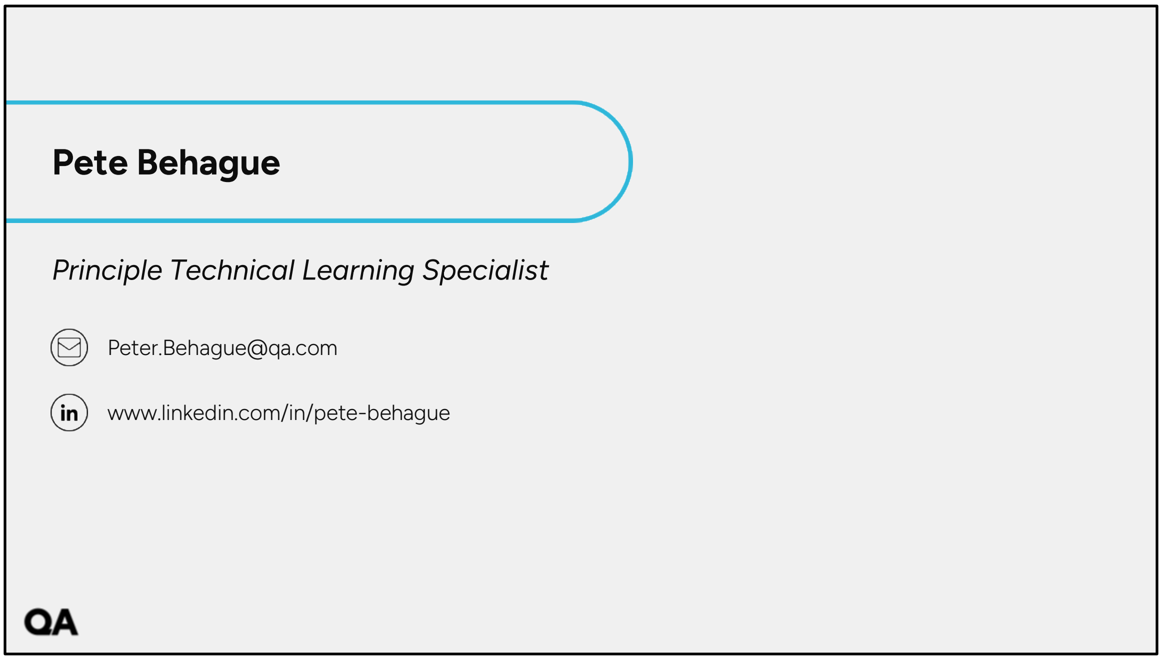
Learn. To Change.

SAMMAN TRAINING

Day 1 – Techniques and concepts



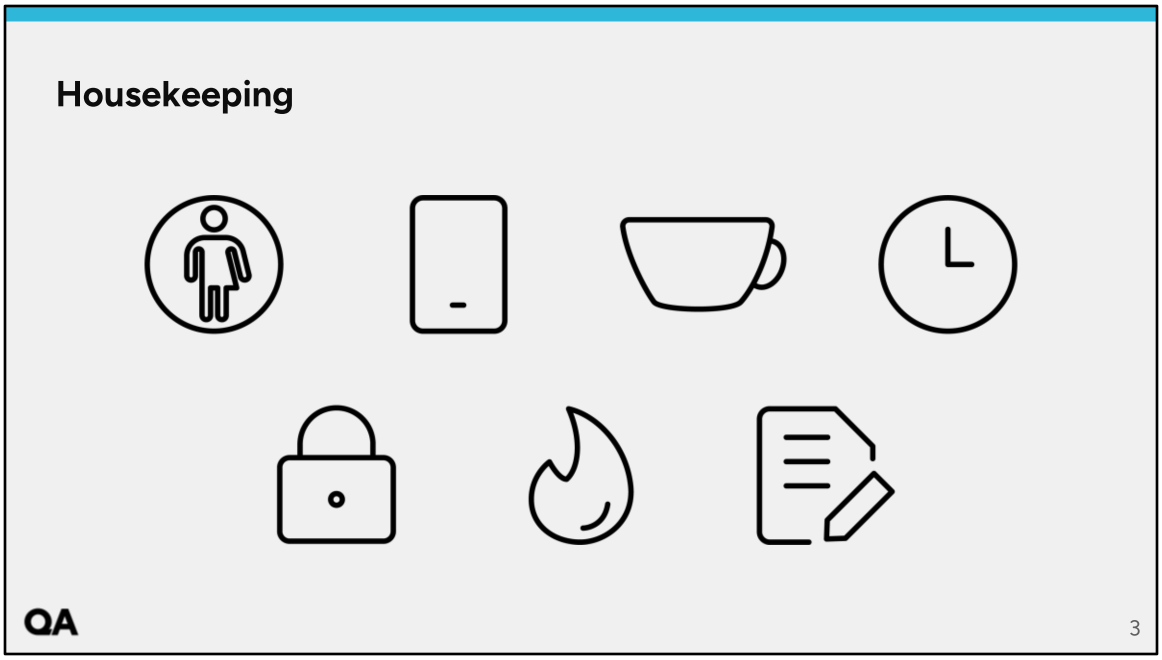
Pete Behague

Principle Technical Learning Specialist

Peter.Behague@qa.com

www.linkedin.com/in/pete-behague

QA



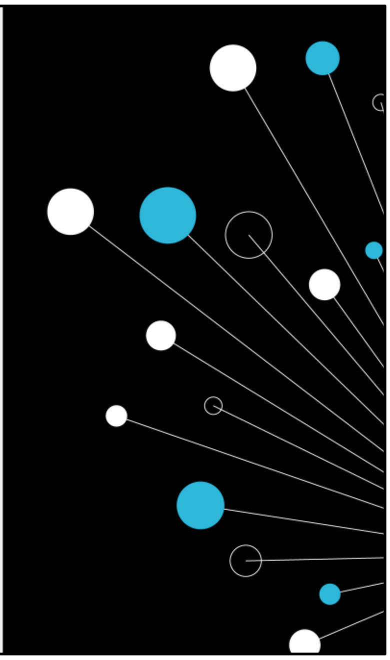
Housekeeping

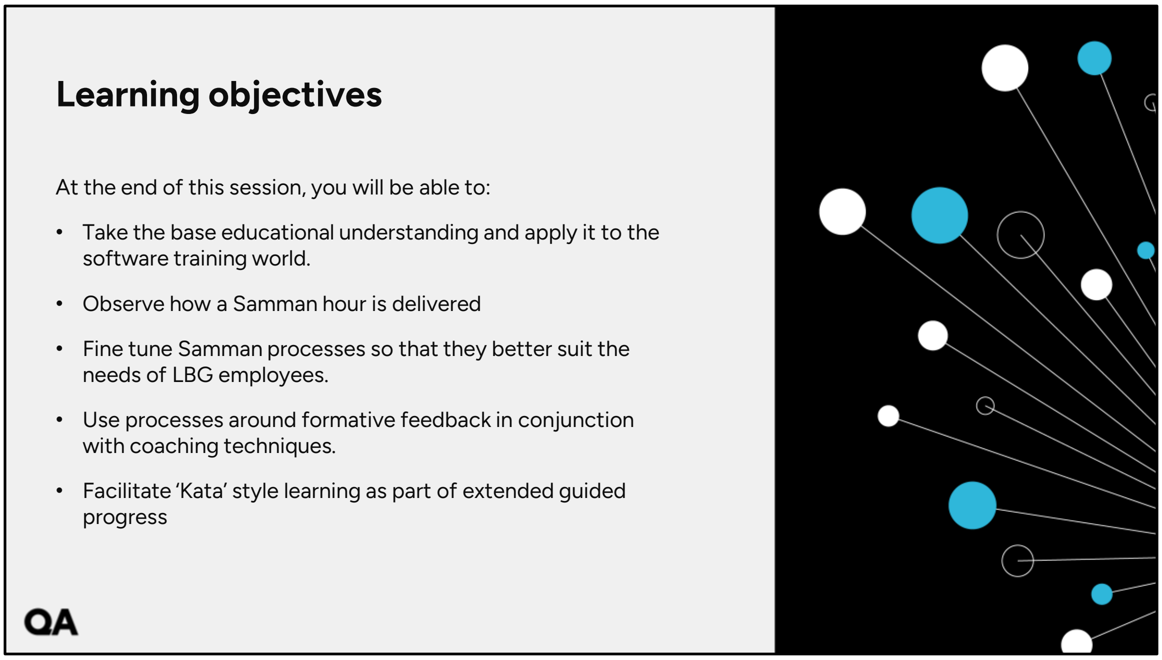
3

Programme Learning objectives

- Understand the value of the Samman training hour and how to apply it
- Understand how to apply the 4C training model – Connect, Concepts, Concrete Practice and Conclusions
- Understand how mentoring / coaching can help engineers to learn coding skills
- How to set individual and team technical goals
- Develop facilitation skills to run technical workshops on TDD, Refactoring in ensemble and paired set ups
- Be able to contextualise technical content to the team needs
- Learn how to embed continuous learning and experimentation in engineering work

QA



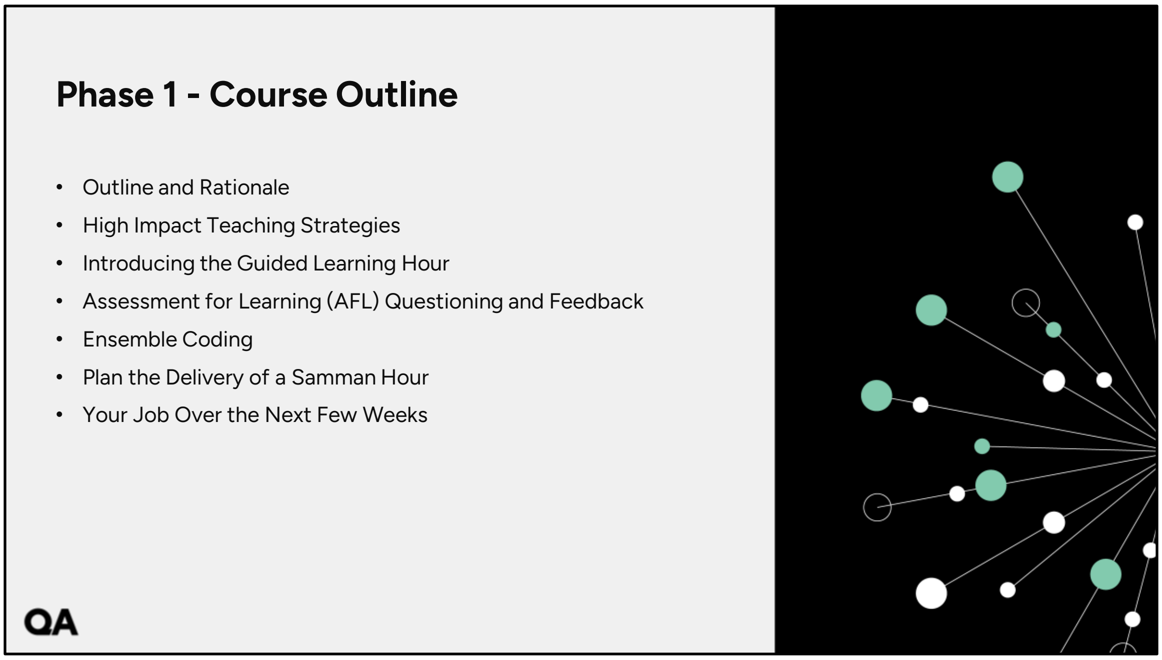


Learning objectives

At the end of this session, you will be able to:

- Take the base educational understanding and apply it to the software training world.
- Observe how a Samman hour is delivered
- Fine tune Samman processes so that they better suit the needs of LBG employees.
- Use processes around formative feedback in conjunction with coaching techniques.
- Facilitate 'Kata' style learning as part of extended guided progress

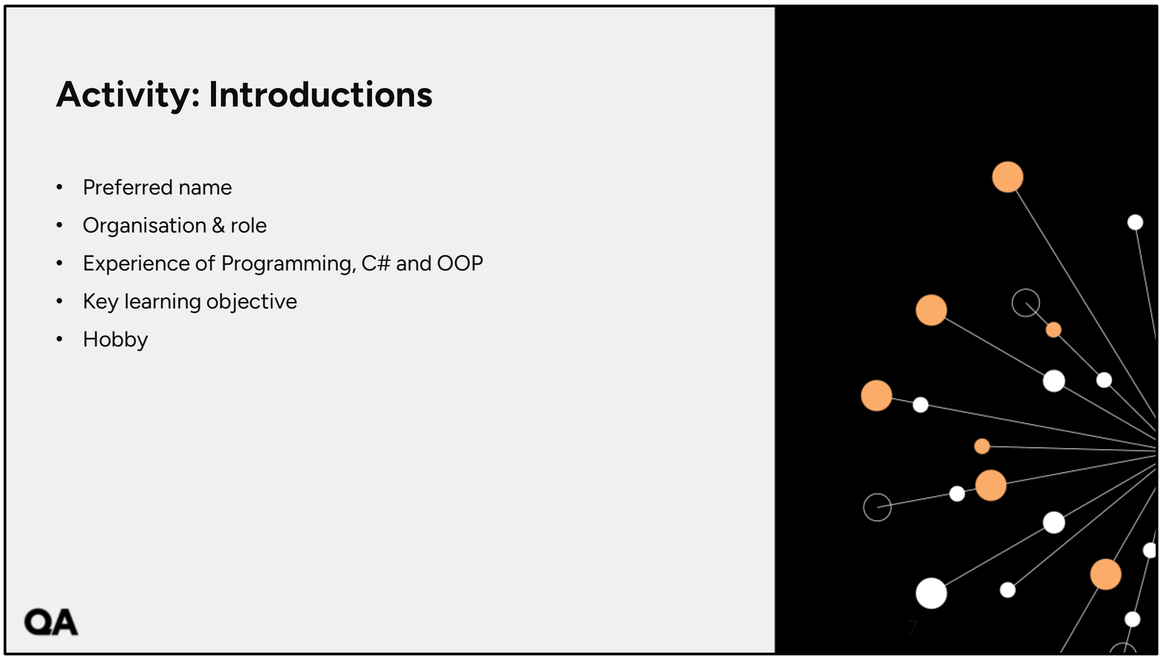
QA



Phase 1 - Course Outline

- Outline and Rationale
- High Impact Teaching Strategies
- Introducing the Guided Learning Hour
- Assessment for Learning (AFL) Questioning and Feedback
- Ensemble Coding
- Plan the Delivery of a Samman Hour
- Your Job Over the Next Few Weeks

QA



Activity: Introductions

- Preferred name
- Organisation & role
- Experience of Programming, C# and OOP
- Key learning objective
- Hobby

QA

Questions

Golden rule

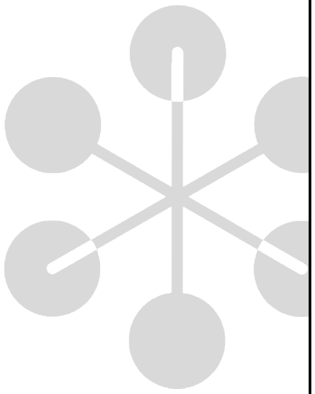
- ‘There is no such thing as a stupid question’

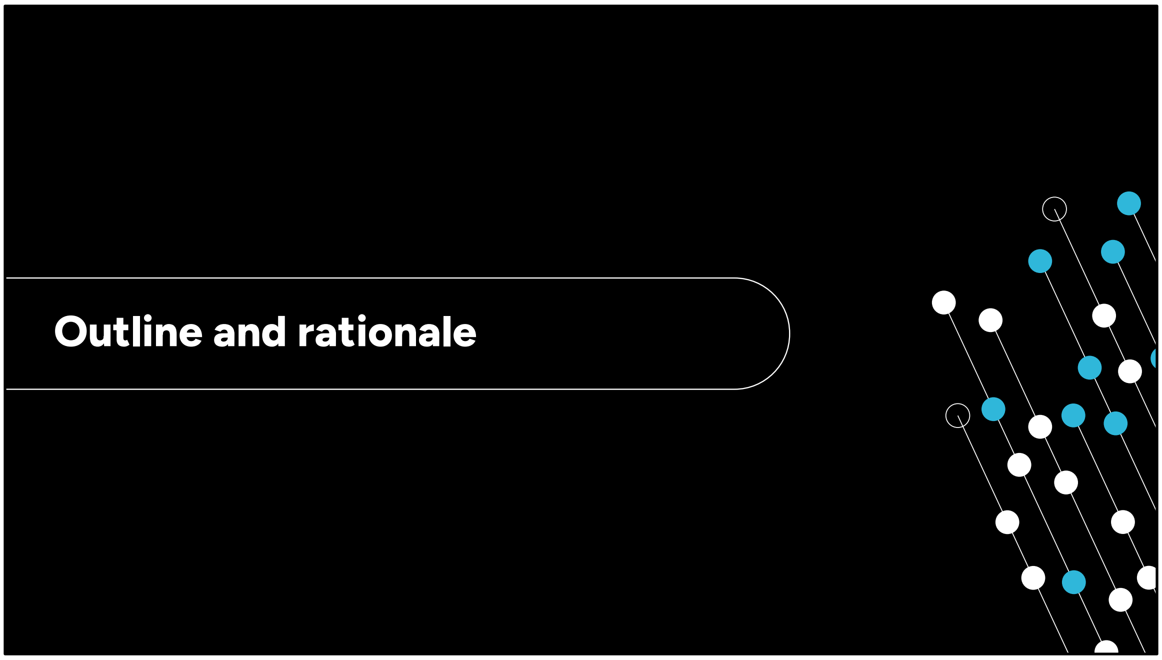
First amendment to the golden rule

- ‘… even when asked by an instructor’
- Please have a go at answering questions

Corollary to the golden rule

- ‘A question never resides in a single mind’
- By asking a question, you're helping everybody





Outline and rationale

What is Samman Technical Coaching

Technical coaches help software developers to adopt better coding practices. The focus is on the code and the way it is being written. A technical coach will themselves be a competent software developer able to mentor and teach specific coding practices. For example Test-Driven Development, Refactoring and the use of Design Patterns.

A technical coach works with an individual or team of developers to help them to achieve their goals. The coach does not usually write production code or tests directly. They contribute indirectly by helping others to do a better job of writing code. They facilitate, mentor and teach.

There may be other coaches working with the same software development team who have other competencies and focus areas. For example process, product, leadership or team dynamics. A technical coach may also coach those areas if they are also skilled in them

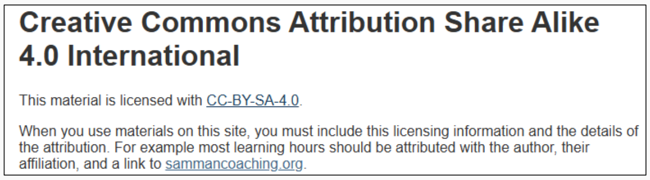
Emily Bache

Samman Technical Coaching
https://sammancoaching.org/

QA

License and Disclaimer

License: CC-BY-SA-4.0. Attribution: sammancoaching.org

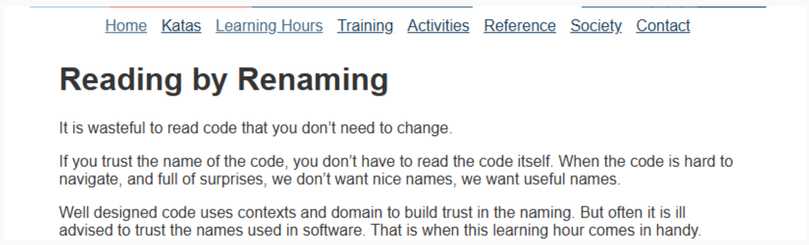

Creative Commons Attribution Share Alike 4.0 International

This material is licensed with CC-BY-SA-4.0.

When you use materials on this site, you must include this licensing information and the details of the attribution. For example most learning hours should be attributed with the author, their affiliation, and a link to sammancoaching.org.

