

KANBAN

Are we there yet? If not, which way?



About Me



Helen Meek

Coach & Trainer

helen.meek@ripple-rock.com

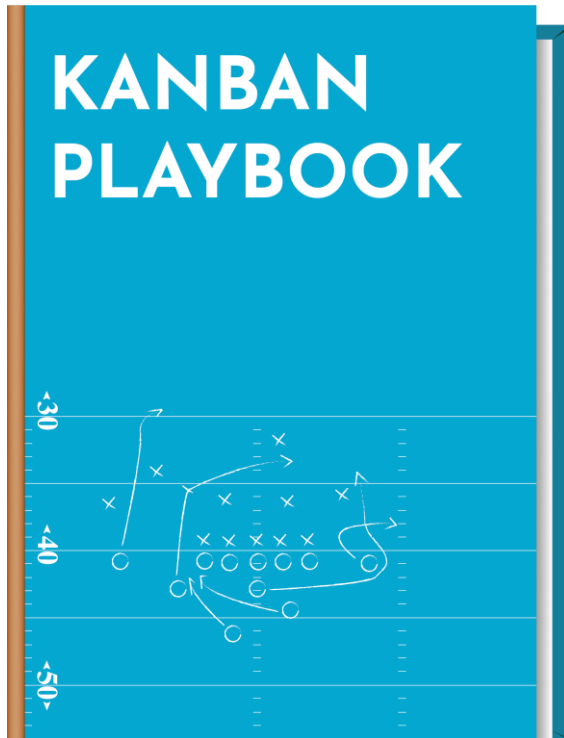
[@Helen_J_Meek](https://twitter.com/Helen_J_Meek)



Kanban Journey



The Kanban Playbook



- Created in 2019 to bring together 14 years of experience and knowledge.
- Helps drive successful evolution for organisations seeking to become fitter for their purpose.
- Maps 150+ Kanban practices against 7 levels of organisational maturity.

The Pillars



The Pillars - Culture



“How We Live”

- Principles we follow
- Things people value and stand for
- Behaviours that we see at all levels



The Pillars - Outcomes



“How Effective We Are”

- Demonstrated results
- Externally observable (customer-valued) benefits
- Internally observable (organisational) benefits
- How sustainable, robust, and likely to survive over the long term
- Resilience and demonstrate our ability to recover from setbacks and unexpected events.

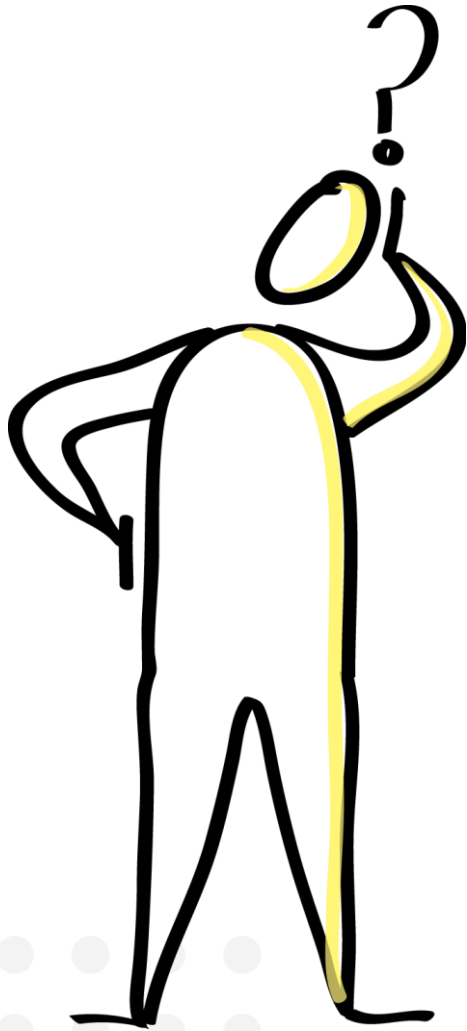
The Pillars - Practices



“How We Do Things”

- Routine activities
- Observable patterns of interactions, measures, metrics, decision frameworks etc.
- Settled, regular Kanban practices
- Habits

Discussion



What is the danger if we only look at Kanban as a set of practices?

How do we even get started? I hear you ask....



