How to Measure Ad Hoc Testing

Jonathan Bach
Satisfice, Inc.
http://www.satisfice.com

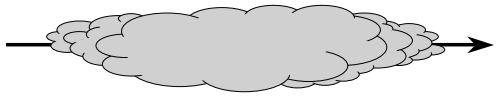
Ad hoc testing (AKA exploratory testing) relies on tester intuition. It is unscripted, unrehearsed, and improvisational.

But...

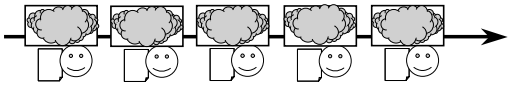
How do I, as test manager, understand what's happening, so I can direct the work and explain it to my clients?

One Solution: Test in Sessions

1) Charter
2) Time Box
3) Reviewable Result
4) Debriefing



VS.



Charter:
A clear mission for the session

- A charter may suggest what should be tested, how it should be tested, and what problems to look for.
- A charter is not meant to be a detailed plan.
- General charters may be necessary at first:
  - *"Analyze the Insert Picture function"*
- Specific charters provide better focus, but take more effort to design:
  - *"Test clip art insertion. Focus on stress and flow techniques, and make sure to insert into a variety of documents. We're concerned about resource leaks or anything else that might degrade performance over time."*

Time Box:
Focused test effort of fixed duration

Short: 60 minutes (+-15)
Normal: 90 minutes (+-15)
Long: 120 minutes (+-15)

- *Brief enough for accurate reporting.*
- *Brief enough to allow flexible scheduling.*
- *Brief enough to allow course correction.*
- *Long enough to get solid testing done.*
- *Long enough for efficient debriefings.*
- *Beware of overly precise timing.*

Debriefing:
Measurement begins with observation

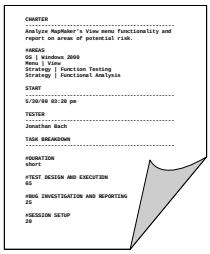
- The manager reviews ***session sheet*** to assure that he understands it and that it follows the protocol.
- The tester answers any questions.
- Session metrics are checked.
- Charter may be adjusted.
- Session may be extended.
- New sessions may be chartered.
- Coaching happens.

Agenda: "PROOF"

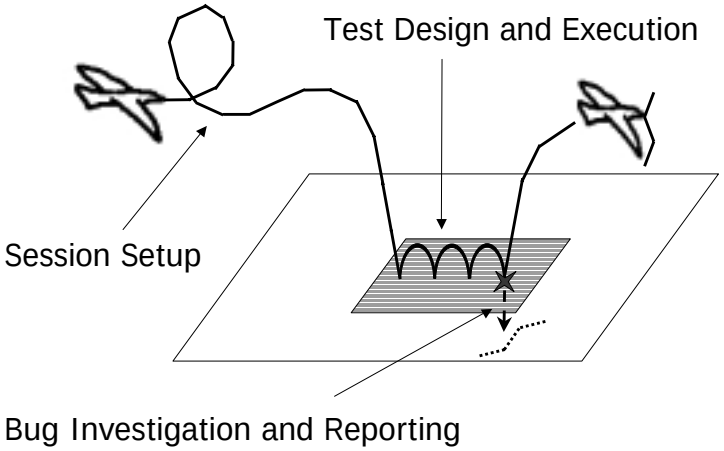
1. Past
2. Results
3. Obstacles
4. Outlook
5. Feelings

Reviewable Result:
A scannable session sheet

- Charter
  - *#AREAS*
- Start Time
- Tester Name(s)
- Breakdown
  - *#DURATION*
  - *#TEST DESIGN AND EXECUTION*
  - *#BUG INVESTIGATION AND REPORTING*
  - *#SESSION SETUP*
  - *#CHARTER/OPPORTUNITY*
- Data Files
- Test Notes
- Bugs
  - *#BUG*
- Issues
  - *#ISSUE*