Sammancoaching.org

Straight in with a Samman hour


Home Katas Learning Hours Training Activities Reference Society Contact

Reading by Renaming

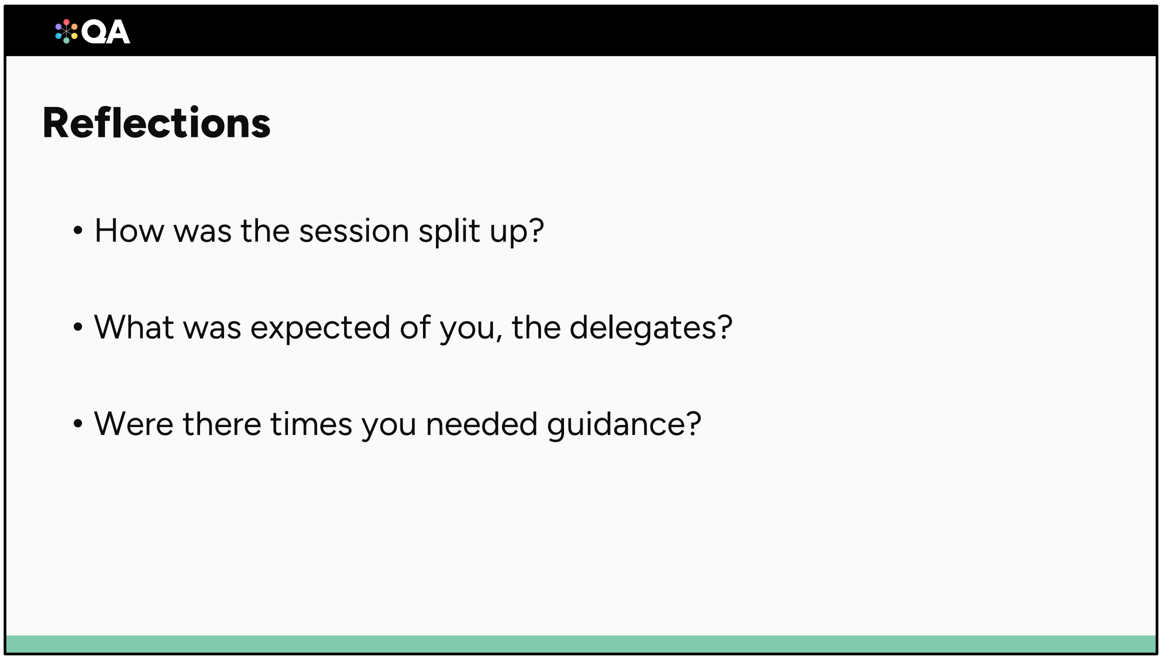
It is wasteful to read code that you don't need to change.

If you trust the name of the code, you don't have to read the code itself. When the code is hard to navigate, and full of surprises, we don't want nice names, we want useful names.

Well designed code uses contexts and domain to build trust in the naming. But often it is ill advised to trust the names used in software. That is when this learning hour comes in handy.

Reading by Renaming
https://sammancoaching.org/learning_hours/testable_design/naming.html

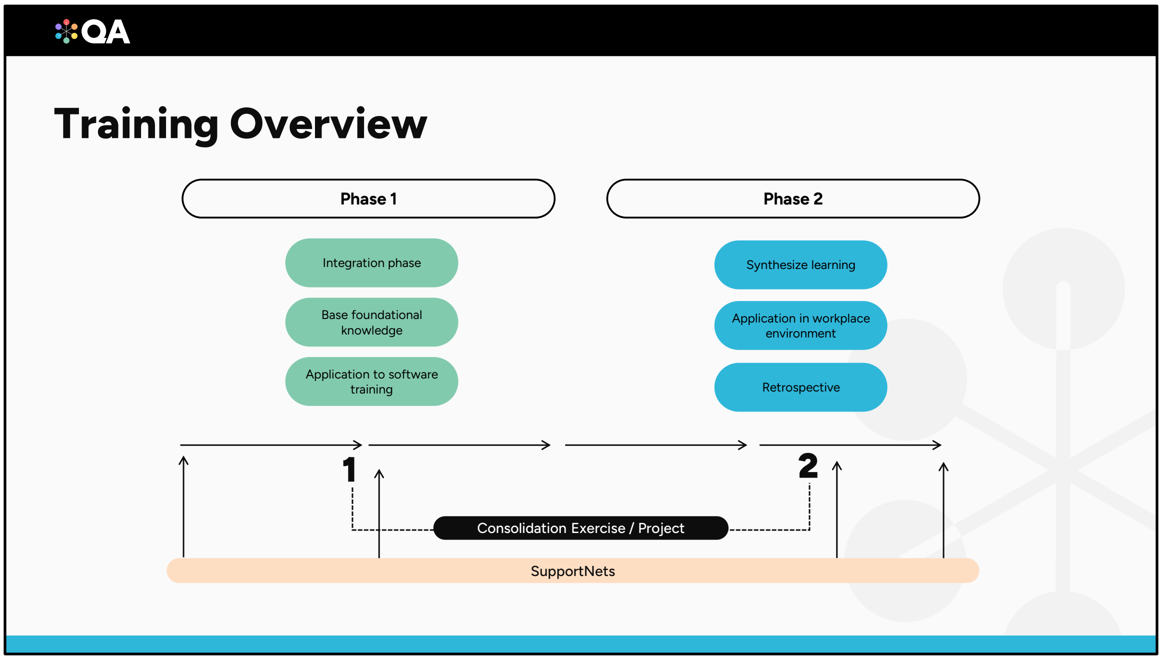
Naming as a Process (Article 1) | Deep Roots
https://www.digdeeproots.com/articles/naming-process/naming-as-a-process/



QA

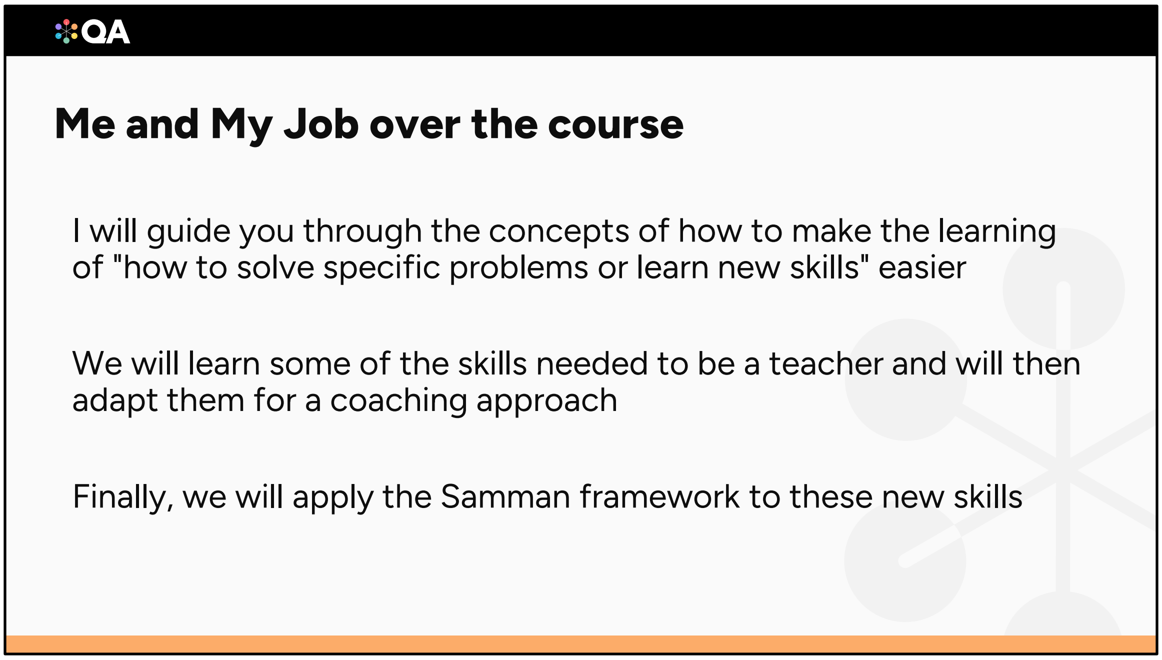
Reflections

- How was the session split up?
- What was expected of you, the delegates?
- Were there times you needed guidance?



Training Overview

Phase 1
Integration phase
Base foundational knowledge
Application to software training
Phase 2
Synthesize learning
Application in workplace environment
Retrospective
1
2
Consolidation Exercise / Project
SupportNets

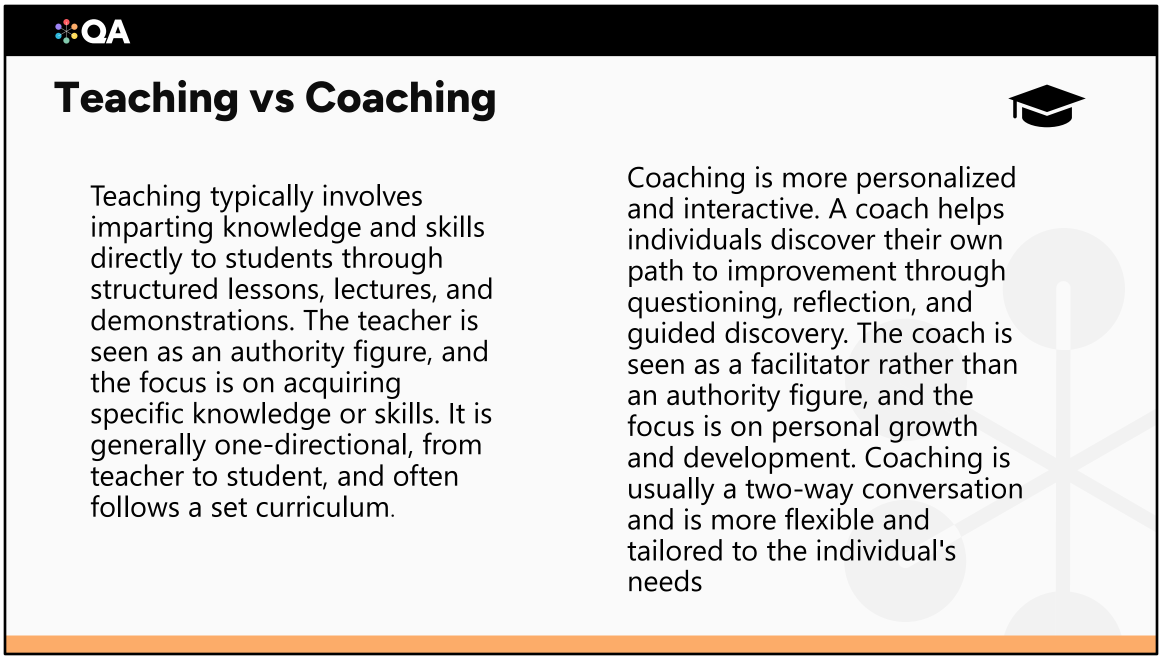


Me and My Job over the course

I will guide you through the concepts of how to make the learning of "how to solve specific problems or learn new skills" easier

We will learn some of the skills needed to be a teacher and will then adapt them for a coaching approach

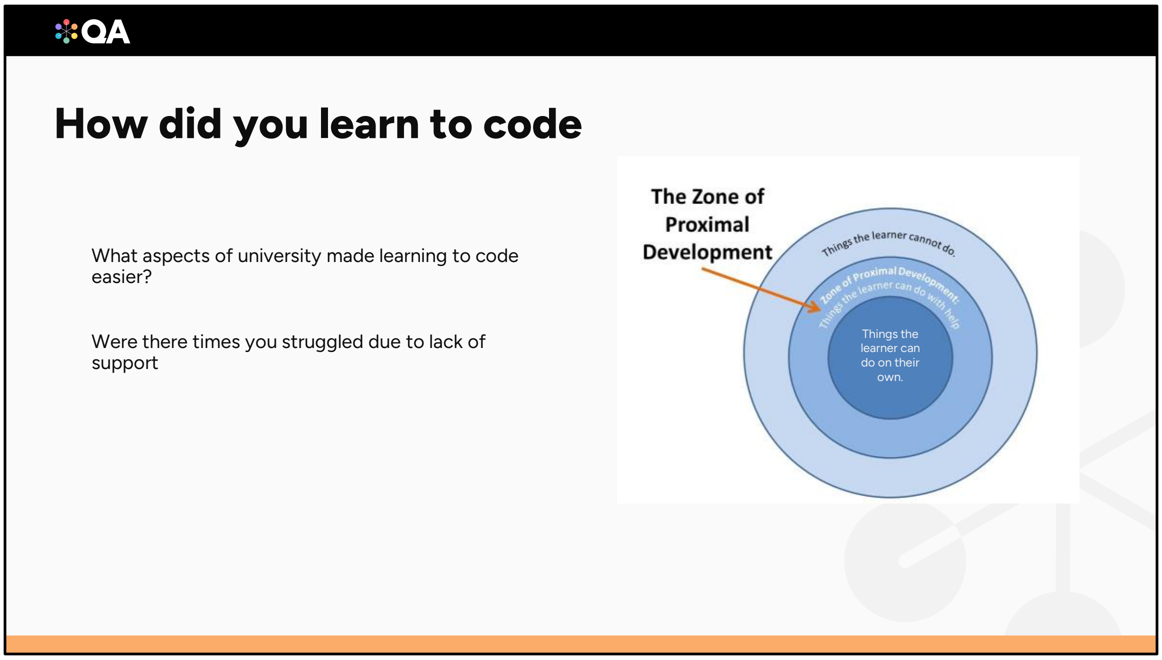
Finally, we will apply the Samman framework to these new skills



Teaching vs Coaching

Teaching typically involves imparting knowledge and skills directly to students through structured lessons, lectures, and demonstrations. The teacher is seen as an authority figure, and the focus is on acquiring specific knowledge or skills. It is generally one-directional, from teacher to student, and often follows a set curriculum.

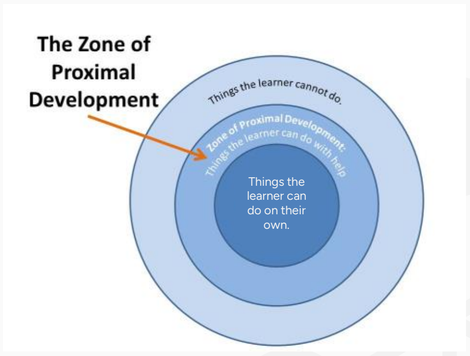
Coaching is more personalized and interactive. A coach helps individuals discover their own path to improvement through questioning, reflection, and guided discovery. The coach is seen as a facilitator rather than an authority figure, and the focus is on personal growth and development. Coaching is usually a two-way conversation and is more flexible and tailored to the individual's needs



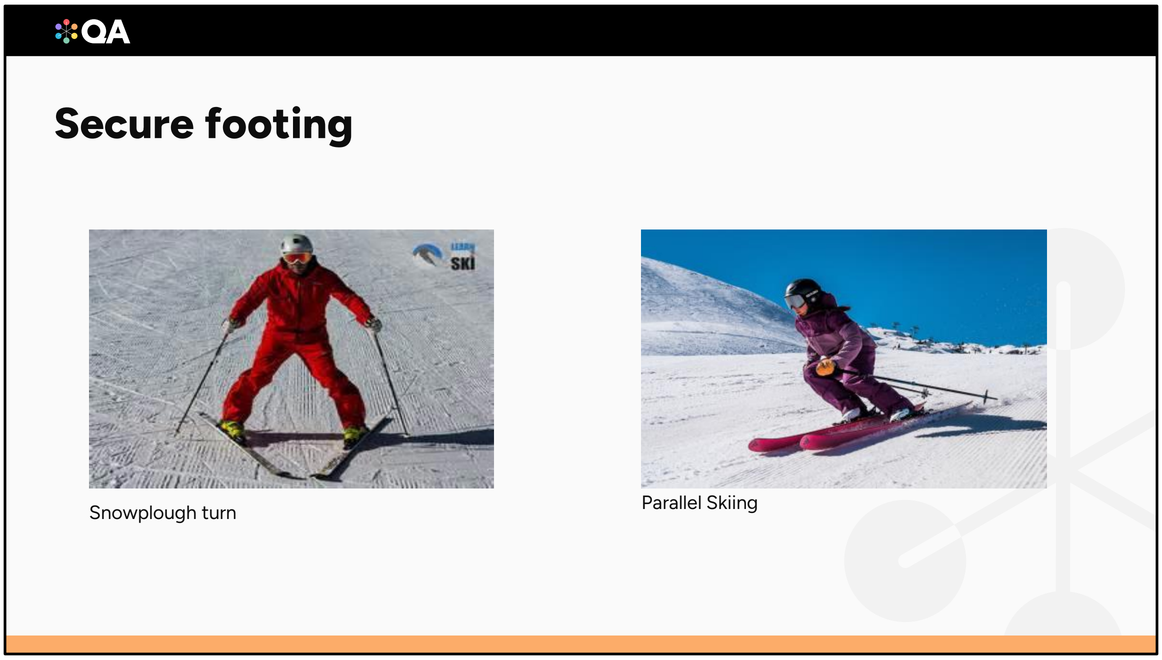
How did you learn to code

What aspects of university made learning to code easier?

Were there times you struggled due to lack of support


The Zone of Proximal Development
Things the learner cannot do.
Zone of Proximal Development: Things the learner can do with help
Things the learner can do on their own.

"Zone of proximal development"
Vygotsky's Theory of Cognitive Development in Social Relationships - Sprouts - Learning Videos - Social Sciences
There is a gap between what people can do on their own and what they can't do.



QA

Secure footing

Snowplough turn

Parallel Skiing

Emily Bache in a TED talk talks about learning to ski. So, Aim is to get someone from a basic level of ability to more advanced skills.
If you are interested:
Snowplough (or wedge turn):
In skiing, a snowplough turn is a basic technique used for controlling speed and changing direction on skis. It involves positioning the skis in a "V" shape, with the tips of the skis together and the tails spread apart, similar to a snowplow or pizza shape. By adjusting the pressure on each ski, a skier can control the speed and direction of the turn.

To progress from snowplough turns to parallel skiing, focus on gradually reducing the angle of your skis, shifting your weight to the outside ski, and practicing steering both skis together. Gradually decreasing the snowplough width allows for more speed and control, eventually leading to parallel turns.
Here's a step-by-step breakdown:
1. Start with a narrower snowplough:
Instead of a wide "V" shape, gradually bring your skis closer together, reducing the angle between them.
2. Weight Transfer:
Shift your weight to the outside ski (the one further down the hill) during the turn. This will allow the inside ski to rotate more easily and come into a parallel position.
3. Steer with your legs:
Imagine steering both skis as if they were arrows, pointing them in the direction you want to turn.
4. Practice edge control:

Learn to roll your ankles and knees to engage the edges of your skis effectively. This will help you initiate and control your turns.

5. Gradual Transition:

Don't try to force parallel turns right away. Start with a narrow snowplough and gradually bring your skis closer together as you gain confidence.

6. Practice, practice, practice:

The key to mastering parallel skiing is repetition and practice on varied terrain.

