



CONTENTS

[Module 1: Agile - what, why and how?](#)

[Module 2: Small Batches - Iterations](#)

[Module 3: Small Batches - Stories](#)

[Module 4: Limiting Work in Progress](#)

[Module 5: Data-Driven Retrospectives](#)

[Format & Delivery](#)

[Price](#)

[Additional Services](#)

For more information contact:

amanda@betterfasterhappier.com

betterfasterhappier.com

MODULE 1: AGILE - WHAT, WHY AND HOW?

Learning outcomes and topics	Activities
What is agile and why should we adopt it? <ul style="list-style-type: none">● Agile is about working in short feedback loops● Short feedback loops increase value and improve quality● Short customer feedback loops increase value● Short development feedback loops improve quality	Exercise <u>“What is agile”</u> Post-it capture & sort
How to create short feedback loops <ul style="list-style-type: none">● Small Batches + Limiting WIP = Short Feedback Loop● Feedback loops require validation criteria● Cycle time is the metric for measuring feedback loops	Exercise <u>Validating Manhattan</u> Defining the problem before you define the solution
Mindset <ul style="list-style-type: none">● Agile is full of “dumb ideas” because it’s highly counter-intuitive● Adopting agile requires an open mind and willingness to change	Exercise <u>Understanding Mindset</u> Card sort identifying agile vs non-agile mindsets

MODULE 2: SMALL BATCHES - ITERATIONS

Learning outcomes and topics	Activities
Working in small batches: why and how? <ul style="list-style-type: none">● Why smaller is faster● Reducing batch size is a 3 step process<ul style="list-style-type: none">○ Features -> Iterations○ Iterations -> User Stories○ User Stories -> Very small sprint stories	Exercise <u>The Coin Game</u> Coin flipping to understand the impact of batch size
Defining iterations with Story Mapping <ul style="list-style-type: none">● Small iterations shorten the customer feedback loop to increase value● Story Mapping is a technique for breaking large scale work into iterations	Exercise <u>Story Mapping Introduction</u> Applying story mapping to your morning routine Exercise <u>Story Mapping Practice</u> Create a story map for a hotel booking site
Scoping and Tracking Iterations <ul style="list-style-type: none">● The qualities of a good iteration:<ul style="list-style-type: none">○ Has an outcome○ Has clearly defined scope○ Can be completed in 2-3 weeks○ Can be released to a customer OR is a system that works end-to-end● Define iteration scope with true/false acceptance criteria● Track and manage iterations in Jira with epics and epic boards	

MODULE 3: SMALL BATCHES - STORIES

Learning outcomes	Activities
<p>Defining stories test first</p> <ul style="list-style-type: none"> • Small stories shorten the development feedback loop to improve quality • BDD encourages you to work test-first. It uses natural language to define behaviour, not implementation • Use the Gherkin syntax to write test cases 	
<p>Vertical slicing</p> <ul style="list-style-type: none"> • Vertical stories: <ul style="list-style-type: none"> ○ Connect all components ○ Have at least one input and one output ○ Can be tested independently • The qualities of a good story: <ul style="list-style-type: none"> ○ Contribute to the iteration value ○ Independently testable ○ And deployable (behind a feature flag) ○ Small ○ Estimable • Managing planning and delivery in parallel • Tracking and managing refinement with a Refinement Board • Using issue types in Jira • Definition of Ready • Definition of Done 	<p>Exercise</p> <p>Hotels.com - Part 1 Break a simple iteration into 3 vertical stories with test cases</p> <p>Hotels.com - Part 2 Break the 3 stories down further, into at least 8 very small stories with test cases</p> <p>Hotels.com - Part 3 Deliver the stories in 4 x 10 minute sprints</p>
<p>Bugs and incidents are just missing tests</p> <ul style="list-style-type: none"> • Defining tests first reduces the number of bugs created • Use 5 Whys to root cause bugs and improve quality in future 	<p>Exercise</p> <p>Bugs Are Missing Tests Root cause bugs and identifying missing tests</p>
<p>Sprint Estimation</p> <ul style="list-style-type: none"> • Small stories are easy to estimate • Keep stories small enough to complete in 3 days or less • Use 1 point per day when estimating 	
<p>Cut-Out-And-Keep: Define Your Ways of Working</p> <ul style="list-style-type: none"> • Create your Definition of Ready • Create your Definition of Done • Design your refinement process and agree roles & responsibilities 	<p>Cut-Out-And-Keep</p>

