

How To Get What You Want From Testing

A Personal View on How Testing Can Help CTOs

Michael Bolton
DevelopSense
<http://www.developsense.com>
@michaelbolton
michael@developsense.com

James Bach
Satisfice
<http://www.satisfice.com>
@jamesmarcusbach
james@satisfice.com

How To Get What You Want (CTOs) - 1

I'm Michael Bolton



How To Get What You Want (CTOs) - 2



What Do I Do?

- I help people to solve testing problems they didn't know they could solve, and I teach them how they can do that themselves.
- I teach Rapid Software Testing
 - <http://www.developsense.com>; <http://www.rapid-software-testing.com>
- I'm focused on advancing the craft of software testing, and its value to organizations
- And I need *your* help!

How To Get What You Want (CTOs) - 5

What are YOU doing?

Directing people
Acquiring resources
Applying judgement
Making decisions
Gathering information
Massaging egos
Weighing opinions
Balancing budgets
Co-ordinating work
Dealing with emotions
Handling problems
Delegating



How To Get What You Want (CTOs) - 6

What Keeps CTOs Awake at Night?

1. Being seen as being behind the curve.
2. Getting on the front page of Report on Business
(for bad stuff)
3. Losing money.

How To Get What You Want (CTOs) - 7



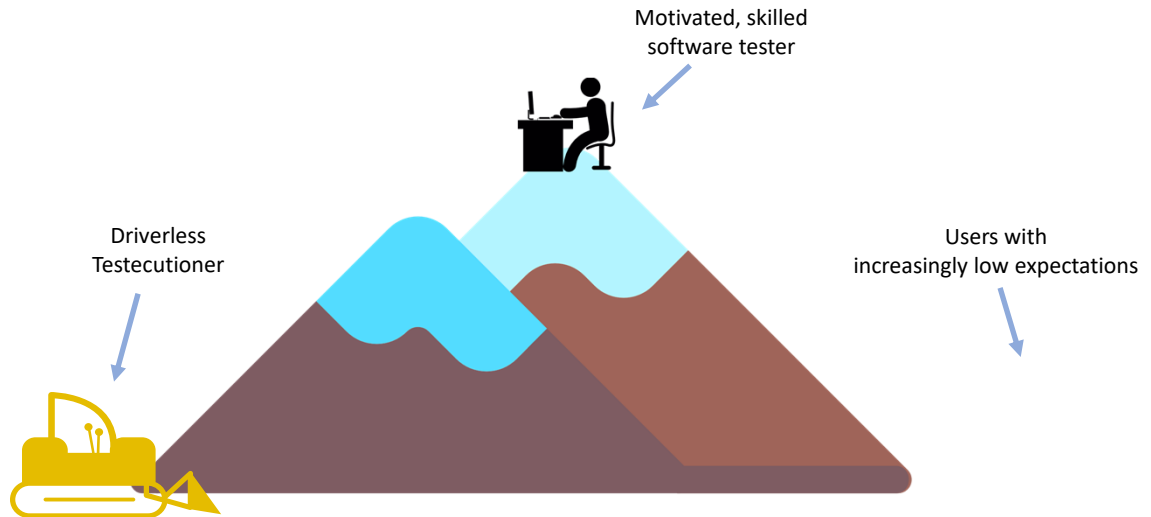
I think How to Get *everybody in my opinion should* ~~What You~~ Really Want from Testing (CTOs' Edition)

Michael Bolton
DevelopSense
<http://www.developsense.com>
@michaelbolton
michael@developsense.com

James Bach
Satisfice
<http://www.satisfice.com>
@jamesmarcusbach
james@satisfice.com

How To Get What You Want (CTOs) - 8

Demand for skilled testing is eroding.



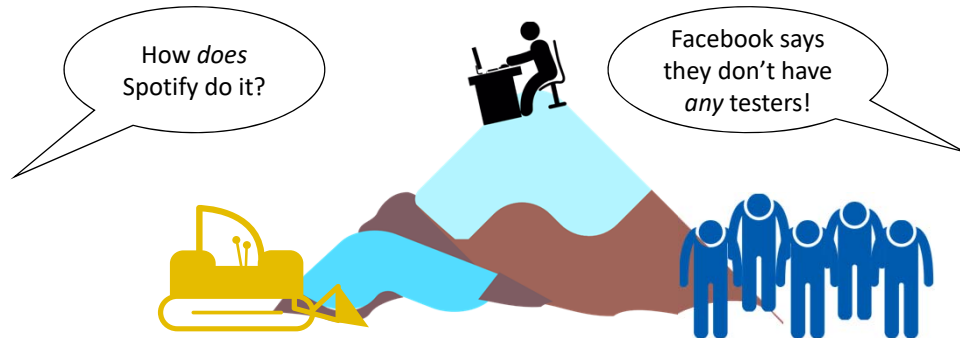
How To Get What You Want (CTOs) - 9

Demand for skilled testing is eroding: shiny "automation" on one side, and low quality standards on the other.



