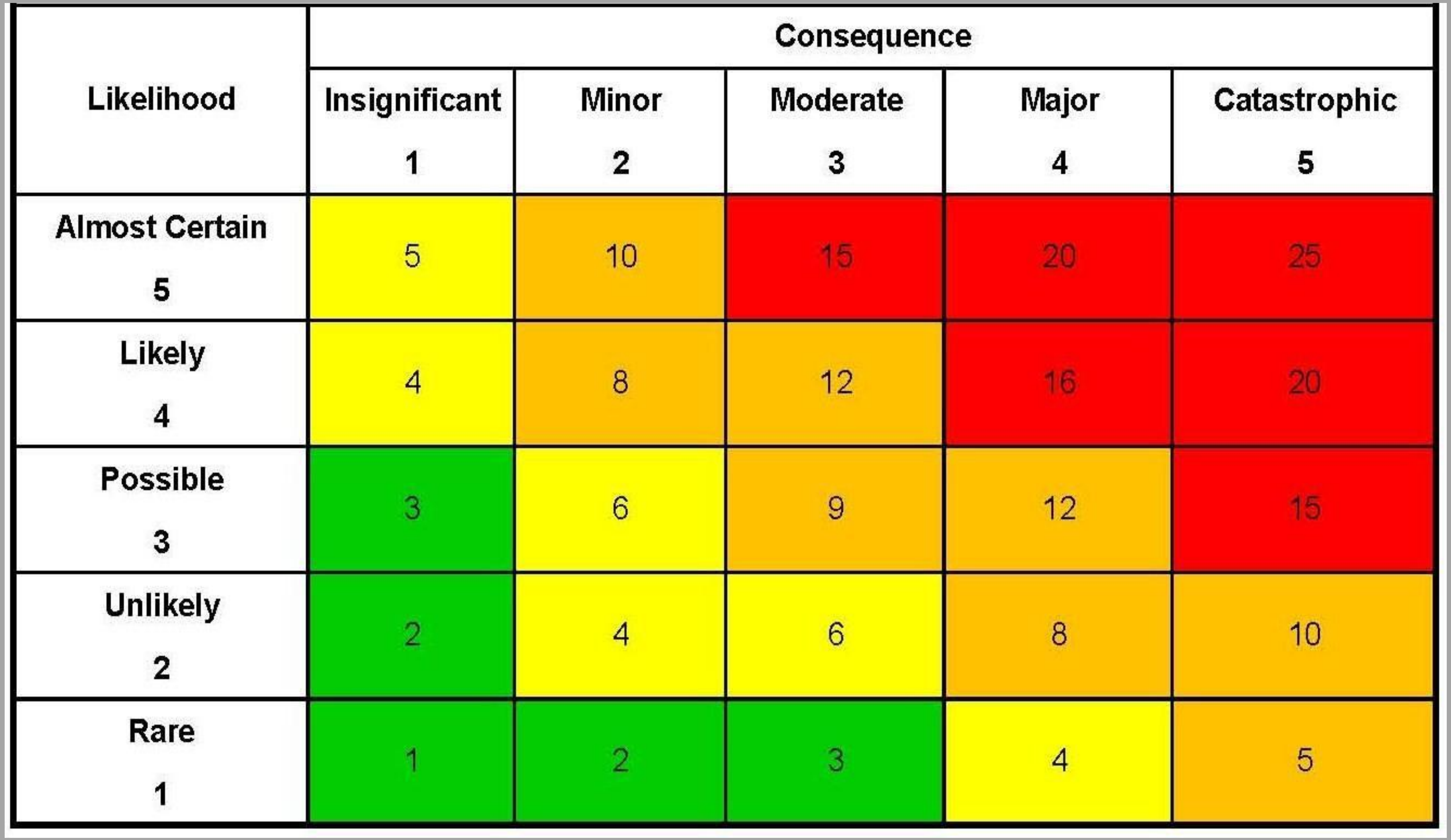
- impacted human

# RISK



Heuristic Risk-based Testing by James Bach - Satisfice.com

# RISK MATRIX (PRIORITIZATION)



| Likelihood | Consequence | | | | |
|---|---|---|---|---|---|
| | Insignificant<br>1 | Minor<br>2 | Moderate<br>3 | Major<br>4 | Catastrophic<br>5 |
| Almost Certain<br>5 | 5 | 10 | 15 | 20 | 25 |
| Likely<br>4 | 4 | 8 | 12 | 16 | 20 |
| Possible<br>3 | 3 | 6 | 9 | 12 | 15 |
| Unlikely<br>2 | 2 | 4 | 6 | 8 | 10 |
| Rare<br>1 | 1 | 2 | 3 | 4 | 5 |

# ORGANIZE TESTING AROUND RISKS

- Risk watch list
  - Risk list to review periodically for what testing has revealed
- Risk/task matrix (sorted by most important at top)

| Risks | Mitigation Tasks |
|---|---|
| If we're worried about risk X | we should invest in tasks Y |

- Component risk matrix

| Component | Risk Judgement | Risk Heuristics |
|---|---|---|
| Printing | Normal | distributed, popular |
| Reporting | Higher | new, strategic, third-party, complex, critical |
| Installation | Lower | popular, usability, changed |

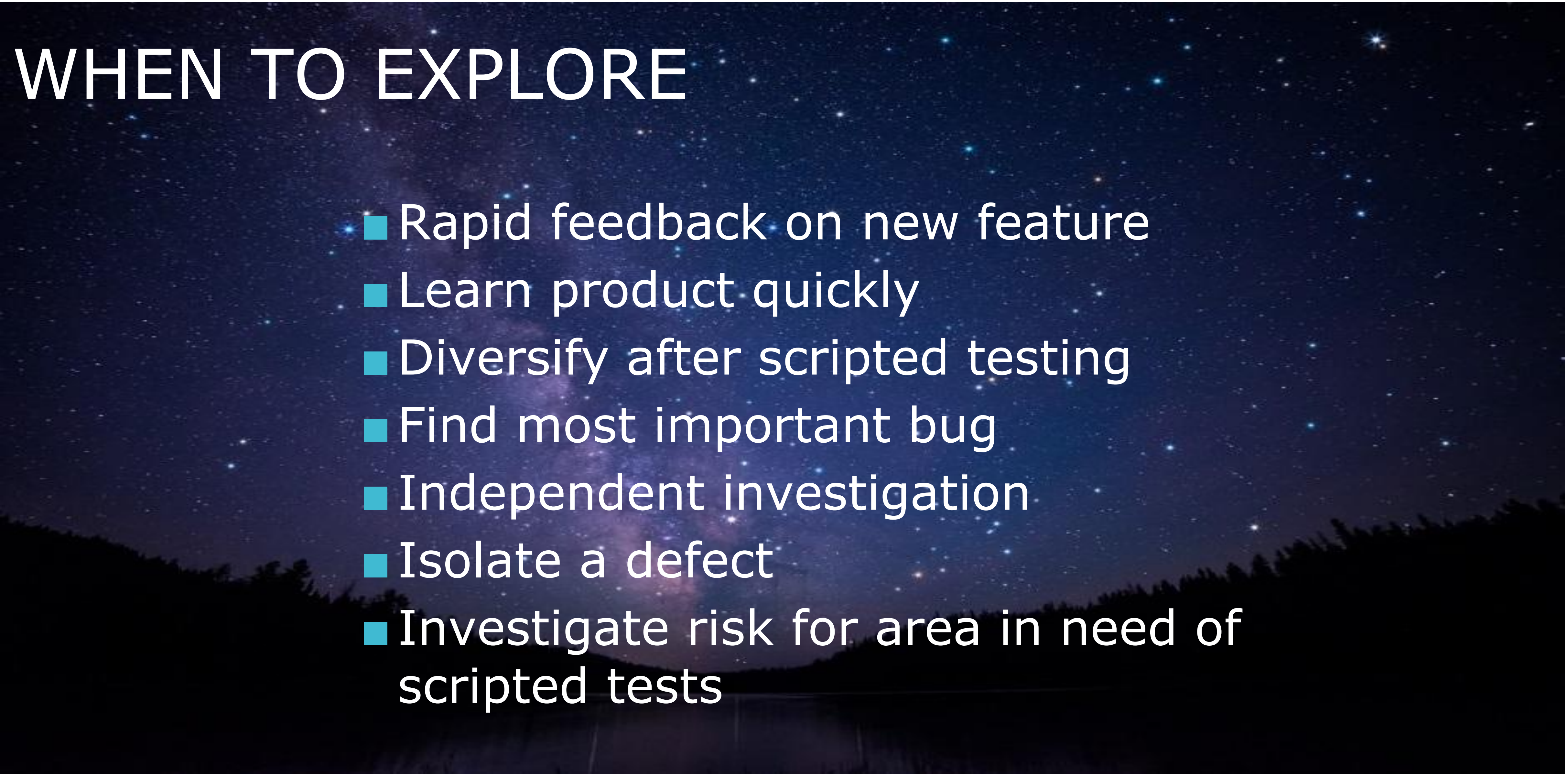
Risk-based Testing by James Bach

# EXPLORING / TESTING

# EXPLORING

- Traversing through an <u>unknown </u>area
- as a <u>self-directed</u> free agent
- learning, studying, investigating, examining, analyzing, inquiring, and searching
- to discover information
- "… guided and structured by the **person** performing that activity."
- "… **new knowledge** feeds into **choice** of which **action** to perform next."

