Amplify your Experience: How to Start Writing and Speaking

Tonie Huizer

DevOps Consultant

Promicro

www.promicro.nl

toniehuizer

https://sessionize.com/seaql2026/

Tony Davis

Editor

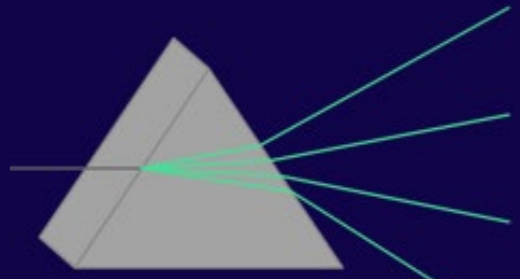
Redgate Software

...redgate.com/simple-talk/author/tony-davis

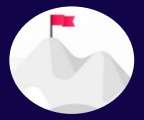
tonytheeditor

Tonie and Tony Show:
https://www.youtube.com/redgate/playlists

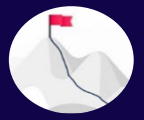
The Amplify journey



Your audience

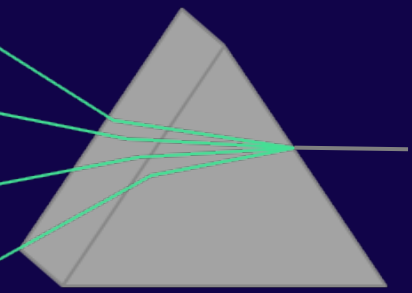


The problem



Your purpose

Your story



A Vague idea

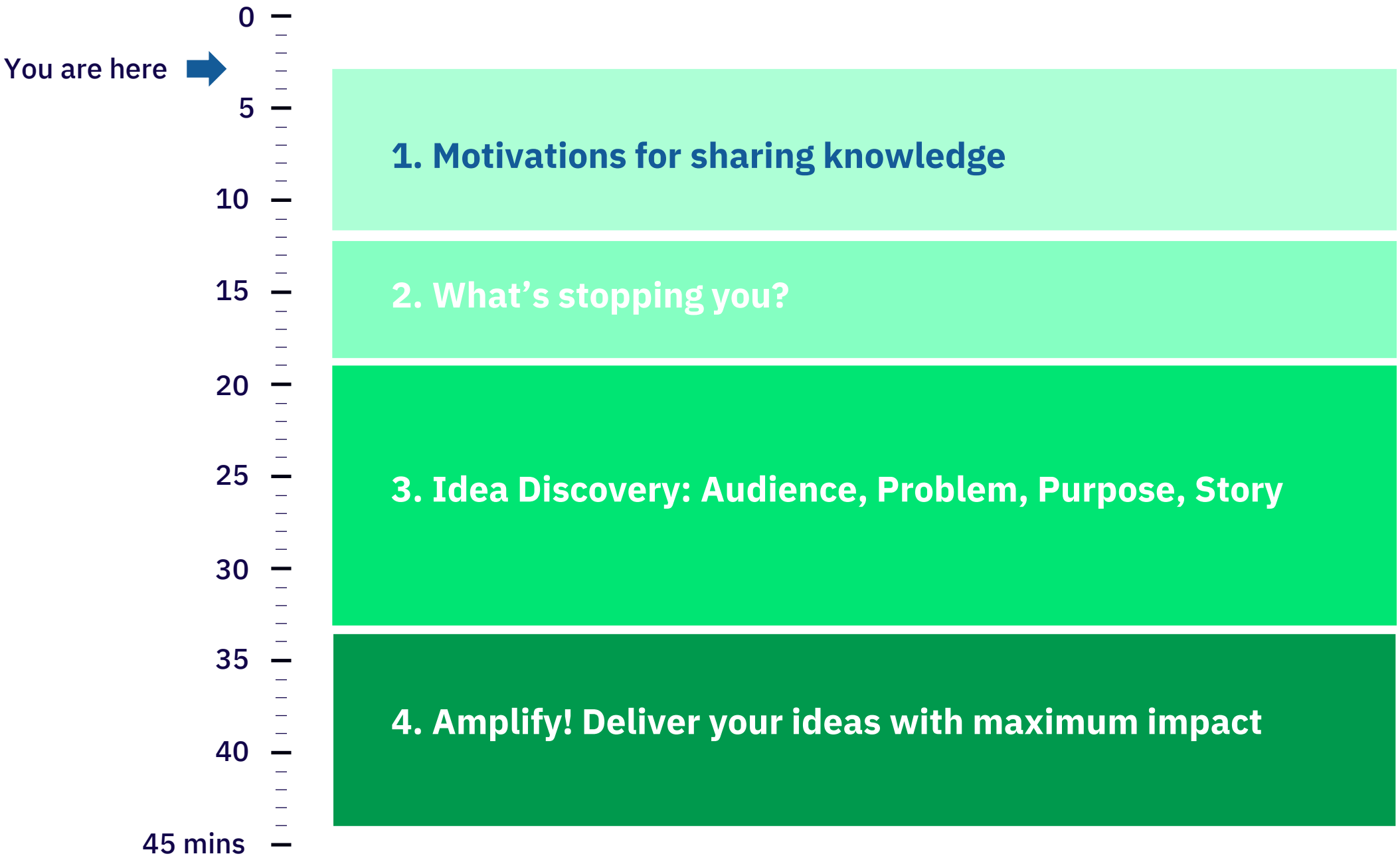
Ambiguous • chaotic • hard to explain

A Good Idea!

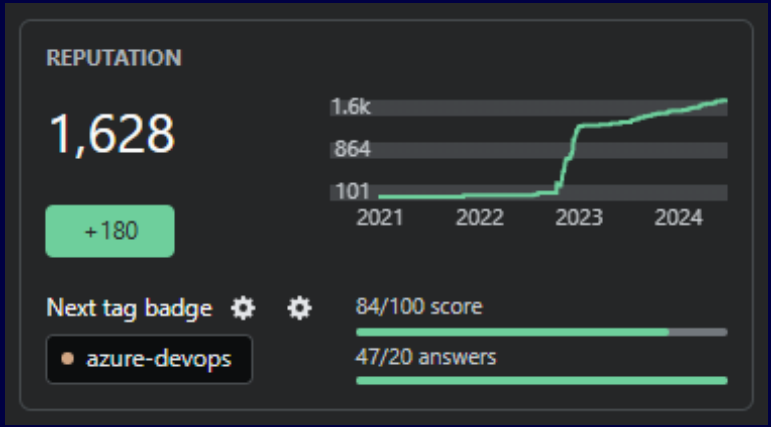
Clear • Structured • High Impact

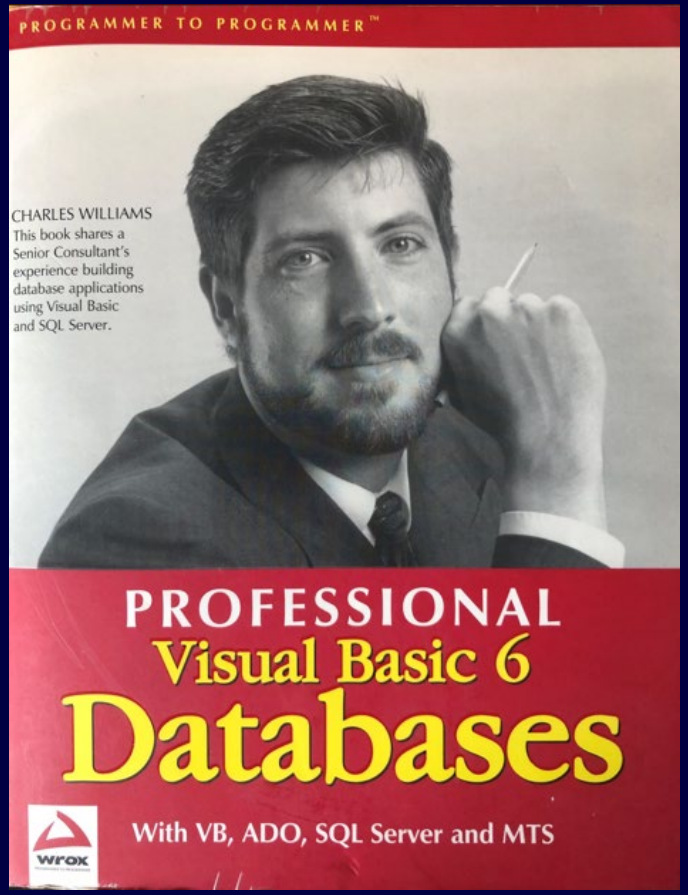
Amplify!