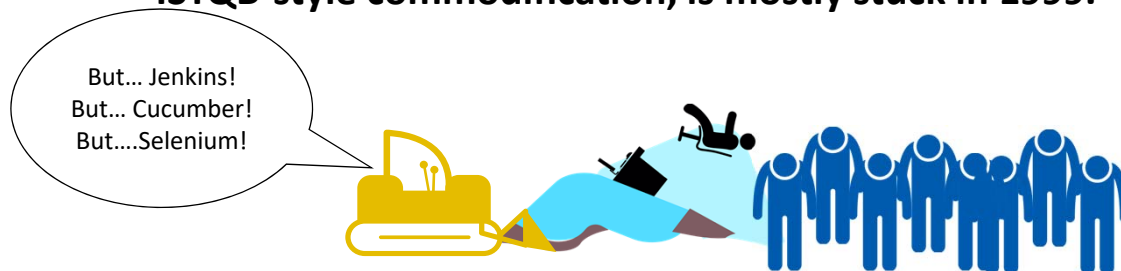
How To Get What You Want (CTOs) - 10

Demand for skilled testing is eroding
shiny “automation” on one side,
and low quality standards on the other.
**The software industry has become dazzled by
values and practices appropriate for
experimental, non-critical software.**



How To Get What You Want (CTOs) - 11

Demand for skilled testing is eroding:
shiny “automation” on one side,
and low quality standards on the other.
The software industry has become dazzled by
values and practices appropriate for
experimental, non-critical software.
**Meanwhile, testing culture, already weakened by
ISTQB-style commodification, is mostly stuck in 1999.**



How To Get What You Want (CTOs) - 12

**Big Problem:
Managers, developers, and executives
don't understand testing.**

**Big Problem:
Testers don't know how to talk about testing.**

**Bigger Problem:
Most testers don't really understand testing either.**

(This is a great way to make myself popular, isn't it?)

Why I'm becoming a grumpy old guy:

Increasingly, testing is confused with “checking builds”.

Our fixation on “test automation” is causing us to lose connection with the human, social purposes of software development and testing.

Tools are great. **We should use them.** We should use them **a lot** to help us develop an understanding of our products.

Tools can help us to be powerful.

But what I'm seeing at conferences and in talk about testing often looks like elaborate attempts to **avoid making contact** with the software, our clients, our customers, and our mission.

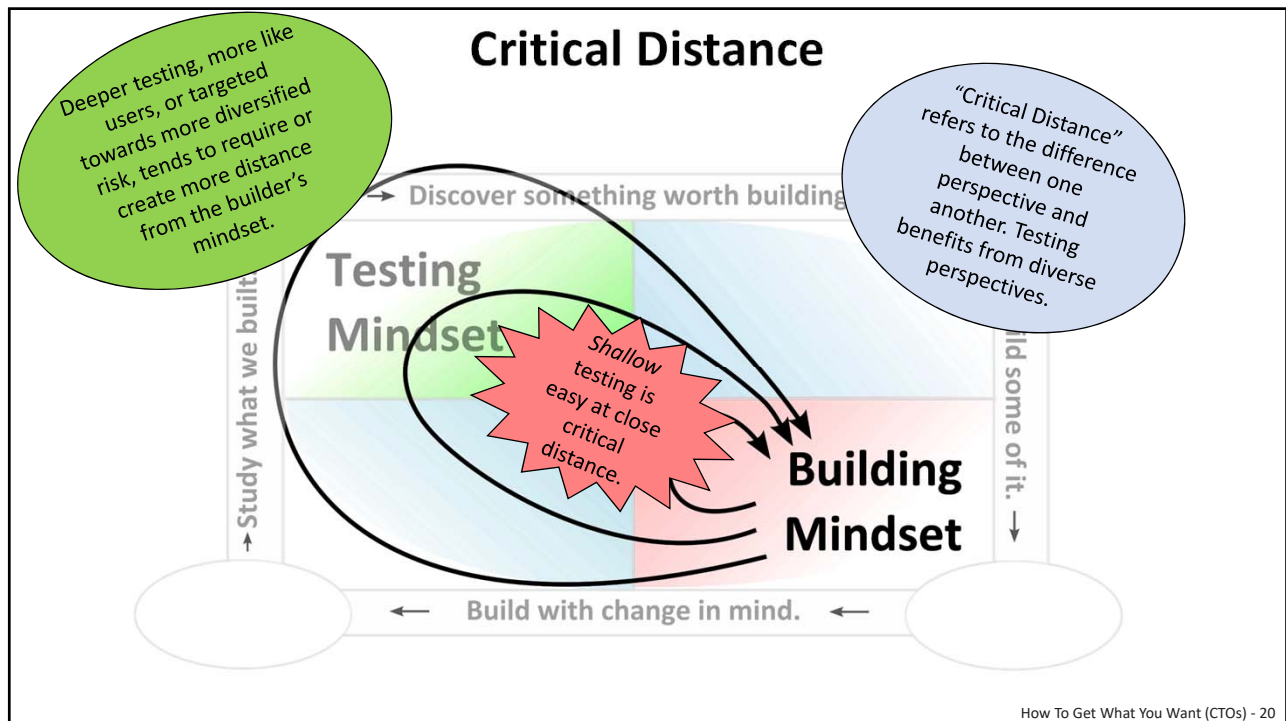
How To Get What You Want (CTOs) - 17

**You can't release a product without developers,
but you *could* release one without testers.**

How To Get What You Want (CTOs) - 18

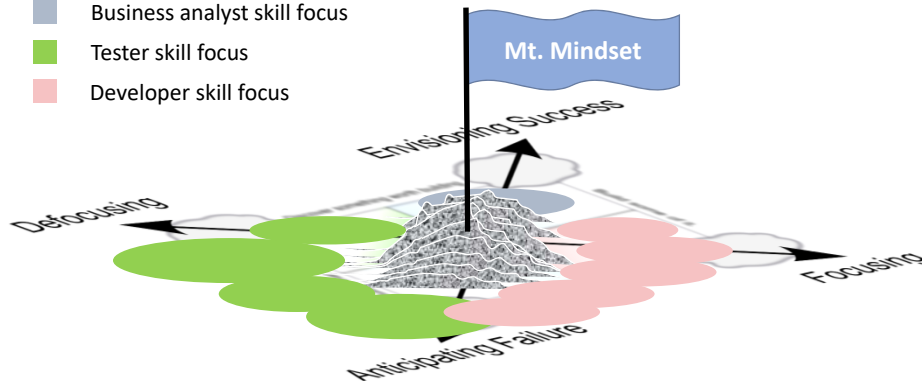
**So why have testers?
Because management wants
an expert answer to this question:**

Are there problems
that threaten the value of the product,
or the on-time successful completion of the project?