[What exploratory testing is not Part 1 ](#) – Michael Bolton

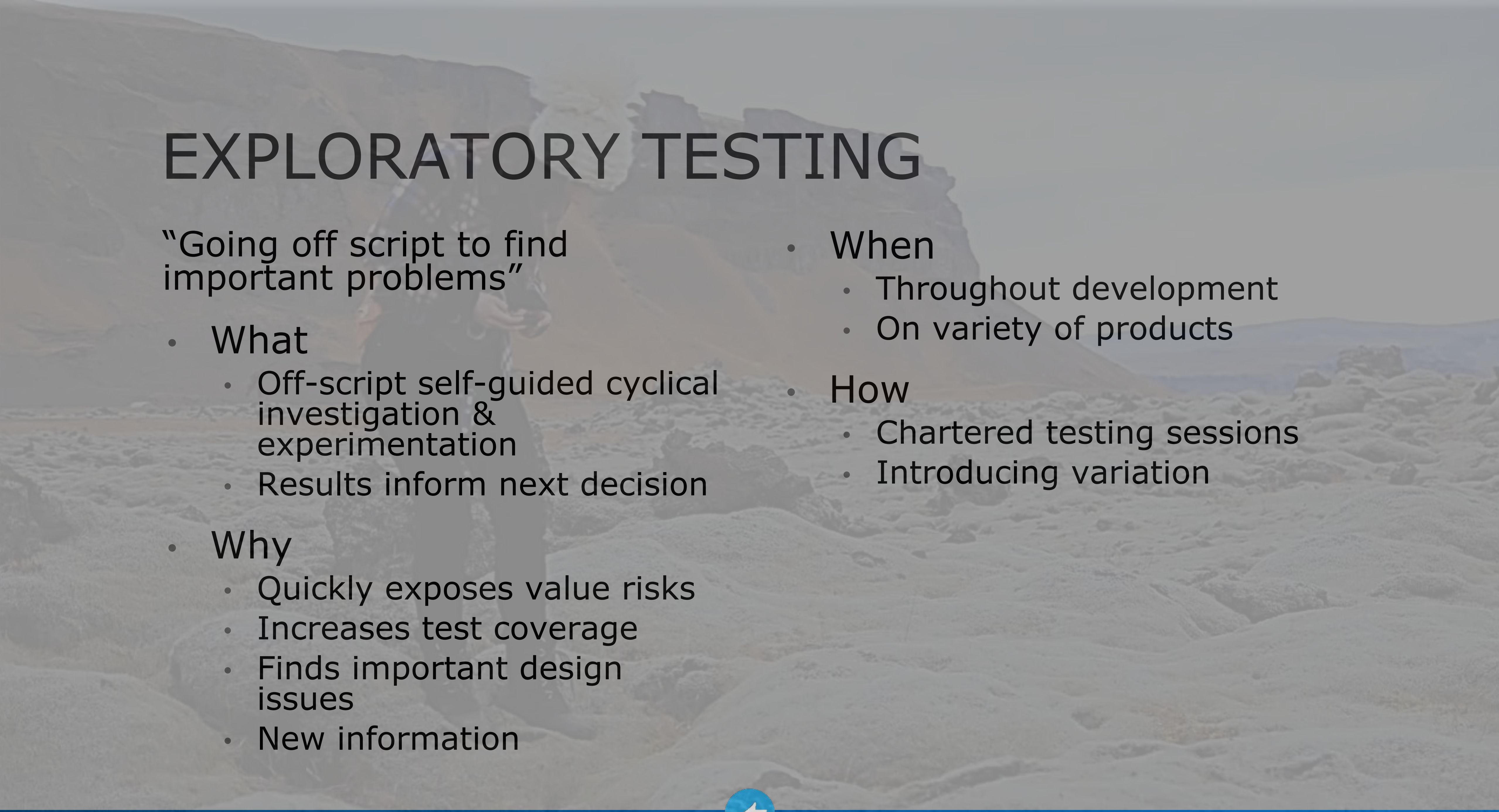


# WHEN TO EXPLORE

- Rapid feedback on new feature
- Learn product quickly
- Diversify after scripted testing
- Find most important bug
- Independent investigation
- Isolate a defect
- Investigate risk for area in need of scripted tests

# EXPLORATORY TESTING BENEFITS

- Exposes value risks quickly (2011 empirical study)
- Increases test coverage
- Includes domain experts in testing
- Provides new information
- Speeds product learning
- Develops testing skills
- Tests many work product types



# EXPLORATORY TESTING

"Going off script to find important problems"

- What
  - Off-script self-guided cyclical investigation & experimentation
  - Results inform next decision
- Why
  - Quickly exposes value risks
  - Increases test coverage
  - Finds important design issues
  - New information
- When
  - Throughout development
  - On variety of products
- How
  - Chartered testing sessions
  - Introducing variation

# HISTORY OF EXPLORATORY TESTING

**History of Definitions of ET** - James Bach / Michael Bolton

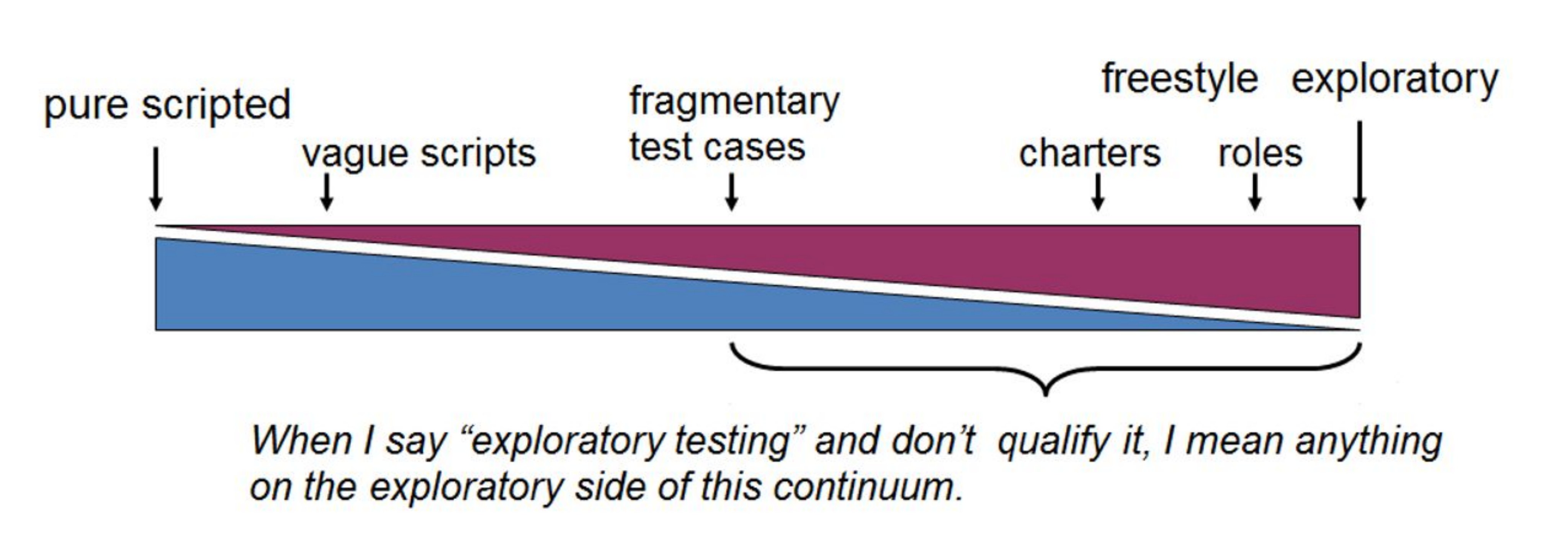
| | | |
|---|---|---|
| 1972 | Dr. William C. Hetzel<br>- "Program Test Methods"<br>- Forms of testing models/artifacts □ scripting / test factories | |
| 1983 | **Cem** Kaner, Professor of Software Engineering at Florida Institute of Technology Coined "exploratory testing" to describe the practice of some of the best testers in Silicon Valley | |
| 1987 | James Bach, Test manager at Apple Computer<br>- Ad hoc effective at bug finding<br>- Highly scripted testing ineffective | |
| 1988 | - "Testing Computer Software" by Cem Kaner, Jack Faulk, & Hung Quoc Nguyen : "quick tests", "whatever comes to mind", "guerilla raids" | |
| 1995 | **Cem** Kaner mentioned the phrase "exploratory testing" in a Usenet **message and James** Bach – "learning, planning, and testing all at the same time" | |

# HISTORY OF EXPLORATORY TESTING

History of Definitions of ET - James Bach / Michael Bolton

| | |
|---|---|
| 2003 | "Simultaneous learning, test design, and test execution" |
| 2006 | "An approach to software testing that emphasizes the **personal freedom** and **responsibility** of each tester to **continually optimize** the **value** of his[her] work by treating learning, test design and test execution as **mutually supportive** activities that run in **parallel** throughout the project." |
| 2015 | Term "Exploratory Testing" is deprecated and replaced with "Testing" |

