SAMPLE

Building Tomorrow's QA Engineers

The Repro's Keeper in the Machine Era

Aiman JoeHanz

Free sample: Introduction & Chapter 1

putrabooks.github.io

Buy the full book:

Carousell carousell.sg/p/building-tomorrow-s-qa-engineers-ebook-pdf-1471387055

Ko-fi

Payhip

Contents

Introduction

Chapter 1 — The Repro

Chapter 2 — The Two Folklores

Chapter 3 — What Is Actually Held

Chapter 4 — The Three Commitments

Chapter 5 — The Territories

Chapter 6 — The Window

Chapter 7 — The Session

Chapter 8 — The Instruments

Chapter 9 — The Year

Chapter 10 — The Long Arc

Introduction

The question this book keeps returning to is small and awkward: how would you know?

A tester is told that a screen is broken. She opens it, and it works. She asks what the person did, and the account is vague because the person was doing their job rather than watching their hands. Somewhere between the report and the working screen there is a fact, and until somebody produces it, everybody in the conversation is guessing about different things.

The fact is what this trade calls a repro: the shortest reliable sequence that makes a failure happen on demand. It is the unit of this work the way a measurement is the unit of an experiment, and everything else a tester does, from exploratory sessions to release decisions, is downstream of whether one exists.

Two habits argue against it, and both are old. The first is the hunt, in which testing is a search for defects and the tester is measured by the count. The second is the stamp, in which testing is a step near the end that a build passes through, and its output is a mark on a form. Neither is dishonest and both are expensive, because both can be performed without establishing anything a second person could check.

What replaces them is a discipline with four parts. Describe the failure precisely enough that somebody else can cause it. Separate what the product does from what it was meant to do, and keep both recorded. Decide which failures matter, phrased in the terms of the people who would feel them. Leave behind evidence that survives your departure, so the next person inherits a verified picture rather than an assurance.

None of that requires a particular tool, a particular process model, or permission. It requires that a claim about software be treated as a claim, with the awkward consequence that claims need support.

The chapters that follow describe the work as it is done: the two folklores this trade argues with, the four things it is actually holding, the territories inside it, the window that keeps closing, the craft of the session, the instruments, a year in the life of one engineer, and the shape of a career that machine-generated code is changing faster than it is changing the need to know whether the code is right.

One warning belongs here rather than at the end. The tools in this trade are

moving quickly, and the parts of this book that will date are the ones about tools. The parts that will not date are the ones about evidence: what it takes to establish that something is true about a system, and what it costs to skip that step. That has been the same since software was written by hand, and it will be the same when most of it is not.

Chapter 1 — The Repro

The Defect That Cannot Be Shown

The most expensive object in a software organisation is a defect nobody can demonstrate, and its cost is paid in disagreement.

It arrives as a complaint from a customer or a colleague, phrased as an impression: it was slow, it lost the order, it did something odd on the second screen. Somebody attempts to reproduce it, fails, and the report becomes a matter of belief. The person who reported it is certain; the person who could not reproduce it is doubtful; and the two positions are compatible because neither has established a fact.

The test is not whether the reporter is credible. It is whether the sequence has been captured precisely enough that somebody else can cause the failure. Until that exists, the conversation is about perceptions, and perceptions cannot be fixed, scheduled, or closed.

This is why the craft's first product is a reproduction rather than a fix. A defect with a repro is a work item: it can be assigned, argued about, prioritised and verified when it is done. A defect without one is a mood, and moods accumulate in a backlog where they rot.

The disagreement is more expensive than it looks, because it does not stay between two people. A failure that cannot be shown is taken to a meeting, and the meeting decides what to do about a thing nobody present has seen. The options in that meeting are belief or dismissal, and both are wrong instruments: belief sends somebody to repair a defect that may not exist as described, and dismissal closes a report from somebody who may be entirely correct. The organisation pays either way, and it pays again when the same report returns next month wearing different words.

What the meeting was missing was not intelligence or effort. It was an object small enough to carry into the room: a series of steps, a stated result, a stated expectation, and a version. Anybody who has sat through a long argument about a screen nobody could make fail knows that the argument ends the moment somebody says the words that start the sequence, because the room stops debating character and starts reading a script.

What Makes a Sequence Reliable

A reliable sequence has three properties, and a tester who can tell which is missing has skipped most of the usual argument.

The first property is completeness. The steps include the precondition as well as the action: the account state, the data already present, the session's history, the browser or device, the version deployed. Practitioners who record only the actions produce sequences that work on their own machine and nowhere else, because the precondition was the part doing the work.

The preconditions most often left out are the ones that feel like the world rather than the test. An account that had a previous failed attempt. A record that already carried a flag. A first-of-the-month batch that had run before the user opened the screen. None of these was typed by the person reporting, which is exactly why they carry the failure, and a sequence that omits them reproduces nothing except on the machine where the world happened to be right.