Why roles? Because changing mindsets is HARD.

- Business analyst skill focus
- Tester skill focus
- Developer skill focus



NOTE: We do NOT claim that different kinds of work *must* be done by different people, or that the people *must* have different titles.

We DO claim that having skilled people *focused* on testing is a powerful heuristic for addressing the mindset switching problem.

How To Get What You Want (CTOs) - 21

Problem: Misbegotten Ideas about Programs

~~A program is a set of instructions for a computer.~~

A program is a communication among people, mediated by hardware and software.

(Credit: Cem Kaner, Software Testing as a Social Science)

How To Get What You Want (CTOs) - 22

What Testing IS

- “Gathering information in order to inform a decision” (Weinberg)
- “An empirical, technical investigation of software, done on behalf of stakeholders, with the intention of revealing quality-related information of the kind that they seek” (Kaner)
- Applied critical thinking... “thinking about thinking with the intention of avoiding being fooled.” (Bach and Bolton)
- (but there’s more, later...)

How To Get What You Want (CTOs) - 23

What Testing IS NOT



Confirmation



Demonstration



“Breaking the product”

How To Get What You Want (CTOs) - 24

Testing is not Quality Assurance

“QA” = Questions + Answers

- Testing does not *assure quality*
 - *YOU, dear executive, do that!*
- Testing does not *improve quality*
 - *unless someone changes something, quality stays the same*
- Testing *informs decisions* about quality
- Testing *raises questions*
 - “Is there a problem here?”
 - “Is everyone OK with this?”
- Testing *gets answers*
 - but not *complete* answers
 - “partial answers that might be useful” (Cem Kaner)



How To Get What You Want (CTOs) - 25

Testing Is *Social Science*



“Computers and their software are two things. As collections of interacting cogs they must be ‘checked’ to make sure there are no missing teeth and the wheels spin together nicely.

Machines are also ‘social prostheses’, fitting into social life where a human once fitted. It is a characteristic of medical prostheses, like replacement hearts, that they do not do exactly the same job as the thing they replace; the surrounding body compensates.

Harry Collins, Abstract, “Machines as Social Prostheses”, EuroSTAR 2013

How To Get What You Want (CTOs) - 26

Testing Is *Social Science*



“Contemporary computers cannot do just the same thing as humans because they do not fit into society as humans do, so the surrounding society must compensate for the way the computer fails to reproduce what it replaces.

This means that a complex judgment is needed to test whether software fits well enough for the surrounding humans to happily ‘repair’ the differences between humans and machines. This is much more than a matter of deciding whether the cogs spin right.”

Harry Collins, Abstract, “Machines as Social Prostheses”, EuroSTAR 2013

How To Get What You Want (CTOs) - 27

In other words...

Is the product we’ve got
good enough
for people to be happy with it?

How To Get What You Want (CTOs) - 28

Problem: Misbegotten Ideas about Quality

~~Quality is conformance
to requirements.~~

Quality is
value to some person(s).

(Credit: Jerry Weinberg, *Quality Software Management*)

How To Get What You Want (CTOs) - 29

Reification (reducing ideas to things)

~~Requirements
=
the requirements
document!~~

Some of the requirements are
described, *to some degree*,
in the requirements
document.

How To Get What You Want (CTOs) - 30

Problem: Mistaken Missions

Testing is about
quality control, or
quality assurance.
Testers are quality
gatekeepers.

Testers can't assure quality.
The conflict between
what we're *called*
and what we *do*
causes confusion and pain.

Problem: Mistaken Missions

Testing is about
quality control, or
quality assurance.
Testers are quality
gatekeepers.

Testing is about
exploring and investigating
risk.

Mistaken Models of Testing

~~Testing is about preventing problems.~~

Development is about preventing problems.
Testing is about *discovering problems development didn't prevent.*

Mistaken Models of Testing

~~Testing is about preventing problems.~~

Early in development, testers can help by *anticipating risks and problems* in a way that *helps* developers to prevent them.

Shallow Models of Risk

There ~~might be an error in~~
the code!

There can be *many problems*
in the relationships between
people and the product.

How To Get What You Want (CTOs) - 35

Risk Story Elements

- Some PERSON(S)
 - user, customer, developer, tester, businessperson, bystander...
 - (a group, a business, a community, society at large...)
- will EXPERIENCE
 - be affected, in the context of an event or situation, at least once by ...
- a PROBLEM
 - that leads to bad feelings (annoyance, frustration, confusion), loss, harm, or diminished value...
- with respect to SOMETHING DESIRABLE
 - like capability, reliability, performance...
- that CAN BE DETECTED
 - by a feeling, principle, tool, comparison to a document or picture...
- in SOME SET OF CONDITIONS
 - perhaps always, perhaps only sometimes,...
- because of a VULNERABILITY
 - a bug, a missing feature, an inconsistency...
- in the SYSTEM
 - some result, process, component, feature, environment...

Stakeholders

Context

Problem

Quality Criteria

Oracles

Test Conditions

Theory of Error

Product Factors

How To Get What You Want (CTOs) - 36

Problem: Mistaken Missions

~~Testing is showing that
the product works.~~

Testing is learning about
the product; searching for
problems; finding them;
reporting on them.

Problem: Mistaken Missions

~~Testing is about
confirming that the
product works.~~

Testing is about discovering
how the product
doesn't work, or
might not work.

Problem: Mistaken Missions

~~Testing is about
confirmation.~~

Testing is about
investigation.

Problem: Mistaken Missions

~~Testing is about
building confidence.~~

Testing is about
demolishing
unwarranted confidence.