Deliver your ideas with maximum impact


0
You are here
5
10
15
20
25
30
35
40
45 mins
1. Motivations for sharing knowledge
2. What’s stopping you?
3. Idea Discovery: Audience, Problem, Purpose, Story
4. Amplify! Deliver your ideas with maximum impact

What you know will help others





Tech comms is a core IT skill

Better content

- Hits its audience, gets noticed

Expanded network

- Community and Corporate events

Better opportunities

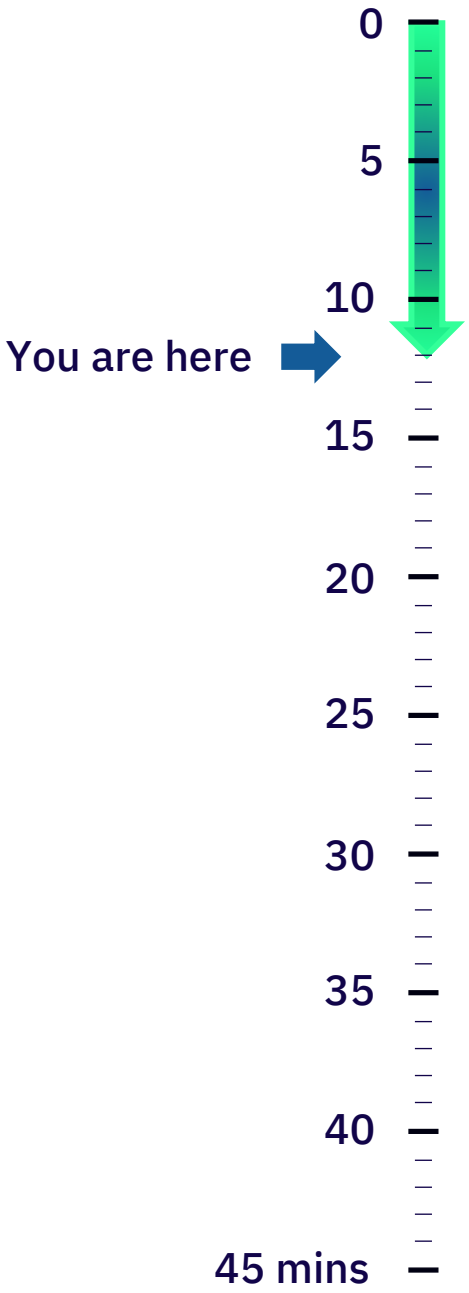
- Community and professional

Better tech communication

More effective teamwork

Better Communication with managers

Project proposals that get accepted



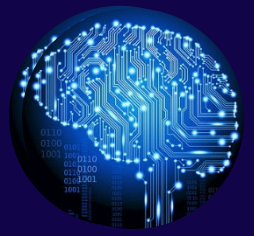
1. Motivations for sharing knowledge

2. What's stopping you?

3. Idea Discovery: Audience, Problem, Purpose, Story

4. Amplify! Deliver your ideas with maximum impact

Confidence Barriers



“I'm not an Expert...”

“I have nothing unique or interesting enough to say”

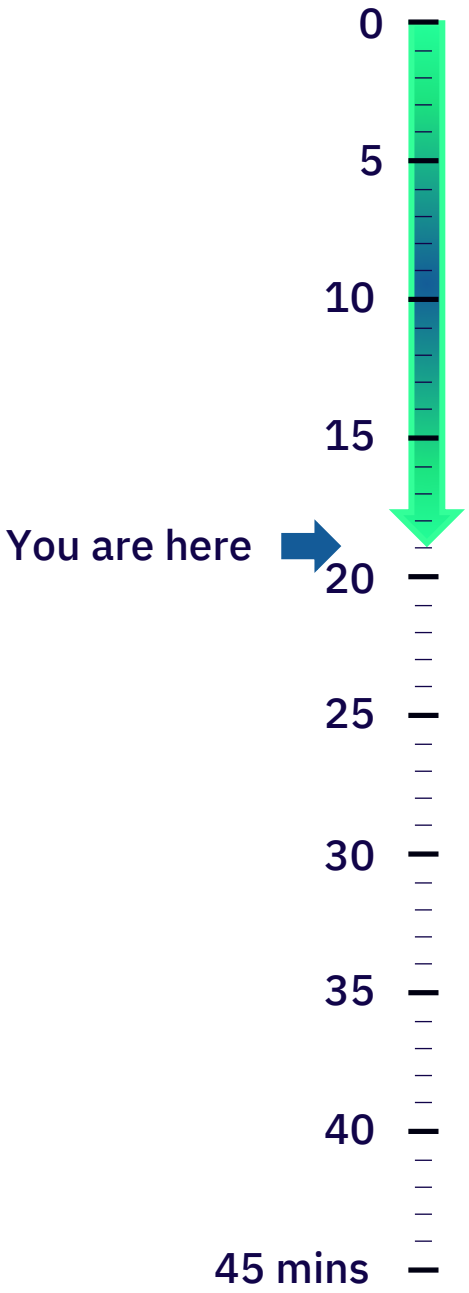
“What if I get it wrong and embarrass myself?”

Communication Barriers

“I struggle to express my ideas clearly”

“I’m no Grant Fritchey”

“I tried but no-one was interested”



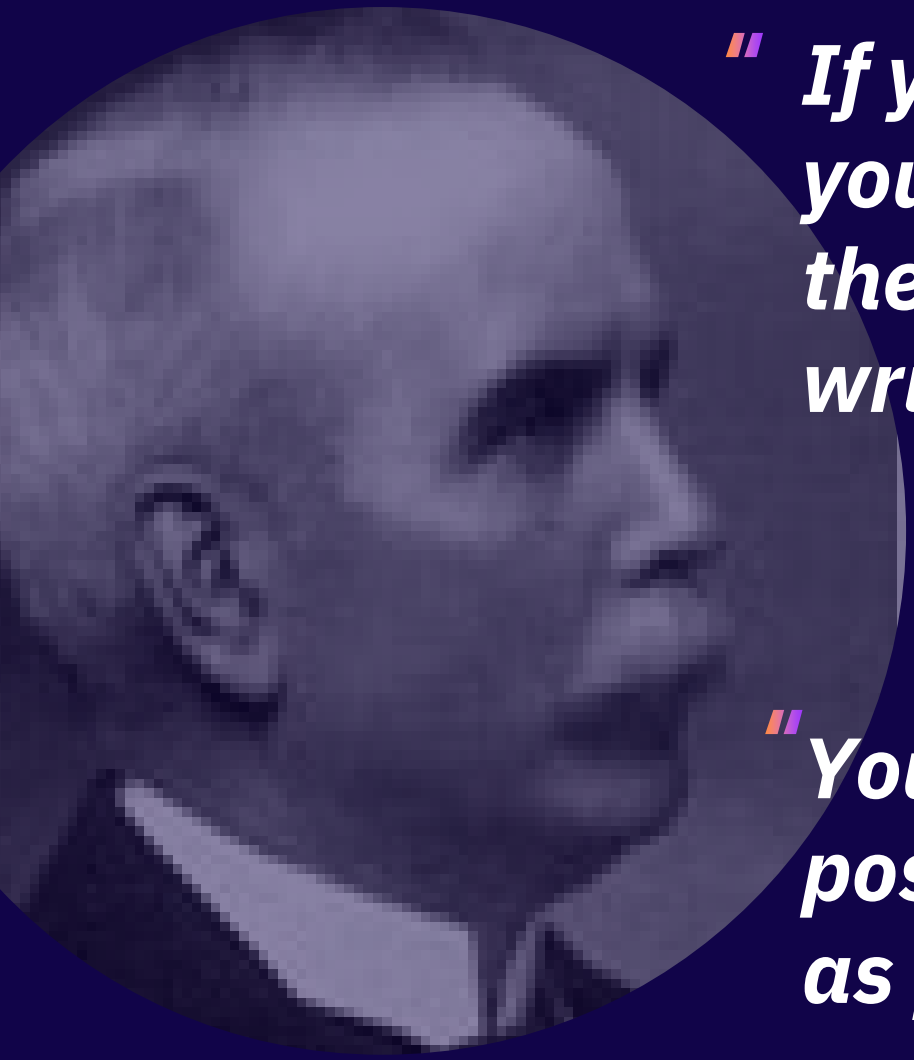
1. Motivations for sharing knowledge

2. What's stopping you?

3. Idea Discovery: Audience, Problem, Purpose, Story

4. Amplify! Deliver your ideas with maximum impact

The Wisdom of Phil Factor



"If you haven't put yourself in the shoes of your audience, tried to imagine what's on their mind when they find your content, then writing is the hardest of all skills"

