What does exploratory testing even mean?
Why do we press harder on the remote when the batteries are dead?
Why does the dentist talk to you if you can’t respond?
These two things are basically the same!?
If you don’t understand modern art, it’s not your fault.
Everyone talks about it. Nobody really knows how to do it. Everyone thinks everyone else is doing it. So everyone claims they are doing it.

Exploratory testing is like teenage sex
Formal Testing
Informal Testing
Ad-hoc Testing
Exploratory Testing
Unstructured
Manual Testing
Structured
Test Automation
Formal Testing
Exploratory Testing

« Testing as performance »

Intelligent Testing
Create new test ideas based on what you have learned

Analyze Potential Risks
Focus the unknown based on options, assumptions, beliefs

High Information Value
Learn something new

Low Risk Coverage
Hard to scale continuously because of human reliance

Problem Detector

Formal Testing

« Testing as artifact creation »

Mechanical Testing
Process pre-defined data in pre-designed steps

Monitor Known Risks
Confirm what you already know (measureable things)

Low Information Value
Repeat what you have already learned

High Risk Coverage
Easy to scale because it’s parallelizable

Change Detector
The purpose of testing a program is to find problems in it.
Agile Testing Equation

Checked
Efficient Formal Testing

+ 

Explored
Effective Exploratory Testing

= 

Tested
Productive Testing
Your system under test.
Why are we **missing** these issues?

» What do we have to do to **proactively** find them? «

Your **system under test**.
I'm an automated test case.

Automation is doing, what automation does.
Risks

Automated Testing
The **narrower** the **view**, the **wider** the **ignorance**.

- The miss-the-forest-for-the-trees problem

What you are usually doing is hard to automate in any **meaningful way**.

- The unbalanced-effort-versus-value problem

Developer testing is like a **lawn service**. I could do it myself, I simply choose not to.

- The unwilling-tester problem

You don’t write automation you **should** write, you write the automation you **know how** to write.

- The line-of-the-least-resistance problem

Automated testing is not automated **manual testing**.

- The no-brain-in-the-loop problem

Automated Testing
I'm an automated test case.

Automation is doing what automation does.

Automated Testing
I'm the same **test case** executed **manually**.
Exploratory Branching

The art of deciding on what to test next

Exploratory Testing

...acts on itself, formal testing doesn't

Plan as you test!

Story-Based
Motivating
Credible

Test Cases vs. Scenarios
Pre-specified inputs vs. hypothetical situation

Exploratory Testing
Testing is blind without a mind.

You've got to focus on specific issues.

You've got to be well-prepared.

You have to bring knowledge.

You got to have a brain in the loop.
Visiting London

~100 Billion Billion Billion Billion Billion Billion Billion Billion Billion Billion Billion Combinations

\[ \approx 50 \]
Setting **goals** is the first step in turning the **invisible** into the **visible**.

Tony Robbins
Be involved in **every stage** of the data's life cycle.

A **tour** is an exploration of a product that is organized around a **theme**.
Think **superficially**! Whatever you do, **don't go beyond skin deep.**
Money Tour
Focus on money-generating features.

Landmark Tour
Focus on key feature combinations.

FedEx Tour
Track data through various stages.

Back Alley Tour
Focus on the least-used features.

All-Nighter Tour
Run scenarios over & over again.

The Saboteur
Sabotage system resources.

Intellectual Tour
Ask the hardest questions to test the limits.

Supermodel Tour
Take a single-minded focus on the UI.

Collector’s Tour
Generate & document all possible outputs.

Rained-Out Tour
Start, stop & cancel time-consuming tasks.

Museum Tour
Focus on legacy code in new environments.

Couch Potato Tour
Do as little as possible, process default values.

Guidebook Tour
Follow the user manual’s advice to its letter.

Antisocial Tour
Focus on least-likely & known bad inputs.

After-Hours Tour
Focus on batch functionality.
A distinctive **characteristic** possessed by someone

A **state** of being free from defects and/or deficiencies

The **degree** (measure) of excellence of something

**Quality**
What does it actually **mean**?
Quality is inherently **subjective**!

**Different stakeholders** will perceive the same product as having different levels of quality.

We must look for **different things**, for different stakeholders.

We must **diversify** testing!
Polychrome Testing
Testing from various perspectives
SIX THINKING HATS
EDWARD DE BONO
THE INTERNATIONAL BESTSELLER
that has changed the way the world's
most successful business leaders think
*WE GIVE OUR MINDS A SHAPED FOR CONTEXT" REMINDING US THAT
THINKING IS A SKILL AND CAN BE IMPROVED.—EDWARD DE BONO
REVISED AND UPDATED

www.debonogroup.com

Factual
Emotional
Critical
Positive
Creative
Control

Polychrome Testing
A detective searches for **clues**, for **evidence** and for **facts** that will help them to effectively solve the case they are working on.

It calls for **information** known or needed. It concentrates on **facts** and **figures**, and things we believe but don’t yet know. It’s about **questioning facts**, just the facts.