The second property is determinism. The sequence produces the failure every time, or the practitioner knows the rate. A failure that occurs once in five attempts is still real and it needs to be described as a rate, because a reviewer who cannot reproduce it on the first run will treat it as noise. "Three times out of five" is a fact; "I saw it once" is a rumour.

The third property is minimality. The sequence contains nothing irrelevant: no steps that were incidental to the session, no detours that happened to precede the failure. Minimality is what makes a defect diagnosable, because the fix usually lives in the step that turns out to be necessary, and an eight-step sequence hides which one it is.

Minimality is also what makes a sequence teachable. A colleague who receives four steps reads them and runs them. A colleague who receives fourteen reads them, decides the failure is complicated, and forwards the report to somebody else. The shorter sequence travels further through an organisation, and a defect that travels is a defect that gets seen by somebody able to fix it.

The three properties pull against each other in practice, and the craft is the reconciliation. Completeness adds steps and minimality removes them. The resolution is to move the precondition out of the steps and into the setup: a sentence about the state of the world before the sequence begins, then the shortest run of actions that crosses the failure. The result reads as two paragraphs and behaves as an instrument.

The Failure That Comes and Goes

Some failures refuse to arrive on schedule, and the trade has a specific

practice for them, which is to hunt the condition rather than the failure.

The instinct when a defect will not repeat is to run the sequence again, and again, until either it happens or patience ends. The instinct is almost right and it wastes most of its information. Between the run that failed and the run that succeeded, something differed, and the discipline is to vary one thing at a time and record the result: the account, the data set, the time of day, the load, the sequence of screens visited beforehand. Each controlled variation is an observation, and the observations accumulate into a condition.

The condition, once found, converts an anecdote into a reproduction. A failure that happens only for accounts created before a migration, only in the first second after login, only when two actions race, is no longer intermittent; it is specific, and specificity is what a repair needs. Many of the most serious defects in commercial software are of this kind precisely because their conditions were rare enough to survive every scheduled pass.

There is a habit worth naming here, because it costs organisations years. A failure that cannot be reproduced on demand is sometimes closed as unreproducible, which converts a question into an answer without producing any evidence. The honest closure is a recorded rate and condition: did not repeat in twenty attempts, occurred twice in a week for one customer, correlated with the morning batch. That record lets somebody recognise the defect when it next appears, and recognition is the beginning of reproduction.

The second honest practice is the instrumented attempt. When a failure cannot be summoned, the tester arranges to catch it in the act rather than to summon it: logging enabled on the path, the account placed under watch, the support contact briefed to press a marked button when it happens again. The tester is no longer waiting for luck; the product is being prepared to testify.

Four Sources of a Repro

Tests produce reproductions in four ways, and knowing which one you are using determines how much confidence to place in the result.

The deterministic path comes from a written test: an automated check that performs a sequence and asserts an outcome. It is the strongest evidence because it is repeatable by a machine and it can be run forever. Its limitation is that it only covers what somebody thought to assert.

The exploratory path comes from a session at the product with a purpose in mind, in which the tester is watching behaviour rather than following a script.

It produces reproductions of the unexpected kind, which are the ones no test was written for, and it depends entirely on the tester's judgement about where to look.

The reported path comes from somebody else: a user, a support agent, a colleague who noticed something. It produces the highest-value reproductions in ordinary commercial work, because these are failures that users actually encounter, and it also produces the vaguest initial descriptions.

The inferred path comes from reasoning: a tester reads a change, predicts what will break, and constructs the sequence that proves it. It is the source that separates a diagnostician from an operator, and it is the only one that produces evidence before anybody is harmed.

The four sources also differ in what they cost to convert into evidence, and the conversion is where most of the value is realised. A reported failure arrives as a paragraph of impressions and becomes a reproduction through an interview: what screen, what account, what time, what did the message say. An exploratory finding arrives as a tester's own memory and becomes a reproduction the same evening, while the steps are fresh. An inferred reproduction is written before the failure is ever caused, which means it is the only kind whose sequence can be checked for completeness before it is run.

Practitioners differ in which source dominates their practice, and the differences show in their blind spots. The tester who works mostly from reports knows the users' world and underweights the design's corners. The tester who works mostly by inference knows the code's neighbours and underweights the ordinary. A team is well staffed when its members read from all four sources, because each source fails in a place the others reach easily.

The Repro as a Contract

A well-written reproduction is a contract between the person who found the failure and the person who will repair it, and the terms are worth stating.