"You need as few ideas or messages as possible, and they must be as memorable as you can make them"

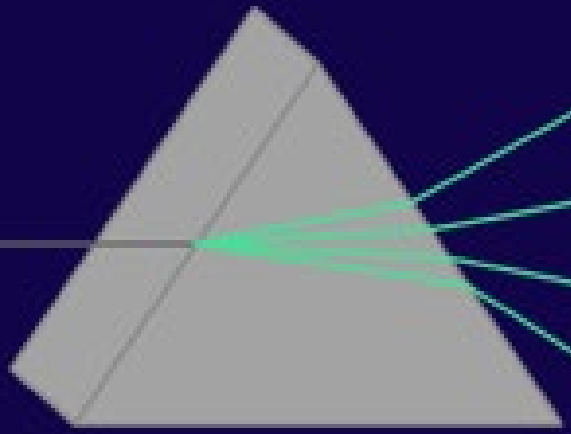
I have an idea…let's go!

"Why developers should be using execution plans to investigate poor query performance"

2 months later

Idea Discovery

I have an idea

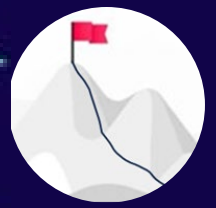


Your Audience

Who needs to know this and why?

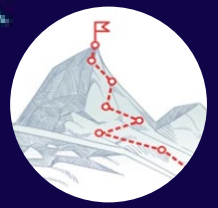
The Problem

What issue do you want to tackle?



Your Purpose

What above all else do you want to achieve?



Your Story

Problem → Purpose → Solution

Is it a *good* idea?

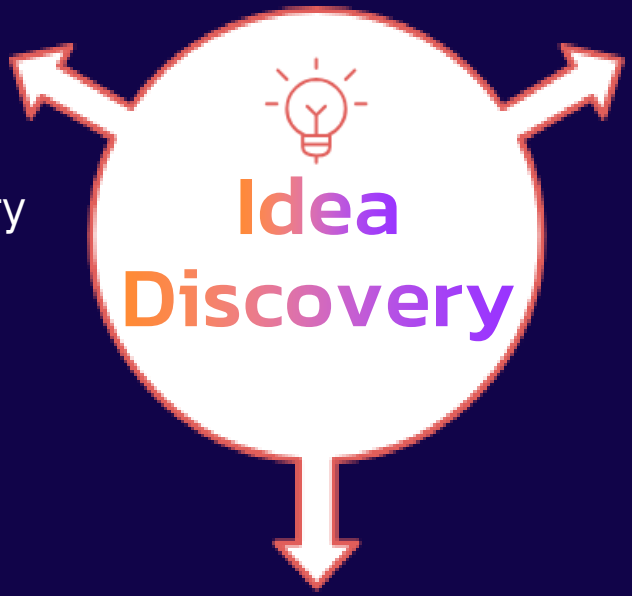
Test out idea on peers (work, events, local user groups...)

Start Slack threads

Join online discussions

3 creative routes to great ideas

Freeform or
Visual Discovery



Structured
Exploration

Informal
Discovery

Structured Discovery

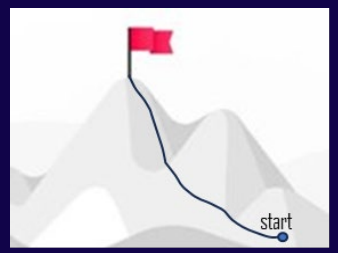
"Why developers should be using execution plans to investigate poor query performance"

Your Audience

The Problem

.NET devs don't use exec plans

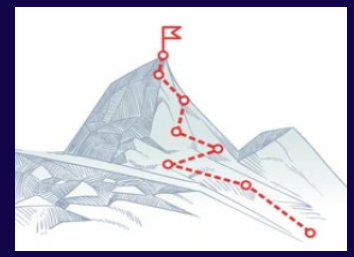
- *"Don't fit in my workflow"*
- *"Too heavyweight"*
- *"Too much effort to learn"*
- *"Quicker just to 'rip it up and try again"*



Your purpose

Lower the effort. Demo the rewards

Convince .NET devs to give exec plans another try, with a fast, simple way to retrieve and read plans that slots into their workflow and helps them spot common, easily-fixed SQL mistakes.



Your story

Smart fix first, then rewrite

Think "exec plans aren't worth the effort"? This session will change your mind. Rewrites waste time if the SQL is fine. I'll show you a quick, low-friction way to spot simple mistakes before you rip up your code.

The Problem

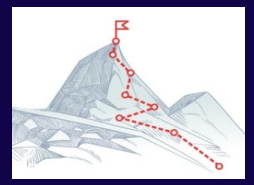
.NET devs don't use exec plans

- *"Don't fit in my workflow"*
- *"Too heavyweight"*
- *"Too much effort to learn"*
- *"Quicker just to 'rip it up and try again"*

Your purpose

Lower the effort. Demo the rewards

Convince .NET devs to give exec plans another try, with a fast, simple way to retrieve and read plans that slots into their workflow and helps them spot common, easily-fixed SQL mistakes.



Your story

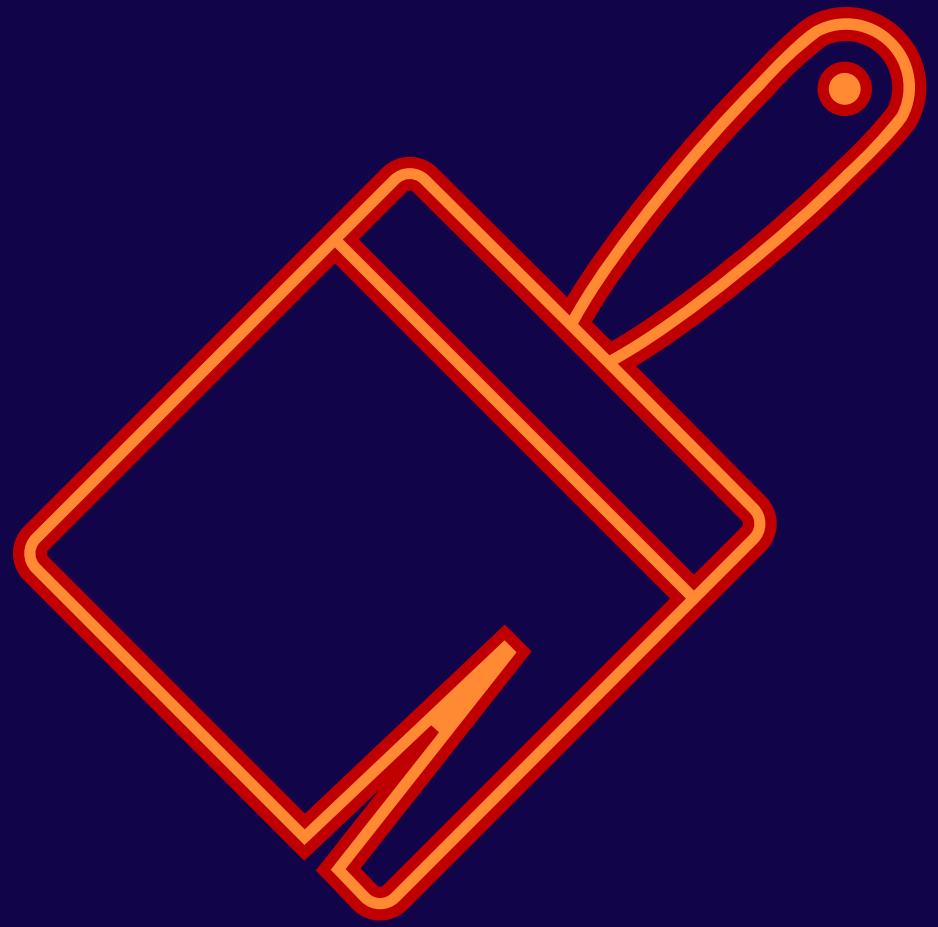
Smart fix first, then rewrite

Think "exec plans aren't worth the effort"? This session will change your mind. Rewrites waste time if the SQL is fine. I'll show you a quick, low-friction way to spot simple mistakes before you rip up your code.