“

Start with What
You Do Now

”

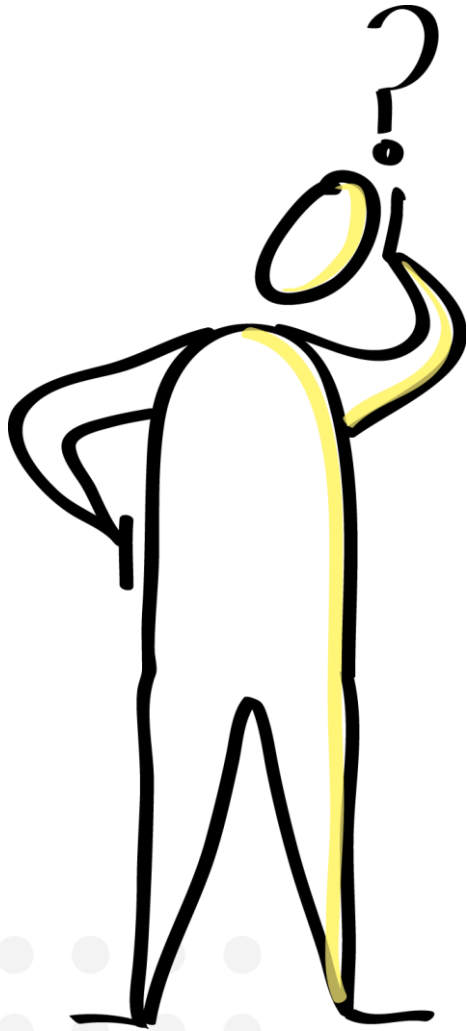
Identify Where We Are Now?