Mistaken Missions

~~Testing is about
reducing damaging
uncertainty.~~

Testing is about
*preserving appropriate
skepticism.*

Problem: Mistaken Models of Testing

~~Testing is all about the
button-pushing, which can
be done more quickly by
machinery.~~

Testing is about *learning*,
which can only be done by
humans with intentions.
**But tools can be powerful
aids to testing.**

Call This “Checking”, Not Testing

operating a product
algorithmically to check
specific facts about it...

Think
“spelling
checker”

means

Observe

Evaluate

Report

Interact with the product in
specific, *algorithmic* ways to
collect specific observations.

Apply *algorithmic*
decision rules to those
observations.

Report the outputs of
the evaluations
algorithmically.

How To Get What You Want (CTOs) - 43

A check can be performed...



by a machine that *can't* think
(but that is quick and precise)



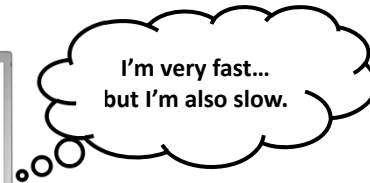
by a human who has been told *not* to think
(and who is slow and variable)

Notice that “quick” and “slow” here refer only to the speed of
observable behaviours and algorithmic evaluations.
The machine is *infinitely* slow at recognizing unanticipated trouble.

How To Get What You Want (CTOs) - 44

Testing Is *More Than* Checking

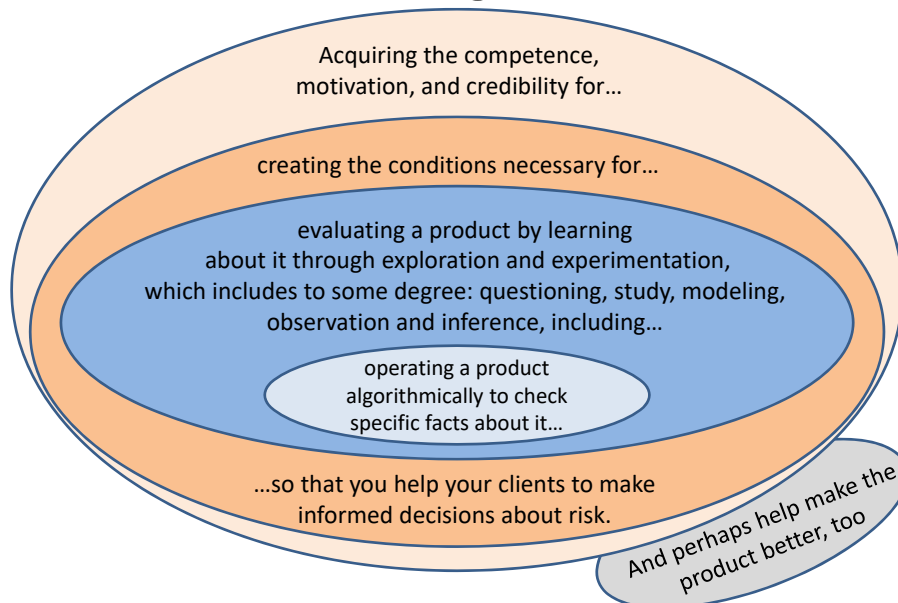
- *Checking* is okay, but it is mostly focused on confirming what we know or hope to be true.
- To understand our products and the risk of problems that matter to people, we must do more than output checking; we must *test*.



See <http://www.developsense.com/2009/08/testing-vs-checking.html>

How To Get What You Want (CTOs) - 45

Testing is...



Why it's important to distinguish testing and checking

- Because *checking* is mechanistic. It can be made completely **explicit** and automated. It is *inside* testing. It is a *tactic* of testing.



People don't
confuse biting
with eating!

Biting can be
done by
tools... but
eating can't.



How To Get What You Want (CTOs) - 47

Why it's important to distinguish testing and checking

- Because *checking* is mechanistic. It can be made completely **explicit**, encoded, and automated. It is *inside* testing. It is a *tactic* of testing.
- Because *testing* involves **tacit** and **social skills** that cannot be encoded. Testing skills and must be developed through socialization, practice, and increasingly challenging work, not via rote procedures.
- Because talk about efficiency and effectiveness makes for very different conversations when we're talking about explicit vs. tacit skills.
- Because for checking to be *truly* excellent, it must be embedded in excellent testing. Developing valuable checks requires skill!
- Programmers have resisted marginalization for years!
(They no longer call compilers "autocoders" and programming languages are no longer called "autocodes".)

How To Get What You Want (CTOs) - 48

Problem: Unhelpful Categories

~~“Manual testing”~~

Testing is
neither manual
nor automated.

Testing is neither manual nor automated!



~~Manual~~ Doctoring



~~Manual~~ Parenting



~~Manual~~ Research



~~Manual~~ Management

What About AI?!

~~AI will replace testers!~~

Nonsense. AI is only
“algorithm improvement”;
sophisticated software.
We’ll need to test the
daylights out of it.

How To Get What You Want (CTOs) - 53

Problem: Testing Reduced to Test Cases

~~“We write test cases.
Passing test cases show
the product is good.
Failing ones show the
product is bad.”~~

Testing is NOT
about test cases.

How To Get What You Want (CTOs) - 54

Testing is not test cases!



Piloting Cases



Parenting Cases



Research Cases



Management Cases

How To Get What You Want (CTOs) - 55

Why to resist framing testing as test cases:

- Testing is about exploration, experimentation, discovery, investigation, learning, and reporting.
- *Test cases* tend to focus on output checking, confirmation and demonstration; showing that something *works*, rather than prompting a search for problems that matter.
- When people turn testing into test cases, they start counting them.
- When people turn testing into *counting*, the information loss, dysfunction and distortion begins.