A Structure

1. **Smart fix before rewrite – a quick demo of value**
2. **What's in a plan? A simple visual introduction**
3. **Retrieving a plan without the pain– integrated with workflow** ⭐
4. **Reading a plan – remove the fear factor**
5. **.NET demos –five common SQL mistakes** ⭐
   - **how to spot the 'clues' in the plan**

Freeform or Visual Idea Discovery



Say it, draw it...

- ✓ Write down your idea
- ✓ Pin down your idea
  - ✓ Purpose
  - ✓ Audience
  - ✓ Story
- ✓ Structure
- ✓ Write or present

Developing and testing applications are in most c
your career everyone has come across developin
representative.

I still remember the example of a software vendc
production which was only tested against IDs bel
than 10k of IDs. What happened was a down tim

That copy of production

So, we can agree that getting your development,
the right test data is necessary to deliver a high s

But creating the right test data is hard and we do
Luckily many of us can achieve the same with a c
Or was it the year before or wait was it even olde
We might have to ask a colleague that works on t
shared development database is.

While the age of the structure of the database ca
workflow in place, the data in the database is a w

For starters, is it necessary to have all this data in
and faster?
And what about the reusability of this "copy of pi
data with the last test run?

And don't get me started about personal informa
Be honest, is this copy even GDPR compliant?

Maybe working with the "copy of production" so
this really true?

Step-by-step working on alternatives

The first improvement that needs to be done is u
one of your co-workers! Your own personal datak
minimum to get it right.

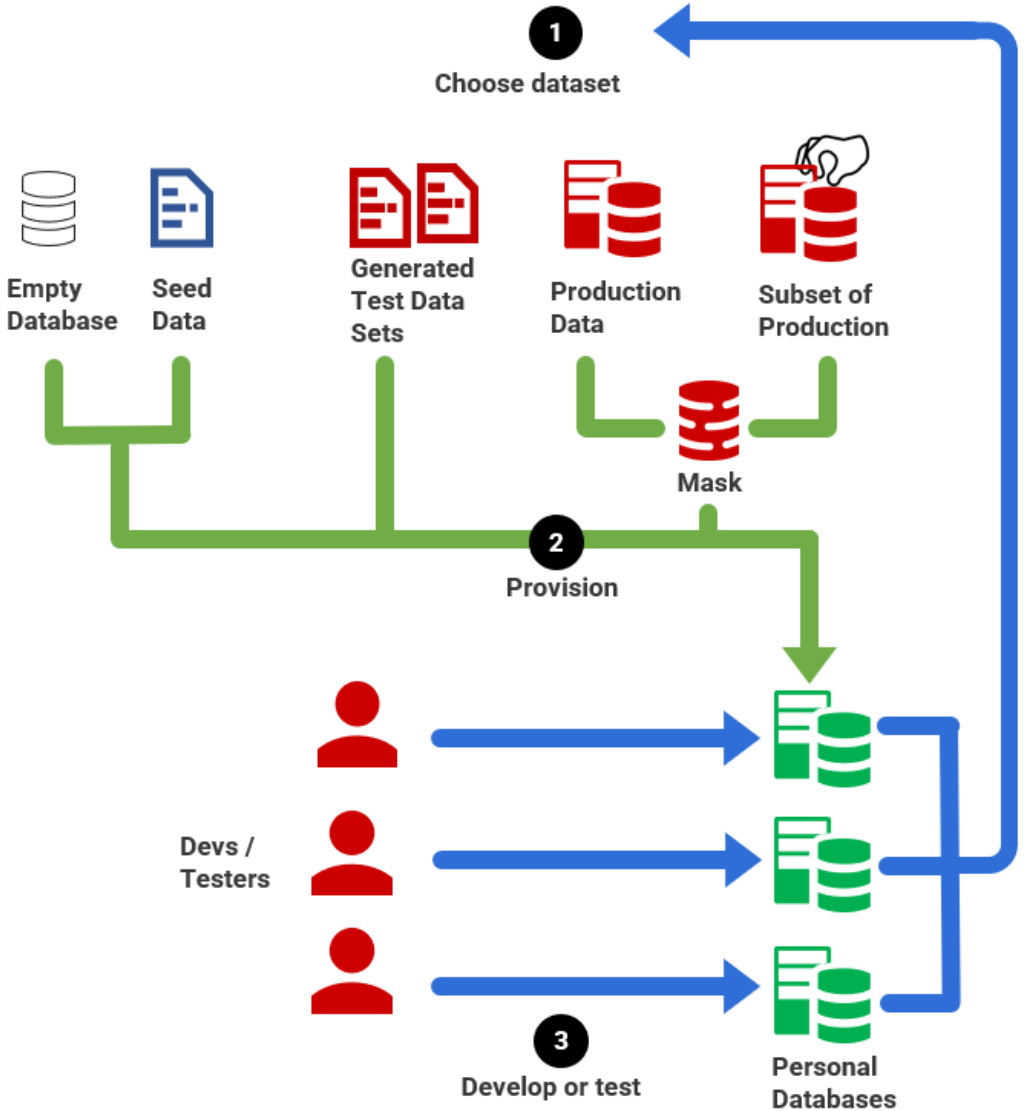
The database must be easy to create and, just as
Over, and over again.
Every feature should be developed and tested ag

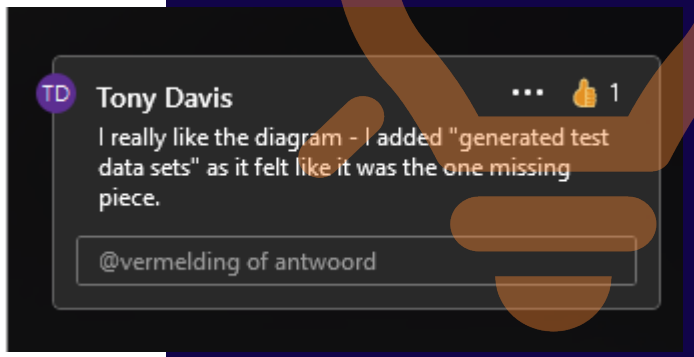
With your own database environment in place, it
data! What data and how much is really needed

That depends on the use case. Sometimes all the
subset of the data. If you're developing app whic
using the production data might be the best choi
masked (obfuscated), we want to prevent GDPR

Creating scripts to insert the exact amount of dat
so-called subsets of data can be:

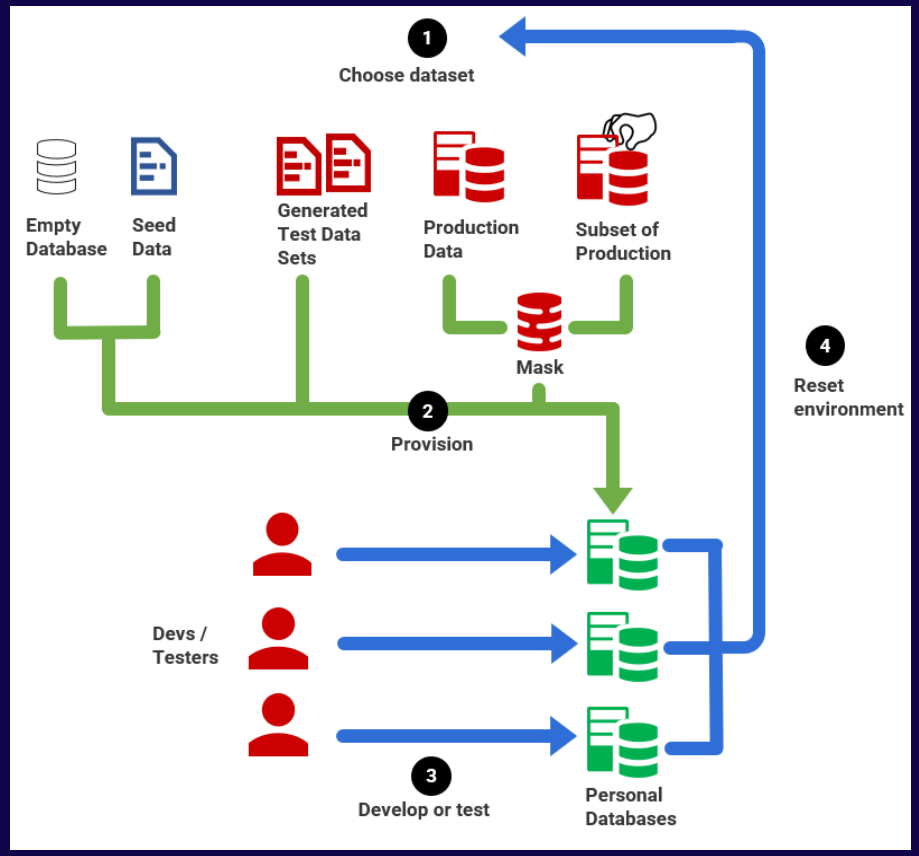
- Written, and maintained (!), during the d

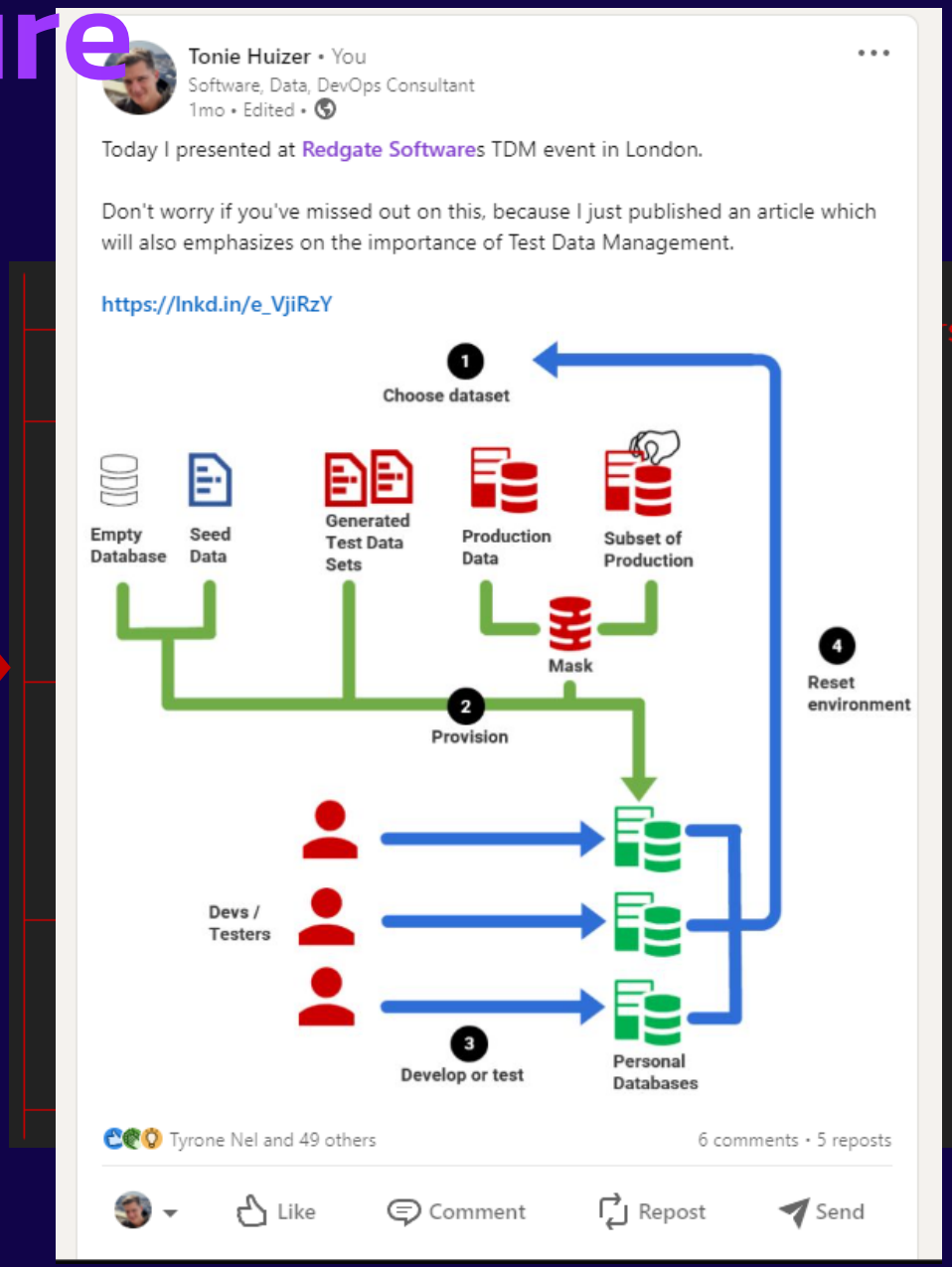




Develop a structure

- ✓ Write down your idea
- ✓ Pin down your idea
  - ✓ Purpose
  - ✓ Audience
  - ✓ Story
- ✓ Structure
- • Write or present





Informal idea discovery

"A blog about tracing Power BI datasets, and debugging DAX performance, so we can do tuning and avoid throttling in the PBI environment"

Problem, Story ↓ *+ Who, Why, When*

PBI developers need a way to spot report killers *before* they crush the live service!