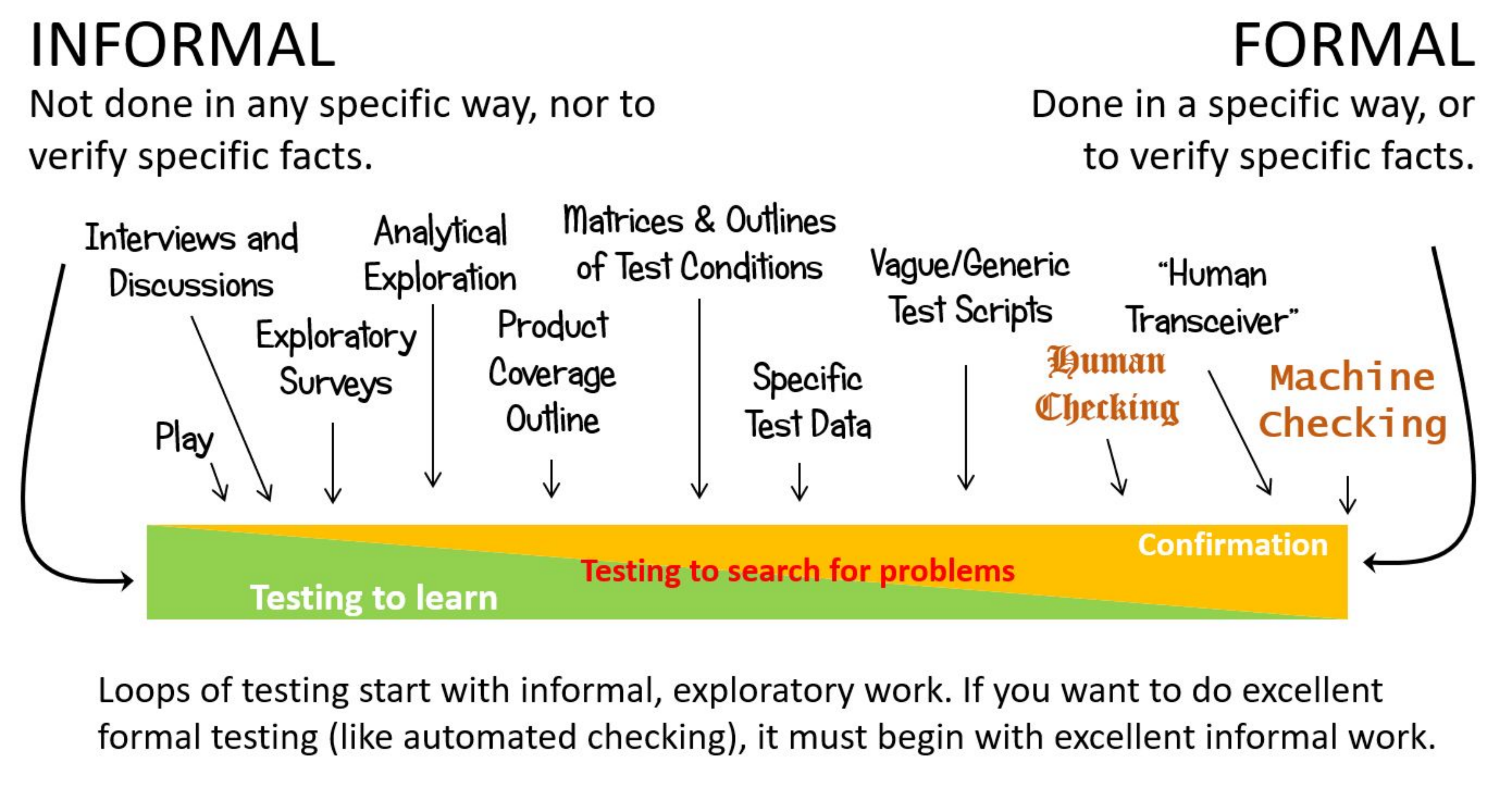
# SCRIPTED / EXPLORATORY CONTINUUM



*When I say "exploratory testing" and don't qualify it, I mean anything on the exploratory side of this continuum.*

[Exploratory Testing 3.0](#) **– Michael Bolton**

# TESTING FORMALITY CONTINUUM



INFORMAL

Not done in any specific way, nor to verify specific facts.

FORMAL

Done in a specific way, or to verify specific facts.

Interviews and Discussions

Play

Exploratory Surveys

Analytical Exploration

Product Coverage Outline

Matrices & Outlines of Test Conditions

Specific Test Data

Vague/Generic Test Scripts

Human Checking

"Human Transceiver"

Machine Checking

Testing to learn

Testing to search for problems

Confirmation

Loops of testing start with informal, exploratory work. If you want to do excellent formal testing (like automated checking), it must begin with excellent informal work.

The Sock Puppets of Formal Testing - Michael Bolton

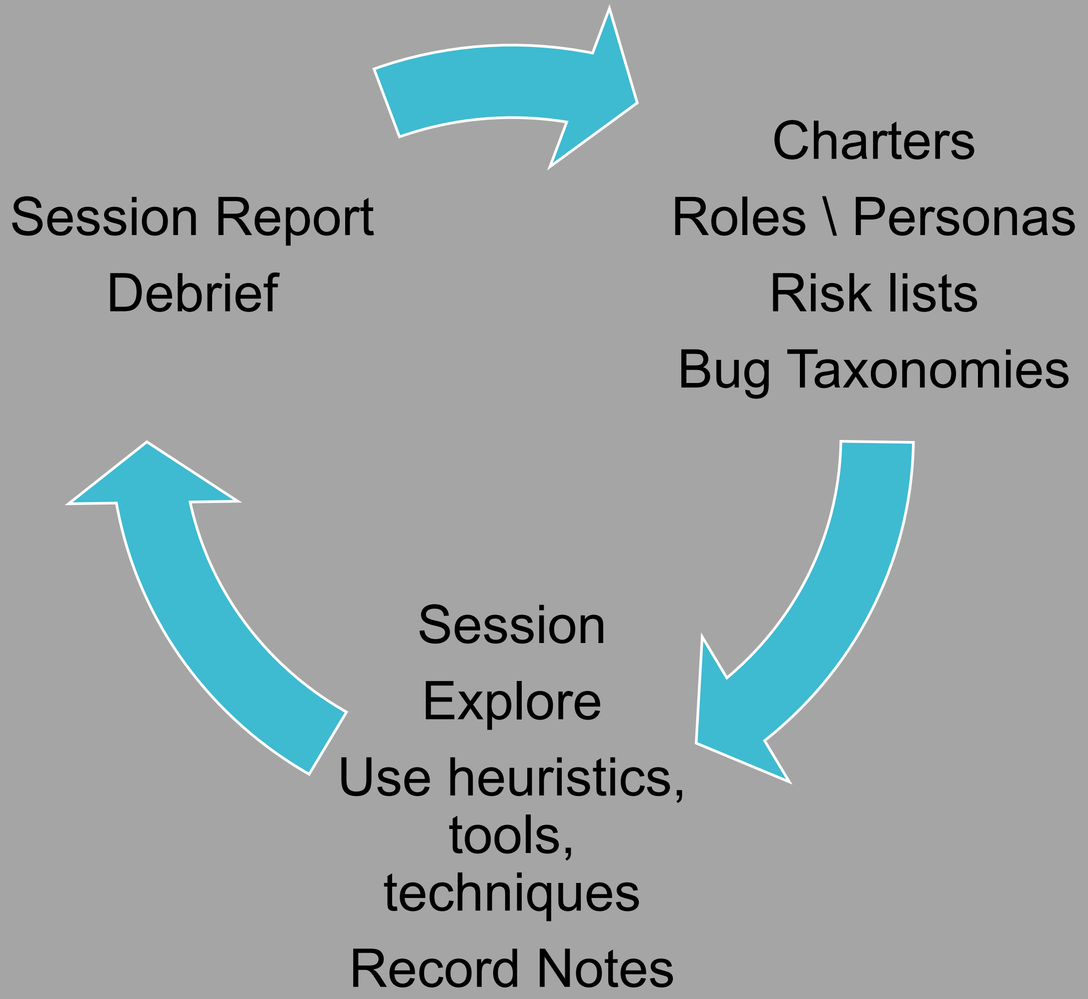
# EXPLORATORY TESTING IS NOT…

- Unstructured playing around
- Touring
- After-everything-else testing
- Tool-free testing
- Quick tests ("attacks")
- Undocumented testing
- Limited to black box testing
- Limited to function testing
- Limited to an example

# DIMENSIONS OF TESTING

- Design, execution, interpretation and learning
  - Happen together
  - Performed by same person
- Tester makes own choices about what, when and how to test
- All prior learning informs the tester’s choices about the next test
- Revealing new information vs. confirming existing product knowledge
- Varying test aspects vs. repeating tests

# SESSION BASED TESTING



Charters

Roles \ Personas

Risk lists

Bug Taxonomies

Session

Explore

Use heuristics, tools, techniques

Record Notes

Session Report

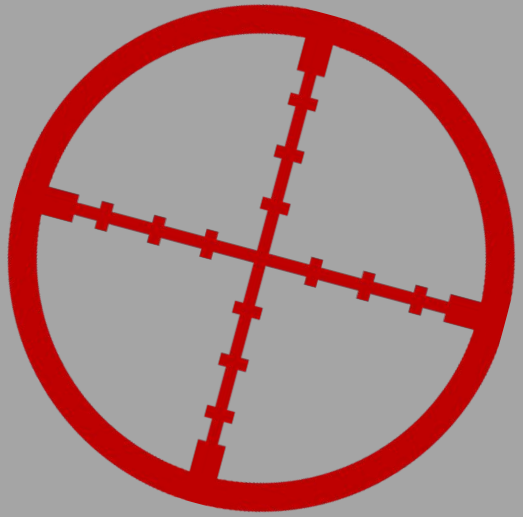
Debrief

# TEST MISSIONS

> “Different objectives require different testing tools and strategies, and will yield different tests, different test documentation, and different results.”
> – Cem Kaner