How To Get What You Want (CTOs) - 56

Reification (reducing ideas to things) (again)

Testing is all about
test cases.

A test is not an artifact.
A test is *a performance*.

How To Get What You Want (CTOs) - 57

**Testing is not about correctness, or
passing or failing test cases.
Skilled testers focus on two questions.**

To themselves, referring to
the product:

Is there a problem here?

To the team and to
management, referring to
issues and obstacles:

Are you okay with this?

How To Get What You Want (CTOs) - 58

Problem: Failure to Tell the Testing Story

~~The testing story is
all about bugs.~~

The testing story has three parts: the status of the product; how testing has been done; and what makes testing harder and slower.

How To Get What You Want (CTOs) - 59

The Testing Story Is Three Braided Stories

A story about the status of the PRODUCT...

...about what it does, how it failed, and how it might fail...
...in ways that matter to your various clients.

A story about HOW YOU TESTED...

...how you operated and observed the product...
...how you recognized problems and their significance...
...what you have testing so far *and have not tested yet*...
...what you won't test at all (unless things change).

A story about how GOOD that testing was, or could be...

...the risks and costs of testing or not testing...
...how testable (or not) the product is...
...what made testing harder or slower...
...what you need and recommend for faster, higher-value testing.

Bugs

Oracles

Coverage

Issues

Image credit: iStockphoto.com

How To Get What You Want (CTOs) - 60

The “Measuring Quality” Trap

How should we measure
quality?

You can measure attributes
that might have a bearing on
quality, but you can't
measure quality itself.

How To Get What You Want (CTOs) - 61

The “Measuring Quality” Trap

How should we measure
quality?

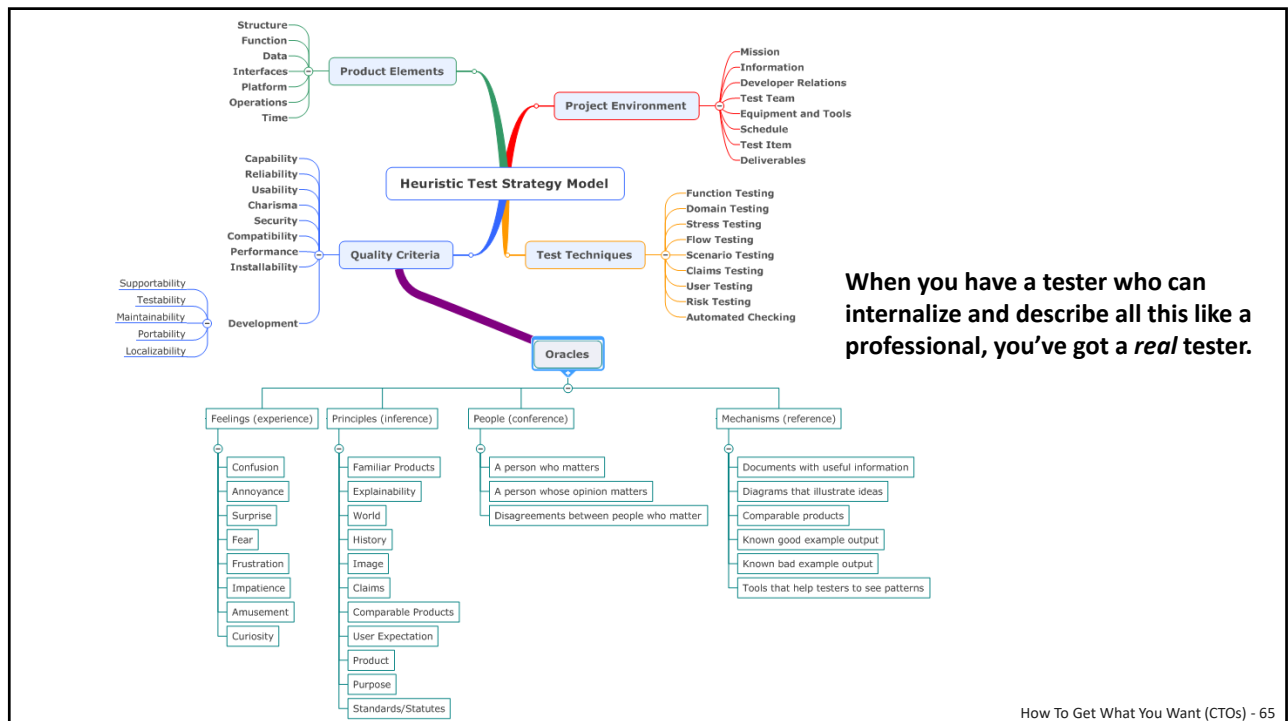
You can't measure quality.
But you can report on it, and
you can discuss it.

How To Get What You Want (CTOs) - 62

Problem: Shallow Test Strategy

~~“We read the specs, and then we write test cases.”~~

Excellent test strategy requires rich models of context, quality criteria, product factors, oracles, and test techniques.