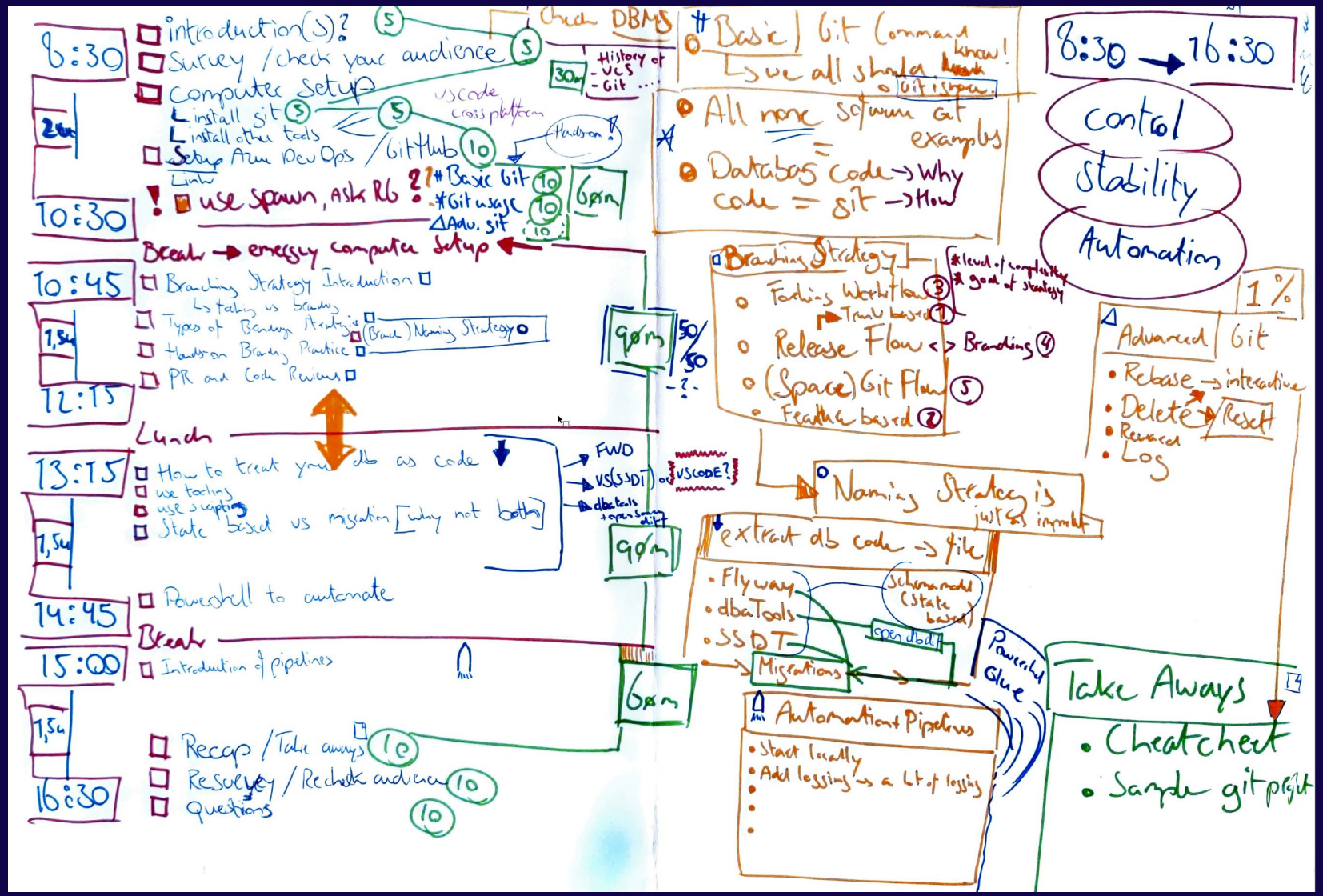
"My team of devs recently ran a single report that caused such severe throttling that it brought down an entire P3 Premium Capacity PBI service. It had 100 measures, nested up to 15 levels. Trying to rewrite it was not fun".

Audience, Purpose ↓ *+ Who, What, How*

Give PBI developers a way to audit and manage performance of their reports

1. **How to set up customizable SQL Traces for large PBI datasets**
2. **How to Identify DAX queries using most resources on the PBI service**
3. **How to tune queries on prem using tools within DAX Studio and Power BI Desktop**

Idea Discovery for your presentation!



Structure

08:30
- Intro
- Know the audience (survey)
- **Get set for control**
  - Git, GitHub
  - Git tools
  - Azure DevOps

10:30

Break -> emergency laptop setup

10:45
- You need a Branching strategy
- Branching best practices
- Importance of PR and Code reviews

12:15

Lunch

13:15
- How to treat your DB as code
  - Use tooling
  - Use scripting

14:45
- State-based, migrations, both?

Break

15:00
- Intro to pipelines
- Building Automation pipelines

....etc....

Story

Working with Git

- Git commands you need to know
- Get a Git space
- DB code = Git code

Branching strategies

- Forking walkthrough
- Release flow / Gitflow
- feature branching
- Naming is important

Extract DB code -> file

- Flyway – schema model
- dbaTools
- SSDT

Automation Pipelines

- Start locally
- You will need **logging**

08:30 –> 16:30

Control

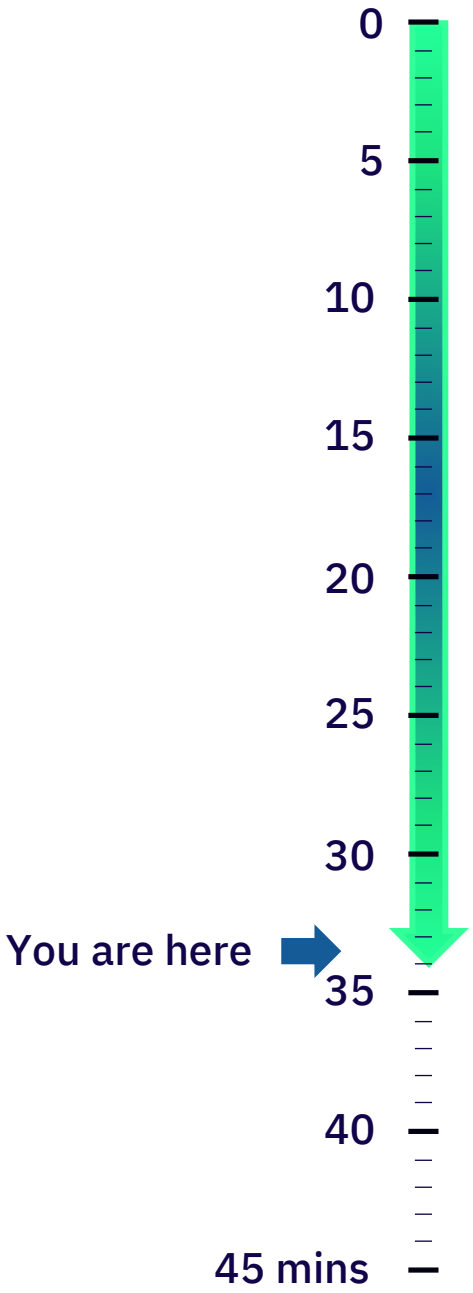
Stability

Automation

Purpose

Takeaways

- Cheat-sheet
- Sample git project



1. Motivations for sharing knowledge

2. What's stopping you?

3. Idea Discovery: Audience, Problem, Purpose, Story

4. Amplify! Deliver your ideas with maximum impact

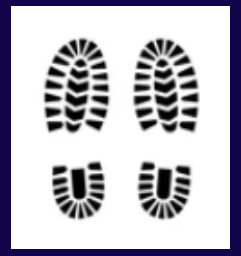
More Wise Words of Phil Factor

"If you can't be brilliant, be clear and simple. Your readers will be grateful and continue to read."

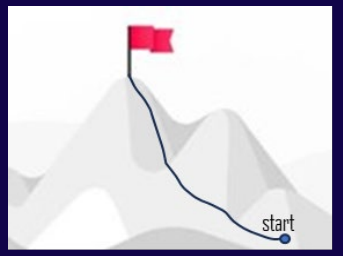
Remember your audience

1. Audience first, ego second

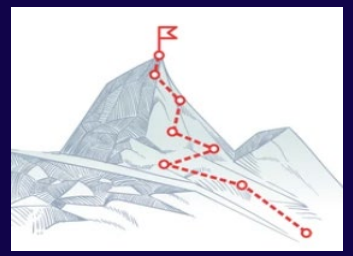
Amplify your reach



Audience



Purpose



Story

2. Start Strong

The reader will give you a narrow window of attention: don't miss it

Explain very succinctly what you have to say and why the reader should persist in reading

Is this relevant?

Is this moving the narrative forward?

3. Stay on track

Don't digress or get sidetracked

Simple logical train of thought – every paragraph has a point

Never skip an important part of the technical information

Refine for Impact and Clarity

Be clear, precise but never dull

Quicksand

"As a rule, non-production personnel should not have access to controlled or sensitive environments using any database role that has elevated administrative permissions..."

Concrete

"There's no good reason for a developer to have sysadmin on Staging..."

Refine for Impact and Clarity

No periphrastic perambulations

Quicksand

"Leveraging Business Intelligence solutions for actionable insights is game-changing for modern enterprises, but an effective data-driven, decision-making framework requires large volumes of data for accurate trending and forecasting"

Concrete

"Business Intelligence solutions help companies make better decisions by analysing large data sets to identify current trends and predict future ones..."