MODULE 4: LIMITING WORK IN PROGRESS

Learning outcomes	Activities
<p>Reducing delays by limiting WIP</p> <ul style="list-style-type: none"> • Delays increase cycle time • Limiting work in progress reduces delays and shortens feedback loops • Applying WIP limits 	<p>Exercise</p> <p><u>The Coin Game</u> Flipping coins to understand the impact of bottlenecks</p> <p>Exercise</p> <p><u>The Multi-Tasking Name Game</u> Understanding the impact of high WIP</p>
<p>Using Pairing & Swarming to Reduce WIP</p> <ul style="list-style-type: none"> • Pairing and swarming improves quality and increases speed • Tips for successful, sustainable pairing 	<p>Exercise</p> <p><u>Pairing Patterns</u> Mapping different scenarios with different pairing patterns</p>
<p>Using Stand-Ups and Sprint Planning to Limit WIP</p> <ul style="list-style-type: none"> • Stand-ups are a planning session not a status update • Use stand-ups to focus on finishing • Stand-Up tips 	<p>Exercise</p> <p><u>Let's Make Stand-Ups Great Again</u> Review a stand-up plan and improve it to get more finished.</p>
<p>Sprint Planning</p> <ul style="list-style-type: none"> • Understand capacity by tracking working days and average points per developer per day • Use a Capacity Planning Calculator • How to carry work over between sprints • Use sprint planning to define opportunities to pair and swarm • Sprint Planning tips 	<p>Exercise</p> <p><u>Calculate Sprint Capacity with a Capacity Calculator</u></p> <p>Exercise</p> <p><u>Treat Every Day Like the Last Day of the Sprint</u> Plan the first day of the sprint aiming to maintain the lowest WIP</p>

MODULE 5: DATA-DRIVEN RETROSPECTIVES

Learning outcomes	Activities
<p>Retros: What and why?</p> <ul style="list-style-type: none"> Retrospectives are a feedback loop to improve process Retros exist to prevent us making the same mistake twice 	
<p>Data-driven Process Improvement</p> <ul style="list-style-type: none"> Process health data is a tool to learn and improve Key metrics give a picture of overall health - they are like alerts Key metrics: <ul style="list-style-type: none"> Cycle Time (feedback loop) Volume (throughput) Predictability (sprint predictability) Leading indicators help us debug process problems - they are like logging Leading indicators: <ul style="list-style-type: none"> Focus (WIP) metrics Batch size metrics 	<p>Exercise</p> <p><u>Map the Metrics</u> What can we learn from each metric</p>
<p>Identifying and Root Causing Problems</p> <ul style="list-style-type: none"> How to use data to identify process problems How to root cause process problems Data can highlight both problems within and external to the team Retro tips 	<p>Exercise</p> <p><u>Using Data to Identify Process Problems</u> Review a set of metrics to identify opportunities to improve</p> <p>Cut-Out-And-Keep Review your team metrics, identify and root cause problems, define actions</p>
<p>Agile Guard Rails: Minimum Viable Agility</p> <ul style="list-style-type: none"> The purpose of your process is to work in short feedback loops Minimum Viable Agility guard rails ensure you remain agile whilst still giving you huge flexibility to tailor your process to your team and context MVA guard rails: <ul style="list-style-type: none"> Define goals and measure success Measure platform health 	

<ul style="list-style-type: none"> ○ Limit Work in Progress ○ Work in small batches ○ Visualise work ○ Hold a daily stand-up ○ Measure cycle time ○ Hold regular retrospectives 	
<p>Taking Ownership: Autonomy and Accountability</p> <ul style="list-style-type: none"> ● You have the freedom to define your process, but you are responsible for the health of that process 	<p>Cut-Out-And-Keep</p> <p>Decide who will take responsibility for agile ceremonies</p>

FORMAT & DELIVERY

The training consists of 5 modules totalling 3 days of training delivered face to face.

The training can be delivered in one go, but it is recommended to deliver it in 2-3 parts. This avoids information overload, improves engagement and increases successful adoption.

In addition to hands-on practical exercises, “Cut-Out-And-Keep” activities are included to allow teams to define:

- How their ways of working will evolve in order to put new learnings into practice
- Roles, responsibilities and ownership for process health and effectiveness.

These help ensure that teams don't simply revert to current ways of working when the training is over.

At the end of the training participants will receive:

- PDF copies of the training decks
- A summary of key learnings
- A comprehensive list of follow-up reading
- A set of guides to running agile ceremonies effectively

PRICE

Training

£800 per person for 3 days training*

Maximum 16 people per session

*Discount available for multiple teams

Training can be customised to meet the needs of your organisation.

Optional Follow-Up Coaching & Troubleshooting

£150 per hour

2 hours per team per month for a 3 month period is usually sufficient (£900/team)

ADDITIONAL SERVICES

Process Health Metrics Set-Up

Visualise process health and empower your teams and leadership to identify and debug process problems fast with agile metrics. The price includes:

- A team process metrics dashboard with key process metrics and leading indicators
- A sprint capacity calculator
- A video guide to updating metrics
- A guide to using metrics to debug process problems

A sample is available on request.

Agile Knowledge Base

Provide on-demand support for your teams with a comprehensive agile knowledge base located in Confluence or the equivalent. The Knowledge Base includes:

- Guides to common agile practices including breaking work down, BDD, TDD and pair programming
- A self-learning playlist of blogs and talks covering all aspects of agile working
- A recommended reading list for people to explore topics in more depth
- A guide to using Confluence

A sample is available on request.

Jira Project Template Set-Up

Get new teams set up quickly and ensure consistency where required with a template Jira project. This is provided together with a guide to setting up a new project from the template.

Agile Onboarding Workshop

Get new people up to speed with your ways of working with an agile onboarding workshop. The price includes the workshop design and training your team to run it.

For more information contact:

amanda@betterfasterhappier.com

betterfasterhappier.com