Identifying The Levels



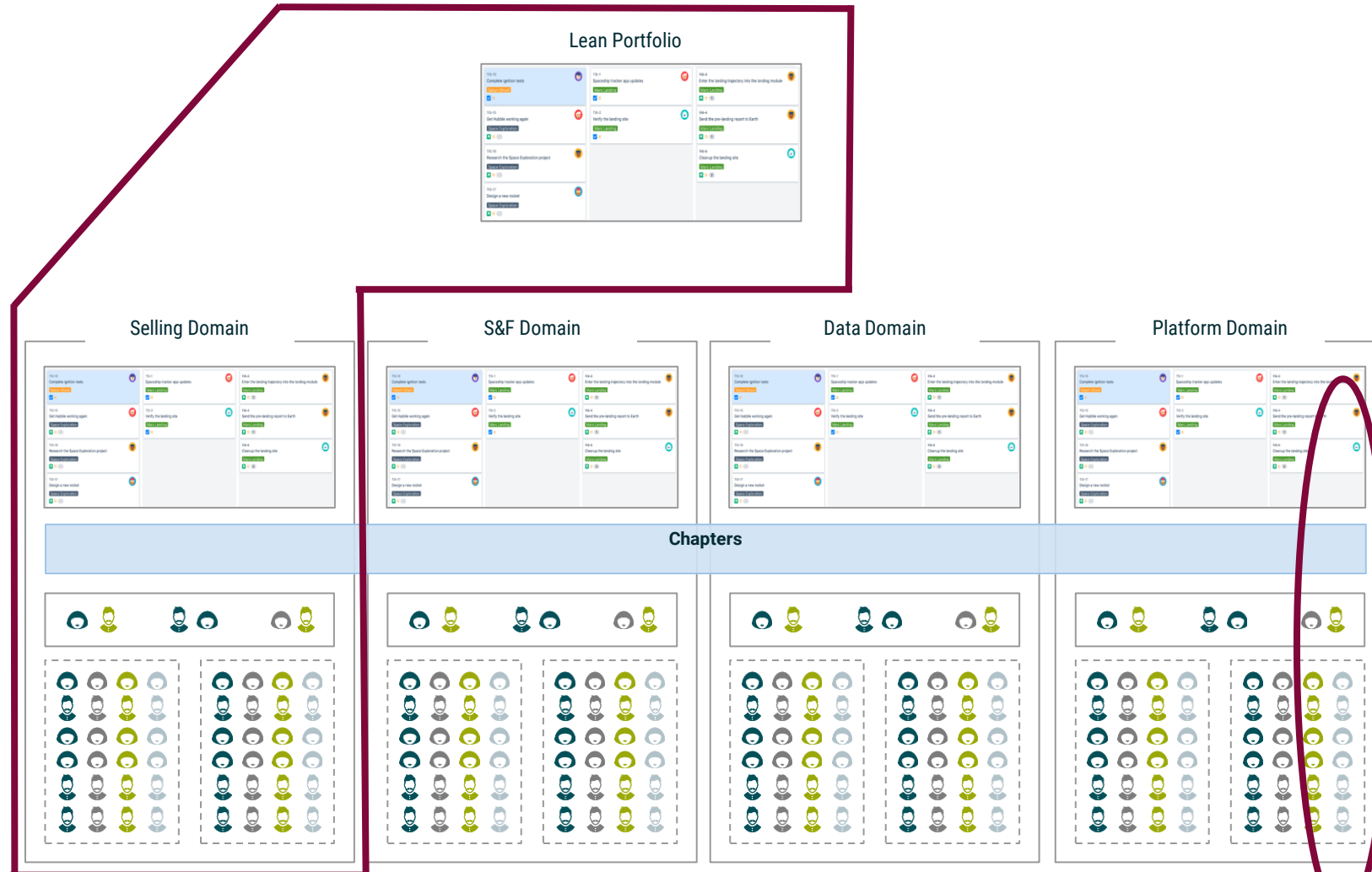
Discussion



When looking at the levels, do you look at them as a whole organisation, team or something else?



Examples



ML1 – Team Managed

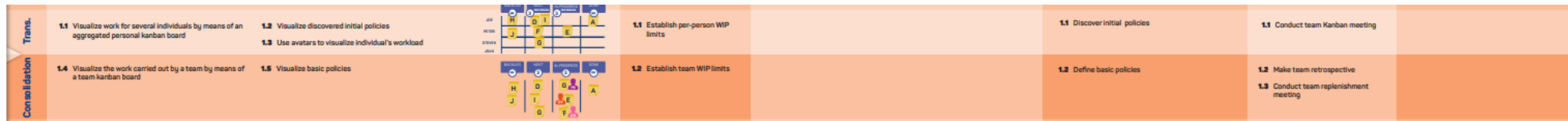
- Emerging culture of transparency, collaboration and teamwork. But teams are still unconnected.
- Emerging process, still inconsistent, individuals typically manage tasks, some teams start managing work items
- Focus on starting, rather than finishing
- Business outcome depend on individual's skills
- Metrics focused on applying the practices and individuals



Collaboration
Taking Initiative
Transparency

Deliverable

1 Team focused
· Emergent process
· Inconsistent outcomes
· Team Kanban



ML2 – Customer Driven



- Managerial heroics
- Processes, workflows, policies, and decision frameworks are understood, defined, and are repeatable
- Fail to meet customer expectations consistently

Product | Service

2 Customer-Driven

- Consistent process
- Inconsistent outcomes
- “Routine”
- Delivery Kanban
- Discovery Kanban
- End-to-end flow

	Transition								
Transition	<ul style="list-style-type: none"> 2.1 Visualize progress using a horizontal position on an emergent workflow kanban board 2.2 Visualize work types by means of card colors or board rows 2.3 Visualize blocked work items, defects and rework 	<ul style="list-style-type: none"> 2.4 Visualize work item aging 2.5 Visualize dependencies on another service or system 2.6 Visualize basic service policies 				<ul style="list-style-type: none"> 2.1 Define work types based on customer requests 2.2 Define basic services 2.3 Map upstream and downstream flow 2.4 Collect flow-related data 2.5 Capture the desired delivery date 		<ul style="list-style-type: none"> 2.1 Define flow-related metrics 2.2 Define basic service policies 	<ul style="list-style-type: none"> 2.1 Conduct workflow replenishment meeting 2.1 Identify sources of dissatisfaction
Consolidation	<ul style="list-style-type: none"> 2.7 Visualize constant WIP (CONWIP) on an emergent workflow delivery kanban board 2.8 Visualize concurrent or unordered activities with checkboxes 2.9 Visualize sequential activities where no dependency or preferred sequence exists using rows or vertical spaces 	<ul style="list-style-type: none"> 2.10 Visualize optional multiple unordered activities performed by specialist teams using partial rows 2.11 Visualize defined workflow using a kanban board 2.12 Visualize multiple services by means of aggregated service delivery overview board 				<ul style="list-style-type: none"> 2.1 Establish CONWIP limits on emergent workflow 2.2 Establish WIP limit on the aggregated service delivery overview board 	<ul style="list-style-type: none"> 2.6 Manage blocking issues 2.7 Manage defects and other rework types 2.8 Manage aging WIP 2.9 Implement Flow Manager role 	<ul style="list-style-type: none"> 2.3 Define policies for managing aging WIP 2.4 Define policies for managing blocking issues 2.5 Define policies for managing defects and other rework types 2.6 Define basic policies for dependency management 	<ul style="list-style-type: none"> 2.2 Conduct workflow Kanban meeting 2.3 Conduct blocker clustering 2.4 Conduct flow review 2.2 Identify sources of delay 2.3 Revise problematic policies 2.4 Define actions to develop basic understanding of the process and improve flow