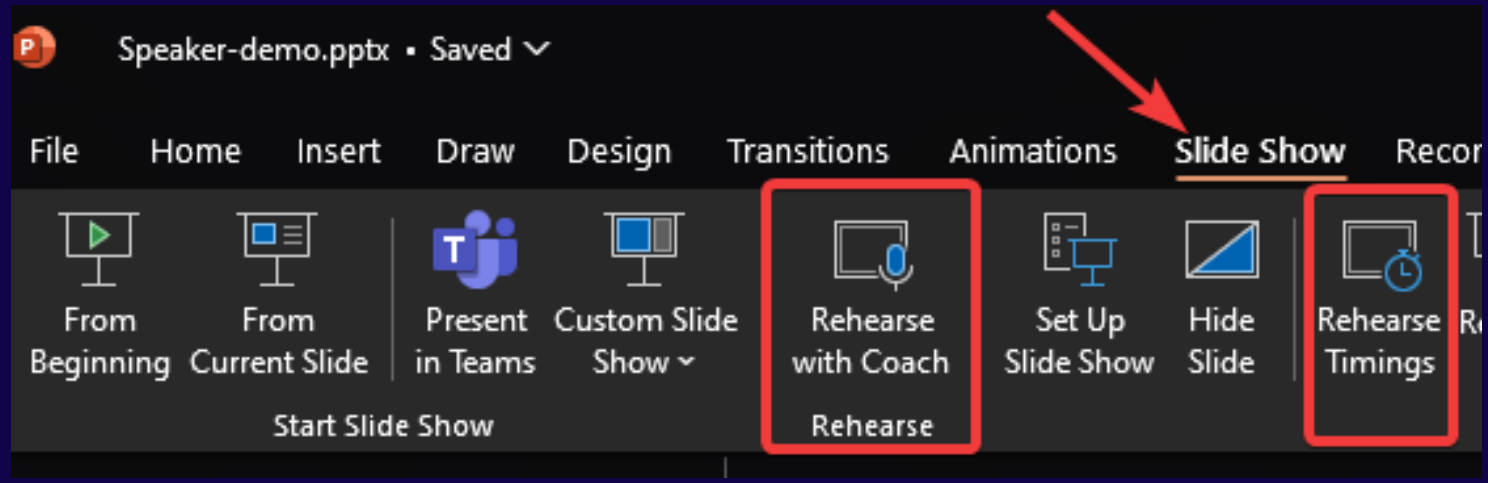
Refine for Impact and Clarity

Keep it plain and simple!

Quicksand		**Concrete**
"Thingamajig" words	→	**Terms used for their precise meaning**
Confusing variation in terminology	→	**Consistent, specific**
"Vague Handwaving"	→	**Concrete examples**
Hyperbole/buzzwords	→	**Proof of value**
Formal, textbook language	→	**Natural, conversational style**
Twisted colloquialisms	→	**Plain, International English**

**** See also: Strunk and White, Steven Pinker or George Orwell – they all agree!***

Refine for Impact and Clarity



- **Rehearse your delivery**
- **Rehearse your timings!**
- **Speak slowly – esp. if English 2nd language**
- **Try to enjoy it – the audience wants you to succeed**

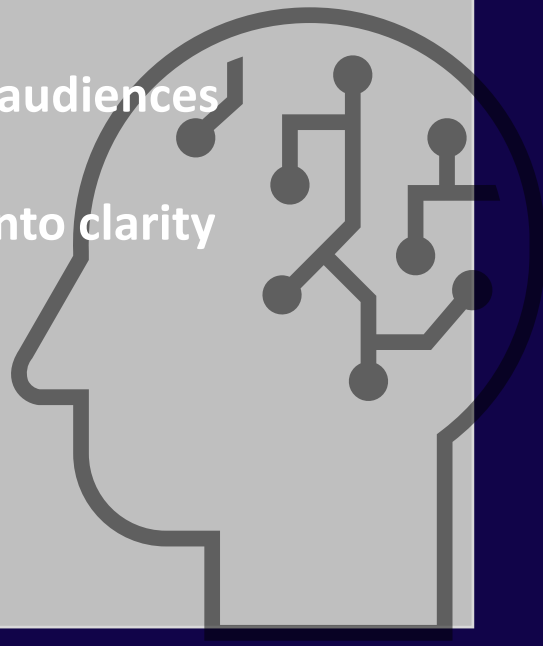
AI as a Companion not a Replacement

Your creative input

Audience, Purpose, Story
Experience & Judgment
Humor, Opinion, Subtlety

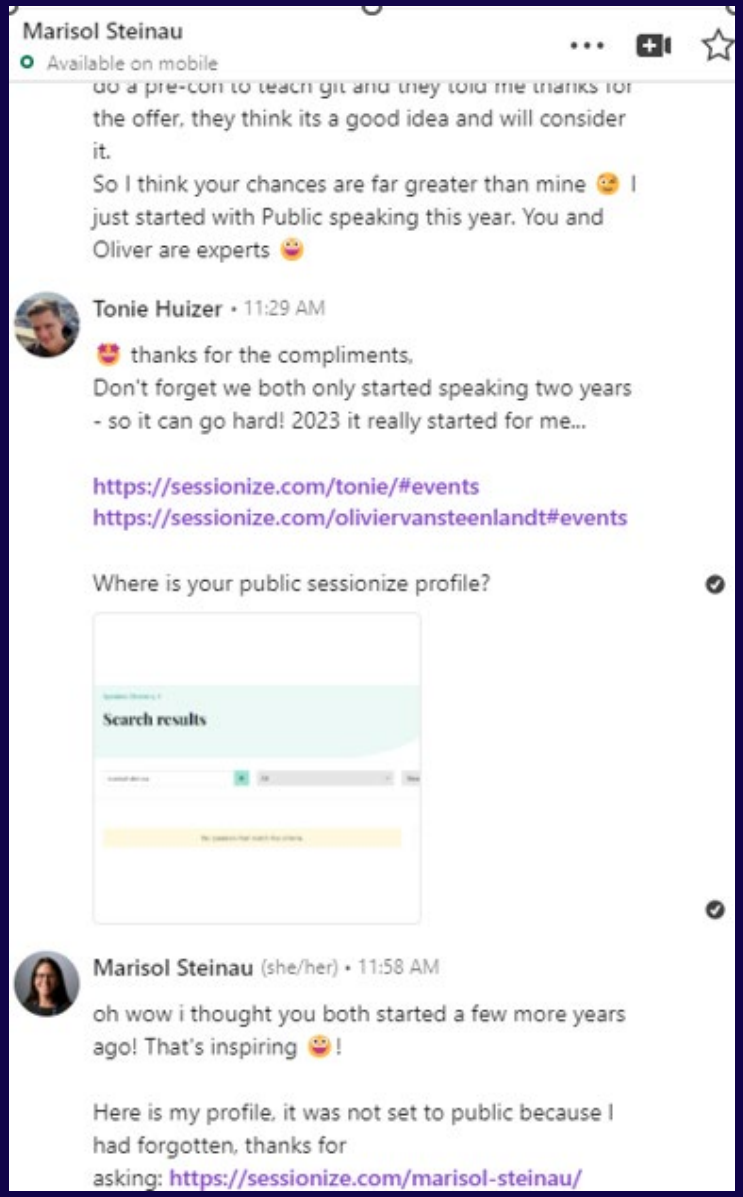
AI's guidance

Reframe for different audiences
Spot patterns & gaps
Turn rough thoughts into clarity



1. **Your insights drive the value**
   AI is only effective if guided by your knowledge of the audience, problem, and purpose
2. **Enhance not generate**
   Use AI to refine, simplify, and test your ideas
3. **People want you not AI**
   Keep your experience, humour, and perspective in the spotlight