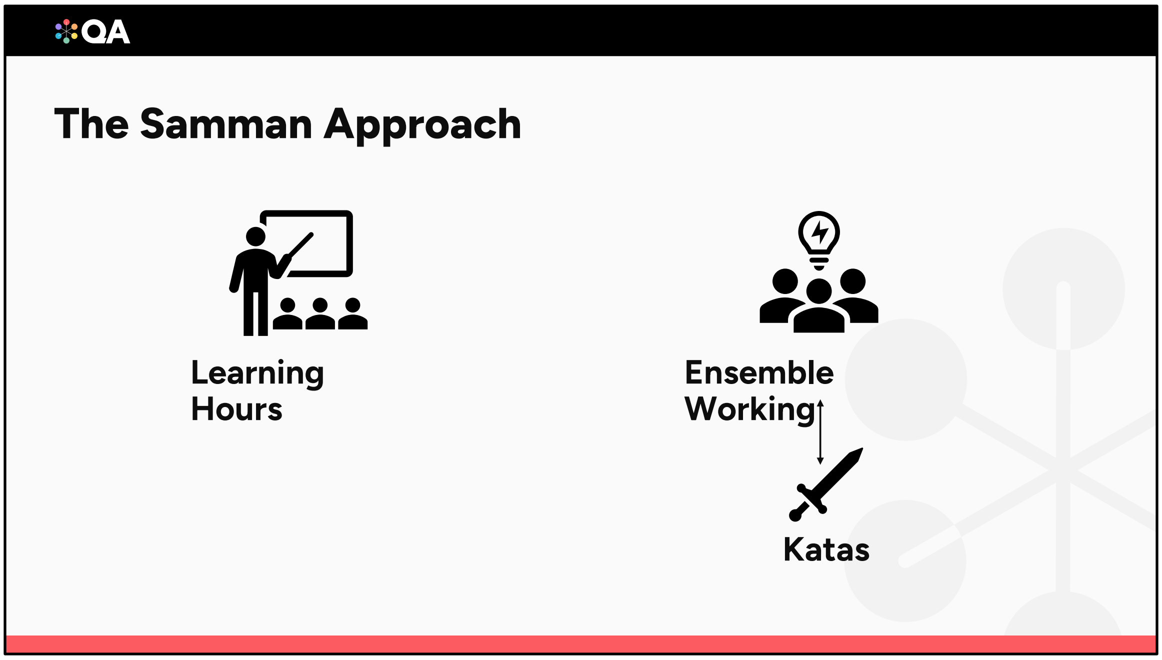
Tips for Success:

Maintain a forward stance: Keep your weight forward, over your shins, and avoid leaning back.

Look in the direction of the turn: This will help you maintain balance and steer effectively.

Use your poles to help with balance and rhythm: Plant your poles at the beginning of each turn to help with momentum and timing.

Be patient and persistent: It takes time and practice to progress from snowplough to parallel skiing. Don't get discouraged if you don't get it right away.



The Samman Approach

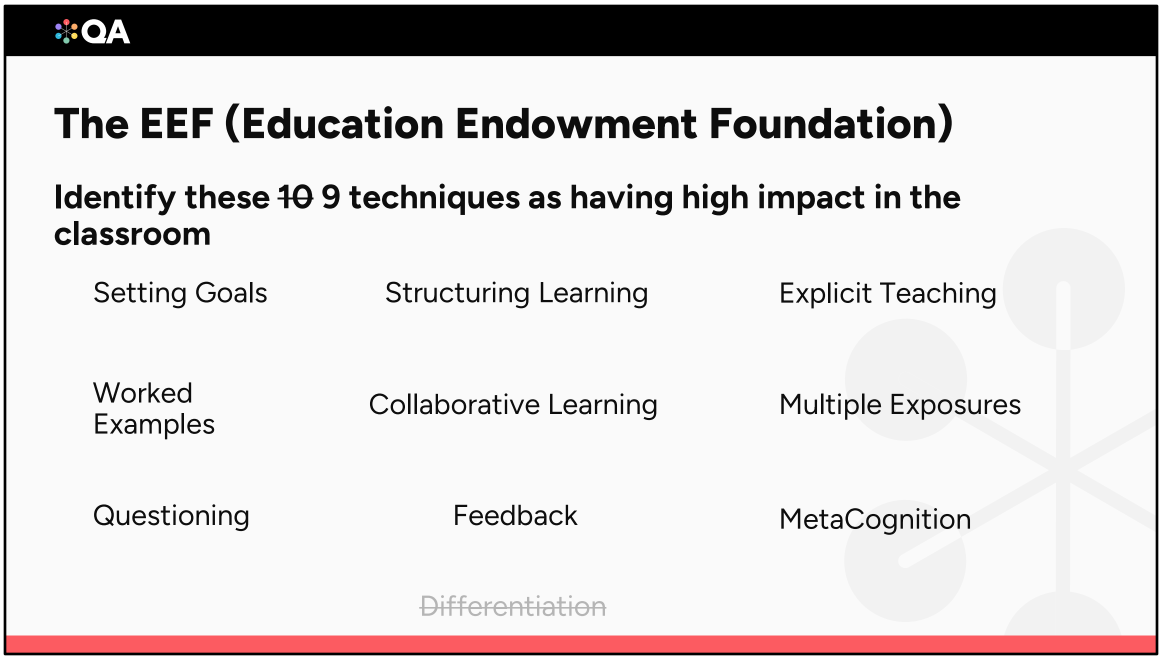
Learning Hours

Ensemble Working

Katas

High Impact Teaching Strategies

Being an effective Technical Coach



The EEF (Education Endowment Foundation)

Identify these ~~10~~ 9 techniques as having high impact in the classroom

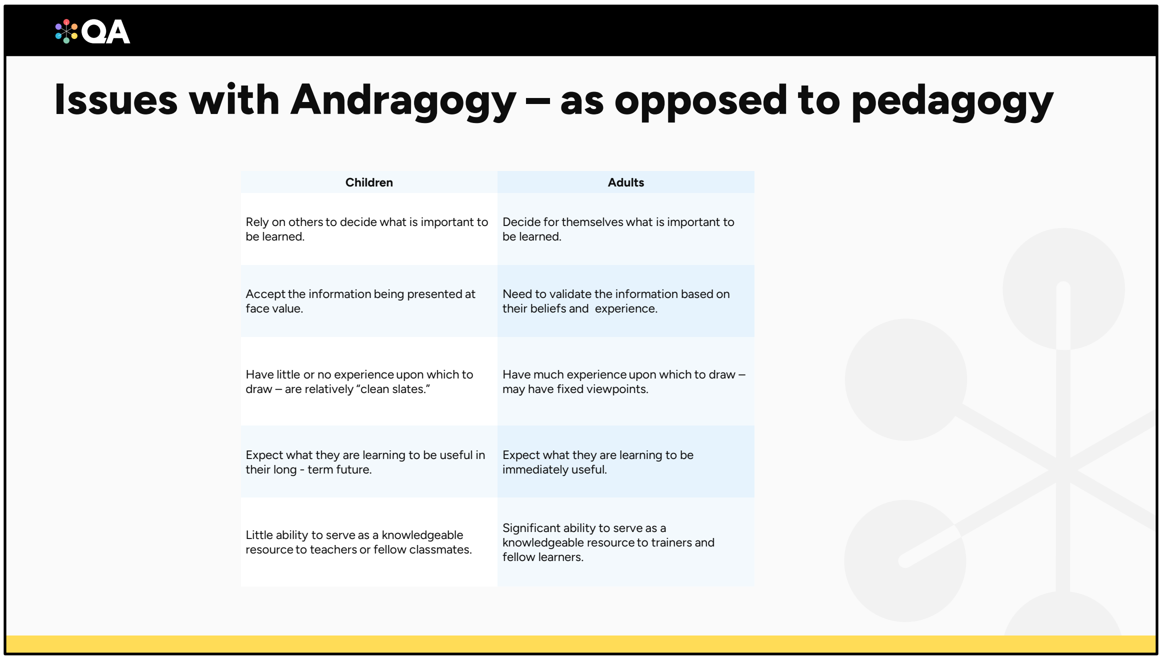
Setting Goals	Structuring Learning	Explicit Teaching
Worked Examples	Collaborative Learning	Multiple Exposures
Questioning	Feedback	MetaCognition

~~Differentiation~~

https://www.getatomi.com/blog/10-high-impact-teaching-strategies

Each of the techniques have research behind them that, amongst other things, has determined their **effect size** on learning. **Effect size** tells you how meaningful the relationship between variables or the difference between groups is. It indicates the practical significance of a research outcome.
While **statistical significance** shows that an effect exists in a study, **practical significance** shows that the effect is large enough to be meaningful in the real world. Statistical significance is denoted by *p* values, whereas practical significance is represented by effect sizes.
Statistical significance alone can be misleading because it's influenced by the sample size. Increasing the sample size always makes it more likely to find a statistically significant effect, no matter how small the effect truly is in the real world.
Effect size ranges: Small 0.2, Medium 0.5, Large 0.8 or greater

https://uploads.getatomi.com/schools/resources/atomi-teaching-with-impact-HITS-guide.pdf?hsCtaTracking=c9b400dc-5a04-4575-b8f6-c5bc483594ae%7C79964e12-d7bc-4e1f-a663-b97b37279e95 https://www.education.vic.gov.au/Documents/school/teachers/support/Expired/0000highimpactteachstrat-expired.pdf



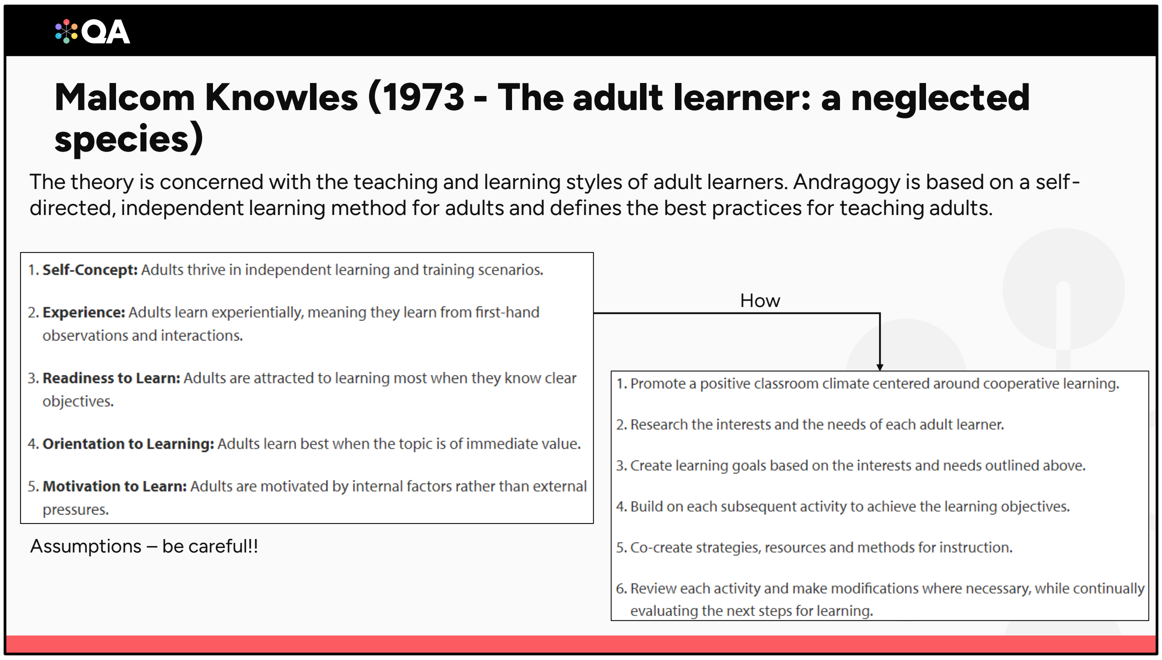
Issues with Andragogy – as opposed to pedagogy

Children	Adults
Rely on others to decide what is important to be learned.	Decide for themselves what is important to be learned.
Accept the information being presented at face value.	Need to validate the information based on their beliefs and experience.
Have little or no experience upon which to draw – are relatively "clean slates."	Have much experience upon which to draw – may have fixed viewpoints.
Expect what they are learning to be useful in their long - term future.	Expect what they are learning to be immediately useful.
Little ability to serve as a knowledgeable resource to teachers or fellow classmates.	Significant ability to serve as a knowledgeable resource to trainers and fellow learners.

Andragogy is the term used to describe a set of principles, methods and practices for teaching adult learners.
Pedagogy is the term used to describe principles and best practices for teaching children.

Source - https://pce.sandiego.edu/15-top-strategies-for-teaching-adult-learners-faqs/

Transition to andragogy occurs around age 16



Malcom Knowles (1973 - The adult learner: a neglected species)

The theory is concerned with the teaching and learning styles of adult learners. Andragogy is based on a self-directed, independent learning method for adults and defines the best practices for teaching adults.

1. **Self-Concept:** Adults thrive in independent learning and training scenarios.
2. **Experience:** Adults learn experientially, meaning they learn from first-hand observations and interactions.
3. **Readiness to Learn:** Adults are attracted to learning most when they know clear objectives.
4. **Orientation to Learning:** Adults learn best when the topic is of immediate value.
5. **Motivation to Learn:** Adults are motivated by internal factors rather than external pressures.

Assumptions – be careful!!

How

1. Promote a positive classroom climate centered around cooperative learning.
2. Research the interests and the needs of each adult learner.
3. Create learning goals based on the interests and needs outlined above.
4. Build on each subsequent activity to achieve the learning objectives.
5. Co-create strategies, resources and methods for instruction.
6. Review each activity and make modifications where necessary, while continually evaluating the next steps for learning.

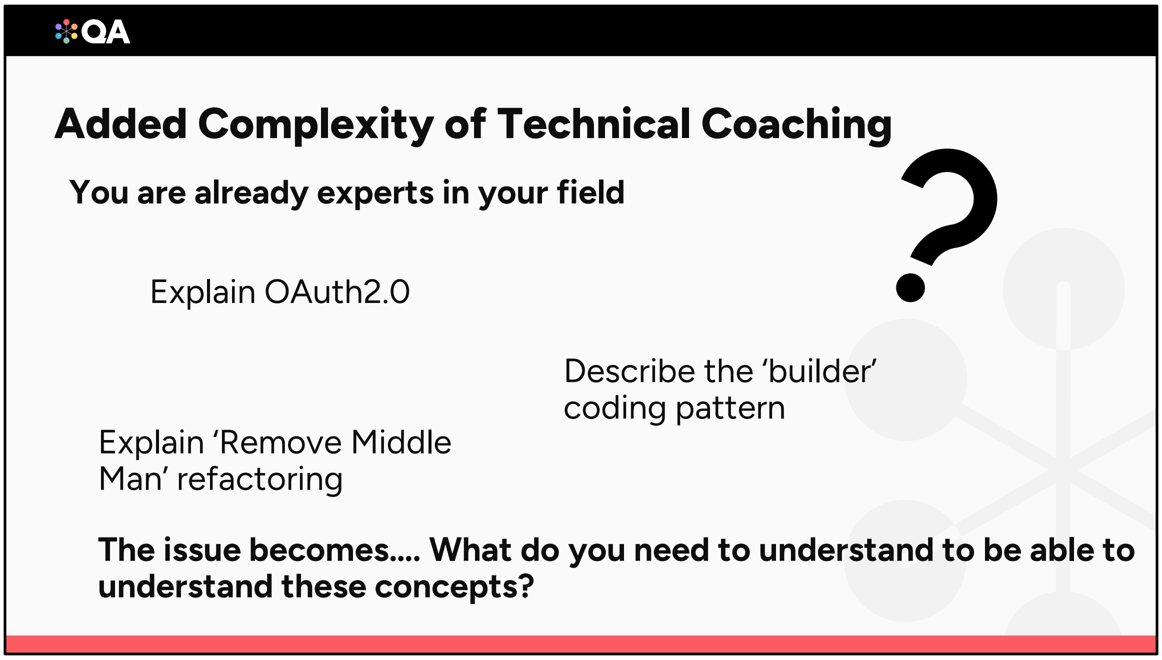
Malcom Knowles was (and still is) a big influence on educational learning.

Source - https://pce.sandiego.edu/15-top-strategies-for-teaching-adult-learners-faqs/

What is Andragogy? Less Than 100 Words - Roundtable Learning
https://roundtablelearning.com/what-is-andragogy-less-than-100-words/#:~:text=Characteristics%20of%20andragogy%20are%20covered%20in%20the%20following%20five%20assumptions.

For more detail on current version of the book's content see:
https://www.taylorfrancis.com/books/mono/10.4324/9780429299612/adult-learner-malcolm-knowles-elwood-holton-iii-richard-swanson-petra-robinson-malcolm-knowles-elwood-holton-iii-richard-swanson-petra-robinson

10 Simple Principles of Adult Learning
https://www.wgu.edu/blog/adult-learning-theories-principles2004.html



Added Complexity of Technical Coaching

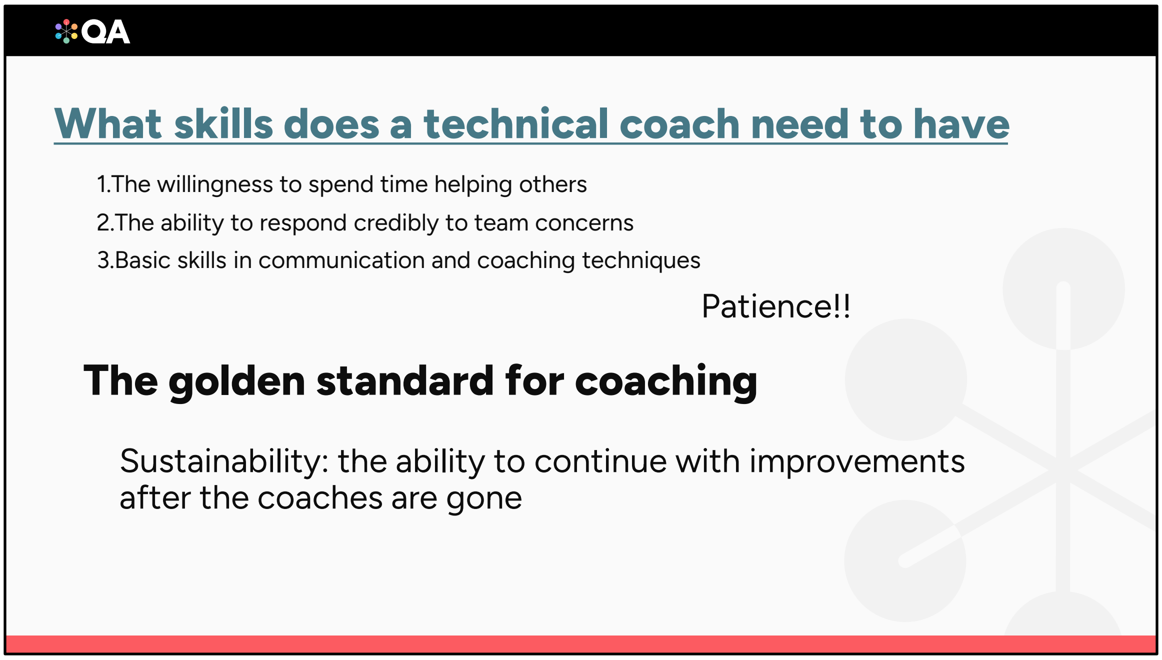
You are already experts in your field

Explain OAuth2.0

Describe the 'builder' coding pattern

Explain 'Remove Middle Man' refactoring

The issue becomes…. What do you need to understand to be able to understand these concepts?



QA

What skills does a technical coach need to have

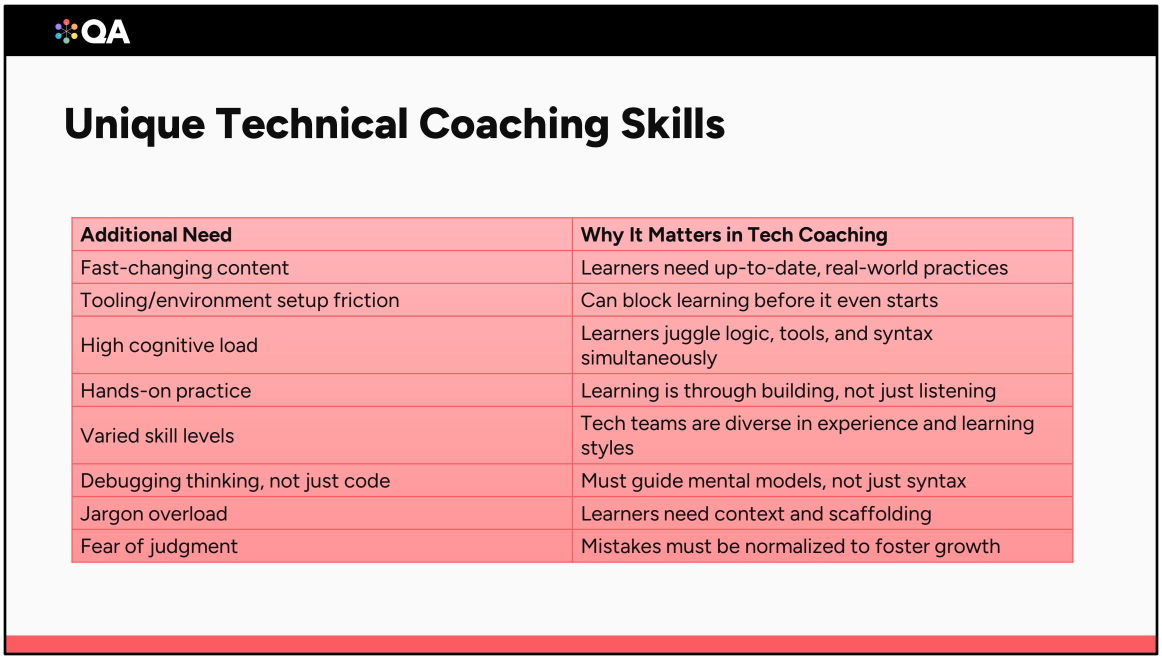
1. The willingness to spend time helping others
2. The ability to respond credibly to team concerns
3. Basic skills in communication and coaching techniques

Patience!!

