



Agile Change Management Cheat Sheet

	A Awareness	D Desire	K Knowledge	A Ability	R Reinforcement
What is it?	<i>Ensure everyone understands the problem Agile solves (speed to value, predictability, quality, responsiveness) and the cost of staying as-is.</i>	<i>Build motivation and safety so people choose to try the new way, not just “attend the training”.</i>	<i>Provide role-based learning that translates into day-to-day behaviour.</i>	<i>Remove barriers and build capability through coaching in the flow of work.</i>	<i>Turn Agile from an “initiative” into “how we run the place”.</i>
3 Practical Tips	<ol style="list-style-type: none"> Lead with customer pain + delivery pain (missed deadlines, rework, handoffs, slow decisions). Show flow data (lead time, WIP, delays, escalation volume) rather than “Agile theory”. Explain what will change and what won’t (this reduces fear). 	<ol style="list-style-type: none"> Make it personal: “What pain will this remove for you?” Start with volunteers / one or two pilot value streams. Use managers as multipliers (enable them to coach, not police). 	<ol style="list-style-type: none"> Train by role (leaders vs PO/PM vs team vs stakeholders). Teach using their work (real backlogs, real priorities, real constraints). Focus on “how we decide and deliver”, not just Scrum mechanics. 	<ol style="list-style-type: none"> Start with a pilot and harden the operating model before scaling. Protect teams from overload: limit WIP, reduce “project thrash”. Make impediments visible and resolve them fast (leadership owns this). 	<ol style="list-style-type: none"> Celebrate outcomes + learning (not just “velocity”). Run cadence health checks (monthly) and deeper retros (quarterly). Build a community of practice and a lightweight quality gate (not bureaucracy).
Success Enablers	<ul style="list-style-type: none"> A clear “North Star” for delivery (e.g., faster value, fewer handoffs, better reliability). A short leader narrative: “what we’re changing in how we work, and why”. A visible “case for change” using 2–3 real work items as examples. 	<ul style="list-style-type: none"> Co-design: teams help define working agreements, cadences, and measures. Leaders model new behaviours (prioritisation, WIP limits, attending reviews). Clear “what’s in it for me” by persona (Ops, Product, Tech, Leadership). 	<ul style="list-style-type: none"> Role-specific paths: leadership, product, delivery, enabling functions. Practical templates: Definition of Ready/Done, backlog hygiene, metrics, DoR for intake. A simple “Agile playbook” for the organisation. 	<ul style="list-style-type: none"> Embedded coaching (first 6–12 weeks) in planning, reviews, retros, refinement. Lightweight governance: clear decision rights + escalation path. Practical tooling norms (single backlog per product/value stream, clear intake). 	<ul style="list-style-type: none"> Visible wins (lead time down, fewer defects, faster decision cycles). Communities of practice (POs, SMS/Agile Leads, Engineers, Delivery Managers). Periodic maturity check-ins and operating model tweaks.
Failure Risks	<ul style="list-style-type: none"> “Agile as a process/theatre” (ceremonies without outcomes). Too abstract (“we’re becoming Agile”) with no concrete operating model changes. Treating Agile as a delivery-team problem, not a leadership/operating model shift. 	<ul style="list-style-type: none"> People feel “done to” (voluntold squads/roles). Middle management fears loss of control and quietly blocks. Agile is positioned as “more meetings” rather than less thrash / faster decisions. 	<ul style="list-style-type: none"> One-off training with no follow-through. Confusion between Agile, Scrum, Kanban, SAFe (“method shopping”). People learn ceremonies but can’t answer: “How do we plan? Prioritise? Measure? Escalate?” 	<ul style="list-style-type: none"> No time to do Agile properly (old work + new way = burnout). Dependencies and approval chains remain unchanged. Metrics incentivise the old behaviours (utilisation, “on time/on scope” only). 	<ul style="list-style-type: none"> Leadership attention moves on after launch. Teams revert when things get busy. “Agile exceptions” become the norm (“we don’t do retro this sprint”).

	A Awareness	D Desire	K Knowledge	A Ability	R Reinforcement
What is it?	<i>Ensure everyone understands the problem Agile solves (speed to value, predictability, quality, responsiveness) and the cost of staying as-is.</i>	<i>Build motivation and safety so people choose to try the new way, not just “attend the training”.</i>	<i>Provide role-based learning that translates into day-to-day behaviour.</i>	<i>Remove barriers and build capability through coaching in the flow of work.</i>	<i>Turn Agile from an “initiative” into “how we run the place”.</i>
Who is impacted?	<i>List the people/roles/teams/areas/personas targeted in this step</i>	<i>List the people/roles/teams/areas/personas targeted in this step</i>	<i>List the people/roles/teams/areas/personas targeted in this step</i>	<i>List the people/roles/teams/areas/personas targeted in this step</i>	<i>List the people/roles/teams/areas/personas targeted in this step</i>
Activities	<i>What activities do you plan to execute to build awareness?</i>	<i>What activities do you plan to execute to create desire?</i>	<i>What activities do you plan to execute to build knowledge?</i>	<i>What activities do you plan to execute to hone ability?</i>	<i>What activities do you plan to execute to reinforce the change?</i>
Key Milestones	<i>What are the key dates/milestones/events/deadlines for awareness?</i>	<i>What are the key dates/milestones/events/deadlines for desire?</i>	<i>What are the key dates/milestones/events/deadlines for knowledge?</i>	<i>What are the key dates/milestones/events/deadlines for ability?</i>	<i>What are the key dates/milestones/events/deadlines for reinforcement?</i>