Acts of Leadership
Customer Awareness
Evolutionary Change
Flow
Narrative
Respect
Understanding (internal)

ML3 – Fit For Purpose



- Consistent process & consistent outcome
- Manage upstream options, triage discipline
- Service delivery management, Classes of Services, Customer-oriented actionable metrics
- Basic operations / project portfolio review
- The service is fit-for-purpose

Agreement
Balance
Customer Service
Fitness for Purpose
Leadership at All Levels
Short-term Results
Understanding (external)
Unity & Alignment

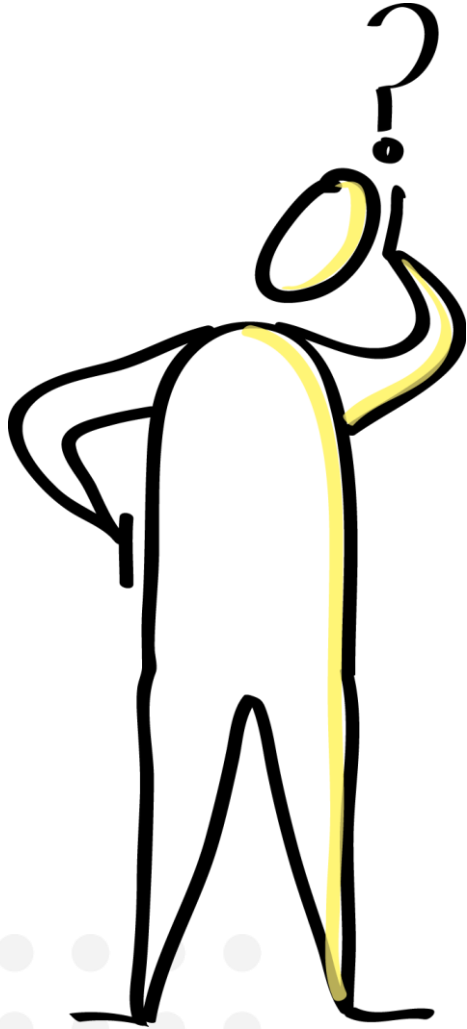
3

Fit-for-Purpose

- Consistent process
- Consistent outcomes
- Meet expectations
- Fit-For-Purpose
- Inconsistent economics