The golden standard for coaching

Sustainability: the ability to continue with improvements after the coaches are gone

Source: https://davenicolette.wordpress.com/2015/04/10/what-skills-does-a-technical-coach-need-to-have/

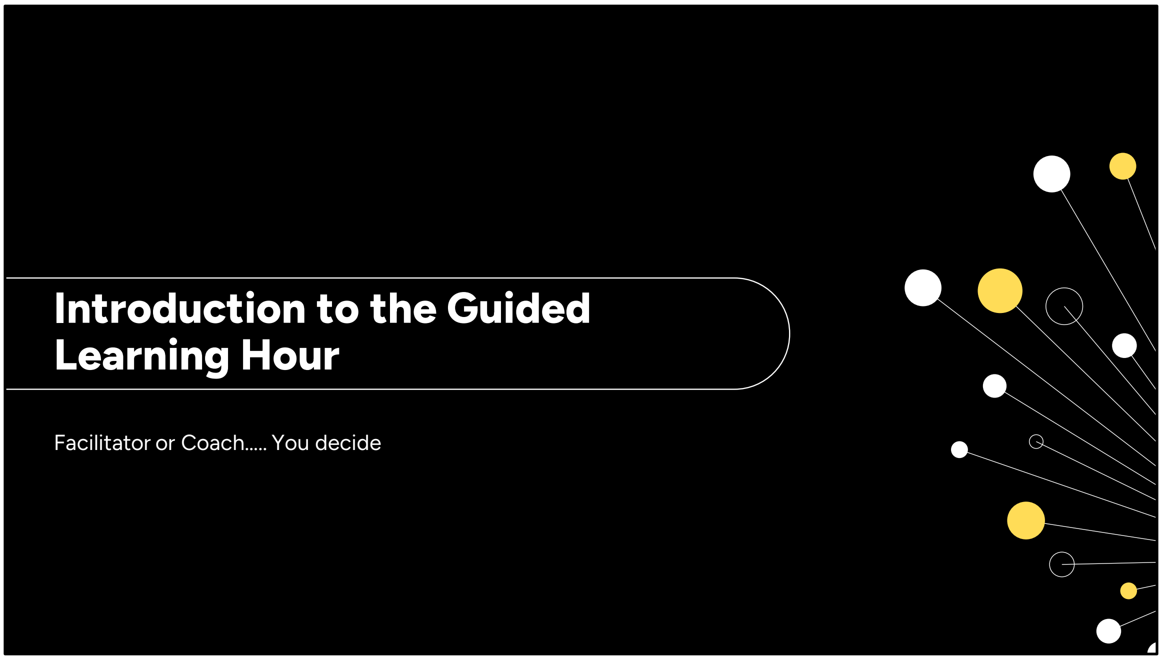

QA

Unique Technical Coaching Skills

Additional Need	Why It Matters in Tech Coaching
Fast-changing content	Learners need up-to-date, real-world practices
Tooling/environment setup friction	Can block learning before it even starts
High cognitive load	Learners juggle logic, tools, and syntax simultaneously
Hands-on practice	Learning is through building, not just listening
Varied skill levels	Tech teams are diverse in experience and learning styles
Debugging thinking, not just code	Must guide mental models, not just syntax
Jargon overload	Learners need context and scaffolding
Fear of judgment	Mistakes must be normalized to foster growth

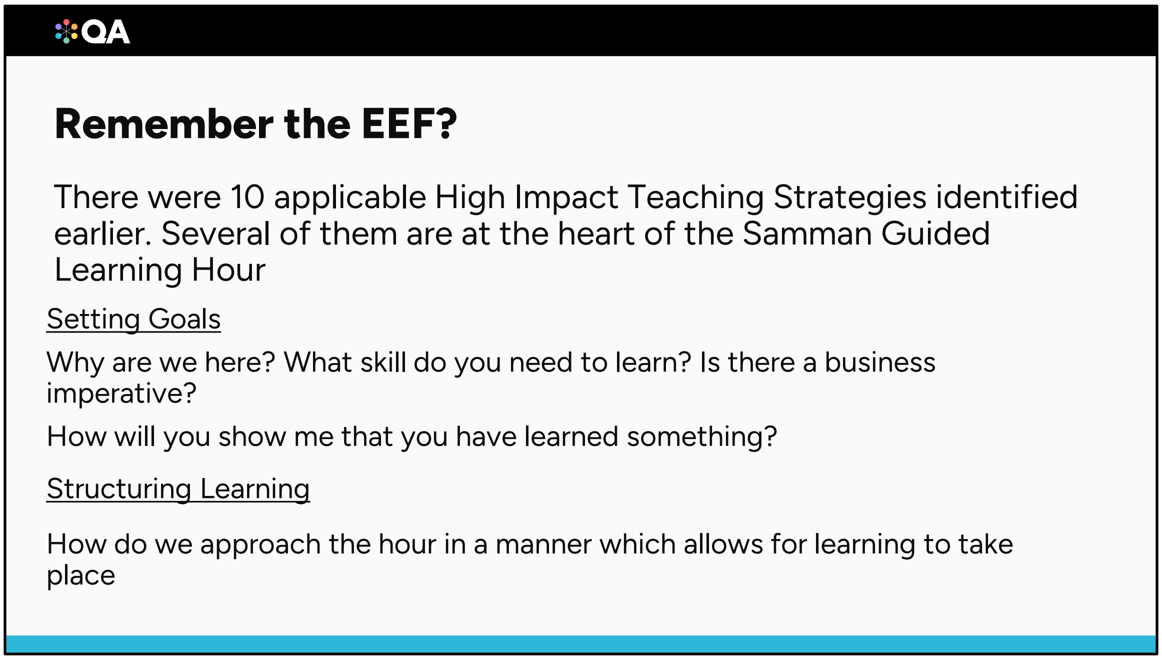
Fast-changing content: Coaches need to stay technically current — tools, frameworks, best practices evolve constantly.
Tooling/environment setup friction: Coaches need to anticipate and troubleshoot technical blockers that are unrelated to the actual learning goal.
High cognitive load: Coaches must break problems into digestible, layered steps and watch for signs of overload. non-technical teaching, concepts typically don't require this level of simultaneous task-switching.
Hands-on practice: A coach must design interactive exercises, live demos, and guided problem-solving. Non-technical subjects may use more reading/discussion-heavy models, where practice is conceptual rather than applied.
Varied skill levels: A good coach tailors support to individuals while maintaining group progress.
Debugging thinking, not just code: A coach often needs to debug not just code, in order to understand the learner’s thinking. This requires asking probing questions like: “What did you expect this line to do?” and “What have you tried already?”
Jargon overload: Technical domains are full of acronyms, jargon, and nested abstractions. Coaches must carefully scaffold explanations, define terms in

context, and avoid assuming prior knowledge.
Fear of judgment: Technical environments can foster perfectionism or fear of “sounding stupid.” Coach must actively create psychological safety and normalize mistakes and questions.



Introduction to the Guided Learning Hour

Facilitator or Coach….. You decide



Remember the EEF?

There were 10 applicable High Impact Teaching Strategies identified earlier. Several of them are at the heart of the Samman Guided Learning Hour

Setting Goals

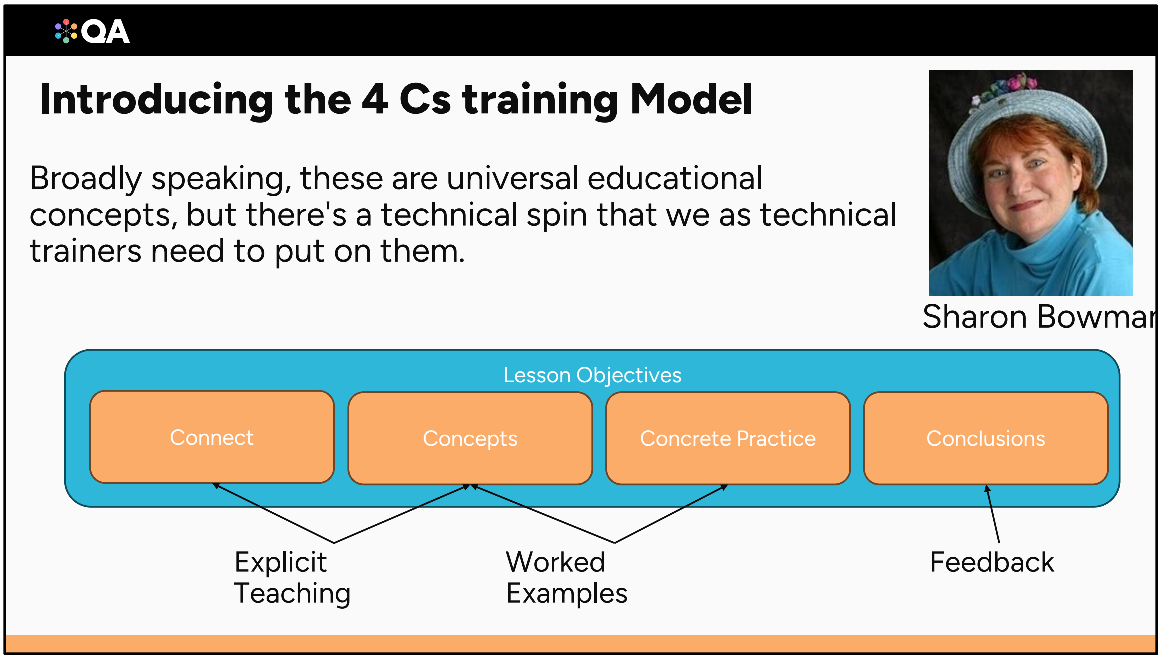
Why are we here? What skill do you need to learn? Is there a business imperative?

How will you show me that you have learned something?

Structuring Learning

How do we approach the hour in a manner which allows for learning to take place

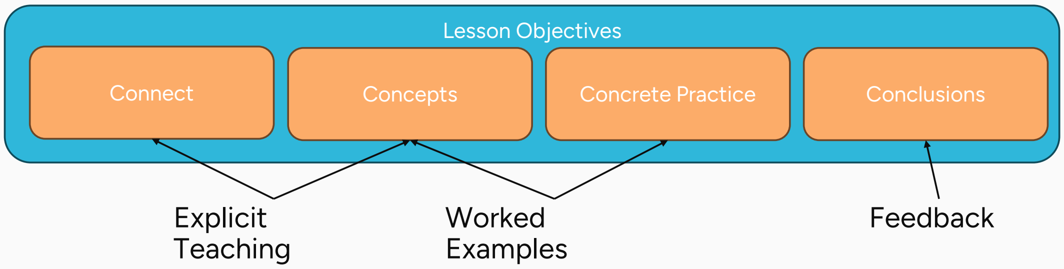
In the second session, we will look into how to go about writing your own learning hours and Katas. This will be very important to consider.



Introducing the 4 Cs training Model

Broadly speaking, these are universal educational concepts, but there's a technical spin that we as technical trainers need to put on them.

Sharon Bowmar

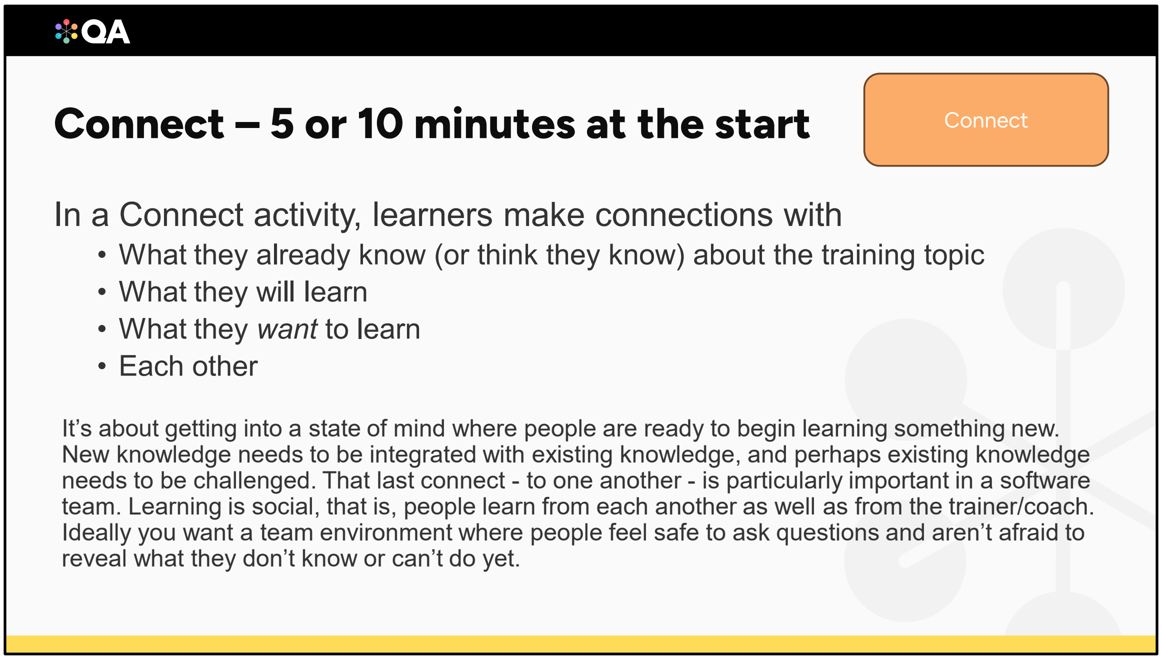


Samman and the 4Cs Training Model:
4C Training model
https://sammancoaching.org/activities/4C_model.html

Sharon L Bowman's 4Cs
The4CsMap2016
https://bowperson.com/images/resources/quick-guide-to-four-c-map.pdf

https://bowperson.com/images/resources/the-four-c-map.pdf

Note: Explicit Teaching, Worked Examples and Feedback are 3 of the high impact techniques specified by the EEF



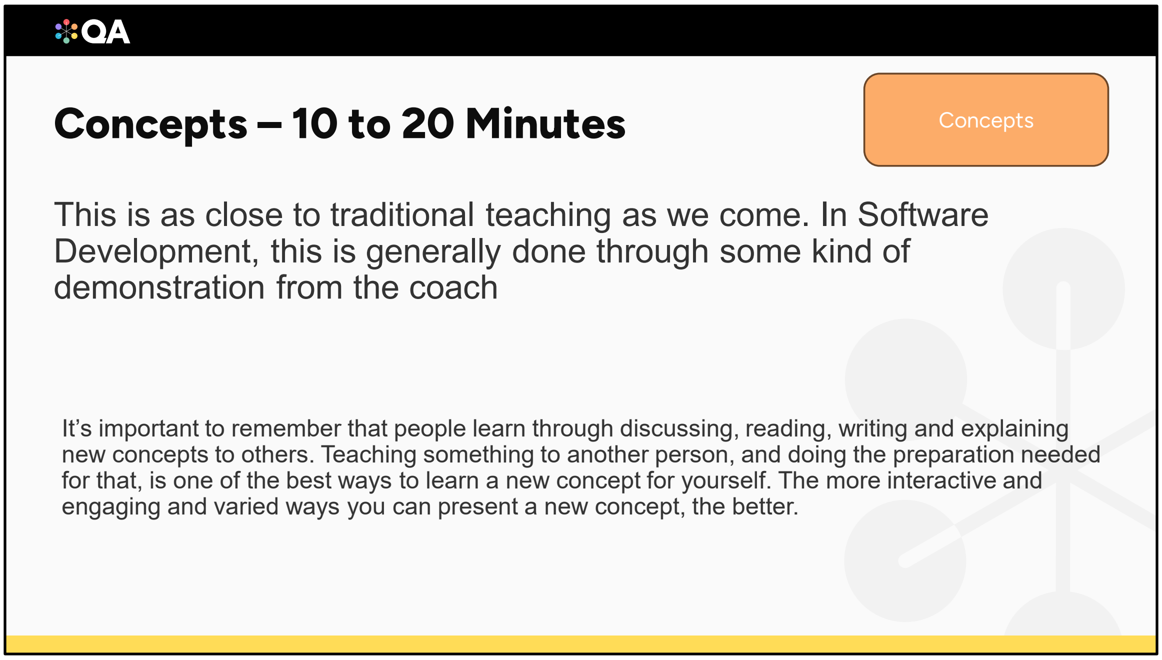
Connect – 5 or 10 minutes at the start

Connect

In a Connect activity, learners make connections with

- What they already know (or think they know) about the training topic
- What they will learn
- What they *want* to learn
- Each other

It's about getting into a state of mind where people are ready to begin learning something new. New knowledge needs to be integrated with existing knowledge, and perhaps existing knowledge needs to be challenged. That last connect - to one another - is particularly important in a software team. Learning is social, that is, people learn from each other as well as from the trainer/coach. Ideally you want a team environment where people feel safe to ask questions and aren't afraid to reveal what they don't know or can't do yet.



QA

Concepts – 10 to 20 Minutes

Concepts

This is as close to traditional teaching as we come. In Software Development, this is generally done through some kind of demonstration from the coach

It's important to remember that people learn through discussing, reading, writing and explaining new concepts to others. Teaching something to another person, and doing the preparation needed for that, is one of the best ways to learn a new concept for yourself. The more interactive and engaging and varied ways you can present a new concept, the better.

QA

Concrete Practice – 20 minutes

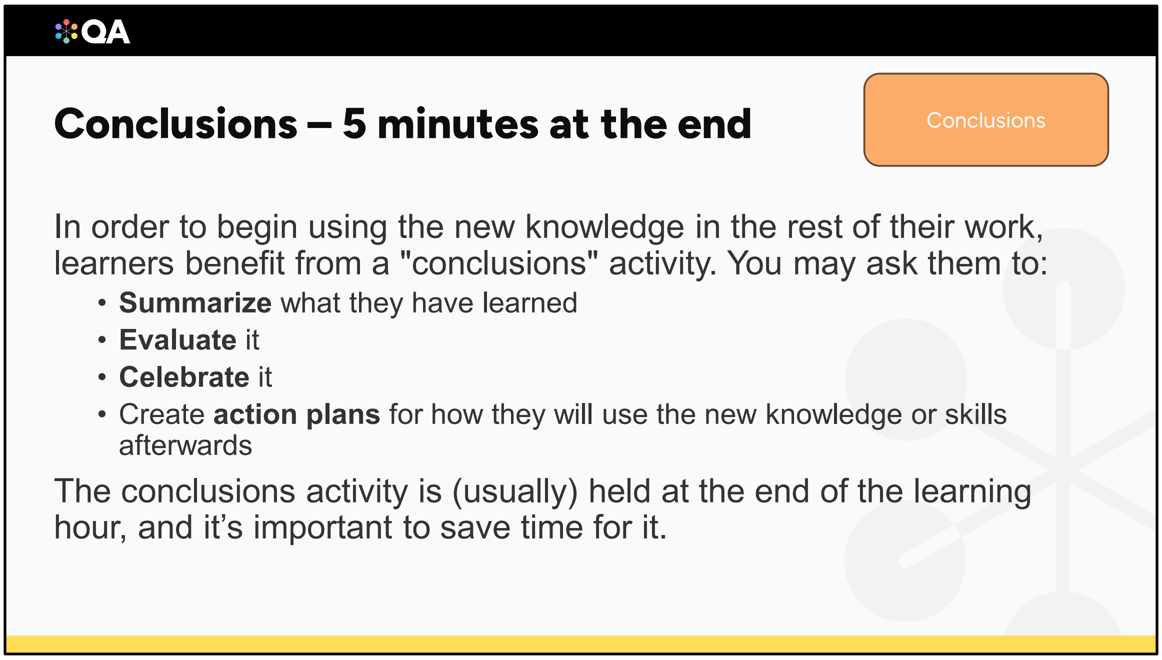
Concrete Practice

Usually in a Learning Hour, this will be a coding exercise, although not exclusively.

