Data Type Attacks

Paths/Files	Long Name (>255 chars) Special Characters in Name (space * ? / \ | < > , . ( ) [ ] { } ; : ' " ! @ # $ % ^ &) Non-Existent Already Exists No Space Minimal Space WriteProtected Unavailable Locked On Remote Machine Corrupted
Time and Date	Timeouts Time Difference between Machines Crossing Time Zones Leap Days Always Invalid Days (Feb 30, Sept 31) Feb 29 in Non-Leap Years Different Formats (June 5, 2001; 06/05/2001; 06/05/01; 06-05-01; 6/5/2001 12:34) Internationalisation dd.mm.yyyy, mm/dd/yyyy am/pm, 24 hours Daylight Savings Changeover Reset Clock Backward or Forward
Numbers	0 32768 (215) 32769 (215 + 1) 65536 (216) 65537 (216 +1) 2147483648 (231) 2147483649 (231 + 1) 4294967296 (232) 4294967297 (232 + 1) Scientific Notation (1E-16) Negative Floating Point/Decimal (0.0001) With Commas (1,234,567) European Style (1.234.567,89) All the Above in Calculations

Data Type Attacks

Strings	Long (255, 256, 257, 1000, 1024, 2000, 2048 or more characters) Accented Chars (àáâãäåçèéêëìíîðñòôõöö, etc.) Asian Chars ( ) Common Delimiters and Special Characters (" ' ` | / \ , ; : & < > ^ * ? Tab) Leave Blank Single Space Multiple Spaces Leading Spaces End-of-Line Characters (^M) SQL Injection ('select * from customer) With All Actions (Entering, Searching, Updating, etc.) Emojis
General	Violates Domain-Specific Rules (an ip address of 999.999.999.999, an email address with no "@", an age of -1). Violates Uniqueness Constraint

Web Tests

Navigation	Back (watch for 'Expired' messages and double-posted transactions) Refresh Bookmark the URL Select Bookmark when Logged Out Hack the URL (change/remove parameters; see also Data Type Attacks) Multiple Browser Instances Open Swipe/Tap/Pinch
Input	See also Data Type Attacks HTML/JavaScript Injection (allowing the user to enter arbitrary HTML tags and JavaScript commands can lead to security vulnerabilities). Check Max Length Defined on Text Inputs > 5000 Chars in TextAreas
Syntax	↗ HTML Syntax Checker ↗ CSS Syntax Checker

Web Tests

Preferences	Javascript Off Cookies Off Security High Resize Browser Window Change Font Size
Accessibility / A11y	Keyboard: Navigation; Skip to link (first tab); No traps (menus / subsections); visible focus indicator; use all functionality; pop ups have focus, can be dismissed. Context: Links (descriptive) ; Alt-text (descriptive or decorative is hidden); Form input labels; Main elements (only one) Country and language defined; plain language used. Content: Capitals in #; No all capitals text; No justified text; Zoom to 200%; Gender neutral; acronyms explained; clear instructions; Good contrast; More than just colour to indicate success e.g. green tick.

API Tests

↗ **BINMEN** (Gwen Diagram & Ash Winter)	Boundary, Invalid Entries, NULL, Method, Empty, Negative
↗ **POISED** (Amber Race)	Parameters, Output, Interop, Security, Errors, Data
↗ **VADER** (Stuart Ashman)	Verbs, Authorisation/Authentication, Data, Errors, Responsiveness

Mobile/Device/Tablet

↗ **Mobile App Testing** (Daniel Knott)	Mobile Device, Orientation, Mobile Browsers, Interrupts, Look, Energy Consumption, Automation, Performance, Personas, Time & Date, Ergonomics, Security, Tracking, Inputs, Network, Platform Guidelines.

This cheat sheet includes original ideas from Elisabeth Hendrickson, James Lyndsay, and Dale Emery. Further ideas from Andrea Jensen, Ady Stokes, Callum Akehurst-Ryan, Dave Harrison, Deborah Sherwood, Mark Winteringham, and Simon Tomes.

Testing Wisdom

- A test is an experiment designed to reveal information or answer a specific question about the software or system
- Stakeholders have questions; testers have answers
- Don't confuse speed with progress
- Take a contrary approach
- Observation is exploratory
- The narrower the view, the wider the ignorance
- Big bugs are often found by coincidence
- Bugs cluster
- Vary sequences, configurations, and data to increase the probability that, if there is a problem, testing will find it
- It's all about the variables
- I am not all humans, not everyone does things as I do

Heuristics

Variable Analysis	Identify anything whose value can change. Variables can be obvious, subtle, or hidden
TouchPoints	Identify any public or private interface that provides visibility or control. Provides places to provoke, monitor, and verify the system
Boundaries	Approaching the Boundary (almost too big, almost too small), At the Boundary
Goldilocks	Too Big, Too Small, Just Right
CRUD	Create, Read, Update, Delete
Follow the Data	Perform a sequence of actions involving data, verifying the data integrity at each step. (Example: Enter → Search → Report → Export → Import → Update → View)
Configurations	Varying the variables related to configuration (Screen Resolution; Network Speed, Latency, Signal Strength; Memory; Disk Availability; Count heuristic applied to any peripheral such as 0, 1, Many Monitors, Mice, or Printers).
Interruptions	Log Off, Shut Down, Reboot, Kill Process, Disconnect, Hibernate, Timeout, Cancel
Starvation	CPU, Memory, Network, or Disk at maximum capacity

