CONTINUUM
LEVEL UP
TOOLKIT
LEVEL
UP

WHAT IS CONTINUUM?

CONTINUUM

=

DRIVING CONTINUOUS IMPROVEMENT

- Understand and improve our current practices and capabilities
- Increase ability to respond to changing business conditions
- Better leverage innovation

WHAT IS CONTINUUM?

Assessing team or organizational capabilities in responding to change

Planning and executing improvements to existing practices

CONTINUUM HELPS YOU LEVEL UP!

A FEW POINTS TO CONSIDER

Continuum provides a platform for a team assessment not an individual assessment.

Continuum should not be used as a tool to compare teams but rather to encourage continuous improvement.

The focus should be on improving the teams current competencies and recognising the steps needed to “level up” rather than focusing on the mature end state.

Competencies can be mastered individually or as a group.

Continuum should not be used as a performance measurement tool.

HOW DOES IT WORK?

SKILL DIMENSIONS

INDIVIDUALS AND INTERACTIONS

- TEAMING
- FEATURE TEAMING

CUSTOMER COLLABORATION

- STRATEGY ALIGNMENT
- SOFTWARE DESIGN
- RELEASE MANAGEMENT

RESPONDING TO CHANGE

- CONTINUOUS INTEGRATION
- PLANNING AND REQUIREMENT MANAGEMENT
- RISK AND ISSUE MANAGEMENT

WORKING SOFTWARE

- ENVIRONMENTS
- QUALITY ASSURANCE
- CODING PRACTICES
- INCIDENT MANAGEMENT

TEAMING

INDIVIDUALS AND INTERACTIONS

Teaming:
A group of craftsmen that work together to achieve a specific goal.

The goal:
To achieve a high level of accountability, collaboration, autonomy and transparency as a team effort.

Examples:
Guilds, CoP, CoI, team of designers.

FEATURE TEAMING

INDIVIDUALS AND INTERACTIONS

FEATURE TEAMING

Feature Teaming:
A self-organised, cross-functional team which consists of individuals involved in the end to end delivery of a product or a service.

The goal:
To develop into a team responsible for all aspects of a feature delivery, from idea stage to production and support.

Example:
Online Share Trading team, Investment and wealth team.

STRATEGY ALIGNMENT

Strategy Alignment:
The ability to align the decision making with the organisation’s business and technical objectives.

The goal:
To advance to a state where the team can actively participate in strategy planning - for Business and IT

Example:
Strategy is used to push back on initiatives that are not prioritised.

SOFTWARE DESIGN

CUSTOMER COLLABORATION

Software Design:
The overall design of a system or the technical decisions involved while creating the system.

The goal:
Leveling up towards the JIT (just in time) design culture with full traceability between requirements and design.

Example:
Design done JIT by the team, designer part of the team.

RELEASE MANAGEMENT

CUSTOMER COLLABORATION

Release Management:
Planning, managing and governing releases of software.

The goal:
Leveling up to the state where releases can be deployed at any time and are fully automated.

Example:
The team controls the release.

CONTINUOUS INTEGRATION

Continuous Integration:
A software development practice whereby each team member frequently integrates their work (at least daily), leading to multiple integrations per day. Each integration is verified by an automated build (including testing) to detect integration errors as quickly as possible. This dimension includes source control and configuration management.

The goal:
To advance to a state where the build pipeline extends directly into production and where the decision to release to the customer is made by Business (just a push of a button).

Example:
Test driven development and test driven infrastructure practices adopted by the team. Automation of build, test, deployment, change, configuration and release processes are in place.

PLANNING AND REQUIREMENT MANAGEMENT

RESPONDING TO CHANGE

Planning and Requirements Management:
The breakdown of work, work prioritisation, estimation techniques, iterative planning and effective management of requirements from definition, through traceability, to delivery.

The goal:
To advance to the state where the team is defining requirements JIT, linked to the objective outcomes and prioritised by business value.

Example:
The team is able to respond to changing requirements without any issues.

RISK AND ISSUE MANAGEMENT

Risk and Issue Management:
The process of identifying, assessing, prioritizing and resolving risks and issues.

The goal:
Leveling up to a state where risk management is used to identify potential opportunities.

Example:
Measure’s are in place to demonstrate the effectiveness of risk management.