Often it will be a code kata with a clear desired outcome, like a refactoring kata or a test design kata.

If you're coding from scratch doing TDD you might:

- Introduce rules and restrictions, starting code and/or checklists.
  - This is to make it less challenging and open than full-on TDD.
- Narrow the focus of the session to a particular aspect you want to learn about.
  - You only have an hour.



Conclusions – 5 minutes at the end

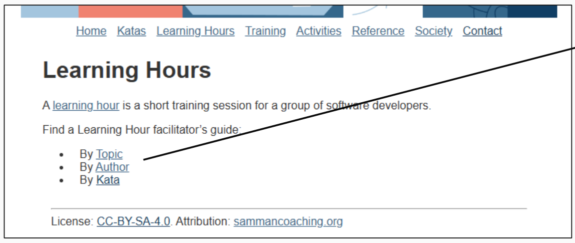
Conclusions

In order to begin using the new knowledge in the rest of their work, learners benefit from a "conclusions" activity. You may ask them to:

- **Summarize** what they have learned
- **Evaluate** it
- **Celebrate** it
- Create **action plans** for how they will use the new knowledge or skills afterwards

The conclusions activity is (usually) held at the end of the learning hour, and it's important to save time for it.

Samman Learning hours


Home Katas Learning Hours Training Activities Reference Society Contact

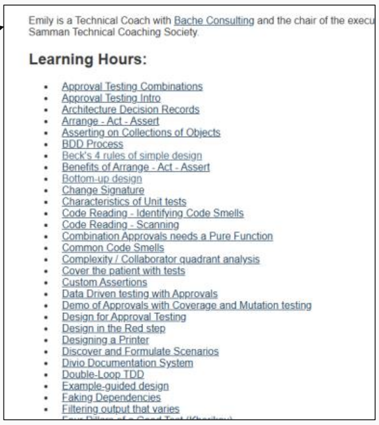
Learning Hours

A learning hour is a short training session for a group of software developers.

Find a Learning Hour facilitator's guide:

- By Topic
- By Author
- By Kata

License: CC-BY-SA-4.0. Attribution: sammancoaching.org


Emily is a Technical Coach with Bache Consulting and the chair of the execu
Samman Technical Coaching Society.

Learning Hours:

- Approval Testing Combinations
- Approval Testing Intro
- Architecture Decision Records
- Arrange - Act - Assert
- Asserting on Collections of Objects
- BDD Process
- Beck's 4 rules of simple design
- Benefits of Arrange - Act - Assert
- Bottom-up design
- Change Signature
- Characteristics of Unit tests
- Code Reading - Identifying Code Smells
- Code Reading - Scanning
- Combination Approvals needs a Pure Function
- Common Code Smells
- Complexity / Collaborator quadrant analysis
- Cover the patient with tests
- Custom Assertions
- Data Driven testing with Approvals
- Demo of Approvals with Coverage and Mutation testing
- Design for Approval Testing
- Design in the Red step
- Designing a Printer
- Discover and Formulate Scenarios
- Divio Documentation System
- Double-Loop TDD
- Example-guided design
- Faking Dependencies
- Filtering output that varies

Take your time to look through and see how many fit the immediate needs of the team. Are there any you have a specific interest in and want to trial?

Sammancoaching.org/learning_hours

Facilitator or Coach

On Emily’s YouTube channel, you will see a number of guided learning hours that she has delivered as if it were live.

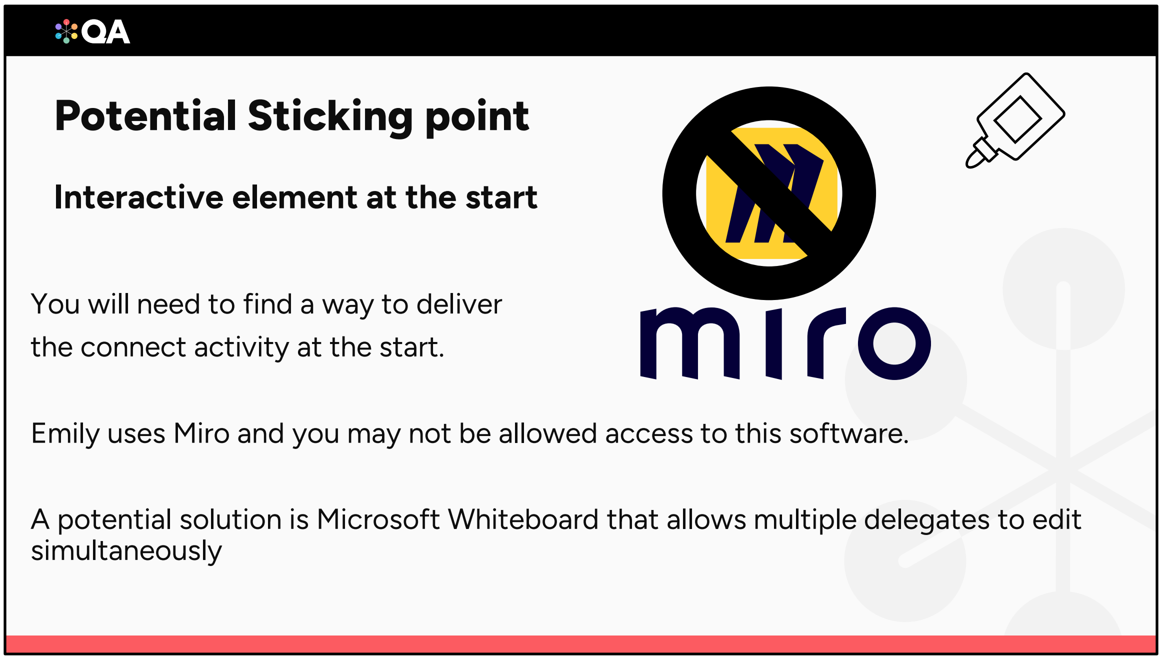
You can act as a facilitator and play these at the front of the GLH if you desire. I highly recommend you have a go at watching how she might deliver it and then coach the hour yourself

Emily Bache

@EmilyBache-tech-coach · 5.84K subscribers · 67 videos

I'm Emily Bache, Software Developer and Technical Coach. What everyone wants is ...more

patreon.com/emilybache and 4 more links



QA

Potential Sticking point

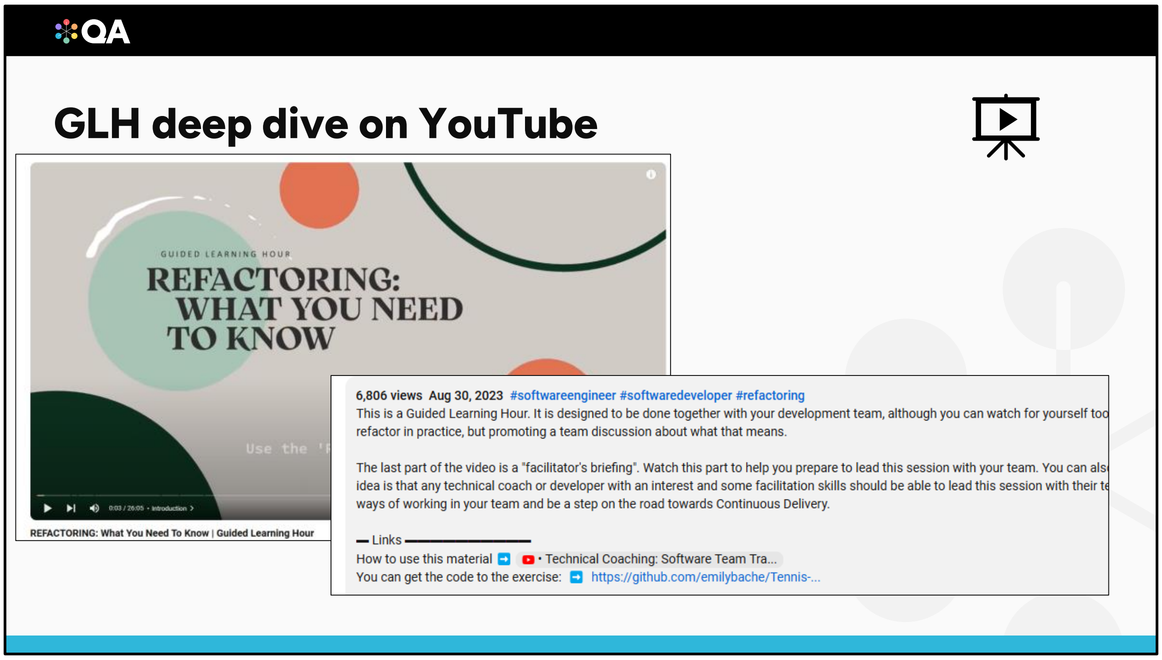
Interactive element at the start

You will need to find a way to deliver the connect activity at the start.

Emily uses Miro and you may not be allowed access to this software.

A potential solution is Microsoft Whiteboard that allows multiple delegates to edit simultaneously

Possible Teams alternative:
[Digital Online Whiteboard App | Microsoft Whiteboard](https://www.microsoft.com/en-us/microsoft-365/microsoft-whiteboard/digital-whiteboard-app)
https://www.microsoft.com/en-us/microsoft-365/microsoft-whiteboard/digital-whiteboard-app



GLH deep dive on YouTube

GUIDED LEARNING HOUR

REFACTORING: WHAT YOU NEED TO KNOW

REFACTORING: What You Need To Know | Guided Learning Hour

6,806 views Aug 30, 2023 #softwareengineer #softwaredeveloper #refactoring

This is a Guided Learning Hour. It is designed to be done together with your development team, although you can watch for yourself too
refactor in practice, but promoting a team discussion about what that means.

The last part of the video is a "facilitator's briefing". Watch this part to help you prepare to lead this session with your team. You can als
idea is that any technical coach or developer with an interest and some facilitation skills should be able to lead this session with their te
ways of working in your team and be a step on the road towards Continuous Delivery.

— Links —

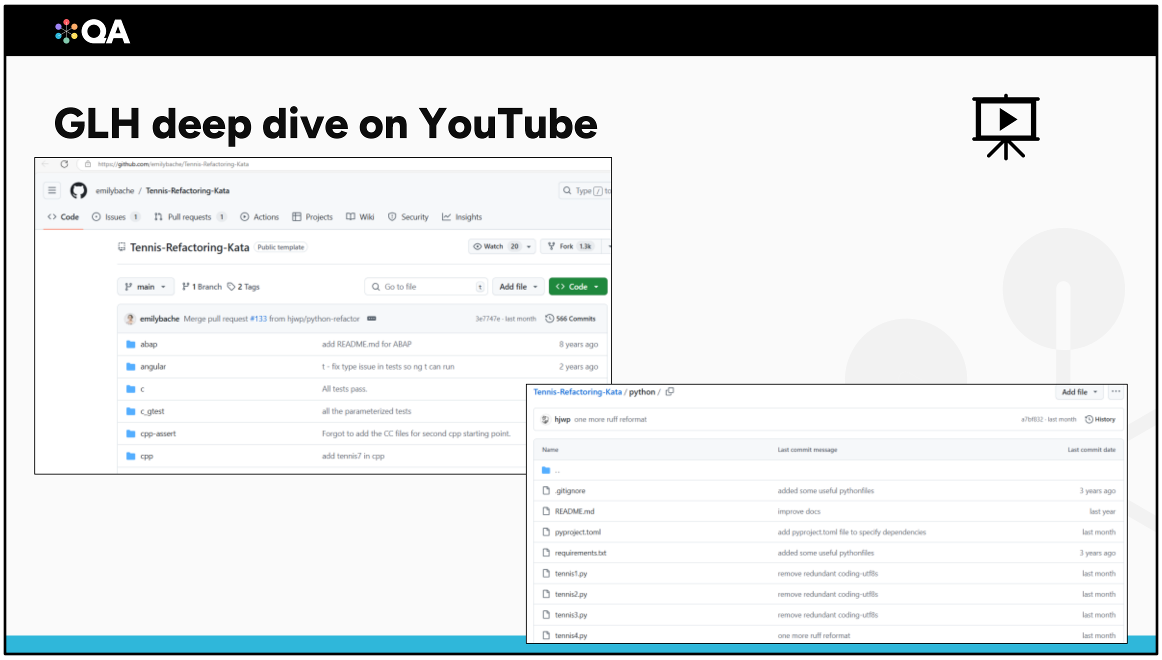
How to use this material ➡ • Technical Coaching: Software Team Tra...

You can get the code to the exercise: ➡ https://github.com/emilybache/Tennis-...

Misconceptions about Refactoring
https://sammancoaching.org/learning_hours/refactoring/misconceptions.html

REFACTORING: What You Need To Know | Guided Learning Hour:
https://www.youtube.com/watch?v=K7xSsNpeM8I

Watch out for the facilitator briefing at the end of the session.



GLH deep dive on YouTube

https://github.com/emilybache/Tennis-Refactoring-Kata

emilybache / Tennis-Refactoring-Kata

Code · Issues 1 · Pull requests 1 · Actions · Projects · Wiki · Security · Insights

Tennis-Refactoring-Kata Public template — Watch 20 · Fork 1.3k

main · 1 Branch · 2 Tags · Go to file · Add file · Code

emilybache Merge pull request #133 from hjwp/python-refactor — 3e7747e · last month — 566 Commits

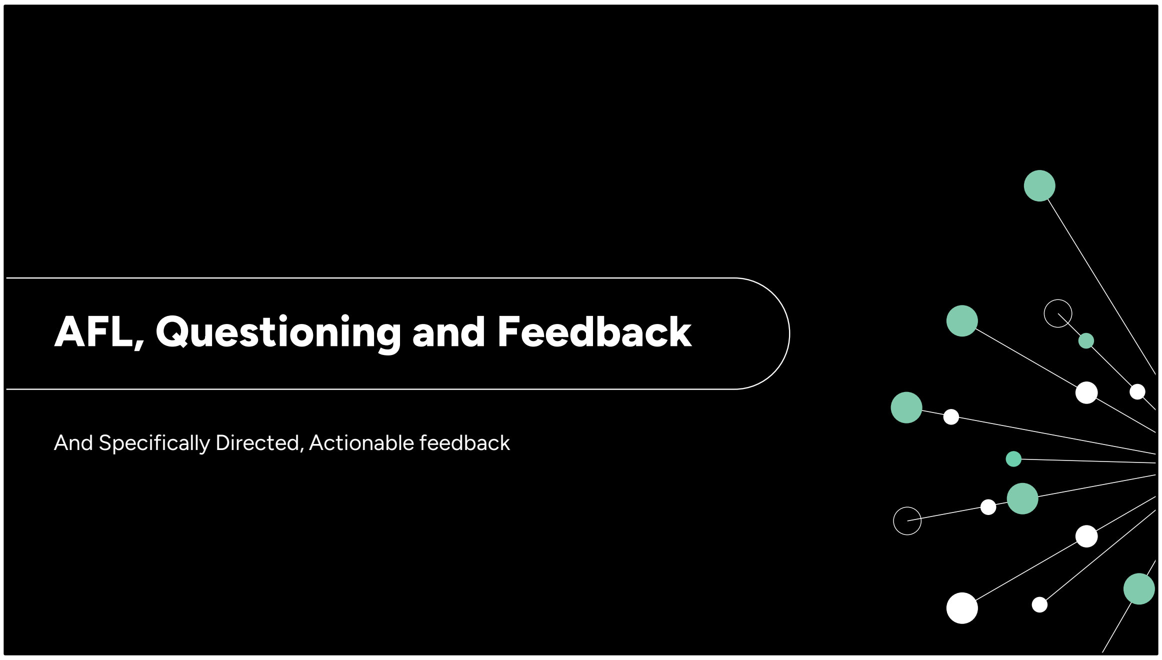
abap	add README.md for ABAP	8 years ago
angular	t - fix type issue in tests so ng t can run	2 years ago
c	All tests pass.	
c_gtest	all the parameterized tests	
cpp-assert	Forgot to add the CC files for second cpp starting point.	
cpp	add tennis7 in cpp	

Tennis-Refactoring-Kata / python /

hjwp one more ruff reformat — a7bf832 · last month — History

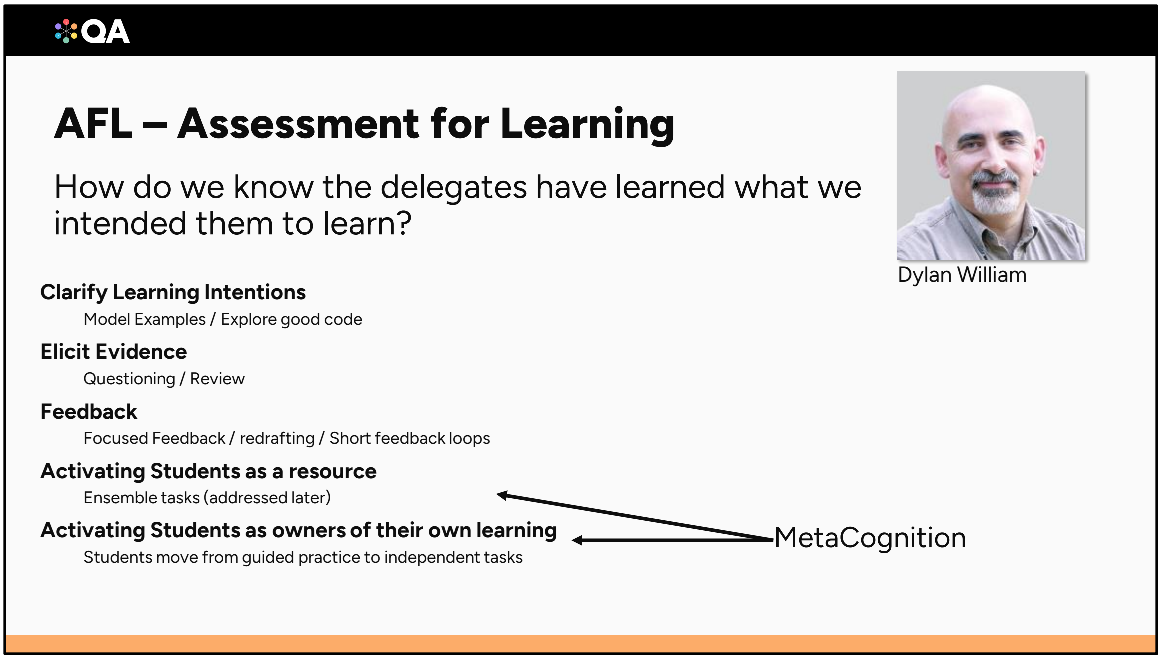
Name	Last commit message	Last commit date
..		
.gitignore	added some useful pythonfiles	3 years ago
README.md	improve docs	last year
pyproject.toml	add pyproject.toml file to specify dependencies	last month
requirements.txt	added some useful pythonfiles	3 years ago
tennis1.py	remove redundant coding-utf8s	last month
tennis2.py	remove redundant coding-utf8s	last month
tennis3.py	remove redundant coding-utf8s	last month
tennis4.py	one more ruff reformat	last month

emilybache/Tennis-Refactoring-Kata: This is a Refactoring Kata based on the rules of Tennis
https://github.com/emilybache/Tennis-Refactoring-Kata



AFL, Questioning and Feedback

And Specifically Directed, Actionable feedback



QA

AFL – Assessment for Learning

How do we know the delegates have learned what we intended them to learn?