Transition	<ul style="list-style-type: none"> 3.1 Visualize "ready to commit" status, also known as "ready to pull" 3.2 Visualize request acceptance criteria, also known as "entry criteria" 3.3 Visualize workflow and team work items by means of aggregated team kanban board 3.4 Visualize upstream options by means of an upstream (discovery) kanban board 3.5 Visualize discarded options on an upstream kanban board 	<ul style="list-style-type: none"> 3.6 Visualize aborted work 3.7 Visualize class of service using ticket colors, board rows or ticket decorators 3.8 Visualize parent-child and peer-peer dependencies 3.9 Use a parking lot to visualize work requests dependent on another service or system currently waiting or blocked 3.10 Visualize pull signals 	<ul style="list-style-type: none"> 3.1 Establish activity based WIP limits 	<ul style="list-style-type: none"> 3.1 Organize around the knowledge discovery process 3.2 Defer commitment (decide before the "last responsible moment") 3.3 Determine the due date 3.4 Measure and analyze service fitness-for-purpose 3.5 Use cumulative flow diagram to monitor WIP 3.6 Use Little's law 	<ul style="list-style-type: none"> 3.7 Report rudimentary flow efficiency 3.8 Gradually eliminate infinite buffers 3.9 Actively close upstream requests which meet the abandonment criteria 3.10 Analyze and report aborted work 3.11 Use classes of service to affect selection 3.12 Analyze and report failure demand 	<ul style="list-style-type: none"> 3.1 Define F4P-related metrics 3.2 Explicitly define request acceptance criteria 3.3 Define upstream request abandonment policies 3.4 Define the meaning of "abandoned" for committed work 3.5 Define basic classes of service based on qualitative cost of delay 	<ul style="list-style-type: none"> 3.1 Conduct improvement suggestions review 	<ul style="list-style-type: none"> 3.1 Solicit change and improvement suggestions 3.2 Identify transaction and coordination costs
Consolidation	<ul style="list-style-type: none"> 3.11 Visualize replenishment signals 3.12 Visualize pull criteria (also known as "pull policies", "definition of done"/"exit criteria") 3.13 Visualize pullable 3.14 Visualize available capacity 3.15 Visualize failure demand versus value demand 3.16 Visualize target date or SLA 	<ul style="list-style-type: none"> 3.2 Use an order point (min limit) for upstream replenishment 3.3 Use a max limit to constrain upstream capacity 3.4 Bracket WIP limits across sub-states states 3.5 Bracket WIP limits across activities 3.6 Create a full kanban system 	<ul style="list-style-type: none"> 3.13 Develop triage discipline 3.14 Manage peer-to-peer or parent-child dependencies 3.15 Use two-phase commit for delivery commitment 3.16 Forecast Delivery 	<ul style="list-style-type: none"> 3.17 Apply qualitative Real Options Thinking 3.18 Implement Service Delivery Manager role 3.19 Implement Service Request Manager role 	<ul style="list-style-type: none"> 3.2 Establish a replenishment commitment point 3.7 Explicitly define pull criteria 3.8 Establish a delivery commitment point 3.9 Establish customer expectations for each work item or a class of work (items) 3.10 Explicitly define fitness-for-purpose and manage it based on metrics 	<ul style="list-style-type: none"> 3.2 Conduct replenishment meeting 3.3 Conduct delivery planning meeting 3.4 Conduct service delivery review (downstream) 3.5 Conduct service request review (upstream) 3.6 Conduct service risk review 	<ul style="list-style-type: none"> 3.3 Analyze blocker likelihood and impact 3.4 Analyze Lead time tail risk 3.5 After meetings: discuss a problem spontaneous bring it to the service delivery review 	

Discussion



What level does your gut instinct tell you that you at?



Where Next?

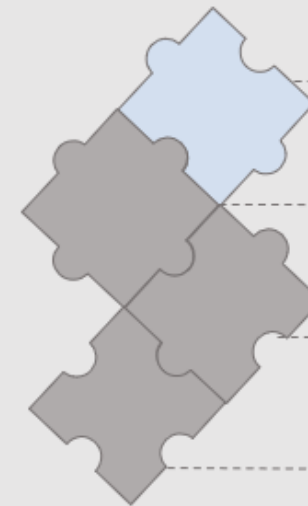
Kanban Maturity Model
A Map to Organizational Agility, Resilience, and Reinvention
Release 1.3

Organizational Maturity Level

Organizational Maturity Level	General Practices					
	VISUALIZE	LIMIT WIP	MANAGE FLOW	MAKE POLICIES EXPLICIT	FEEDBACK LOOPS	IMPROVE COLLABORATIVELY
0 Oblivious	Visualize work in progress	Limit work in progress	Manage flow of work	Make policies explicit	Feedback loops	Improve collaboratively
1 Team focused	Visualize work in progress	Limit work in progress	Manage flow of work	Make policies explicit	Feedback loops	Improve collaboratively
2 Customer-Driven	Visualize work in progress	Limit work in progress	Manage flow of work	Make policies explicit	Feedback loops	Improve collaboratively
3 Fit-for-Purpose	Visualize work in progress	Limit work in progress	Manage flow of work	Make policies explicit	Feedback loops	Improve collaboratively
4 Risk-Hedged	Visualize work in progress	Limit work in progress	Manage flow of work	Make policies explicit	Feedback loops	Improve collaboratively
5 Market Leader	Visualize work in progress	Limit work in progress	Manage flow of work	Make policies explicit	Feedback loops	Improve collaboratively
6 Built for Survival	Visualize work in progress	Limit work in progress	Manage flow of work	Make policies explicit	Feedback loops	Improve collaboratively

From ML1 to ML2

Consider for example to...