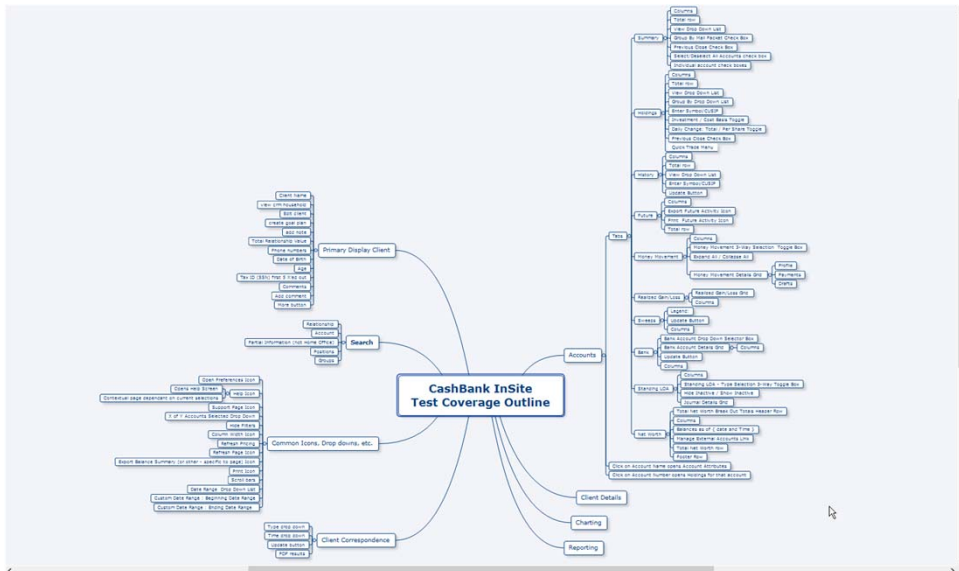


Alternatives to Test Cases: Tables

The screenshot shows an Excel spreadsheet with a test case table. The table is organized into several sections:

- Test Case Master:** Includes fields for Client ID (101), Client or non-client? (Client), Client (72284841), Entitlement Code (A-014), Name, and Mark Number.
- Scenario Description:** A large empty box for describing the test scenario.
- Test Objective:** A large empty box for defining the test objective.
- Expectation:** A large empty box for defining the expected results.
- Transaction Tables:** A large table with columns for Transaction, Date, Rate, Transaction Details, Amount, Deal HD, Profit, Loss, Break Even, Profit Amount, Spread Rate, PWR Table, FIX, and DFC. It includes a 'BUY TABLE' and a 'SELL TABLE' with various currency and instrument details.
- General Ledger:** A summary table at the bottom showing Total Planned, Buy Table, Sell Table, and Servlet Results.

Alternatives to Test Cases: Coverage Outlines



Alternatives to Test Cases: Open Procedures

3.2.2 Fields and Screens

- 3.2.2.1 Start the Zapper Box and the Control Box. (Vary the order and timing, retain the log files, and note any inconsistent or unexpected behaviour.)
- 3.2.2.2 Visually inspect the displays on each box and **VERIFY** conformance to the requirements specifications. Remain alert for the presence of any behaviour or attribute that could mislead or confuse the operator, or impair the performance or safety of the product in any material way.
- 3.2.2.3 With the system settings at *default* values, change the contents of every user-editable field through the range of all possible values for that field. (e.g. Use the knob to change the session duration from 1 to 300 seconds.) Visually **VERIFY** that appropriate values appear and that everything that happens on the screen appears normal and acceptable.
- 3.2.2.4 Repeat 3.2.2.3 with system settings changed to their *most extreme* possible values.
- 3.2.2.5 Select at least one field and use the on-screen keyboard, knob, and external keyboard respectively to edit that field.

How To Get What You Want (CTOs) - 68

Alternatives to Test Cases: More Specific Procedures

- 3.5.2.3 From the power meter log file, extract the data for the measured electrode. This sample should comprise the entire power session, including cooldown, as well as the stable power period with at least 50 measurements (10 seconds of stable period data).
- 3.5.2.4 From the Control Box log file, extract the corresponding data for the stable power period of the measured electrode.
- 3.5.2.5 Calculate the deviation by subtracting the Control Box's reported power for the measured electrode from the corresponding power meter reading (use interpolation to synchronize the time stamp of the power meter and Control Box logs).
- 3.5.2.6 Calculate the mean of the power sample \bar{X} (bar) and its standard deviation (s).
- 3.5.2.7 Find the 99% confidence and 99% two-sided tolerance interval k for the sample. (Use Table 5 of NIST* SOP-QAD-10, or use the equation below for large samples.)
- 3.5.2.8 The equation for calculating the tolerance interval k is:
$$k = \sqrt{\frac{(N-1) \left(1 + \frac{1}{N}\right) Z_{(1-p)/2}^2}{\chi_{\gamma, N-1}^2}}$$

where $\chi_{\gamma, N-1}^2$ is the critical value of the chi-square distribution with degrees of freedom, N-1, that is exceeded with probability γ and $Z_{(1-p)/2}$ is the critical value of the normal distribution which is exceeded with probability $(1-p)/2$.

* See NIST Engineering Statistics Handbook.

How To Get What You Want (CTOs) - 69

Alternatives to Test Cases: Concise Learning Charters

- ...for Intake Sessions (Goal: [negotiate mission](#))
 - “Interview the project manager. Ask about particular concerns or risks.”
 - “Read through all new use cases, and discuss with developers.”
- ...for Survey Sessions (Goal: [learn product](#))
 - “Familiarize yourself with the product by performing a UI tour. Create a Product Coverage Outline.”
- ...for Setup Sessions (Goal: [create testing infrastructure](#))
 - “Develop a library of mindmaps for each major feature area. Use SFDIPOT as a checklist for coverage analysis.”
 - “Identify and list all the error messages in the product.”
 - “Develop a scenario playbook with SMEs and other testers.”
 - “Review use cases, and for each, add several ways in which the user could accidentally or maliciously misuse the feature.”

How To Get What You Want (CTOs) - 70

Alternatives to Test Cases: Concise Testing Charters

- ...for Deep Coverage Sessions (Goal: [find the right bugs](#))
 - “Perform scenario testing based on the scenario playbook.”
 - “Run state-machine-based tours to achieve double-transition state coverage. Find possibilities for programmed checks.”
 - “Perform steeplechase boundary testing on major data items.”
 - “Help developers to set up automated checks for the continuous integration pipeline.”
 - “Generate each identified error message in the product. Look for mismanaged state and error recovery problems, confusing or unhelpful user messages, and missing error codes.”
 - “Develop scripts (working below the GUI) to run transactions continuously and graph results and timings. Make sure many transactions (15%? like production logs?) include invalid data that should be handled and rejected.”