It promises three things: that the failure exists at a named version, that the conditions are described well enough for another person to cause it, and that the observer will know when the failure has occurred. That last clause is the one most reports leave out and the one that earns its keep, because it converts a vague inconvenience into a testable proposition.

It asks one thing in return: that the repairer not treat the resolution as complete until the sequence has been run again on a build that contains the

fix. Without that, the defect is closed on an intention rather than on evidence, and the reopening costs more than the verification would have.

A contract implies two parties with obligations, which is why practitioners who write reproductions well become the people whose reports are trusted. The reputation is not for thoroughness. It is for being right about whether something is broken.

The contract also settles the awkward question of tone. A reproduction that states its terms does not need to persuade, and the persuasion is where reports acquire the smell of advocacy. The sequence says what happened; the expectation says what should have happened; the version says where. A report shaped that way can be read by a stranger with no goodwill towards its author, and being readable without goodwill is the property that makes an artefact survive an organisation's politics.

The Report That Saves a Day

Two reports of the same failure differ enormously in what they cost, and the difference is nearly all in the framing.

The expensive report describes an outcome: the checkout is broken. It starts a conversation about scope, severity, blame and whether the reporter is sure, and none of that conversation produces information. The cheap report describes a sequence and a version, and it starts a repair.

The craft in the framing is the claim's modesty. A report that says what was observed, what was expected and how to make both happen leaves the diagnosis to a specialist and does not prejudge it. A report that includes a theory invites an argument about the theory, which is a conversation the reporter will usually lose and which has nothing to do with the defect.

Experienced testers learn to withhold the diagnosis with effort, because having worked out what is wrong feels like value delivered. The value is in the sequence; the diagnosis is a guess that a colleague may disprove in an hour and that may cost a day of argument in the meantime.

The same modesty applies to reports that say the product is correct. A tester who has exercised a path and seen it behave properly holds something valuable, and its value is spent by saying precisely what was exercised rather than by saying the feature works. Tested is a word that means something only when it is followed by a scope: these flows, that data, this version. Unscoped praise from a tester is worse than silence, because it borrows the credibility

the tester has earned by observing carefully and spends it on a claim nobody can check.

From Sequence to Check

A reproduction that exists only in a ticket is spent once, and a reproduction converted into a check is spent for as long as the product lives.

The conversion is small and it is where the two halves of this trade meet. The sequence that demonstrated the defect, reduced to its minimal form, becomes the body of an automated check. The observed failure becomes the assertion's opposite: the check now asserts what the product should have done. The version that carried the defect goes into the check's history, so that anybody reading the suite a year later can see which behaviour it guards.

Not every reproduction deserves the conversion, and the choice is a judgement about recurrence. A defect caused by a configuration that no longer exists, in a feature scheduled for removal, does not need a permanent witness. A defect caused by a wrong assumption about data, or by an interaction two components had never performed together, is evidence of a fragility that will be rediscovered by the next change in the same area, and a check converts that discovery from an incident into a red line.

The conversion is also where the trade's memory accumulates. A suite of checks is often described as insurance, and the metaphor is wrong in a useful way: insurance pays after the loss, while a converted reproduction prevents the same loss specifically. What the suite actually holds is the organisation's catalogue of its own past surprises, each one written down as a sequence that must never fail again. Reading a suite that way changes how it is maintained, because deleting a check is not trimming cost; it is forgetting a surprise on purpose.

A practitioner who ends each defect with the question of whether it becomes a check is running a quiet, compounding practice. After a year the question has produced a suite that guards exactly the failures this product actually produced, which no generic template can match, and the practitioner's judgement about which sequences recur is the skill the whole book keeps returning to.

The Sequence That Crosses a Boundary

Most failures a commercial tester meets live inside one screen, and the ones that cost the most usually cross a boundary: the product hands something to a

provider, a queue, a file system or a colleague's service, and the failure lives in the handing.

Sequences that cross boundaries need one addition to the classical repro, which is the trace. The steps carry the tester as far as the product's edge, and from there the sequence describes what should have crossed: the request, its identifying fields, the expected response, and where each side recorded the exchange. A defect of the boundary kind presents as a disagreement between two systems that both behave correctly alone, and the reproduction is the evidence that settles which side of the disagreement is the product's.

The identifying fields matter more than the steps. A payout, a refund, a batch item: each carries an identifier that both systems recorded, and the tester who writes the identifier into the repro has written the one line that the provider's engineer can search for. The practitioner's habit at the boundary is to capture identifiers the way a photographer captures frames, before anything is restarted, because the identifier is the only part of the failure that both sides keep.

Boundaries also produce the trade's most instructive class of near miss: the failure that a retry fixes. The tester reruns the sequence, the second attempt succeeds, and the temptation is to close the investigation with the word intermittent. The discipline instead is to keep both attempts and their identifiers, because a first attempt that failed and a second that succeeded is a fact about a timeout, a duplicate, an idempotency assumption or a race, and each of those is a design decision that deserves a question rather than a shrug.