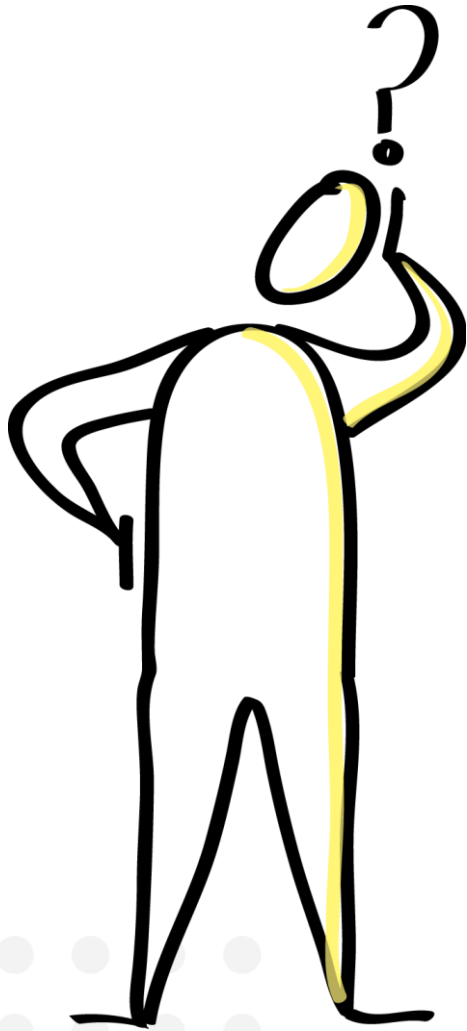
Balance demand vs. capabilities

Manage blocking issues

Establish replenishment

Establish activity-based WIP limits

Discussion



What is the risk if you try to jump too many steps in the model?



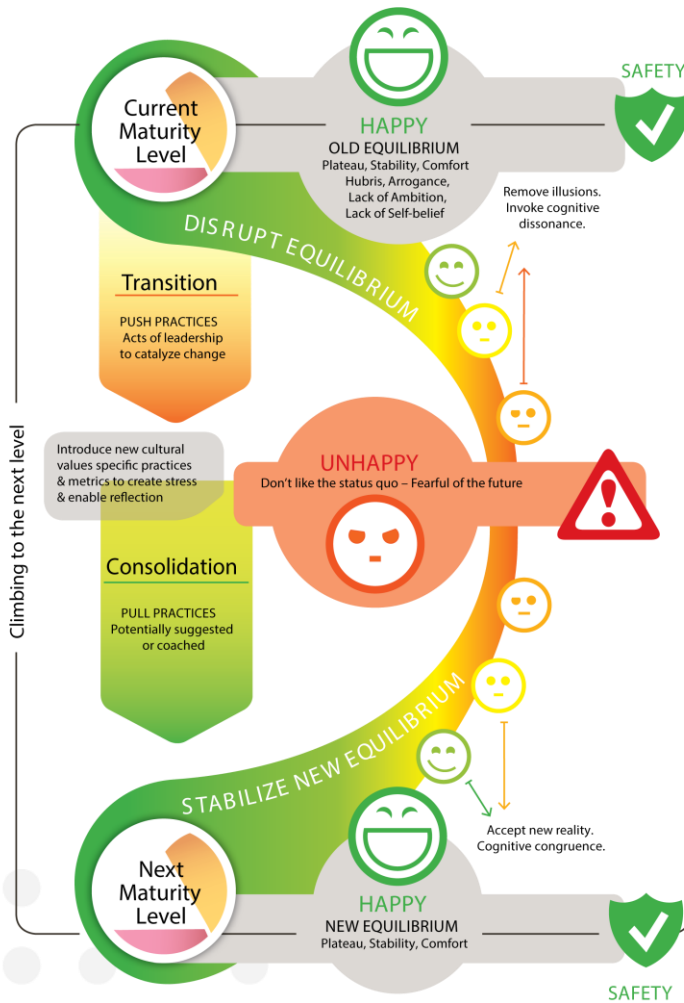
Over Reaching!



Tips to Mitigate Overreaching

- Focus on solving the problem, on the system, not on individuals
- Patience, recognising that a small improvement is still an improvement
- Make policies explicit around decision making

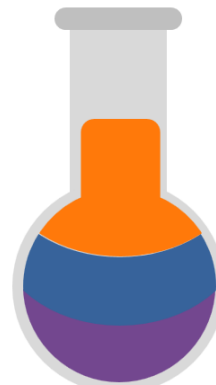
Techniques To Help You



- ## The Evolutionary Change Model
- Encourages evolving in small steps!
 - Codifies the guidance to drive evolutionary change.