The Ongoing Revolution in Software Testing – Cem Kaner

MISSIO
N:



- Find problems that might threaten the value of the product
- Assess quality
- Conform to regulations
- Assess conformance to specification
- Evaluate our product and its interaction with a related product
- Inform ship/no-ship decisions
- Block premature product releases
- Competitive evaluation
- Find safe scenarios and workarounds for problems
- Minimize safety-related lawsuit risk

# CHARTER

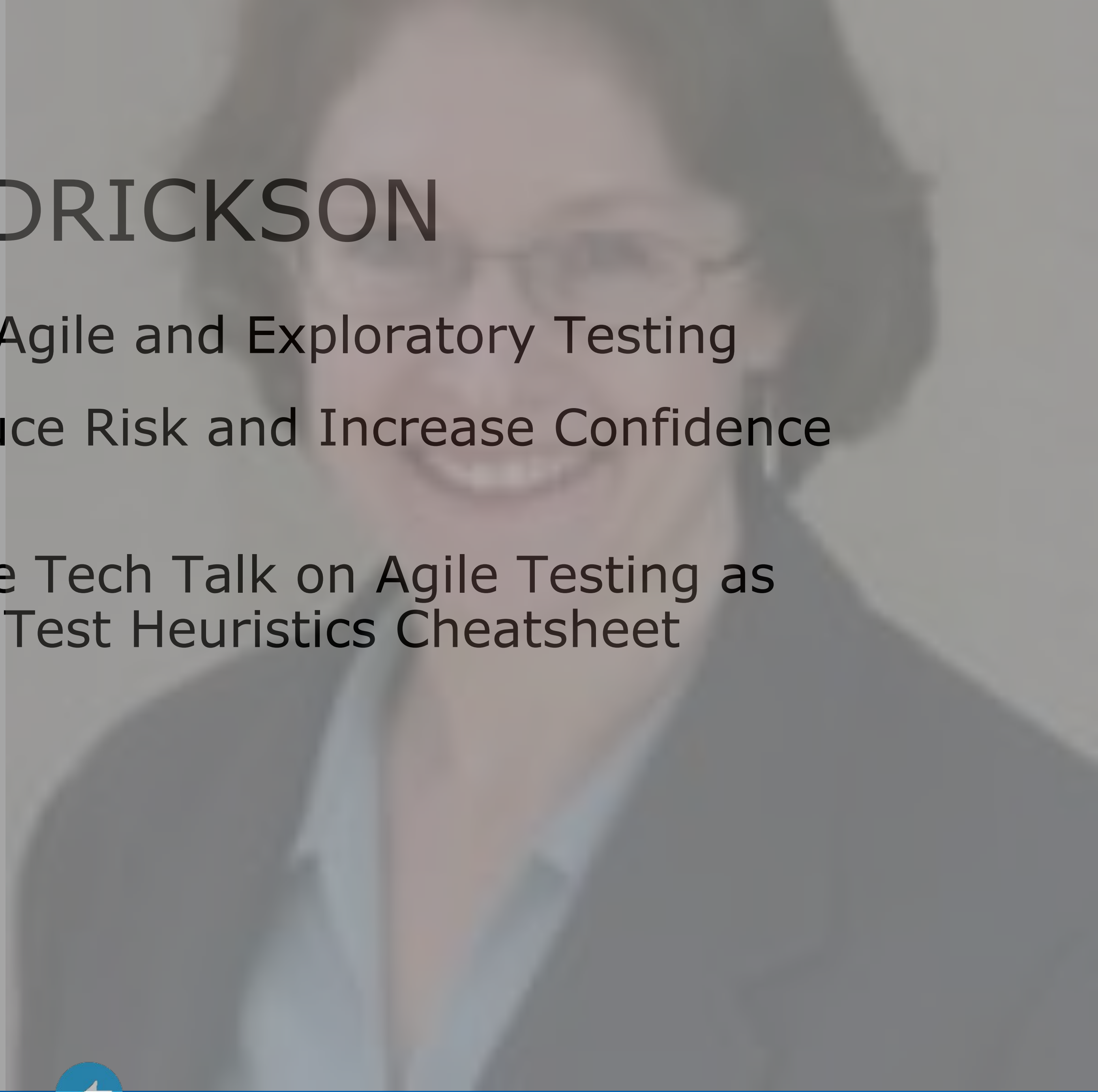
## Brief purpose / mission

**Explore (target)** feature, requirement, or module

**With (resources)** tool, data set, technique, configuration

**To discover (information)** quality issues

# ELIZABETH HENDRICKSON



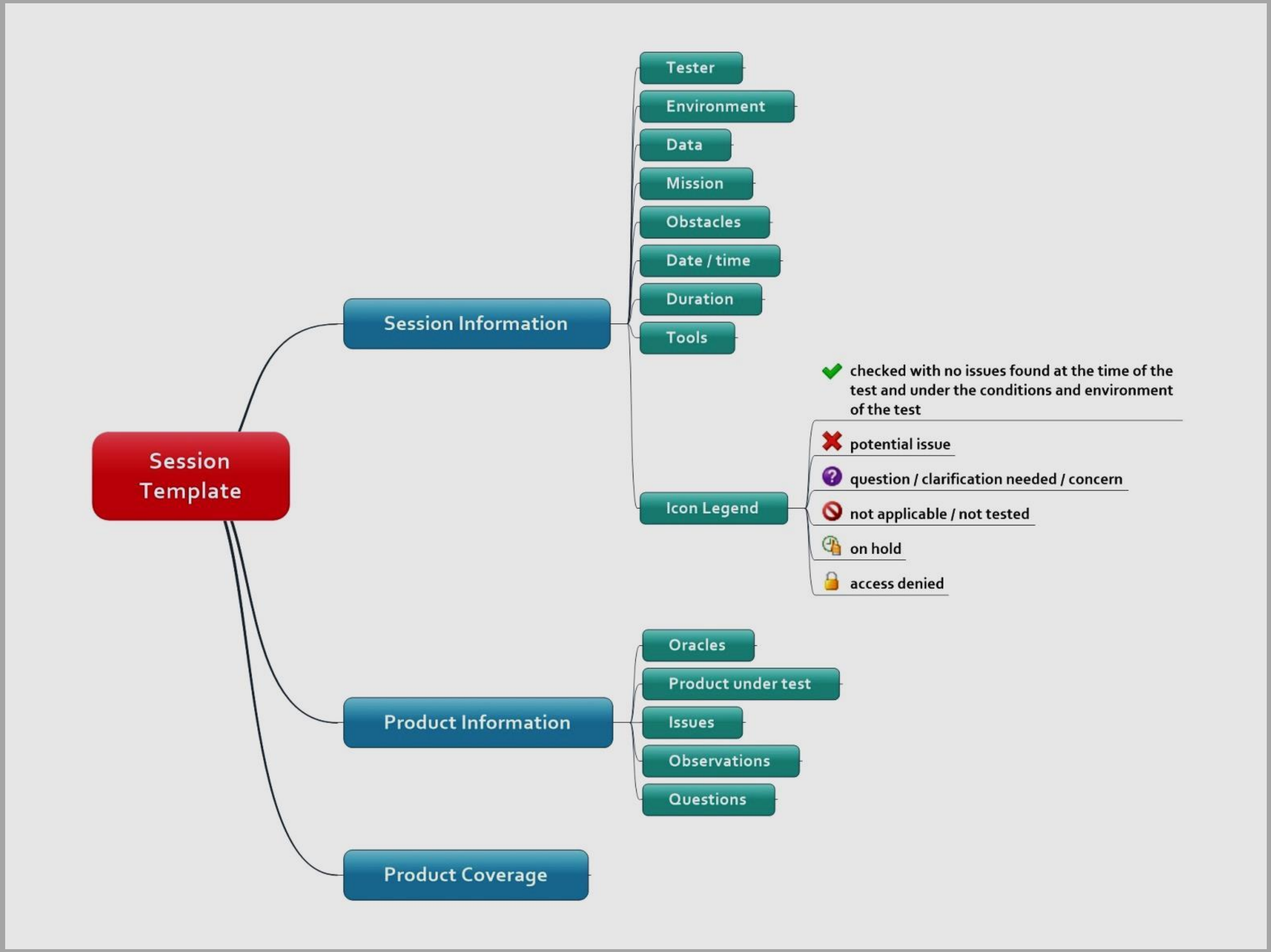
- Consultant and trainer in Agile and Exploratory Testing
- Author of Explore It! Reduce Risk and Increase Confidence with Exploratory Testing
- Best known for her Google Tech Talk on Agile Testing as well as her wildly popular Test Heuristics Cheatsheet
- http://testobsessed.com

# SESSION BASED TESTING

- Uninterrupted Time-box
  - Normal is 90 minutes +/- 15
- Charter – brief purpose/mission or problem to solve
- Progress – Kanban: each card is a session charter
- Reviewable results
  - Test notes
- Debriefing – lead/manager
  - What testing was done or is in progress
  - Results/metrics
- Metrics – count number of sessions (not test cases)
- Retrospective with participants