The Repro That Travels

A reproduction usually stays within one team, and the highest-value ones travel, which changes what they must contain.

The travelling repro goes to a provider whose service is involved, to a vendor whose platform carries the failure, or to a partner whose integration surfaces it. Its reader has no access to the product, no knowledge of its screens and no obligation to be generous, and it must therefore carry its own context: the timestamps, the identifiers, the request as sent, the response as received, and the expectation with its source stated. What the internal report can leave implicit, the travelling report must say, because implicit is a courtesy that only works between people who share a building.

The travelling repro also has a tone to manage, and the craft is neutrality under asymmetry. The reader on the other side is an organisation with its own incentives, and a repro that reads as an accusation produces a defence rather than a repair. The practitioners who get boundary failures fixed write the sequence the way chapter four asks for all reports, with observation and expectation separated and the question addressed to whoever owns the intent, and they add the one sentence that keeps the correspondence short: what we have established, and what we cannot see from here.

There is a final property of the travelling repro that makes it worth practising even for failures that never leave the building. Writing for a stranger is the strictest form of the discipline: it strips out every assumption the team's shared context was quietly carrying. Practitioners who write their internal repros to the standard of the travelling one discover that a proportion of their defects were never as reproducible as the team's familiarity made them look, and that discovery, uncomfortable as it is, is the repro doing its oldest job, which is to tell the truth about what is actually known.

When the Repro Points at the Wrong Thing

A reproduction is a strong instrument and not an infallible one, and the trade's honesty includes the case where the sequence is right and the conclusion drawn from it is wrong.

The usual shape is the symptom that leads. A failure is reproduced beautifully through the screen that displays it, the report names the screen, and the repair is aimed there. The cause lives two services away, in a default nobody read or a rounding nobody chose, and the screen was merely the surface the cause rose through. The repro was never false; it demonstrated exactly what it said. What failed was the leap from demonstrated behaviour to located cause, and the leap is a diagnosis, which chapter one has already asked reports to withhold.

The repair for the wrong-leap is not more caution but more attributes. A repro that carries its identifiers, timestamps and versions gives the repairer a starting line rather than a conclusion, and the repairer's trace from those attributes usually finds the two-services-away cause faster than any argument about screens. The tester's contribution to the wrong-leap is therefore to keep the sequence complete while the interpretation stays modest, which is the report form from later in this chapter doing quiet diagnostic work.

There is a rarer case that deserves its own paragraph: the repro that stops

working. The product changes, the failure disappears, and the ticket closes with the words cannot reproduce. The discipline's handling of that case has already been described as a rate and a condition, and one addition belongs here. A failure that departs when the product changes is not evidence that it was never there; it is a correlation between the change and the disappearance, and that correlation is occasionally the most valuable fact in the file. The tester who records which change coincided with the departure has handed a future investigator the exact lead that a closure note would have burned.

The Repro as a Teaching Instrument

One last use of the reproduction belongs in the first chapter, because it explains why the artefact is worth the practitioner's whole care even in weeks when no defect needs one.

A team's repro folder, read by a new joiner, is the product's real documentation. Each sequence shows a path through the product as a series of steps with consequences, each condition shows where the assumptions live, and each closed entry shows how a disagreement was argued and settled. A new tester who reads the last year's repros before touching the product arrives with a map of the terrain that no overview document supplies, because the map was drawn by the failures themselves.

The repro also teaches the craft's grammar faster than any instruction. A joiner who reads fifty reports learns the two-sentence form by osmosis, sees what a modest claim looks like, and starts writing that way before anybody asks them to. The practitioners who inherit a good folder notice it in their first filed defect, and the practitioners who inherit a bad one spend their first year unlearning the impression that a report is a complaint.

That is the last argument for the care this chapter has asked for. The repro is not only evidence for one defect; it is the trade's ongoing demonstration of what evidence looks like, given to every colleague and successor who will ever have to produce it or read it. A practitioner who writes repros as contracts is teaching an organisation to argue from facts, one small document at a time, and that quieter effect outlasts every defect the documents were written about.

ABOUT THE AUTHOR

Aiman JoeHanz is the author of the Building Tomorrow's People series, in which each book gives one profession a working definition and holds it to that standard. Here the profession is quality assurance, and the argument is that its product was never the test but the reproduction: the smallest set of conditions that makes a failure happen again on demand. Readers who expect a book about coverage percentages will find something more uncomfortable, a trade defined by its ability to describe what went wrong precisely enough that somebody else can watch it go wrong too. He writes for practitioners who are measured on what escaped rather than what was caught.