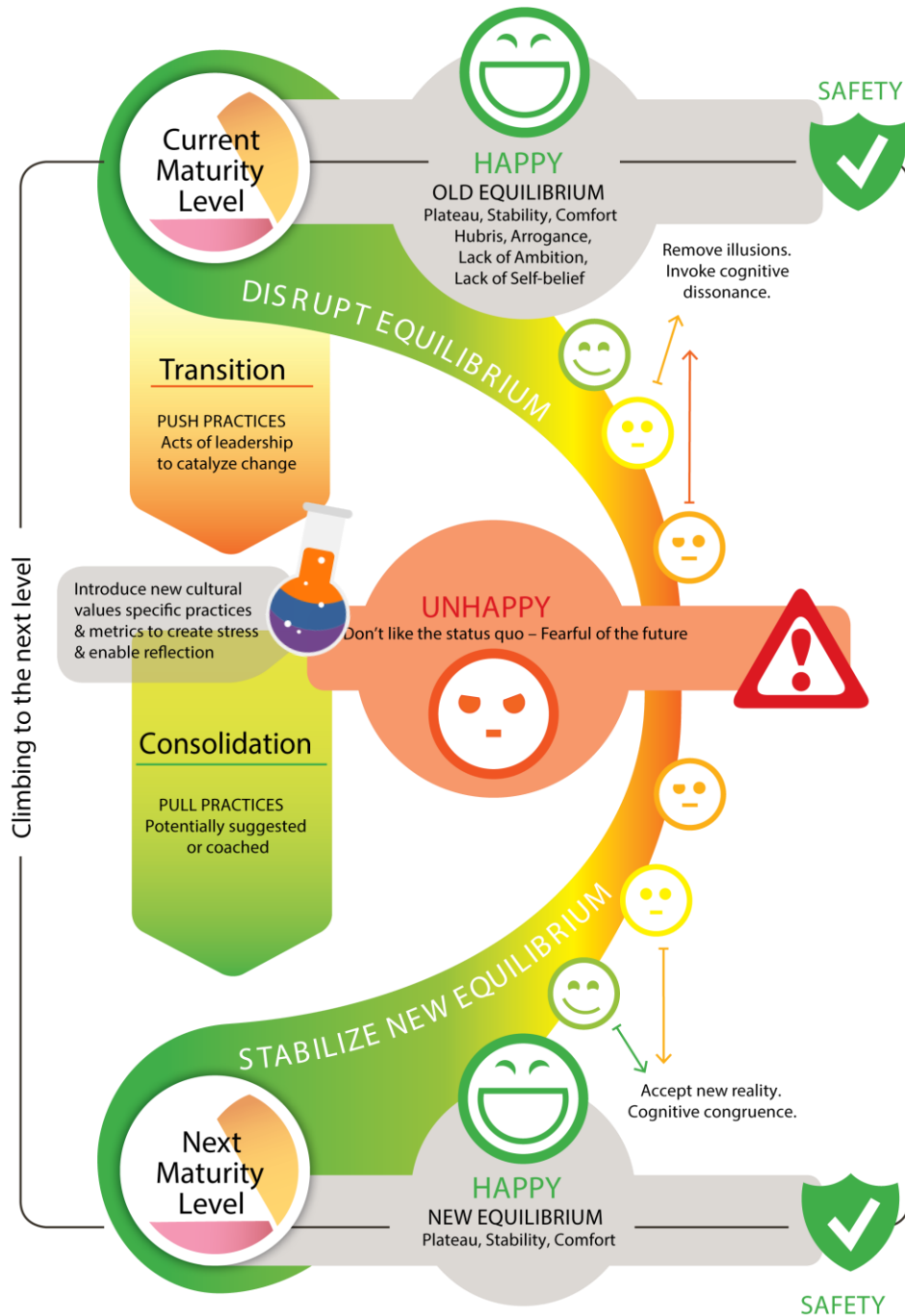
Evolutionary Change Formula

How to lead change in times of equilibrium.



The Change Potion





How to apply the Evolutionary Change Formula through the Transition Practices **to disrupt the stasis**, fostering willingness to improve, and the associated risks.

How to **stabilise a new equilibrium** using Consolidation Practices together with the associated risks.