From Session Based Test Management by James Bach and Michael Bolton

# SESSION TEMPLATE



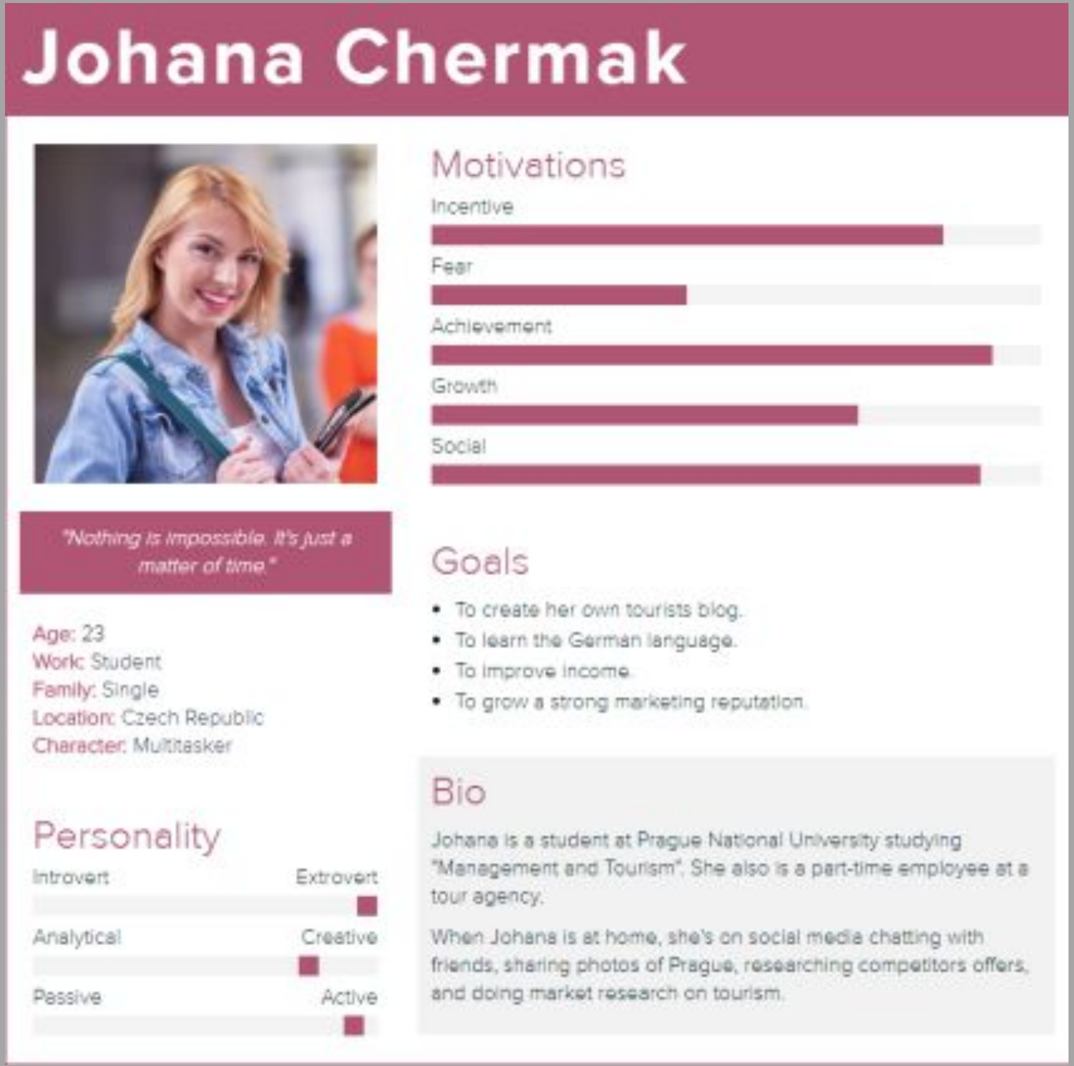
Session Template

- Session Information
  - Tester
  - Environment
  - Data
  - Mission
  - Obstacles
  - Date / time
  - Duration
  - Tools
  - Icon Legend
    - ✓ checked with no issues found at the time of the test and under the conditions and environment of the test
    - ✗ potential issue
    - ? question / clarification needed / concern
    - 🚫 not applicable / not tested
    - ⏸ on hold
    - 🔒 access denied
- Product Information
  - Oracles
  - Product under test
  - Issues
  - Observations
  - Questions
- Product Coverage

# PERSONAS

- Fictitious example of a user based on knowledge of real users
- Not a use case actor, but typical instances of an actor
- Real name, personality, motivations, goals and photo
- Not “invented” but “discovered” by requirements investigation
- Represent behavior patterns, experience goals

From “The Inmates Are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity” by Alan Cooper



## Johana Chermak

*"Nothing is impossible. It's just a matter of time."*

Age: 23
Work: Student
Family: Single
Location: Czech Republic
Character: Multitasker

### Personality

Introvert — Extrovert

Analytical — Creative

Passive — Active

### Motivations

Incentive

Fear

Achievement

Growth

Social

### Goals

- To create her own tourists blog.
- To learn the German language.
- To improve income.
- To grow a strong marketing reputation.

### Bio

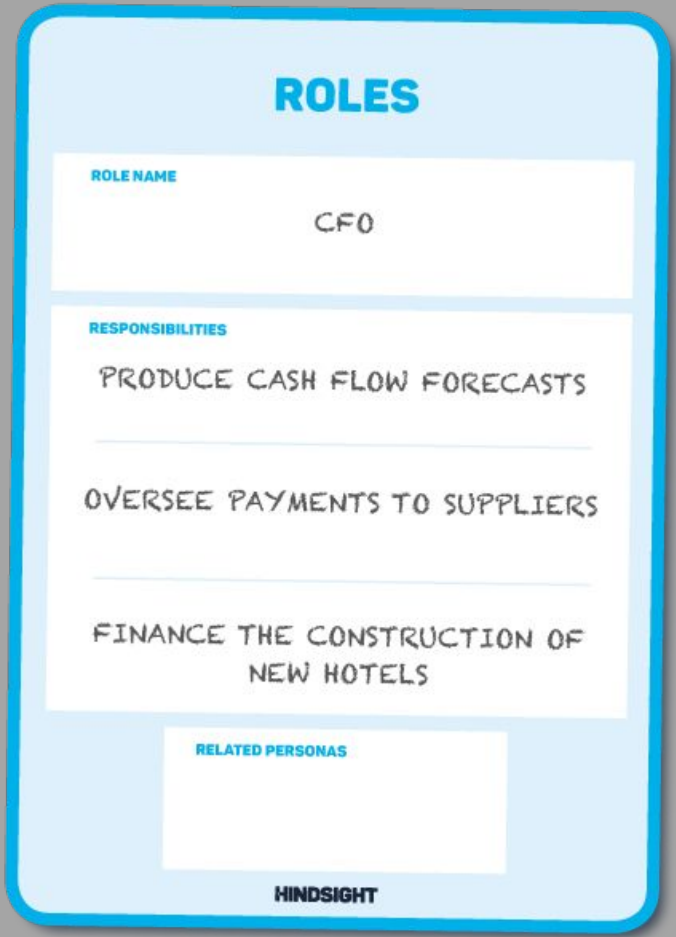
Johana is a student at Prague National University studying "Management and Tourism". She also is a part-time employee at a tour agency.

When Johana is at home, she's on social media chatting with friends, sharing photos of Prague, researching competitors offers, and doing market research on tourism.

# PERSONA TEMPLATE

| Picture & Name | Details | Goal |
|---|---|---|
| Representative picture and name to develop sympathy for the persona. | Relevant characteristics.<br>Relevant behaviors.<br>Demographics, job, lifestyle, hobbies, attitudes, common tasks, etc. | Why would they use the product?<br>What problems for the persona should the product solve? (put main problem at top) |