Dylan William

Clarify Learning Intentions
  Model Examples / Explore good code

Elicit Evidence
  Questioning / Review

Feedback
  Focused Feedback / redrafting / Short feedback loops

Activating Students as a resource
  Ensemble tasks (addressed later)

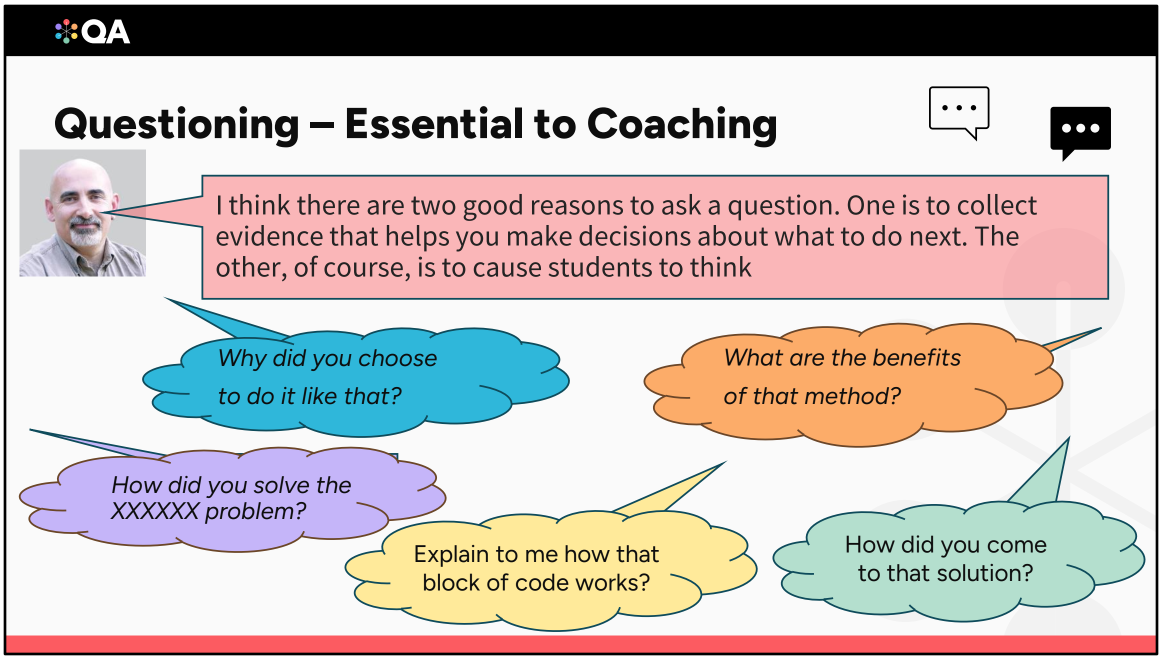
Activating Students as owners of their own learning
  Students move from guided practice to independent tasks

MetaCognition

Welcome to Dylan Wiliam's website
https://www.dylanwiliam.org/Dylan_Wiliams_website/Welcome.html

Dylan Wiliam – Wikipedia
https://en.wikipedia.org/wiki/Dylan_Wiliam

Revisiting Dylan Wiliam's Five Brilliant Formative Assessment Strategies. – teacherhead
https://teacherhead.com/2019/01/10/revisiting-dylan-wiliams-five-brilliant-formative-assessment-strategies/



QA

Questioning – Essential to Coaching

I think there are two good reasons to ask a question. One is to collect evidence that helps you make decisions about what to do next. The other, of course, is to cause students to think

Why did you choose to do it like that?

What are the benefits of that method?

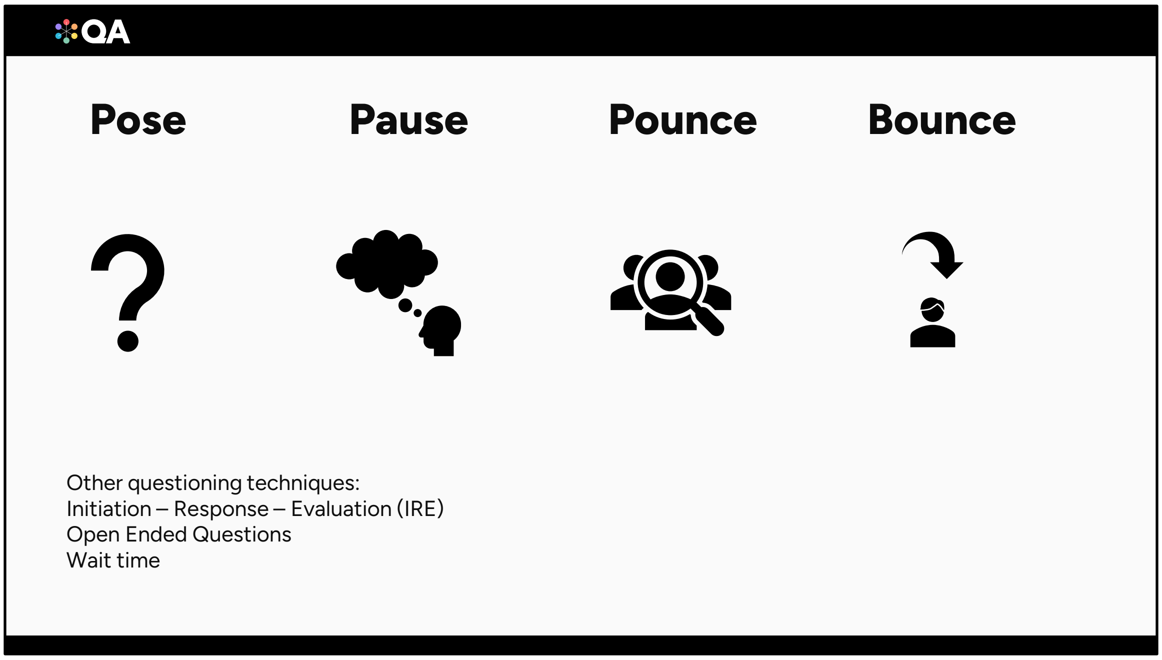
How did you solve the XXXXXX problem?

Explain to me how that block of code works?

How did you come to that solution?

Quote taken from [Podcast Special: Dylan Wiliam on effective questioning in the...](https://www.teachermagazine.com/au_en/articles/teacher-podcast-dylan-wiliam-on-effective-questioning-in-the-classroom) https://www.teachermagazine.com/au_en/articles/teacher-podcast-dylan-wiliam-on-effective-questioning-in-the-classroom#:~:text=there%20and%20then.-,%E2%80%A6,to%20cause%20students%20to%20think.

[The Art of Inquiry: 10 Questioning Techniques to Boost Student Engagement | Education World](https://www.educationworld.com/teachers/art-inquiry-10-questioning-techniques-boost-student-engagement)
https://www.educationworld.com/teachers/art-inquiry-10-questioning-techniques-boost-student-engagement



QA

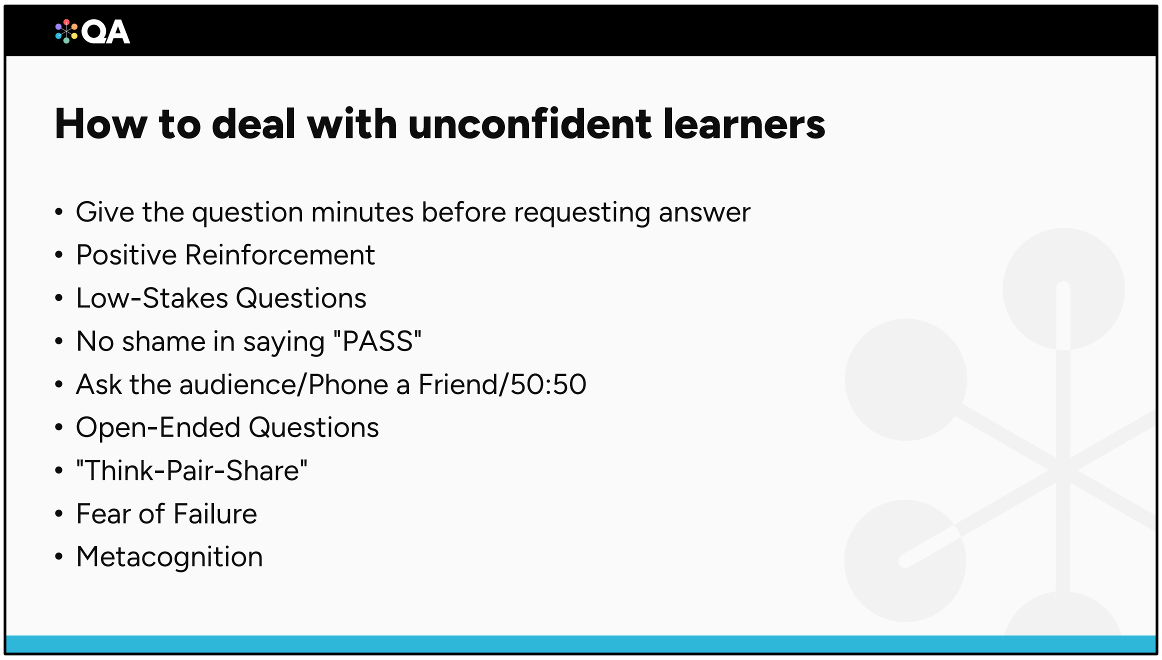
Pose Pause Pounce Bounce

Other questioning techniques:
Initiation – Response – Evaluation (IRE)
Open Ended Questions
Wait time

7.-Pose-Pause-Pounce-Bounce.pdf
https://my.chartered.college/wp-content/uploads/2018/10/7.-Pose-Pause-Pounce-Bounce.pdf
Tutor **poses** a question; **pauses** to allow suitable thinking time; **pounces** on one student for an initial answer; and finally **bounces** the answer to another student who builds on the response.

IRE
Initiation: The teacher poses a question to the class or a specific student.
Response: A student provides an answer to the question.
Evaluation/Feedback: The teacher assesses the response, offering feedback, clarification, or moving on to the next question.
Criticisms of IRE:
Limited student thinking, Passive learning, Potential for superficial understanding:
Open Ended Questions: Questions that require more than a one-word answer.
Wait time – Give people time to digest the question. Research suggests learners often need 3 to 7 seconds of "wait time" to think after a teacher asks a question — especially for higher-order or reflective questions. Many teachers tend to wait

less than 1 second before jumping in, which can reduce student engagement.

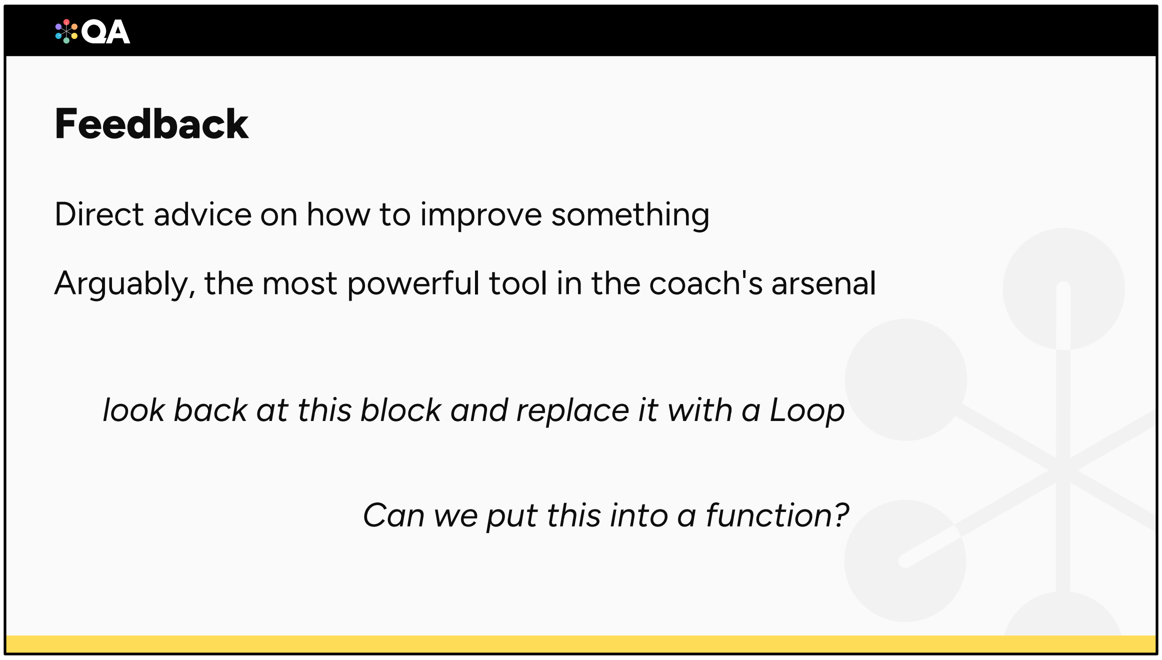


QA

How to deal with unconfident learners

- Give the question minutes before requesting answer
- Positive Reinforcement
- Low-Stakes Questions
- No shame in saying "PASS"
- Ask the audience/Phone a Friend/50:50
- Open-Ended Questions
- "Think-Pair-Share"
- Fear of Failure
- Metacognition

Wait Time: Allow sufficient time for students to process information and formulate responses. Say things like "*In a few minutes, I'm going to ask you …*"
Positive Reinforcement: Focus on effort and progress, not just correctness. Celebrate participation and risk-taking, even if the answers aren't perfect.
Low-Stakes Questions: Start with simpler questions that build confidence before moving to more challenging ones.
Open-Ended Questions: Use questions that require more than a "yes" or "no" answer and encourage explanation and elaboration (e.g., "Why?", "How?", "Can you explain that further?").
"Think-Pair-Share": Provide opportunities for students to discuss questions with a partner before sharing with the larger group.
Think-Pair-Share – Active Learning at King's
https://blogs.kcl.ac.uk/activelearning/2019/05/01/think-pair-share/
Fear of Failure: Remind students that learning involves mistakes and that it's okay to not know everything. Point out the life of a developer is one of constant failure (code rarely works first time).
Metacognition: Encourage students to reflect on their own learning processes and strategies.



QA

Feedback

Direct advice on how to improve something

Arguably, the most powerful tool in the coach's arsenal

look back at this block and replace it with a Loop

Can we put this into a function?

Good approaches:
Radical Candour: Be kind and clear, not sugar-coated. Show that you care personally while challenging directly.
Situation, Behaviour, Impact (SBI): when/where it happened, what the person did, how it affected others or results
Feedforward: Instead of just dwelling on what went wrong, focus on specific suggestions for how to improve next time.

7 Remote Feedback Protocols – Mike Kaechele
https://www.michaelkaechele.com/7-remote-feedback-protocols/

QA

The power of quality, descriptive feedback

https://www.youtube.com/watch?v=E_6PskE3zfQ

Code Review Feedback Scenario

A junior developer submits a pull request for a REST API endpoint.

Round 1 Feedback:
"Great logic! One note: try to avoid duplicating the error handling block — can you refactor it into a helper method?"

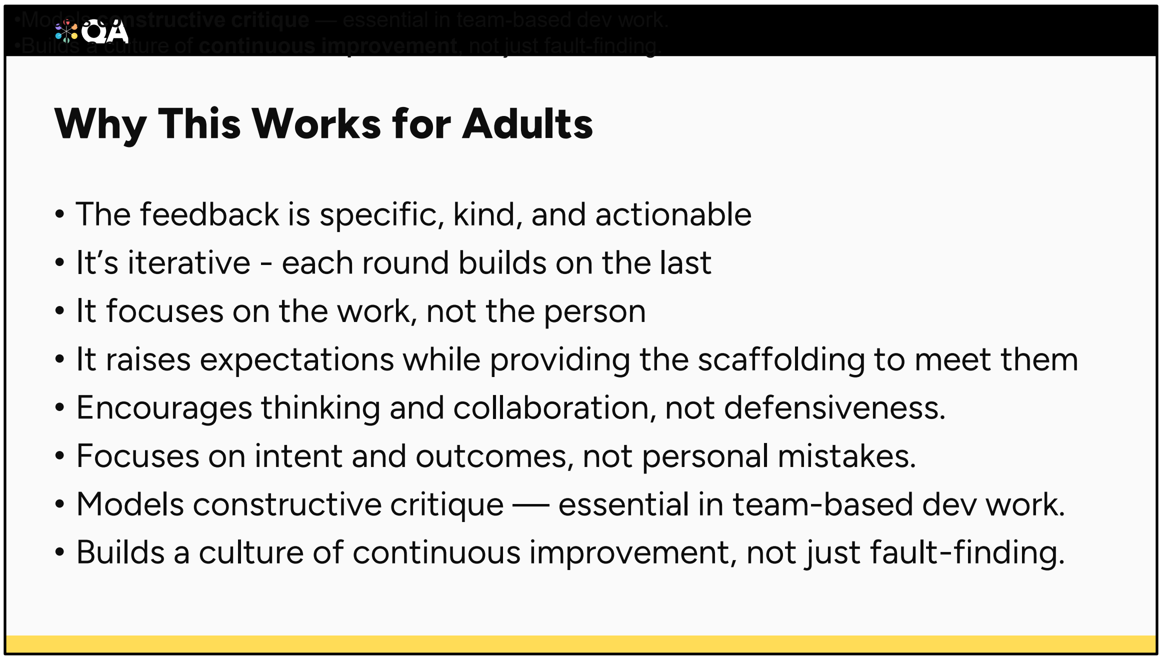
Round 2 Feedback:
"Nice refactor. Now think about using named constants for the status codes — they're more readable and self-explanatory."

Round 3 Feedback:
"Looks good! One polish step: consider adding a unit test for the edge case where the input is null."

Outcome: The final version is cleaner, more maintainable, and better tested — and the learner gains new habits in the process.

Feedback Framing

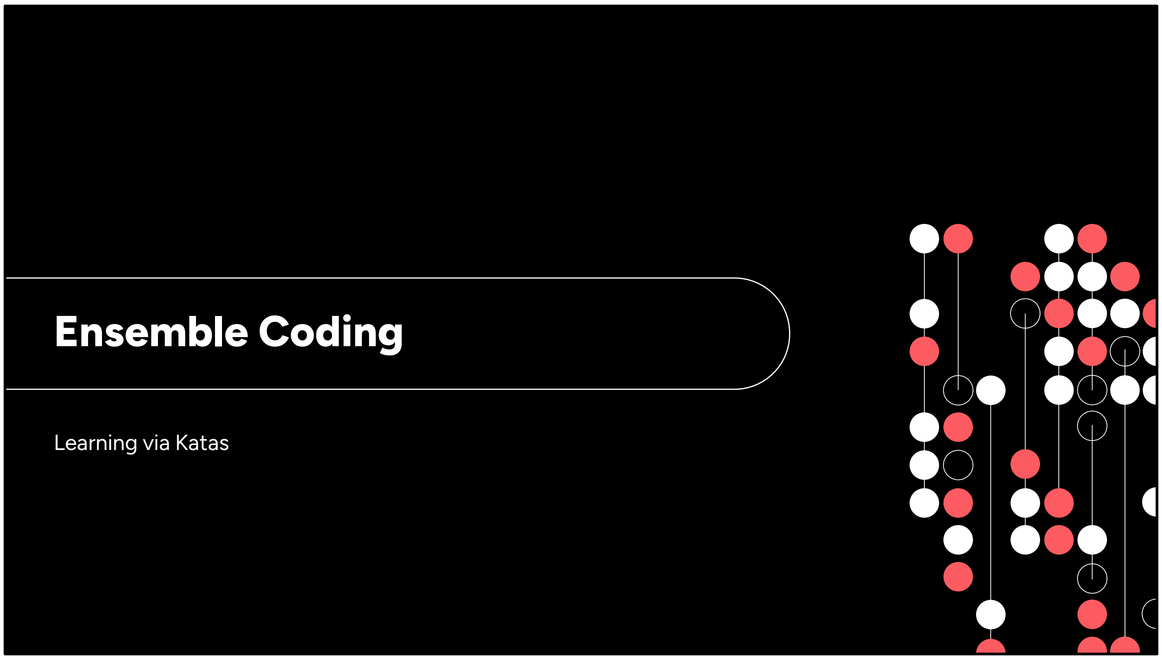
Instead of this...	Try this...
"This code is wrong."	"I think this might throw a null exception — could we add a guard clause?"
"This function is too long."	"What do you think about breaking this into smaller methods for readability?"
"Naming is bad here."	"Would a more descriptive name like "ProcessInvoiceAsync make intent clearer?"
"You should use a different library."	"Have you considered using HttpClientFactory here? It could simplify config and lifecycle."
"This won't scale."	"Interesting approach — how would this behave with 10x the data?"
"You're missing tests."	"Can we add a test for the edge case where the input is empty?"
"This isn't how we do it."	"I've seen us handle similar logic with a strategy pattern — want to try that?"
"Just use async."	"If we make this method async, it should help with thread efficiency under load."