The Architecture

Transition Practices

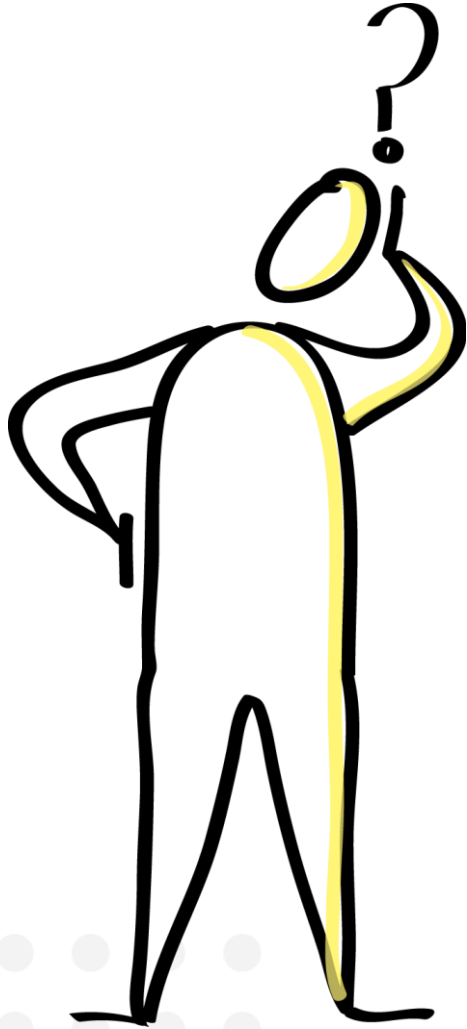
Consolidation Practices

Kanban Practices per level



1 Team focused <ul style="list-style-type: none">• Emergent process• Inconsistent outcomes• Team Kanban	1 → 2	Consolidation	VZ 1.4 Visualize the work carried out by a team by means of a team kanban board	VZ 1.5 Visualize basic policies
		Transition	VZ 2.1 Visualize progress using a horizontal position on an emergent workflow kanban board	VZ 2.4 Visualize work item aging
			VZ 2.2 Visualize work types by means of card colors or board rows	VZ 2.5 Visualize dependencies on another service or system
			VZ 2.3 Visualize blocked work items, defects and rework	VZ 2.6 Visualize dependencies on shared services using avatars
				VZ 2.7 Visualize basic service policies

Break Out Discussions

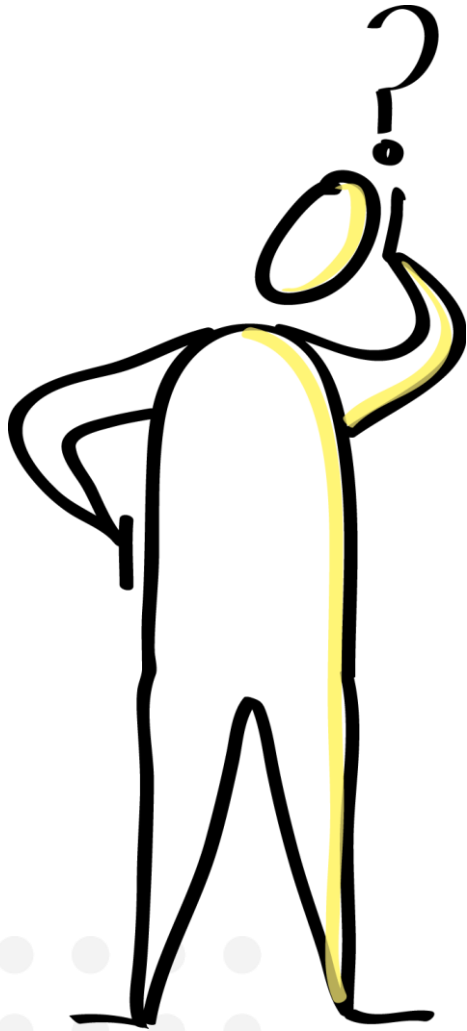


In groups review the practices poster.

Think about where your next evolution could be.



Discussions



Based upon what you have seen,
what is one change you could make
tomorrow?



Kanban Maturity Model



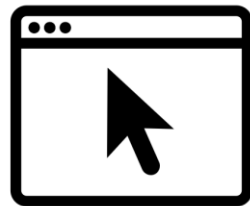
Questions??





My Training Class

<https://ripple-rock.com/training-courses/>
25 % off for BCS. Use code:BCS25



KMM Plus (Includes book, casestudies, posters etc)

<https://kmm.plus/en/>