# ROLES



## ROLES

ROLE NAME

CFO

RESPONSIBILITIES

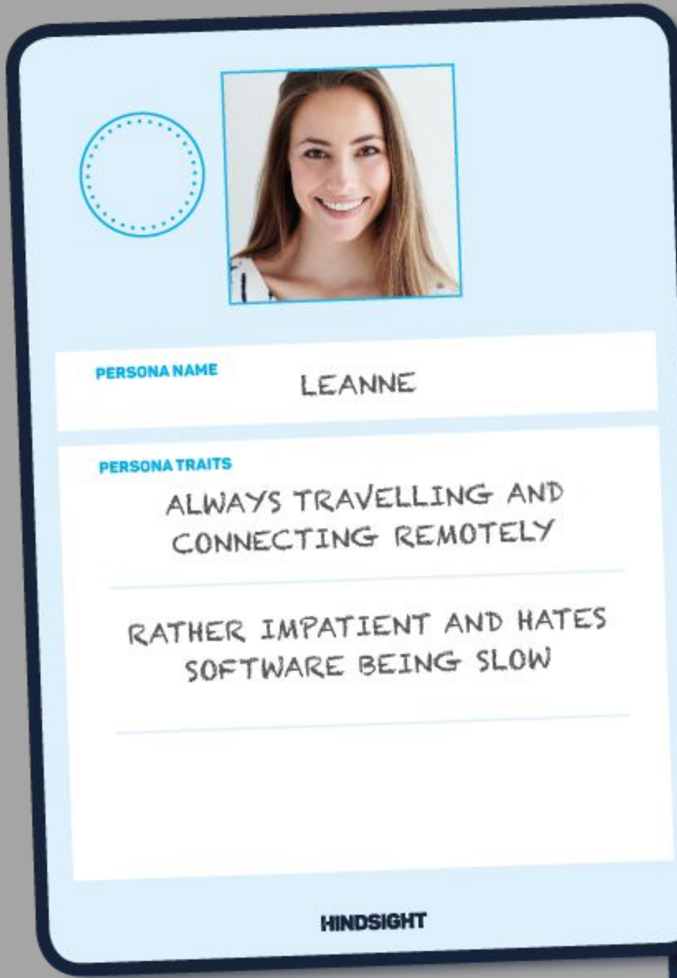
PRODUCE CASH FLOW FORECASTS

OVERSEE PAYMENTS TO SUPPLIERS

FINANCE THE CONSTRUCTION OF NEW HOTELS

RELATED PERSONAS

HINDSIGHT



PERSONA NAME

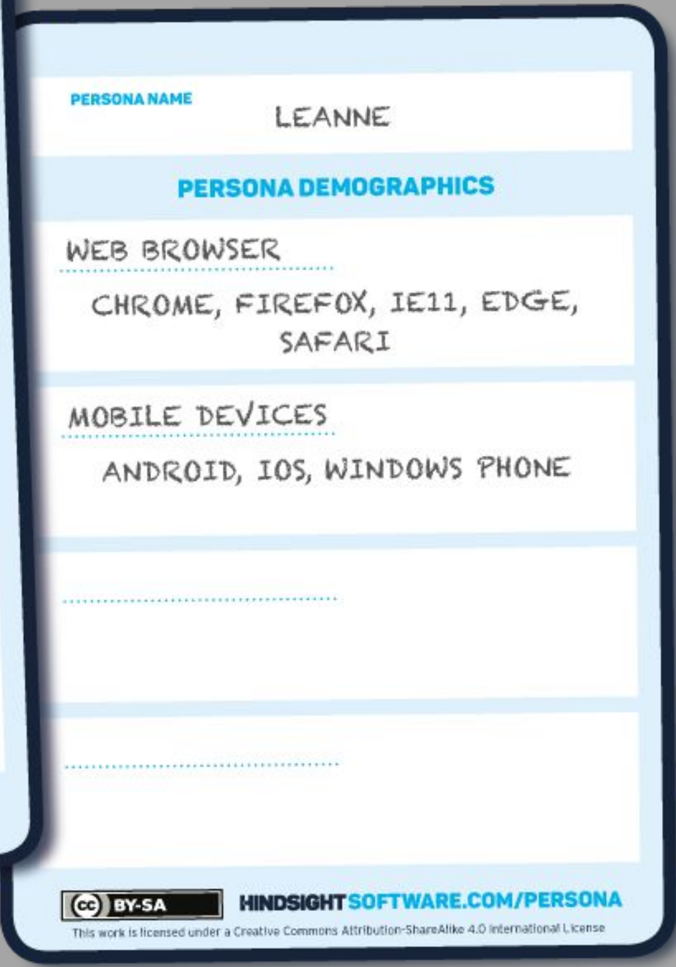
LEANNE

PERSONA TRAITS

ALWAYS TRAVELLING AND CONNECTING REMOTELY

RATHER IMPATIENT AND HATES SOFTWARE BEING SLOW

HINDSIGHT



PERSONA NAME

LEANNE

## PERSONA DEMOGRAPHICS

WEB BROWSER

CHROME, FIREFOX, IE11, EDGE, SAFARI

MOBILE DEVICES

ANDROID, IOS, WINDOWS PHONE

# RISK LISTS

## QUALITY CRITERIA CATEGORIES

- Capability
- Reliability
- Usability & Accessibility
- Charisma
- Security
- Scalability
- Compatibility
- Performance

## RISK CATEGORIES

- Complex
- New/Changed
- Up/downstream dependency
- Critical
- Precise
- Popular
- Strategic
- Third party

9:41
Tuesday
12
Mail
Calendar
Photos
Camera
Weather
News
Maps
Clock
Stocks
Reminders
Home
Notes
Books
TV
App Store
Health

# TOOLS

- XMind
- MindManager

Mind mapping software

- Rapid Reporter
- Test & Feedback

Recording and reporting of test activity

- PerlClip
- Bug Magnet

Input data tools

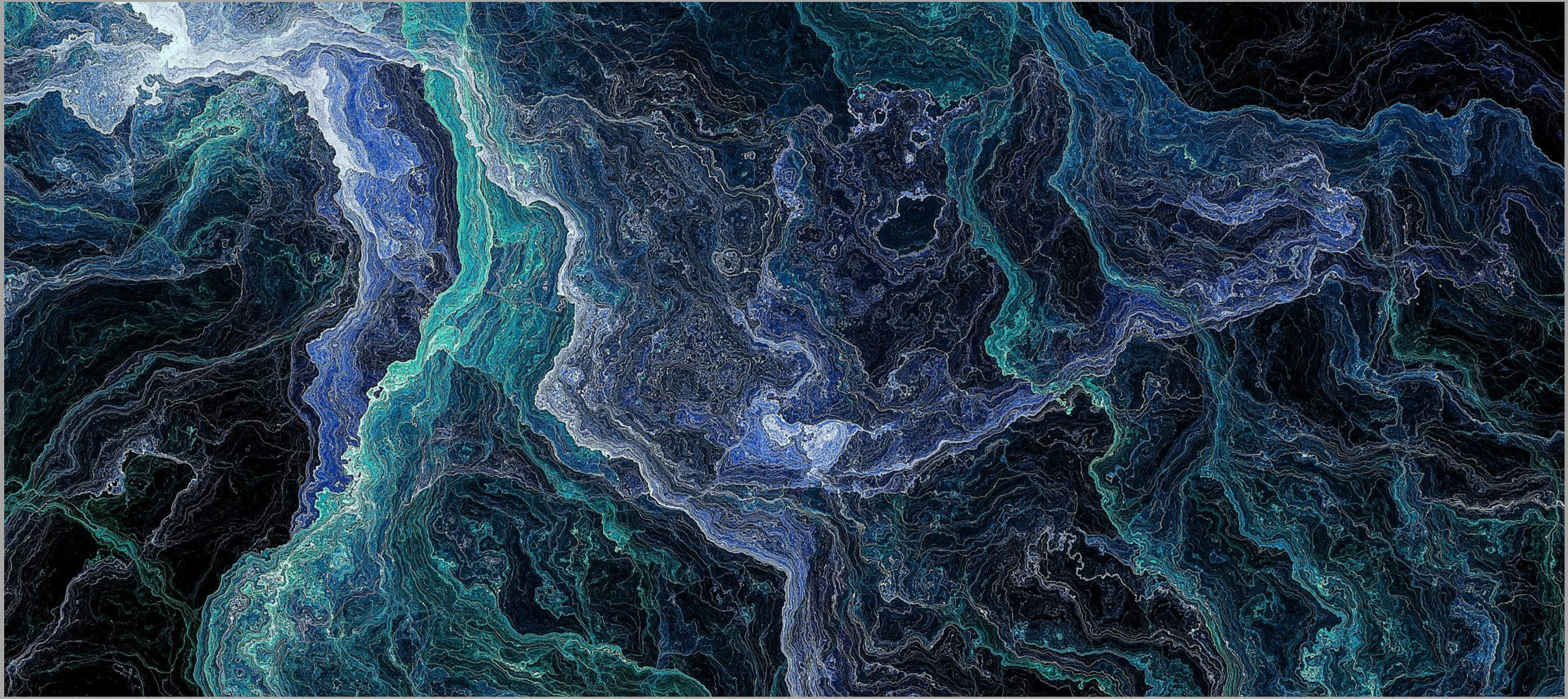
- Fiddler
- Postman

Tools to expose the API

# SOME OTHER TOOLS I HAVE USED

- Own critical thinking mind & senses
- Checklists
- Personas
- MS Word / Excel
- Dev Tools (F12)
- PSR
- Snagit
- Check My Links
- Lighthouse
- Grammar.com
- Time-zone Converter
- Google Translate
- Powershell
- WebAIM Color Contrast Checker
- Dencoder

# QUESTIONS



# ADDENDUM

# EXPLORATORY SKILLS

## SELF-MANAGEMENT

- Chartering work
- Setting procedures & protocols
- Establish success conditions
- Self-care / Self-criticism / Ethics
- Test status evaluation
- Branching work and backtracking
- Focusing / De-focusing work
- Alternating activities
- Keeping useful concise records
- Knowing when to stop

## COLLABORATION

- Getting to know people
- Conversation
- Serving / Guiding other testers
- Asking for help
- Role visiting
- Telling the Testing Story
- Telling the Product Story

[Exploratory Testing Dynamics](#)
[Coding QA Podcast on Exploratory Testing](#)

# EXPLORATORY SKILLS

## LEARNING

- Resources / Web / History
- Reading / Analyzing Documents
- Interviewing
- Pursue Lines of Inquiry
- Indulge Curiosity
- Idea Generation
- Dropping Ideas for Faster Progress
- Recovering Ideas

## TEST PERFORMANCE

- Encountering the Product / Sensemaking / Modeling / Factoring
- Analyzing Product Risk
- Designing Experiments
- Observation / Problem Detection
- Assessing Validity
- Notetaking
- Data Wrangling
- Bug Reporting & Advocacy
- Applying Tools
- Testability Advocacy