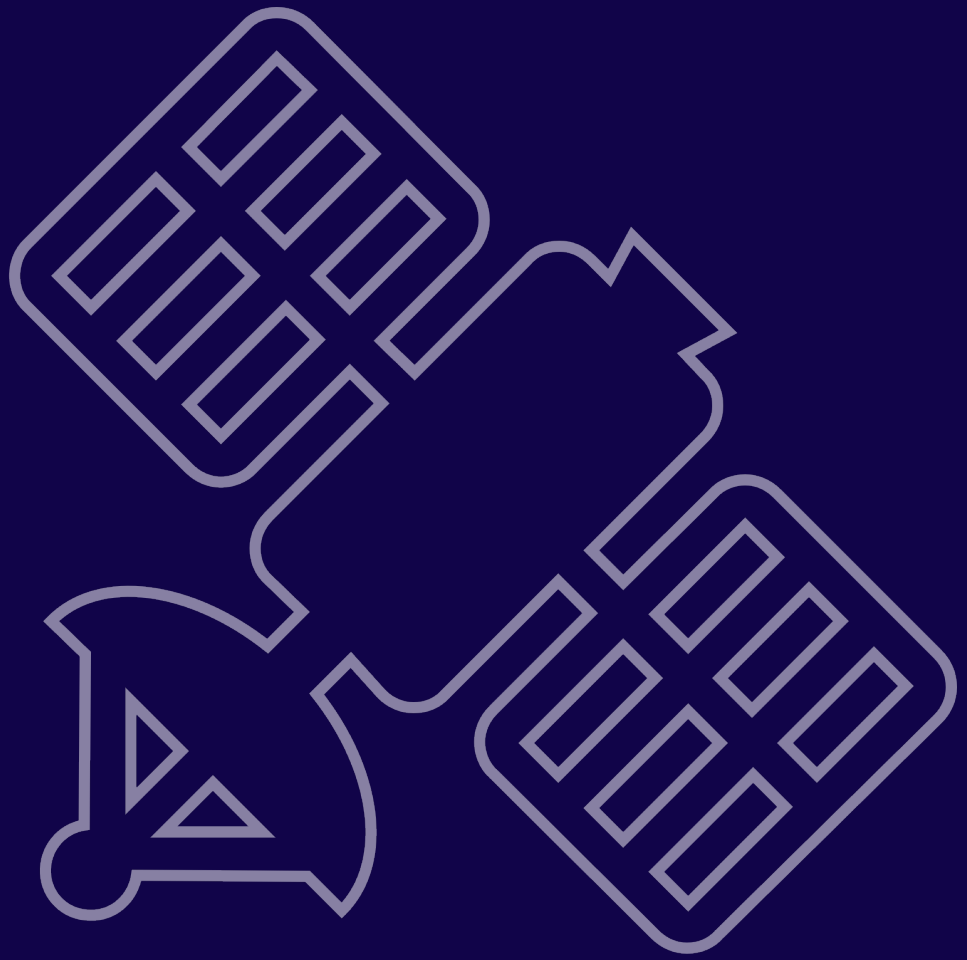
Amplify your profile



You must get comfortable being "out there"

Maybe *no one noticed* because you failed *to help them notice*!

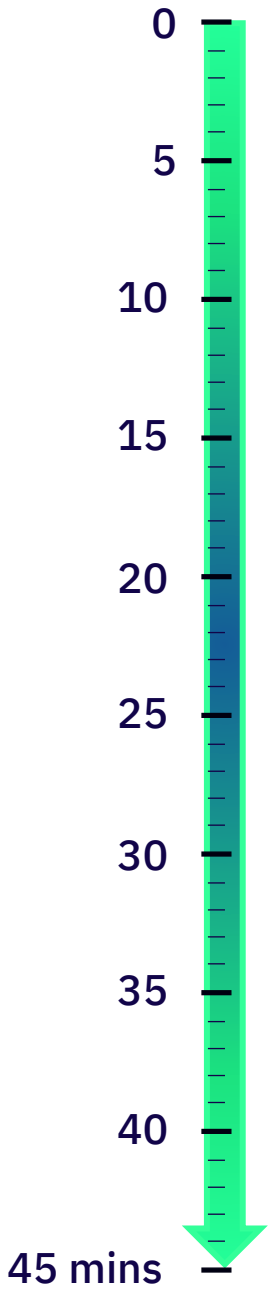
Amplify your reach



SEO ≠ content discovery

Clarity, purpose and structure will get your idea ‘noticed’ by AI

Your content must be where your audience is



1. Motivations for sharing knowledge

2. What's stopping you?

3. Idea Discovery: Audience, Problem, Purpose, Story

4. Amplify! Deliver your ideas with maximum impact

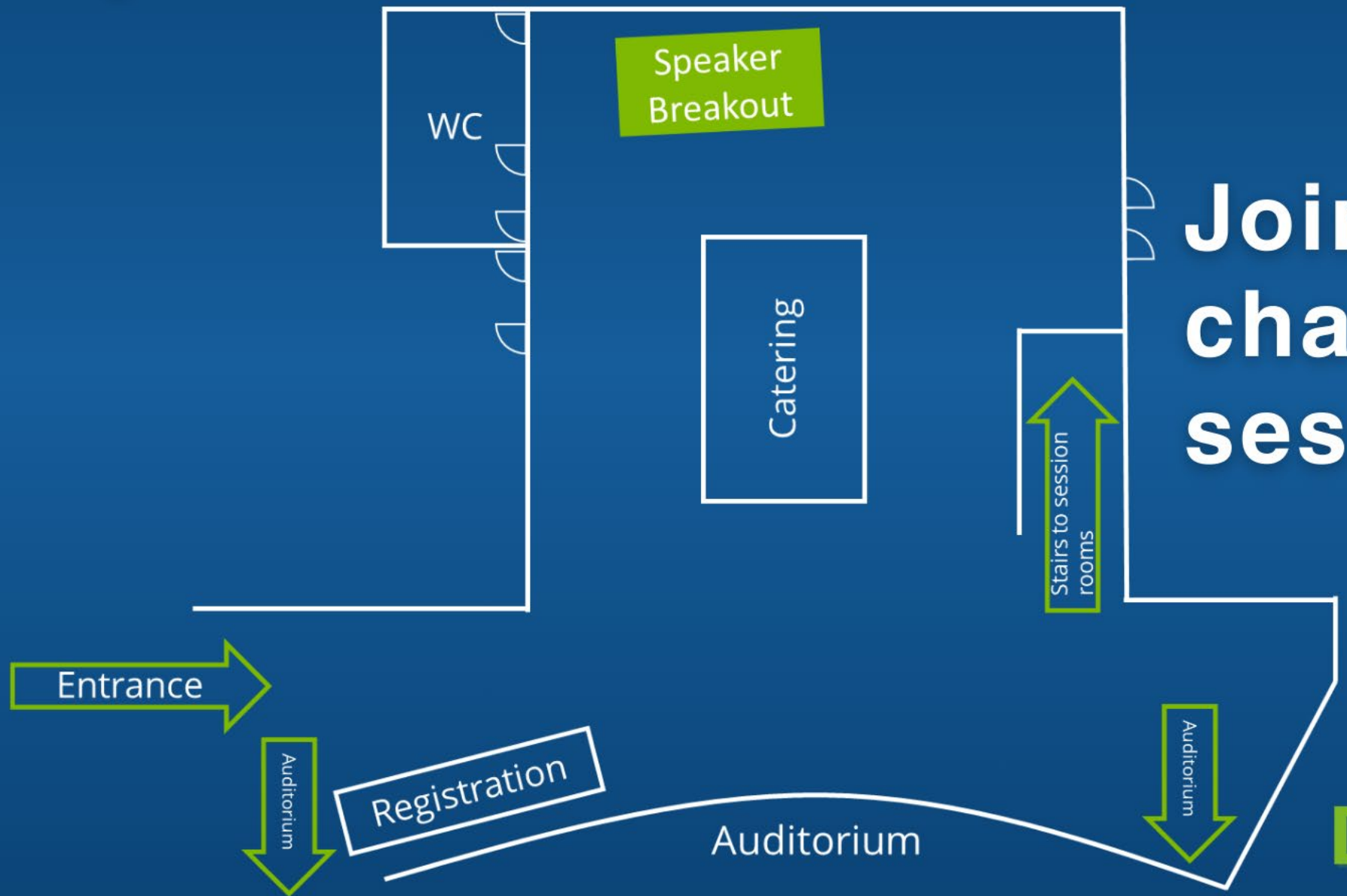
Share what you know!

Session Feedback

Event Feedback

DATA:Scotland

Speaker Breakout



Join me for a chat after this session

DATA:Scotland

DATA:Scotland

ADVANCING ANALYTICS

After Party

Brewdog, Merchant City | 5pm till late