The Breakdown Metrics
Testing is like looking for worms



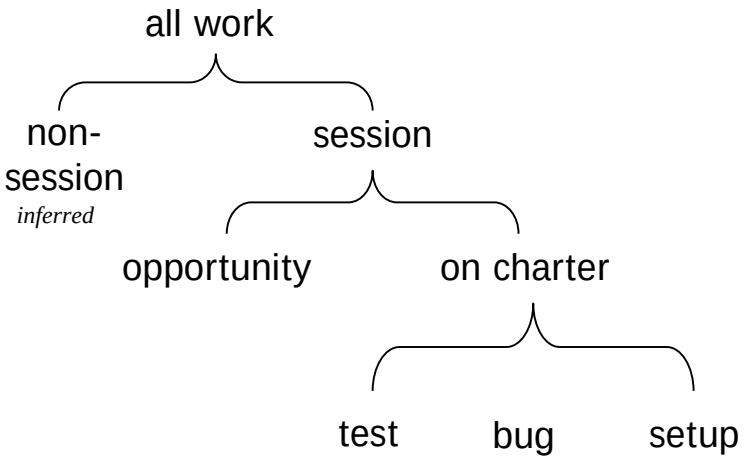
Reporting the TBS Breakdown

A guess is okay, but follow the protocol

- Test, Bug, and Setup are orthogonal categories.
- Estimate the percentage of charter work that fell into each category.
- Nearest 5% or 10% is good enough.
- If activities are done simultaneously, report the highest precedence activity.
- Precedence goes in order: T, B, then S.
- All we really want is to track interruptions to testing.
- Don't include Opportunity Testing in the estimate.

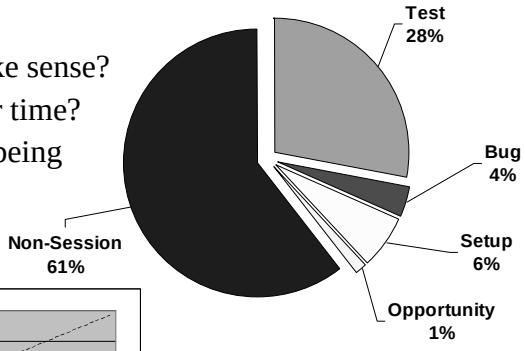
Activity Hierarchy

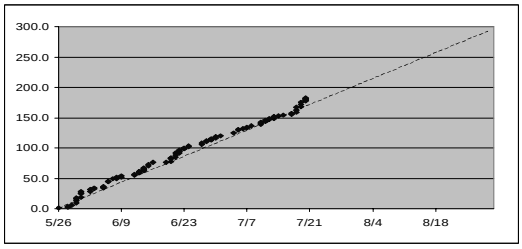
All test work fits here, somewhere



Work Breakdown:
Diagnosing the productivity

- Do these proportions make sense?
- How do they change over time?
- Is the reporting protocol being followed?





Coverage:
Specifying coverage areas

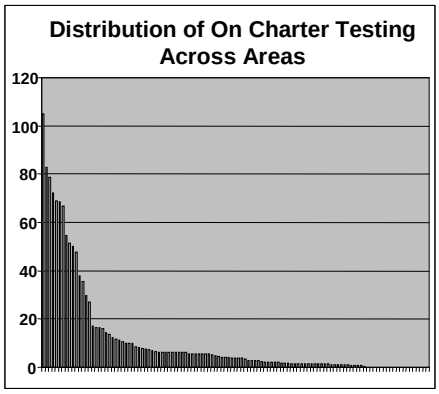
- These are text labels listed in the Charter section of the session sheet. (e.g. “insert picture”)
- Coverage areas can include anything
  - *areas of the product*
  - *test configuration*
  - *test strategies*
  - *system configuration parameters*
- Use the debriefings to check the validity of the specified coverage areas.

Coverage:
Are we testing the right stuff?

- Is this a risk-based test strategy?

or

- Is it a lop-sided set of coverage areas?
- Is it distorted reporting?



Using the Data to Estimate a Test Cycle

1. How many perfect sessions (100% on-charter testing) does it take to do a cycle? *(let's say 40)*
2. How many sessions can the team (of 4 testers) do per day? *(let's say 3 per day, per tester = 12)*
3. How productive are the sessions? *(let's say 66% is on-charter test design and execution)*
4. Estimate: 40 / (12 * .66) = **5 days**
5. We base the estimate on the data we've collected. When any conditions or assumptions behind this estimate change, we will update the estimate.