•Models **constructive critique** — essential in team-based dev work.
•Builds a culture of **continuous improvement**, not just fault-finding.

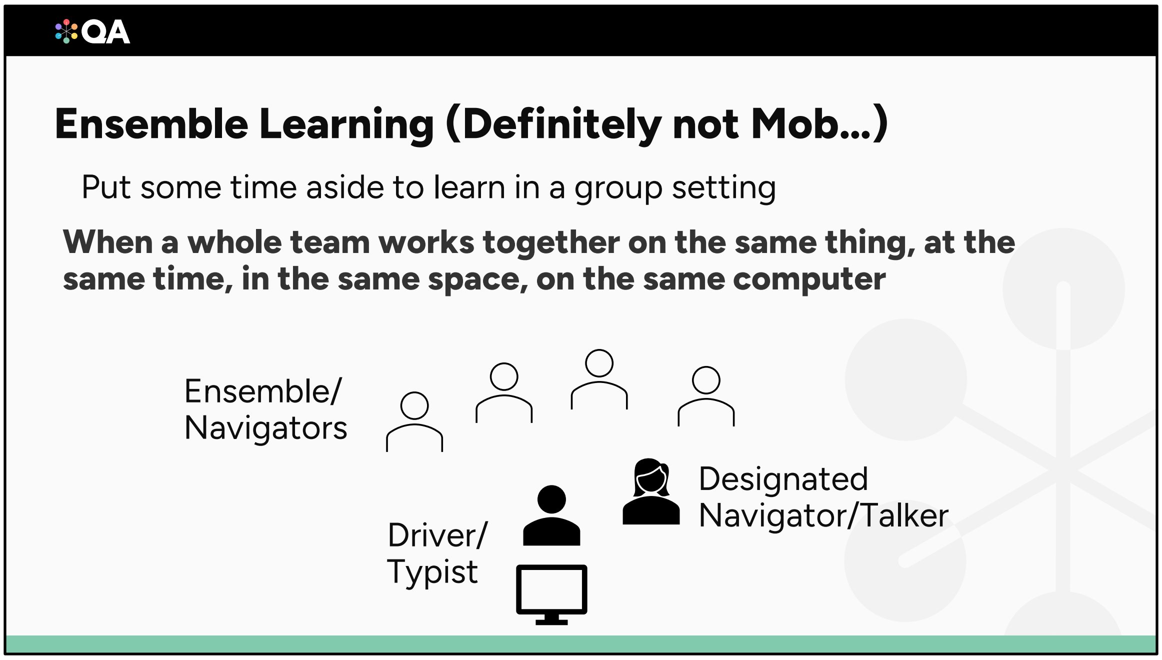
Why This Works for Adults

- The feedback is specific, kind, and actionable
- It’s iterative - each round builds on the last
- It focuses on the work, not the person
- It raises expectations while providing the scaffolding to meet them
- Encourages thinking and collaboration, not defensiveness.
- Focuses on intent and outcomes, not personal mistakes.
- Models constructive critique — essential in team-based dev work.
- Builds a culture of continuous improvement, not just fault-finding.



Ensemble Coding

Learning via Katas



QA

Ensemble Learning (Definitely not Mob…)

Put some time aside to learn in a group setting

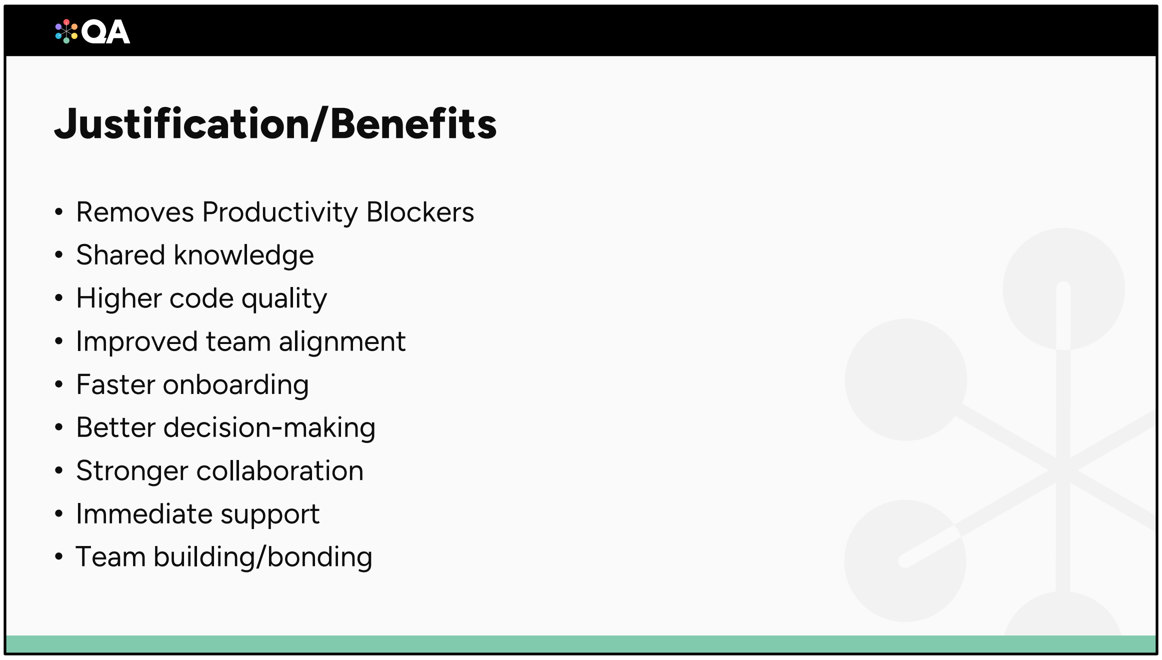
When a whole team works together on the same thing, at the same time, in the same space, on the same computer

Ensemble/ Navigators
Designated Navigator/Talker
Driver/ Typist

A Tester's Journey: Ensemble Is the New Mob
https://www.lisihocke.com/2021/02/ensemble-is-the-new-mob.html#:~:text=Beginning%20of%20February%2C%20one%20colleague,Emily%20Bache%20and%20Lisa%20Crispin.
"The terms "ensemble" and "mob" refer to the same approach. People had been looking for a replacement term for many years to get rid of the negative connotations of the "mob" which was perceived as problematic. Many people had been appalled by the term "mob" and hence didn't want to give it a try. Thinking of bullying or lynch mobs, the term is triggering trauma. Since last year [2019] we now finally have a new term that's a lot more inclusive: the ensemble. It's already been taken up and lived by many leading experts like Emily Bache and Lisa Crispin. "

Excellent blog that describes a first ever ensemble session that highlights positives and negatives:
A Tester's Journey: Our Team's First Mobbing Session
https://www.lisihocke.com/2017/04/our-teams-first-mobbing-session.html

Also this is a great page of resources:
A Tester's Journey: Collaboration
https://www.lisihocke.com/p/collaboration.html



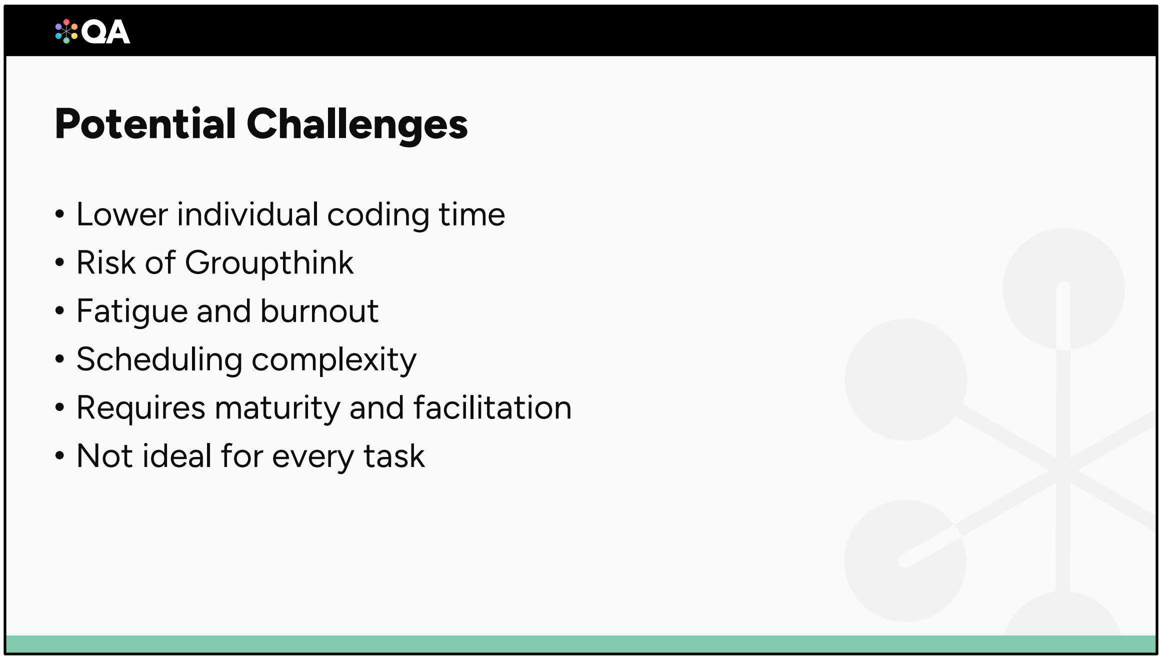
QA

Justification/Benefits

- Removes Productivity Blockers
- Shared knowledge
- Higher code quality
- Improved team alignment
- Faster onboarding
- Better decision-making
- Stronger collaboration
- Immediate support
- Team building/bonding

- Removes Productivity Blockers: ensembles can overcome: Multitasking, Interruptions, Being blocked, Technical debt, Code cruft, Waiting for information or reviews, Unnecessary meetings, Incomplete understanding, or not understanding the problem at all, Context switching, Merge conflicts, Lack of quality, Missing knowledge
- Shared knowledge: Everyone in the team gains a deeper understanding of the codebase and domain, reducing knowledge silos.
- Higher code quality: Continuous peer review and real-time feedback help catch bugs and design issues early.
- Improved team alignment: Promotes a shared vision and consistent coding practices across the team.
- Faster onboarding: New team members ramp up quicker by participating in the work directly with guidance from experienced colleagues.
- Better decision-making: Diverse perspectives lead to more thoughtful design and problem-solving.
- Stronger collaboration: Encourages communication, trust, and empathy within the team.
- Immediate support: If someone gets stuck, others are right there to help, reducing downtime.

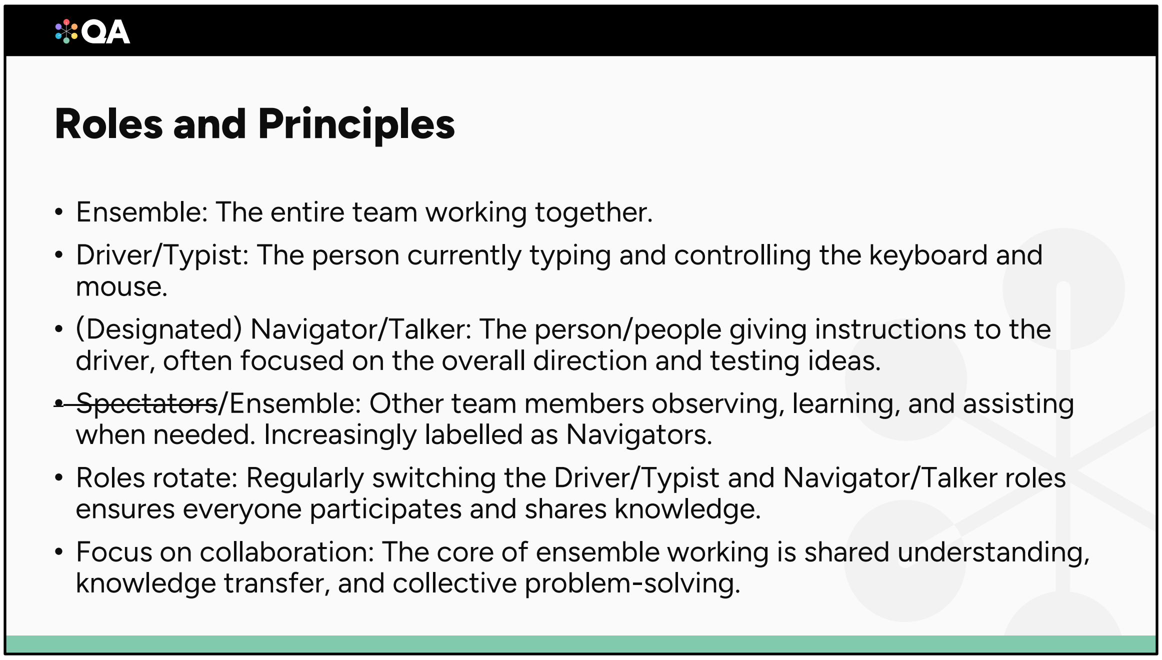
Same benefits as code reviews but done at the time the code is being written!


QA

Potential Challenges

- Lower individual coding time
- Risk of Groupthink
- Fatigue and burnout
- Scheduling complexity
- Requires maturity and facilitation
- Not ideal for every task

- Lower individual coding time: Developers may feel they're not getting enough hands-on practice, especially those who learn best by doing.
- Risk of groupthink: Creative or divergent solutions might get overlooked if everyone leans toward consensus too quickly.
- Fatigue and burnout: Long ensemble sessions can be mentally exhausting; rotating roles and taking breaks is essential.
- Scheduling complexity: Getting everyone in the same room (or virtual room) can be hard, especially with distributed teams.
- Requires maturity and facilitation: Without strong communication and facilitation, sessions can derail or be dominated by stronger personalities.
- Not ideal for every task: Simple, repetitive, or exploratory tasks may not need the full team's attention. However, when working on something "simple" ensembles can innovate and find ways to automate workflow elements.



Roles and Principles

- Ensemble: The entire team working together.
- Driver/Typist: The person currently typing and controlling the keyboard and mouse.
- (Designated) Navigator/Talker: The person/people giving instructions to the driver, often focused on the overall direction and testing ideas.
- ~~Spectators~~/Ensemble: Other team members observing, learning, and assisting when needed. Increasingly labelled as Navigators.
- Roles rotate: Regularly switching the Driver/Typist and Navigator/Talker roles ensures everyone participates and shares knowledge.
- Focus on collaboration: The core of ensemble working is shared understanding, knowledge transfer, and collective problem-solving.

EnsembleProgrammingGuidebook.pdf (includes a chapter on how to be a facilitator)
https://ensembleprogramming.xyz/Download/EnsembleProgrammingGuidebook.pdf

The Mob Programming Role Playing Game gives you the opportunity to investigate additional roles (Researcher, Sponser, Rear Admiral,...) you may want to introduce...
https://github.com/willemlarsen/mobprogrammingrpg

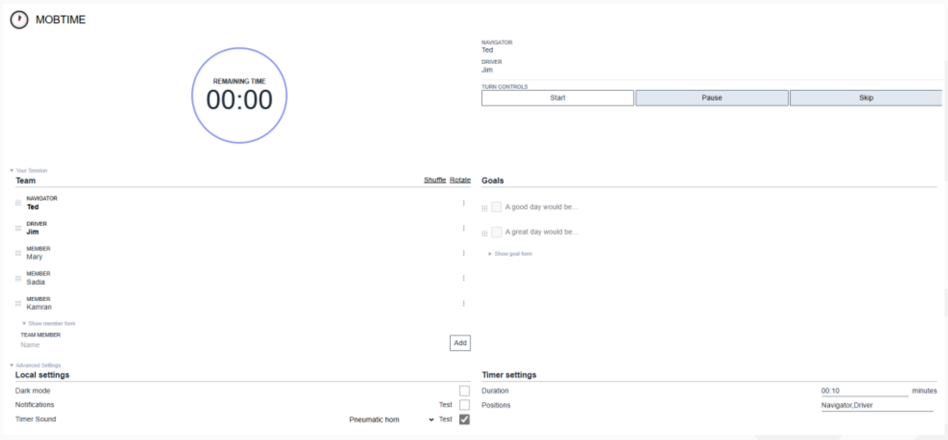
QA

Rules

- Rotate on Timer.
  - 4-15 mins (depending on experience).
- Work done on a previous rotation is enhanced **not** replaced.
- No decisions on the keyboard. The typist <u>**must**</u> be passive.
- Be kind and considerate . Every contribution has value.
- Everyone is either contributing or learning or both.
- Don't expect the first few sessions to be "productive".
- When switching the designated navigator becomes the driver/typist and the driver/typist returns to the ensemble (or vice versa).
- Navigate on highest level of intent the driver can work with.
- Have a retrospective/plenary to summarize and reinforce what's been learned.
- **Have Fun!**

QA

Ensemble Timer



NAVIGATOR
Ted

DRIVER
Jim

Team

NAVIGATOR
Ted

DRIVER
Jim

MEMBER
Mary

MEMBER
Sadia

MEMBER
Kamran

TEAM MEMBER
Name

Goals

- [ ] A good day would be...
- [ ] A great day would be...