Exploratory Testing Dynamics

Coding QA Podcast on Exploratory Testing

# EXPLORATORY SKILLS

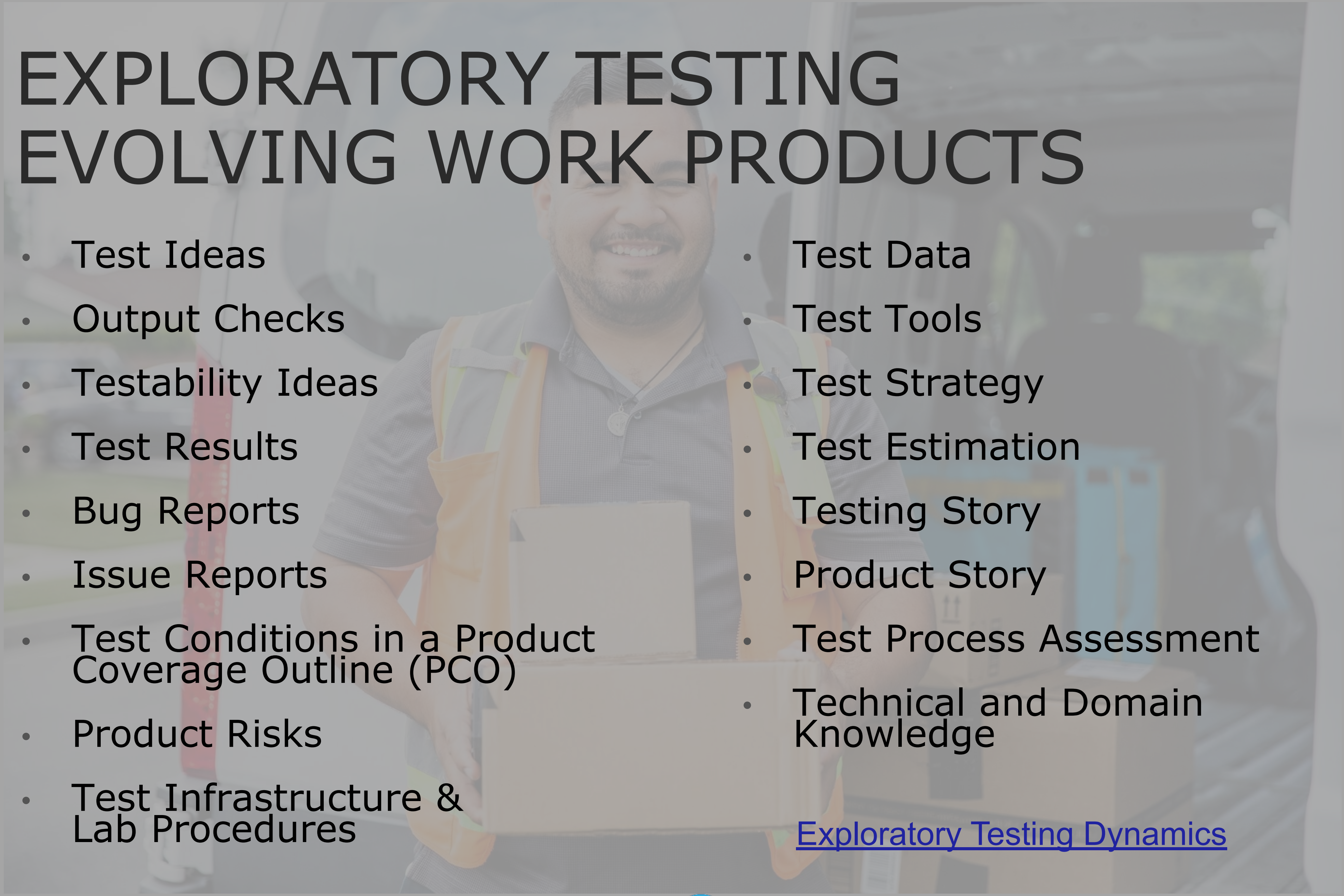
## KNOWLEDGE

- Product
- Technology
- Project
- Domain
- General Systems
- Tools
- Test Technique
- Resource
- People

## OTHER

- Coding
- Design
- Social Science
- Specification Writing
- Math & Logic
- Cognitive Science

Exploratory Testing Dynamics
Coding QA Podcast on Exploratory Testing

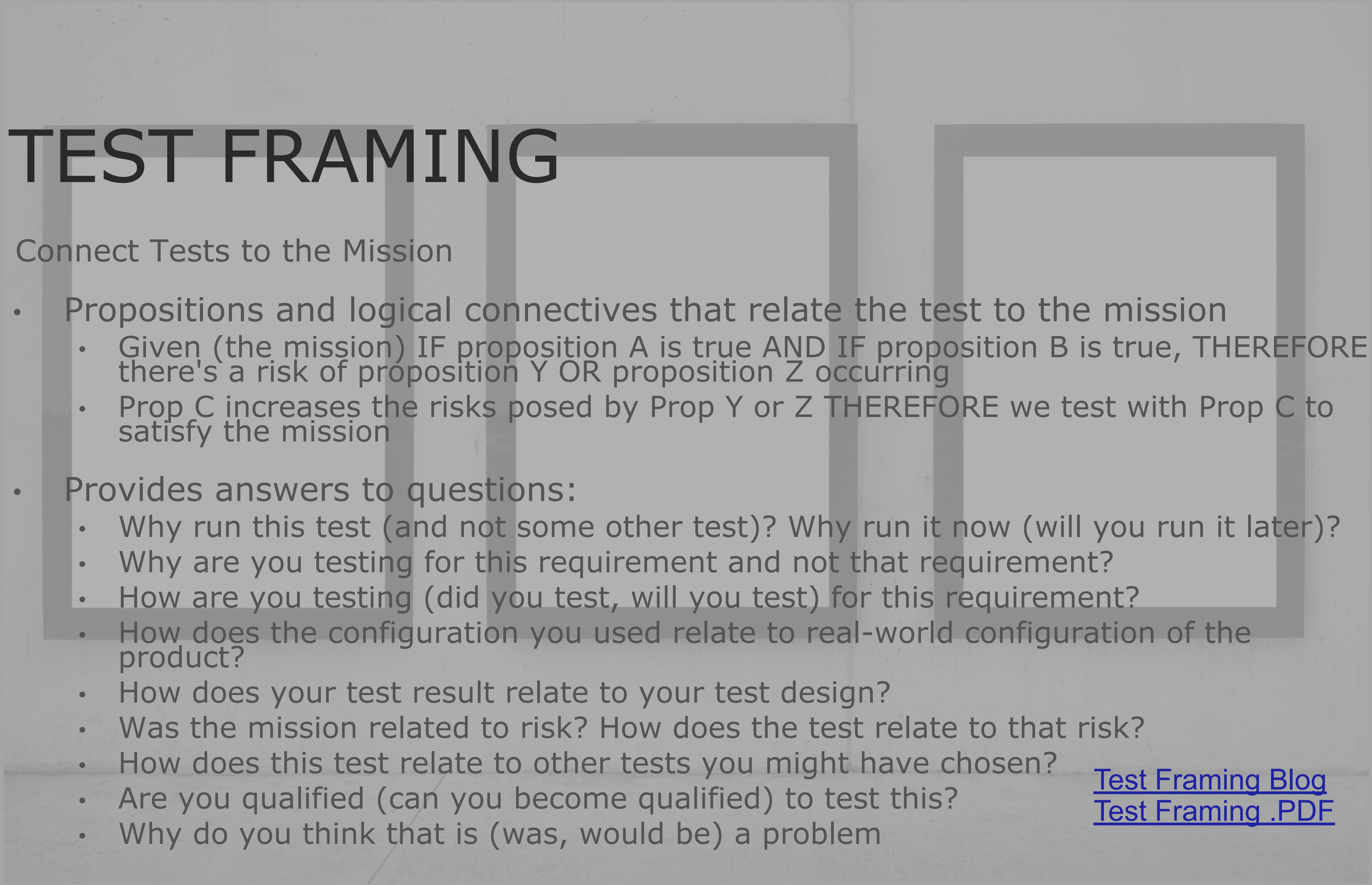


# EXPLORATORY TESTING
# EVOLVING WORK PRODUCTS

- Test Ideas
- Output Checks
- Testability Ideas
- Test Results
- Bug Reports
- Issue Reports
- Test Conditions in a Product Coverage Outline (PCO)
- Product Risks
- Test Infrastructure & Lab Procedures

- Test Data
- Test Tools
- Test Strategy
- Test Estimation
- Testing Story
- Product Story
- Test Process Assessment
- Technical and Domain Knowledge

[Exploratory Testing Dynamics]



# TEST FRAMING

Connect Tests to the Mission

- Propositions and logical connectives that relate the test to the mission
  - Given (the mission) IF proposition A is true AND IF proposition B is true, THEREFORE there's a risk of proposition Y OR proposition Z occurring
  - Prop C increases the risks posed by Prop Y or Z THEREFORE we test with Prop C to satisfy the mission
- Provides answers to questions:
  - Why run this test (and not some other test)? Why run it now (will you run it later)?
  - Why are you testing for this requirement and not that requirement?
  - How are you testing (did you test, will you test) for this requirement?
  - How does the configuration you used relate to real-world configuration of the product?
  - How does your test result relate to your test design?
  - Was the mission related to risk? How does the test relate to that risk?
  - How does this test relate to other tests you might have chosen?
  - Are you qualified (can you become qualified) to test this?
  - Why do you think that is (was, would be) a problem