How To Get What You Want (CTOs) - 71

Alternatives to Test Cases: More Formalized Charters

PROCHAIN ENTERPRISE

SCENARIO TEST CHARTER

UP2: “Check status and perform buffer update”

| | |
|-------------------|--|
| Theme | You are a project manager. You need to update your project to prepare your weekly report on project status. |
| Setup | <ul style="list-style-type: none">- Log in with a user account set up with project manager rights.- Buffer consumption for one of the projects should ideally be in the yellow or red.- At least some of the projects should have multiple project buffers. |
| Activities | <ul style="list-style-type: none"><input type="checkbox"/> View the Standard Projects Status Chart (or custom chart), filter on a set of projects (and turn on name labels). Start a second session in a window next to the first one (log in as the same user), and filter for the same project set. Now you have two project status charts that you can compare.<input type="checkbox"/> Pick one project as “yours”. Now, compare status history of your project to others. Explore the other project details in any way necessary to account for the <i>differences</i> in status.<input type="checkbox"/> View all impact chains for your project, and for some of those tasks:<ul style="list-style-type: none">- view task details- view task links- view task load chart<input type="checkbox"/> If other testers are making task updates during your test session, review those changes and modify some of them, yourself. Otherwise, make at least a few updates of your own. |

How To Get What You Want (CTOs) - 72

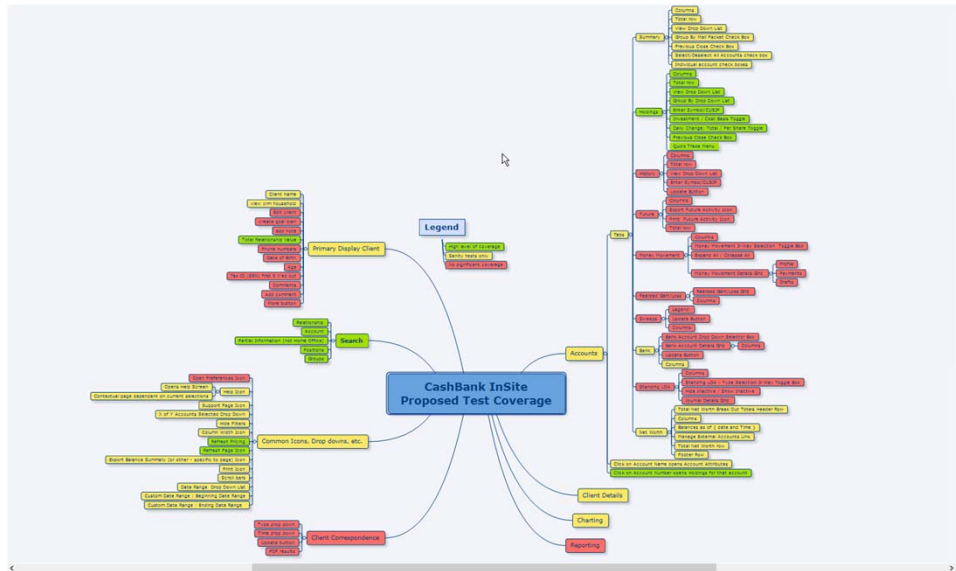
Problem: Failure to report on coverage.

The ~~testing story~~ is about
how many test cases
we have run.

The testing story is about the
what we have covered so far
and
what else could be covered.

How To Get What You Want (CTOs) - 73

Alternatives to Test-Case-Based Reporting



How To Get What You Want (CTOs) - 74

Alternatives to Test-Case-Based Reporting:

Session-Based Test Management Debriefs: **PROOF!**

- **P**ast
 - What happened during the session?
- **R**esults
 - What was achieved? What was covered?
- **O**bstacles
 - What got in the way or slowed things down?
- **O**utlook
 - What's next? What remains to be done?
- **F**eelings
 - How does the tester feel about all this?

Good lab notes are essential!

Logs and logging tools help too!

How To Get What You Want (CTOs) - 75

Alternatives to Test-Case-Based Reporting

Rapid Testing Status
Updated: 05/30 16:30:57
Sessions: 13 (9 reports)
Bugs: 32
[View Completed Session Reports](#)
[View Test Coverage](#)

Y2K Compliance Report
IPAM 6.0

Spot Check Test Report
Prepared by James Bach, Principal Consultant, Softfix, Inc. 8/24/11
1. Overview
This report describes one day of a paired exploratory survey of the Multi-Phase Investigator and Workstation. This testing was intended to provide a spot check of the formal testing already routinely performed on this project. The form of testing we used is routinely applied in court proceedings and occasionally by 3rd-party auditors for this purpose.
Check: we found that there are important vulnerabilities in the test. Look at the test cases.

Game Film Review
BLACK TEXT: What I did
BLUE TEXT: How the system responds
GREEN TEXT: What I was thinking
Charter: "Plunge in and quit" test cycle to investigate an apparent problem in Notepad
Strategy: (Testing) View the problem

How To Get What You Want (CTOs) - 76

Problem: Regression Obsession

~~"But we have to run all the tests after each build!"~~

Running all the tests after each build is probably not a well-considered, risk-focused test strategy.

Consistent Regression Problems Are *Symptoms of Trouble*



- If you see a consistent pattern of regression
 - failing checks or tests probably aren't your biggest problem
 - more likely, the issue is that you've got favourable conditions for regression to happen
 - testing cannot fix this problem; at best testing can only *detect some* regression bugs
 - the programmers are probably working too quickly to understand what's happening

THAT'S A SEVERITY-ONE ISSUE. REPORT IT.

