Windows type of guy, but not doing SharePoint stuff 😊. Might seem strange given it only really has one shell, except for that new fangled “Monad” thing…
Member of ACCU for almost 8 years.
Wanted to find out what tools people are using – Unconscious Incompetence (Unknown unknowns)
Lots written about general programming languages and major tools, not so much about the smaller stuff
Many articles seem very sensational, e.g. Using MongoDB will ruin your life!
StackOverflow is great for definitive answers but has to shy away from the more subjective questions
One size never fits all – have you ever tried a pair of socks that purports to fit anything from a size 6 – 12!
Scott Hanselman’s list is one example of basic tool choices
Part I - Tooling Choices

This talk is a game of two halves
People always say “use the right tool for the job” what does that mean exactly?
Poll what is this – it’s the correct tool for opening a paint tin!
General purpose tools versus custom tools
Examples of other specialist decorating tools, e.g. heat gun, all sorts of brushes and rollers and attachments (ceilings)
Context is everything – often multiple choices available
Tale about rebooting into Linux because it was faster than Windows 3.1 to search the entire hard disk
Not really about right or wrong, but best (most effective) tool
Examples – SharePoint © especially Trello plug-in
Why would you need to use the wrong tool? (refer to Developer Freedom article)
set filename=trello.json
set releaseId=v1.5.2
sed -n " /^	"cards"/ , /^	} , /p" %filename% | ^
egrep -i "idlist|shortUrls|idshort|name" | ^
set -n "/%releaseId%/,/shortUrls/p" | ^
egrep -i "\"idshort\"|\"name\"" | ^
set "s/\s*.idShort.../" | ^
set "s/\s*.name.://" | ^
set "s/,$/" | ^
set "s/\s*/s/\s*$/" | ^
set "s//g" | ^
gawk "!(NR==3) { printf "%03s %%-14s %s\n", first, $0, second } { first=second; second=$0 }" | ^
sort

Mining Trello board (JSON) using SED + AWK until I found JQ
Down the rabbit hole we go, when do we stop?
“If you cannot think of three ways of abusing a tool, you do not understand how to use it.”
-- Gerald Weinberg

From General System Thinking
Know the limitations of any tool to know when you need to look for another
Right tool not always available – build your own – especially custom support tools
Often grow from test tools
Examples – scheduler, calculator, trade filter, DDE bridge, etc.
Functionality packaged as a library allows GUI and/or command line variants (latter for automation, e.g. COM + VBScript or .Net + PowerShell)
Composing small well focused tools
Desktop text file with common one-liners in it – assigned to aliases on mature shell
Avoid writing programs to do what can easily be done with a shell and a simple command or two
A selection of one-liners from my last project
Can be write-only at times and so are often disposable and non-shareable

> PowerShell "gci Build\Release\Company*.dll | % { $_.VersionInfo }

> for /r /d %d in (*.*) do @if exist "%d" rmdir /q "%d" 2>nul

> for /d %d in (*) do @echo "%d" && pushd "%d" && (if exist "clean.cmd" call clean.cmd --all) && (if exist "Source\clean.cmd" pushd Source && call clean.cmd --all && popd) && popd

> sed -n "/<ProjectReference>在内的<!ProjectReference>/p" proj.csproj | egrep "Name" | gawk "match($0,/>(.+)</,ary) {print ary[1]}" | sort > "$temp\csproj-ref" && egrep "dependency" proj.nuspec | gawk "match($0,/>.042(.+)<.042 v/,ary) {print ary[1]}" | sort > "$temp\nuspec-dep" && diff --ignore-case --side-by-side "$temp\csproj-ref" "$temp\nuspec-dep"
Often grow out of one-liners into more maintainable form, i.e. wrapper scripts
Drivers can be build and deployment, e.g. recent NuGet check scripts (check for mismatches, dependencies, unused, etc.)
Talk about the sorts of tools – CI server, static analysis, automated testing of various forms, automated deployment, etc.
Correct return codes are essential (nod to my Robust Software talk 2 years ago)
Using tools to filter noise to allow simple diff-style comparison during testing
Part II – Unlikely Choices
Write stuff down – memory of a goldfish
Can help avoid context switching by tracking stuff for later
Personal TODO list does not have to be replicated on the work board
Diagrams – drawing more than once is ok, the context is often different, so it’s often a different picture
Quotes from Nat Pryce (notebook) and Jason Gorman (Crayola)
Notebook is like a persistent data structure
How many chose to do programming to get away from writing? (My U’s in lit + language)
“Never use the words get, put or nice.”
-- My Wife’s English Teacher

Get, get, get, get, …

However don’t alienate non-native speakers with obscure terms (C++ has obscure terms though, e.g. imbue, emplace, …)

Shift + F7 in MS Word
Questions?
No book to peddle, only my column and blog
Some of my home-grown tools on my GitHub page