Test Framing Blog
Test Framing .PDF

# REFERENCES

- Satisfice, Inc. http://www.satisfice.com/
- DevlopSense http://www.developsense.com/
- Cem Kaner http://kaner.com/
- Gerald M. Weinberg http://www.geraldmweinberg.com
- Context
  - Model: http://www.satisfice.com/tools/satisfice-cm.pdf
- Strategy
  - Model: http://www.satisfice.com/tools/htsm.pdf
- Oracles
  - http://www.developsense.com/articles/2005-01-TestingWithoutAMap.pdf
  - http://www.developsense.com/blog/2012/07/few-hiccupps/
- Exploration (exploratory testing)
  - http://www.developsense.com/blog/2018/07/exploratory-testing-on-an-api-part-1/
  - https://www.youtube.com/watch?v=KzIWVsuc_mI
  - http://kaner.com/pdfs/ExploringExploratoryTesting.pdf
  - http://www.testingeducation.org/BBST/exploratory/BBSTExploring.pdf
  - http://www.testingeducation.org/BBST/testdesign/KanerJohnson_LAWST7StylesOfExploration.pdf
  - Explore It! Reduce Risk and Increase Confidence with Exploratory Testing. By Elisabeth Hendrickson.
  - Exploring Requirements: Quality Before Design, by Don Gause and Gerald M. Weinberg
- Trello Book List for software development and testing
  - https://trello.com/b/WH2I8tCq/book-list-software-development-and-testing

# CONTEXT REFERENCES

- Context-free Questions
  - **Exploring Requirements: Quality Before Design** by Gerald M. Weinberg and Donald C. Gause
  - Michael Bolton's Blog: **Context-Free Questions for Testing**

  http://www.developsense.com/blog/2010/11/context-free-questions-for-testing/

- Context Model
  - By James Bach: http://www.satisfice.com/tools/satisfice-cm.pdf

# RISK REFERENCES

- Risk-based Testing by James Bach
  - https://youtu.be/Sz-WiCV2eh4
  - http://www.satisfice.com/articles/hrbt.pdf
- NIST Risk Management Framework
  - https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-30r1.pdf
  - https://csrc.nist.gov/publications/detail/sp/800-37/rev-2/draft

- Using Bug Taxonomies
  - http://www.testingeducation.org/articles/bug_taxonomies_use_them_to_generate_better_tests_star_east_2003_paper.pdf

- Risk-based Testing Taxonomy
  - https://arxiv.org/pdf/1801.06812.pdf
    - Michael Felderer, Juergen Grossmann, Ina Schieferdecker: Recent Results on Classifying Risk-Based Testing Approaches. CoRR abs/1801.06812 (2018)
  - Evaluation of Comprehensive Taxonomies for Information Technology Threats
    - https://www.sans.org/reading-room/whitepapers/threatintelligence/evaluation-comprehensive-taxonomies-information-technology-threats-38360

# RISK CATALOGS

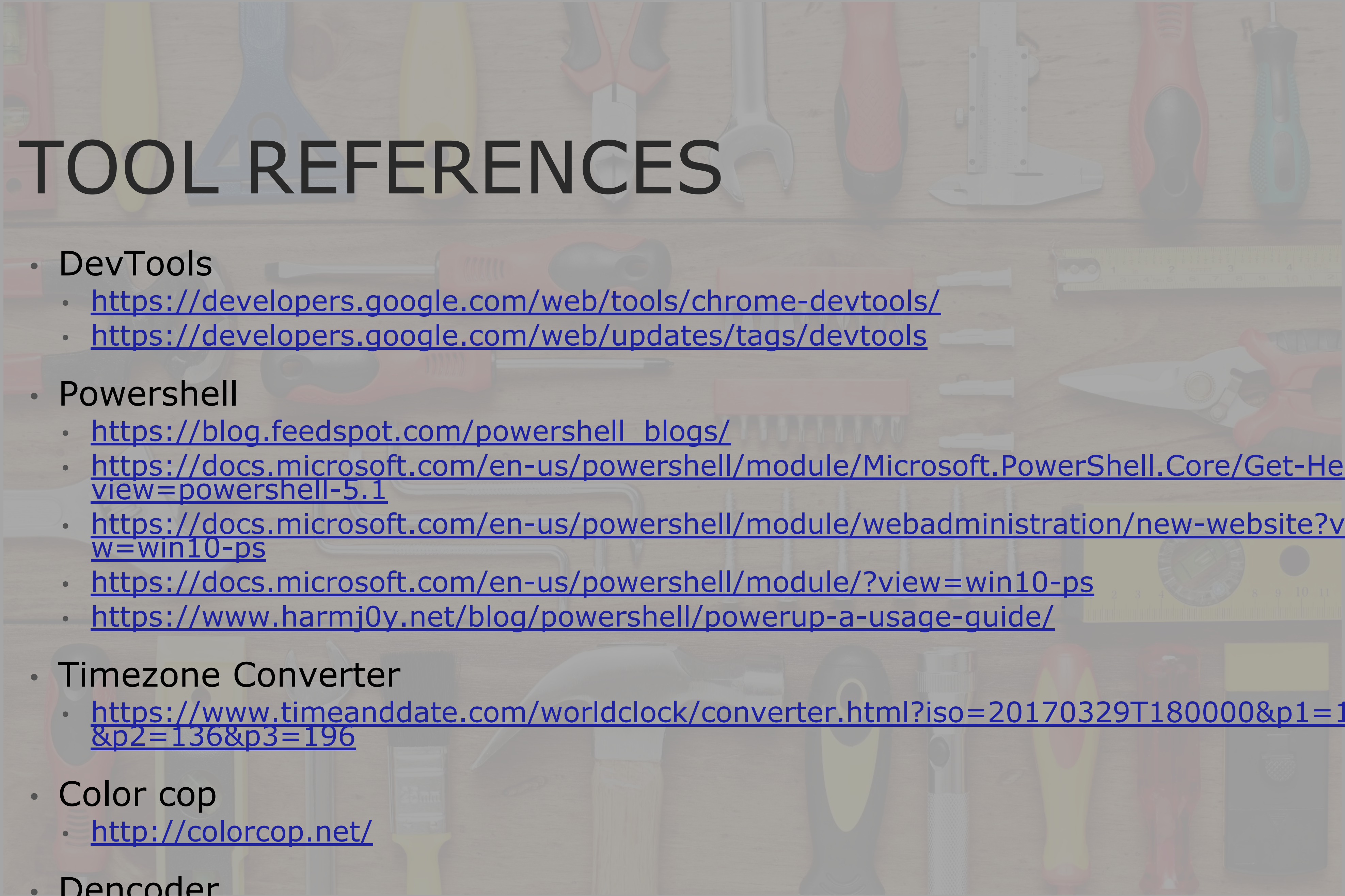
- Web Fault Taxonomy
  - https://www.riverpublishers.com/journal/journal_articles/RP_Journal_1540-9589_842.pdf
    - Marchetto, Alessandro & Ricca, Filippo & Tonella, Paolo. (2009). An Empirical Validation of a Web Fault Taxonomy and its Usage for Web Testing.. J. Web Eng.. 8. 316-345.
- eCommerce Risk Taxonomy
  - A taxonomy of e-commerce risks and failures https://dspace-test.lib.fit.edu/handle/11141/129
    - Vijayaraghavan, G.V. (2003). A taxonomy of e-commerce risks and failures (CS-2003-22). Melbourne, FL. Florida Institute of Technology.
  - Bugs in your shopping cart: A Taxonomy http://kaner.com/pdfs/BugsInYourShoppingCart.pdf
- Cyber Security Threat Risk Taxonomies
  - FAIR Model (Factor Analysis of Information Risk)
    - https://cdn2.hubspot.net/hubfs/1616664/The%20FAIR%20Model_FINAL_Web%20Only.pdf
    - https://www.fairinstitute.org/what-is-fair
  - Open Threat Taxonomy https://www.auditscripts.com/free-resources/open-threat-taxonomy/
  - ENISA Threat Taxonomy
    - https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends/enisa-threat-landscape/etl2015/enisa-threat-taxonomy-a-tool-for-structuring-threat-information
- Broad Risk Catalog
  - https://www.researchgate.net/publication/220689439_Testing_computer_software_2_ed
    - Kaner, Cem & Falk, Jack & Quoc Nguyen, Hung. (1993). Testing computer software (2. ed.)

# HEURISTICS REFERENCES

- Heuristic Test Strategy Model
  - http://www.satisfice.com/tools/htsm.pdf
- Test Heuristics Cheat Sheet
  - http://testobsessed.com/wp-content/uploads/2011/04/testheuristicscheatsheetv1.pdf

# PERSONA REFERENCES

- **The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity** by Alan Cooper
  - **https://www.amazon.com/exec/obidos/ASIN/0672326140/ambysoftinc/**



# TOOL REFERENCES

- DevTools
  - https://developers.google.com/web/tools/chrome-devtools/
  - https://developers.google.com/web/updates/tags/devtools
- Powershell
  - https://blog.feedspot.com/powershell_blogs/
  - https://docs.microsoft.com/en-us/powershell/module/Microsoft.PowerShell.Core/Get-Help?view=powershell-5.1
  - https://docs.microsoft.com/en-us/powershell/module/webadministration/new-website?view=win10-ps
  - https://docs.microsoft.com/en-us/powershell/module/?view=win10-ps
  - https://www.harmj0y.net/blog/powershell/powerup-a-usage-guide/
- Timezone Converter
  - https://www.timeanddate.com/worldclock/converter.html?iso=20170329T180000&p1=104&p2=136&p3=196
- Color cop
  - http://colorcop.net/
- Dencoder
  - http://meyerweb.com/eric/tools/dencoder/