How To Get What You Want (CTOs) - 78

Limited View of Tools

~~Tools are just for
GUI checking, right?~~

Producing test data; obfuscating or cleaning
production data for privacy reasons;
generating interesting combinations of
inputs; generating and representing state
and flow models...

How To Get What You Want (CTOs) - 79

Limited View of Tools

Tools are just for
GUI checking, right?

Setup, configuration, and environment management; submitting transactions; automated checking; creating mocks and simulators; probing the internals; monitoring at the interfaces...

Limited View of Tools

Tools are just for
GUI checking, right?

Sorting, filtering and parsing; visualizing; internal consistency checks; applying oracles; performing statistical analysis...

Limited View of Tools

~~Tools are just for
GUI checking, right?~~

Recording activities; documenting procedures; preparing reports; presenting reports; Mapping strategies; identifying coverage; organizing time and effort; retaining information...

How To Get What You Want (CTOs) - 82

Testers don't decide when testing is done. YOU do.

~~When will the testing
be done!~~

Testing is done when managers and developers are satisfied that there is no more important development work to be done.

How To Get What You Want (CTOs) - 83

Testers don't decide when testing is done. YOU do.

~~When will the testing be done!~~

Testing is never *done*; it only *stops*.
Testing stops when managers decide they can make **their** informed shipping decision.

How To Get What You Want (CTOs) - 84

Not-So-Good Questions for Testers

- Is the product done?
- Are we ready to ship?
- Is it good enough?
 - All three of these are *your* decision, Dear Project Manager.
- How much time do you need to test?
 - This is like asking “How much time do you need to learn about the product?”
- How many tests cases have you run?
- How many test cases are passing and failing?
- How many bugs are in the product?
 - These numbers don't mean anything without a story — and once you have the story, the numbers probably aren't important.

How To Get What You Want (CTOs) - 85

Better Questions for Testers

- What is the product story? What can you tell me about important problems in the product?
- What have you done to obtain the product story?
- What risks I should be aware of?
- What important testing remains to be done?
- What problems are slowing testing down or making it harder to find out what we might need to know?
- What help do you need to speed things up?
- What *specific* aspects of testing are taking time?
- How does your tests link what I need to know?

How To Get What You Want (CTOs) - 86

Talking More Clearly About Testing

Try replacing...

Verify that...

Validate

Confirm that...

Show that it works

Pass vs. fail...

Test case

Counting test cases

Automated testing

Test automation

Use cases

KPIs and KLOCs

with...

Challenge the belief that...

Investigate

Find problems with...

Discover where it *doesn't* work

Is there a problem here?

Test conditions and test ideas

Describing coverage

Programmed checking

Using tools in powerful ways

Use cases AND *misuse* cases AND *abuse* cases AND *obtuse* cases...

Learning from every bug

How To Get What You Want (CTOs) - 87

Talking More Clearly About Testing

Try replacing...

“The environment’s down. We’re stuck.
We can’t test.”

“They didn’t give us good requirements
documents!”

“It’s too hard to test this!”

“We don’ t have enough time to test!”

“We have to...!”

with...

“What can we test, review, or analyze now...
and *are you OK with this situation*, dear
client?”

“Let’s write down what *we* know—and then
they’ll tell us when they think it’s wrong.”

“What can we do in the product and the
project to things more testable?”

“What testing shall we do—what shall we
cover—in the time we *do* have?”

“We choose to...”

Postscript

Problem: Speaking Imprecisely

“flaky tests”

The checks aren't flaky.
But explanations about
inconsistency might be.

Problem: Speaking Imprecisely

“Why is testing
so expensive?”

“Why is *all of development*
so expensive? Let's make a
more testable product.”

Problem: Speaking Imprecisely

“Why is testing taking so long?”

“Why is *all of development* taking so long? Let’s make a more testable product.”

Problem: Speaking Imprecisely

“Can’t we just automate all the testing?”

“Can’t we just automate *all the development*? Let’s ask the developers what they think about that.”

Ignoring the social dimensions

We don't need roles on
Agile teams.

Roles help to focus skills,
commitments, and
relationships.

How To Get What You Want (CTOs) - 94

Why not just eliminate roles?

You have probably experienced this.



Why do some people think a role is a prison or a fortress?

This happens when “role” is defined as
the only things you do and
what no one else does.

HIGH SOCIAL DISTANCE



How To Get What You Want (CTOs) - 96

What a Role Is...

- a commitment to perform some service(s)
- an idea to focus commitments
- a way to help organize effort on a team
- a heuristic for explaining or defining work
- like a hat



What a Role Is NOT

- Not a declaration of the only things you are allowed to do (not a prison)
- Not a declaration of the things that you and you only are allowed to do (not a fortress)
- Not permanent and unchanging
- Not like a tattoo



We prefer to think of roles this way.

A role is like a villa. It is a semi-private space. Someone dwells in it. Someone is responsible. But visitors may come and help.

- Developers help testers.
- Testers help developers.
- But testers are **ACCOUNTABLE** for testing

FLEXIBLE SOCIAL DISTANCE



A Healthy Role Institutionalizes...

- **Competence:** Increases skill over time.
- **Focus:** Marshals energy and concentration to solve difficult problems well; economy of scale.
- **Anticipation:** Identifies future needs and potential problems before its too late.
- **Accountability:** Accepts responsibility for outcomes within scope of the role.
- **Coordination:** Minds the interface with other roles.

See "On a Role" <http://www.developsense.com/blog/2015/06/on-a-role/>

How To Get What You Want (CTOs) - 100

What is the testing role?

- *To test* is to evaluate a product by learning about it through exploration and experimentation.
- When someone is testing, that person has adopted (if only for that time) a testing role.
- *A tester's role* is to
 - to develop one's self as a tester
 - connect with the clients of testing
 - prepare for testing
 - perform testing
 - report the results of testing.

How To Get What You Want (CTOs) - 101