Heuristics

Position	Beginning, Middle, End (Edit at the beginning of the line, middle of the line, end of the line)
Selection	Some, None, All (Some permissions, No permissions, All permissions)
Count	0, 1, Many (0 transactions, 1 transactions, Many simultaneous transactions)
Multi-User	Simultaneous create, update, delete from two accounts or same account logged in twice
Flood	Multiple simultaneous transactions or requests flooding the queue e.g. making/selecting a submit request/button multiple times
Dependencies	Identify "has a" relationships (a Customer has an Invoice; an Invoice has multiple Line Items). Apply CRUD, Count, Position, and/or Selection heuristics (Customer has 0, 1, many Invoices; Invoice has 0, 1, many Line Items; Delete last Line Item then Read; Update first Line Item; Some, None, All Line Items are taxable; Delete Customer with 0, 1, Many Invoices).
Constraints	Violate constraints (leave required fields blank, enter invalid combinations in dependent fields, enter duplicate IDs or names). Apply with the Input Method heuristic.
Input Method	Typing, Copy/Paste, Import, Drag/Drop, Various Interfaces (GUI v. API)
Sequences	Vary Order of Operations → Undo/Redo → Reverse → Combine → Invert → Simultaneous
Sorting	Alpha v. Numeric → Across Multiple Pages
State Analysis	Identify states and events/transitions, then represent them in a picture or table. Works with the Sequences and Interruption heuristics
Map Making	Identify a "base" or "home" state. Pick a direction and take one step. Return to base. Repeat
Users & Scenarios	Use Cases, Soap Operas, Personae, Extreme Personalities

This cheat sheet includes original ideas from Elisabeth Hendrickson, James Lyndsay, and Dale Emery. Further ideas from Andrea Jensen, Ady Stokes, Callum Akehurst-Ryan, Dave Harrison, Deborah Sherwood, Mark Winteringham, and Simon Tomes.

Heuristics

↗ **RCRCRC** (Karen N. Johnson)	Recent - what testing around new areas of code should I think about? Core - what essential functions or features must continue to work? Risky - what features or areas of code are inherently more risky? Configuration Sensitive - what code is dependent on environment settings? Repaired - what code has changed to address defects and potentially created issues? Chronic - what code typically breaks features that need to be tested?
↗ **FAILURE** (Ben Simo)	Functional, Appropriate, Impact, Log, UI, Recovery, Emotions
↗ **WWWWWHKE** (sounds like "wiki" (Darren McMillan)	Who is this for? What is this for? When & by whom is it to be done? Where is it being done? Why is it being done? How is it being achieved? What questions does my Knowledge & Experience produce?
↗ **Diversity & Inclusion** (Callum Akehurst-Ryan) ↗ **Combat Bias with Heuristics of Diversity** (Ash Coleman)	Does this work for me? Does this work for them? Does this work for someone I have never considered or ever met?
↗ **Seven Dwarfs** (Cassandra H. Leung)	Grumpy, Happy, Sleepy, Bashful, Sneezy, Dopey, and Doc
↗ **Specs/Designs Watchlist** (Gerard McCann)	Ambiguity, weasel words (like could, should or may), Fudge (e.g. statements like 'this will be resolved at a later date', but no specifics around who and when), Confusing terminology, jargon or obscure acronyms, Oversimplification, Overcomplication.
↗ **TORCH** (Simon Tomes)	Timer, Oracles, Risks, Consider these questions, Heuristics.
↗ **MCOASTER** (Michael Kelly)	Mission, Coverage, Obstacles, Audience, Status, Techniques. Environment, Risk

This cheat sheet includes original ideas from Elisabeth Hendrickson, James Lyndsay, and Dale Emery. Further ideas from Andrea Jensen, Ady Stokes, Callum Akehurst-Ryan, Dave Harrison, Deborah Sherwood, Mark Winteringham, and Simon Tomes.

Heuristics

Seen and Heard (Ady Stokes)	For everything you can see, is it announced by a screen reader? For everything you hear, can it be read (transcript, subtitles, captions, audio descriptions)
↗ **TuTTu and TaTTa** (Mark Winteringham)	Testing the UI or Testing Through the UI Testing the API or Testing Through the API
↗ **SACRED** (Richard Bradshaw)	State Management, Actions, Codified Oracle, Reporting, Execution, Deterministic
↗ **TRIMS** (Richard Bradshaw)	Targeted, Reliable, Informative, Maintainable, Speedy

Frameworks

Judgement (James Lyndsay)	Inconsistencies, Absences, and Extras with respect to Internal, External – Specific, or External – Cultural reference points
Observations (James Lyndsay)	Input/Output/Linkage
Flow	Input/Processing/Output
Requirements (Gause & Weinberg)	Users/Functions/Attributes/Constraints
Nouns & Verbs	The objects or data in the system and the ways in which the system manipulates it. Also, Adjectives (attributes) such as Visible, Identical, Verbose and Adverbs (action descriptors) such as Quickly, Slowly, Repeatedly, Precisely, Randomly. Good for creating random scenarios
Deming's Cycle	Plan, Do, Check, Act