**Factual white hat thinking**
A heart is a very intuitive organ that senses subtle changes in feeling and emotion when circumstances present themselves. It signifies feelings, hunches and intuition. With this hat we express emotions, feelings and share fears, likes, dislikes, loves, passion, anger and hates.
It's all about judgment. With this hat we are devil's advocates, and spot the difficulties and dangers, where things might go wrong. Probably the most powerful of all hats but a problem if overused.

A Reaper is a mythical creature who brings death and destruction to the living.

Critical black hat thinking
A sun is bright, **happy** and **powerful**, helping to instill life into everything it touches.

It symbolizes sunshine, **optimism**. Under this hat you actively seeking the **positives** and probe for **value** and **benefit**. It's about **constructive** thinking and **encouraging**.

Positive **yellow hat** thinking
A seedling sprouts from the ground and grows persistently - expanding its leaves and branches in many unexpected directions.

It focuses on creativity, on possibilities, on fresh alternatives, and on new ideas without criticism. It's an opportunity to express new concepts and new perceptions. It's all about shaping ideas.

Creative green hat thinking
Control
The Movie Director

It manages the entire thinking process. It provides the focus, purpose & and controls the use of other hats. With this hat we think about our thinking.

Six Thinking Hats
Edward de Bono

The international bestseller that has changed the way the world’s most successful business leaders think.

www.debonogroup.com
1. **Session-Based Testing**
   Structure exploratory testing; make it plannable & applicable for larger teams.

2. **Requirements-Based Testing**
   Limit the scope; make it manageable for skilled & unskilled testers; then iterate.

3. **Tour-Based Testing**
   Set crystal-clear goals to turn the invisible into the visible.

4. **Polychrome Testing**
   Diversify your exploration; explore the product from different viewpoints.

5. **Scenario-Based Testing**
   Capture testing ideas in units of scenarios; translate your findings into a reviewable documentation.
Good testers are Hard to Fool

They are **cautious**, **curious** and **critical**

Great **testers** have the ability to...

... **pose** useful questions.
... **observe** what’s going on.
... **describe** what they perceive.
... **understand** what they observe.
... **interpret** what they find.
... **draw** the right conclusions fast.
... **think** critically about what they know.
... **know** they will never know everything.
... **keep** thinking despite already knowing.
... **focus** on things they don’t know.
... **target** specific issues without losing focus.
... **recognize** and manage bias.
... **form** and test conjectures.
... **scrutinize** illusions they are holding true.
... **analyze** someone else’s thinking.
... **reason** about cause and effect.
... **learn**.
1. What should be explored?
   Business Use Case, System Use Case, Epic, Theme, User Story, Technical Component, …

2. What are the core characteristics?
   Performance, Usability, Stability, Security, Safety, Reliability, Understandability, …

3. Which tour(s) are required?
   Supermodel Tour, FedEx Tour, Money Tour, Landmark Tour, </Custom Tour/>, …

4. Which hat(s) should be considered?
   Black Hat, White Hat, Red Hat, Green Hat, Yellow Hat, </Custom Hat/>

5. Who is required, and who takes which hat?

6. Who owns the session(s)?

7. How long should the session(s) last?

---

User Story

Exploratory Session

Sprint Start

Life-Cycle Integration

Sprint End
Kick-Ass Software Testing

**Pairing Sessions**
- Collaborate Closely
- Share Thoughts
- Extract Testing Recipe

**Split Sessions**
- Explore Separately
- Come Together
- Share Findings

**Blitz Testing**
- Invite People
- Timebox Session
- Elect Winner

**Bug Hunter**
- Select Bug Hunter
- Hunt Buck-Wild
- Share Results

Collaborate Closely
Share Thoughts
Extract Testing Recipe
Explore Separately
Come Together
Share Findings
Invite People
Timebox Session
Elect Winner
Select Bug Hunter
Hunt Buck-Wild
Share Results
Measuring Success

1. Mechanics
Learn the techniques, strengthen what you have learned, then improve.

2. Consistency
Success is the sum of small efforts repeated (day in, day out).

3. Trust
Be patient and trust even though you live in a fast, now, fast, now society.

4. Intensity
Intensity is born from consistency, so do it right then do it fast.
Awesome Results

Measuring Success

overshoot
undershoot
1. **Production Defects**
   How many defects sneak through testing into production? Did you find the right balance between exploratory and formal testing?

2. **Defect Severity**
   It’s not just about the number of defects! Did you find the defects that really matter? How well do you understand the product?

3. **Defect Variety**
   How many different types (e.g. performance, security) of defects did you detect and how does this quantity relate to the production defects?

4. **Mean-Time-To-Feedback**
   How fast (e.g. minutes, hours, days) did it take you to provide feedback to your development?

5. **Formal Testing Leak**
   How many defects that would have been missed by formal testing (e.g. automation) did you detect?
Exploratory testing is not so much a thing that you *do*, it’s far more a way you *think*.

Michael Bolton

*The eyes are useless, when the mind is blind.*
Questions

Because answers exist only to questions...