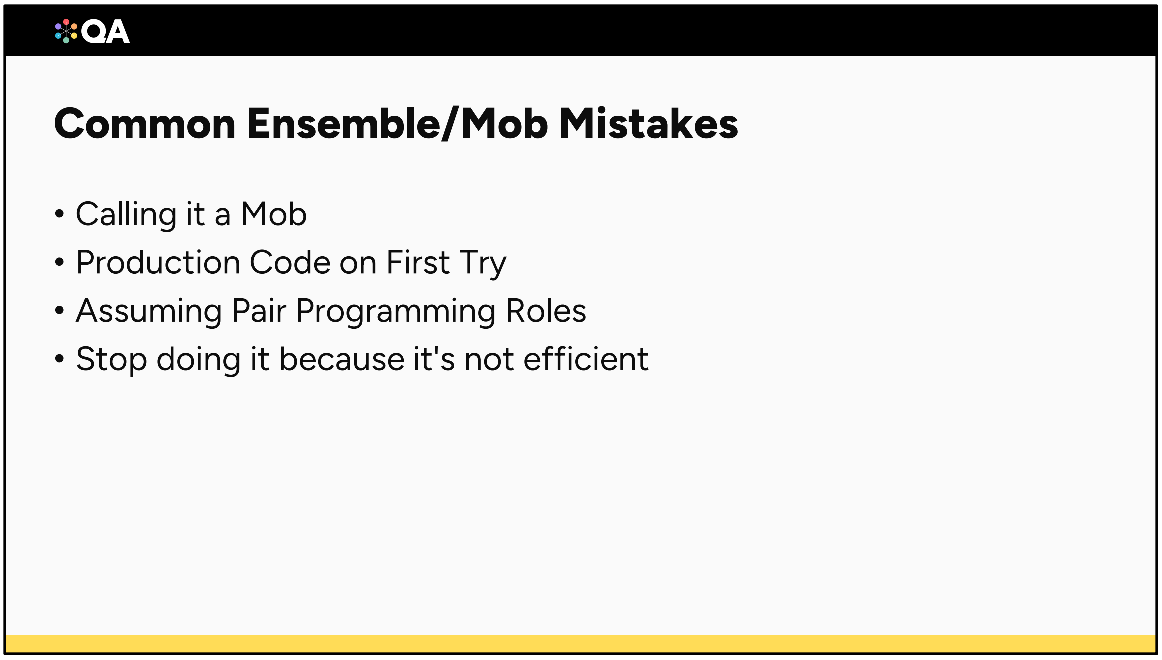
Local settings

Dark mode		☐
Notifications		☐
Timer Sound	Pneumatic horn	✓

Timer settings

Duration	00:10 minutes
Positions	Navigator,Driver

There are a number of timers out there that are designed to be used in ensemble/mob programming sessions.
MobTime
https://mobti.me/


QA

Common Ensemble/Mob Mistakes

- Calling it a Mob
- Production Code on First Try
- Assuming Pair Programming Roles
- Stop doing it because it's not efficient

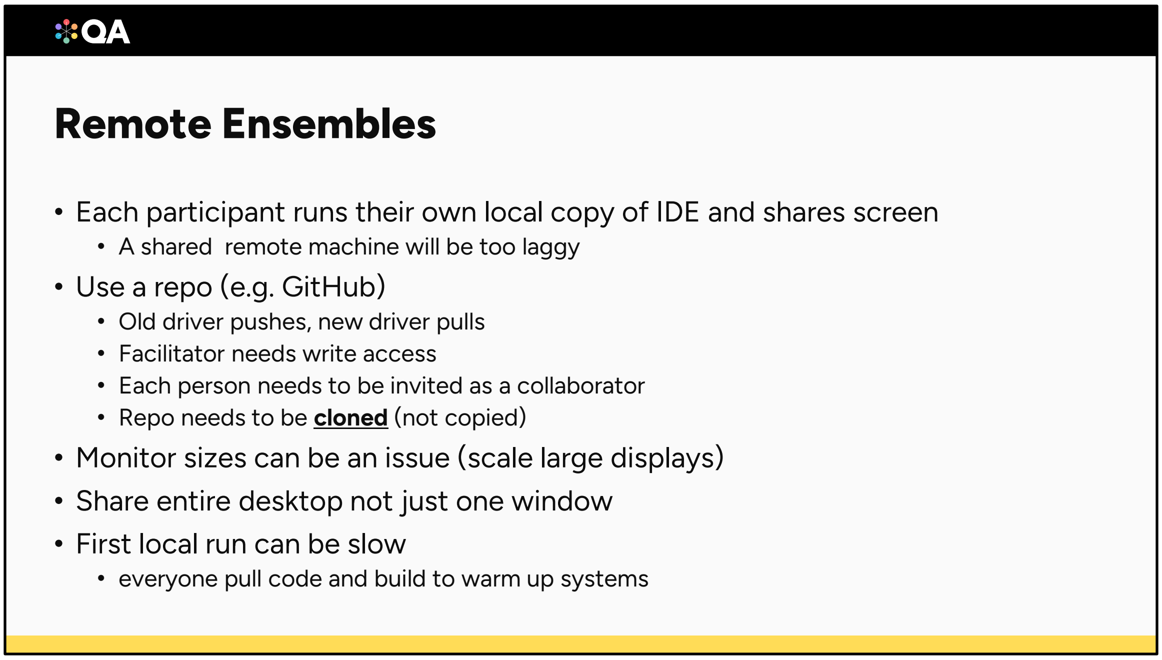
KEYNOTE : Ensemble Working : The ultimate in collaboration - Clare Sudbery
https://events.responsive.se/tidigare-arrangemang/devlin2023/presentationerna-devlin2023/310-keynote-ensemble-working-the-ultimate-in-collaboration-clare-sudbery

Mob Programming & the Power of Flow • Woody Zuill • GOTO 2019
https://www.youtube.com/watch?v=28S4CVkYhWA

Emily Bache: AVOID These Mob Programming / Ensemble Mistakes:
https://www.youtube.com/watch?v=qjE1O9Zdm3U

Ensemble is a skill that needs to be learnt and developed. It takes brain power and effort to learn to do it well.

Mob Mentality Show – YouTube
https://www.youtube.com/@mobmentalityshow/videos


QA

Remote Ensembles

- Each participant runs their own local copy of IDE and shares screen
  - A shared remote machine will be too laggy
- Use a repo (e.g. GitHub)
  - Old driver pushes, new driver pulls
  - Facilitator needs write access
  - Each person needs to be invited as a collaborator
  - Repo needs to be **cloned** (not copied)
- Monitor sizes can be an issue (scale large displays)
- Share entire desktop not just one window
- First local run can be slow
  - everyone pull code and build to warm up systems

Remote Ensemble Programming:
Ensemble programming. Collaborative coding as a remote team | by Stephan Bester | Medium
https://stephan-bester.medium.com/ensemble-programming-c9a4212cd1f2

Remote Ensemble Programming at Meltwater - Meltwater Engineering Blog
https://underthehood.meltwater.com/blog/2023/08/09/remote-ensemble-programming/

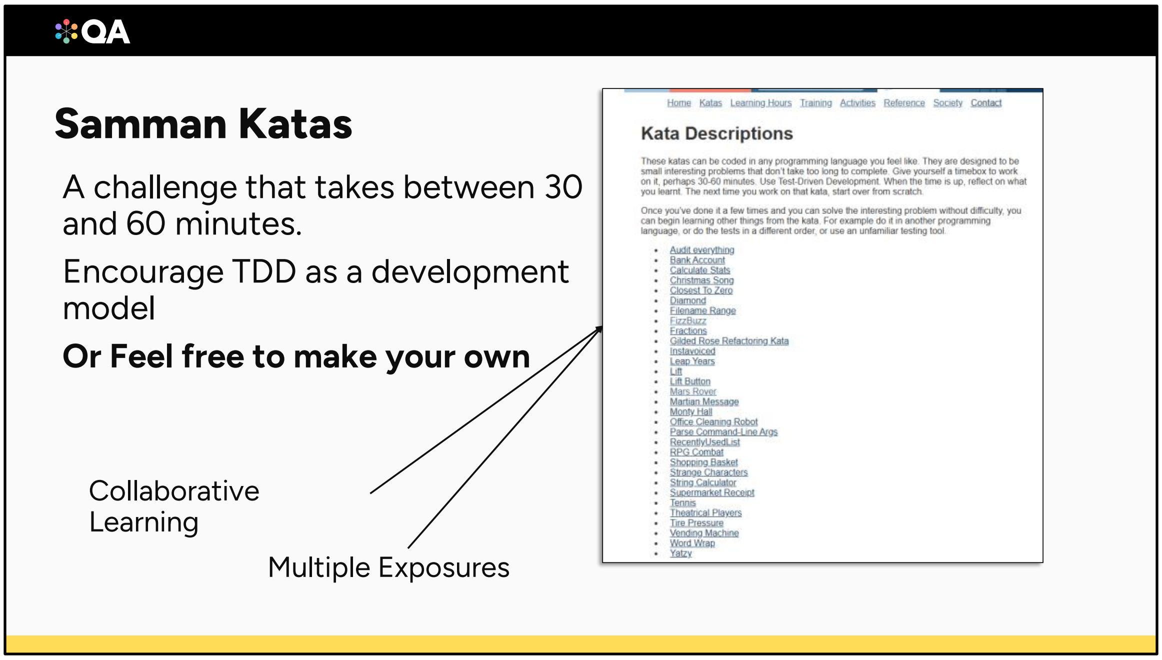
How Remote Ensemble (Mob) Programming Is Working for Me
https://qualitycoding.org/remote-mob-programming/

Remote Mob Programming | How we do Remote Mob Programming. (Has a free PDF book)
https://www.remotemobprogramming.org/

Remote Ensemble **Testing** - How an Experiment Shaped the Way We Work:
https://www.infoq.com/articles/remote-ensemble-working-experiment/

For the first ensemble consider having a practice round where emphasis is on getting things running smoothly.

Fast git handover with mob tool
Fast git handover with mob | Tool for smooth git handover.
https://mob.sh/


QA

Samman Katas

A challenge that takes between 30 and 60 minutes.

Encourage TDD as a development model

Or Feel free to make your own

Collaborative Learning

Multiple Exposures

Home Katas Learning Hours Training Activities Reference Society Contact

Kata Descriptions

These katas can be coded in any programming language you feel like. They are designed to be small interesting problems that don't take too long to complete. Give yourself a timebox to work on it, perhaps 30-60 minutes. Use Test-Driven Development. When the time is up, reflect on what you learnt. The next time you work on that kata, start over from scratch.

Once you've done it a few times and you can solve the interesting problem without difficulty, you can begin learning other things from the kata. For example do it in another programming language, or do the tests in a different order, or use an unfamiliar testing tool.

- Audit everything
- Bank Account
- Calculate Stats
- Christmas Song
- Closest To Zero
- Diamond
- Filename Range
- FizzBuzz
- Fractions
- Gilded Rose Refactoring Kata
- Instavoiced
- Leap Years
- Lift
- Lift Button
- Mars Rover
- Martian Message
- Monty Hall
- Office Cleaning Robot
- Parse Command-Line Args
- RecentlyUsedList
- RPG Combat
- Shopping Basket
- Strange Characters
- String Calculator
- Supermarket Receipt
- Tennis
- Theatrical Players
- Tire Pressure
- Vending Machine
- Word Wrap
- Yatzy

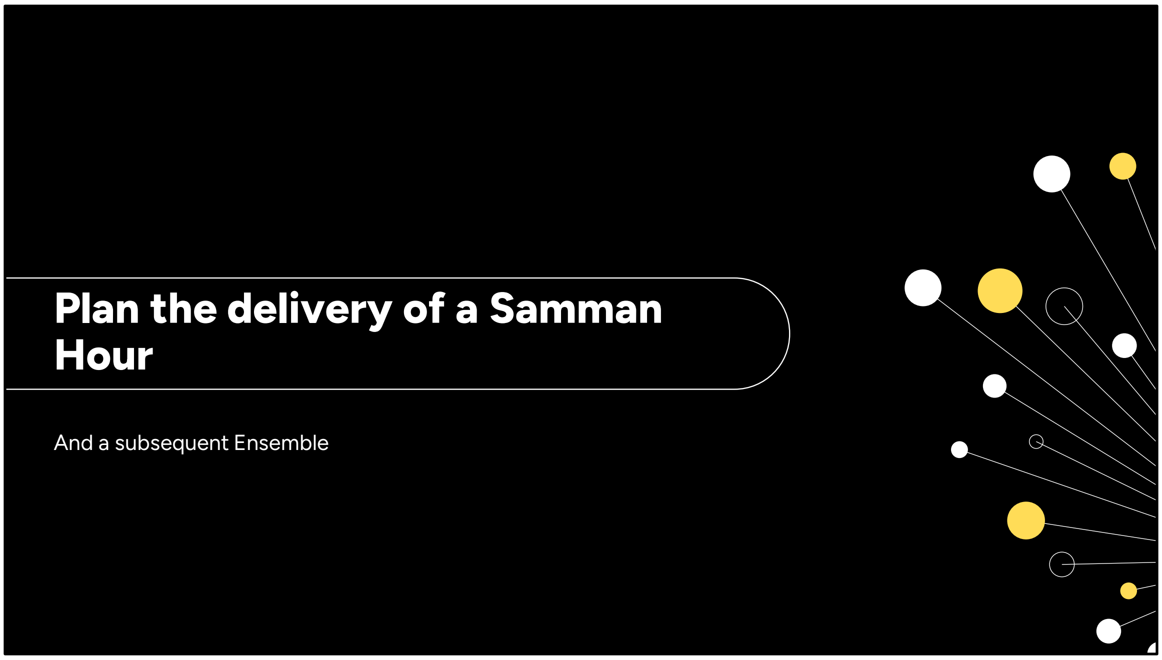
Alternate Saman hour, ensemble one week for each. Therefore 2-week cycles.

Mob Programming Role Playing Game:
willemlarsen/mobprogrammingrpg: A game for exploring the development practice of mob programming
https://github.com/willemlarsen/mobprogrammingrpg

Demo of the Mob Programming and the RPG (Part 1)
https://www.youtube.com/watch?v=ixV8YG5vwyM

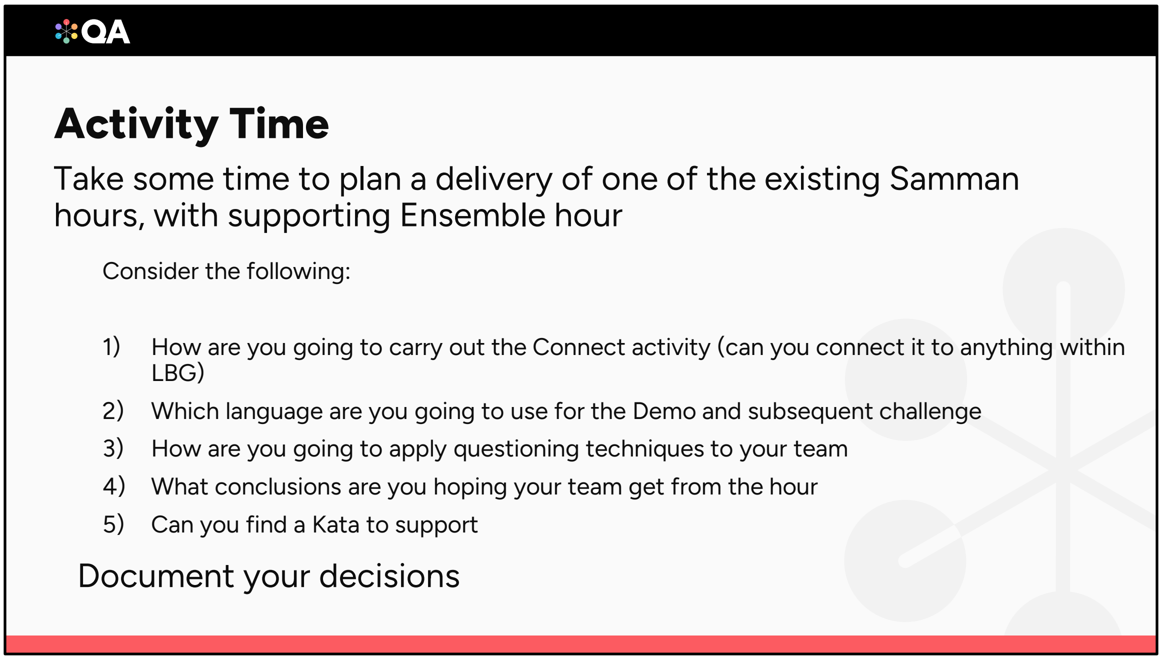
Demo of the Mob Programming and the RPG (Part 2)
https://www.youtube.com/watch?v=V1ZgaX99UJ4

Web App that Automates the Mob Programming RPG
https://gregorriegler.com/mob-programming-rpg/bxmpiwosw0u



Plan the delivery of a Samman Hour

And a subsequent Ensemble



QA

Activity Time

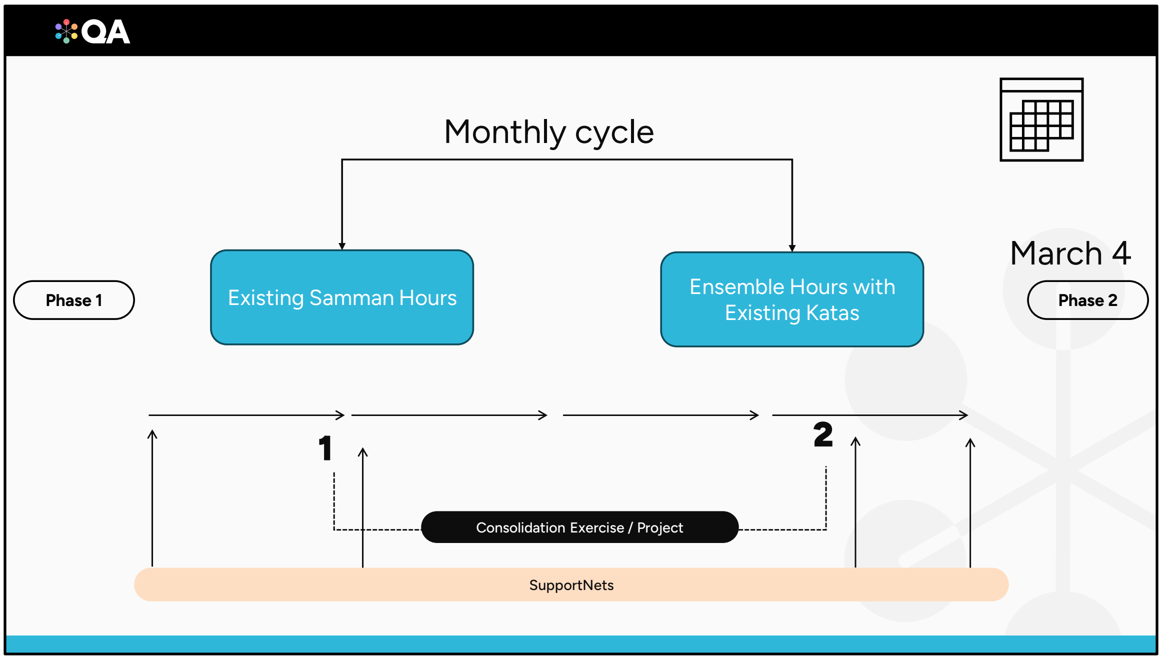
Take some time to plan a delivery of one of the existing Samman hours, with supporting Ensemble hour

Consider the following:

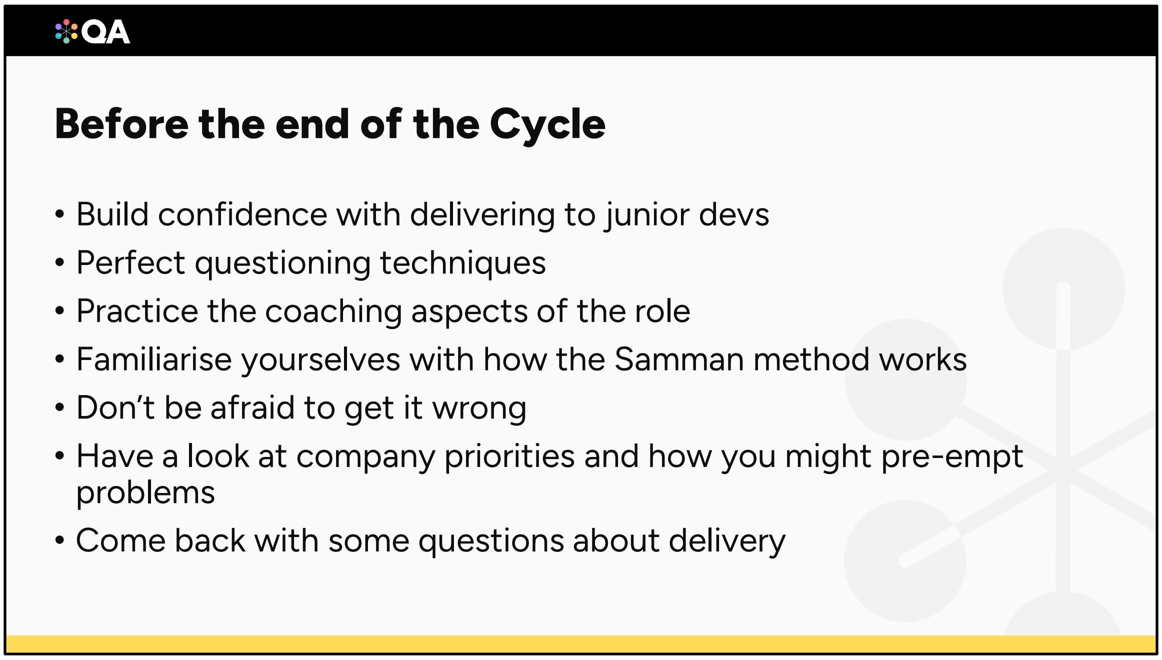
1) How are you going to carry out the Connect activity (can you connect it to anything within LBG)
2) Which language are you going to use for the Demo and subsequent challenge
3) How are you going to apply questioning techniques to your team
4) What conclusions are you hoping your team get from the hour
5) Can you find a Kata to support

Document your decisions

Your Job over the next few weeks



Or try out weekly and get 2 rotations in a single month.



Before the end of the Cycle

- Build confidence with delivering to junior devs
- Perfect questioning techniques
- Practice the coaching aspects of the role
- Familiarise yourselves with how the Samman method works
- Don’t be afraid to get it wrong
- Have a look at company priorities and how you might pre-empt problems
- Come back with some questions about delivery

QA

Concerns or Questions?

Learn. To Change.