ENVIRONMENTS

WORKING SOFTWARE

Environments:
Environments available and easily configurable with accurate and consistent data across platforms. It encompasses all IT environments (from development to production) and data management.

The goal:
To advance to a state where the environments are an integral part of the code build pipeline and managed by the team.

Example:
Provisioning of environments is fully automated. Data and environments are treated as code.

QUALITY ASSURANCE

WORKING SOFTWARE

Quality Assurance:
The Quality assurance dimension ensures that practices and processes are of a high standard. It’s about validation, are we building the right system, and verification, are we building the system right.

The goal:
Create a platform where Tests drive release decisions and the entire team owns quality.

Example:
Tests are written upfront, code is created to pass the tests. There is high level of test automation and code coverage and anyone can execute the tests.

QUALITY ASSURANCE

CODING PRACTICES

WORKING SOFTWARE

Coding Practices:
A set of practices aimed at improving the quality of software.

The goal:
To reach a state where coding standards and practices are regularly reviewed and updated through continuous improvement.

Example:
Constant maintenance and grooming of the code base.
Minimized technical debt.

INCIDENT MANAGEMENT

WORKING SOFTWARE

Incident Management:
Restoring the service as quickly as possible and detecting the underlying cause of an incident to resolve and prevent it from occurring in future.

The goal:
Leveling up to state where the team is able to use normal process to do emergency changes.

Example:
Accept that failure is inevitable, that changes will break things, and that the question becomes “how soon we can restore service?”.

LEVELS

1 TRAVELLER

2 ARTISAN

3 EXPERT

4 PROFESSIONAL

5 MASTER

LEVELS

1 TRAVELLER

2 ARTISAN

3 EXPERT

4 PROFESSIONAL

5 MASTER

LEVELS

The team commences the journey of continuous improvement as a **Traveller**.

At the **Traveller** level we will generally have unrepeatable and poorly controlled processes and reactive behaviour.

Artisan level demonstrates the first step of improvement, obtaining initial craftsmanship skills to enable the journey forward.

Expert level expresses achieving a mature level of continuous delivery practices where you benefit from the larger effects.

Professional level suggests a level of proficiency where metrics and trends are tracked and acted on. It includes practices that will bring substantial value and effect to a higher effort.

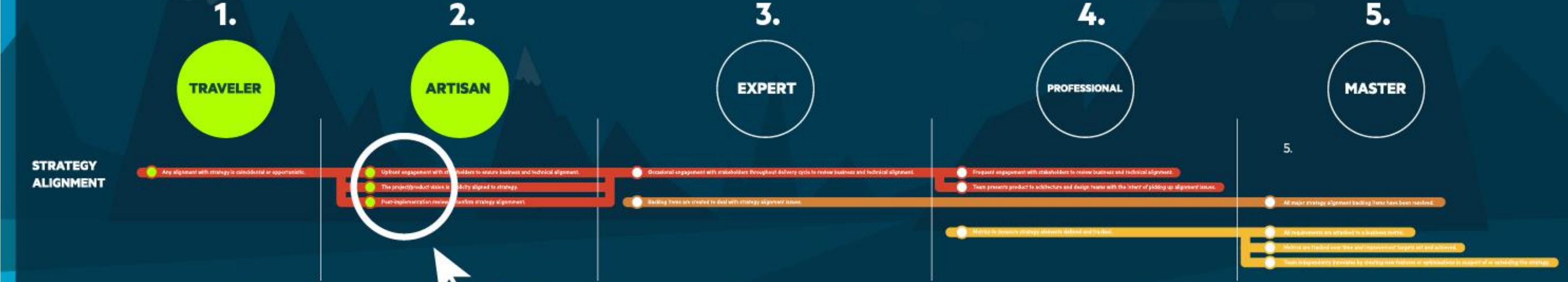
Master level = “continuous improvement”

HOW DOES IT WORK?

1. Assess current practice of all skills

Each member assesses the current performance of the team across all dimensions.
The overall maturity of each dimension is derived from the assessments.

PLOT ON SKILLS TREE


1.
TRAVELER
2.
ARTISAN
3.
EXPERT
4.
PROFESSIONAL
5.
MASTER
STRATEGY ALIGNMENT
Any alignment with strategy is coincidental or opportunistic.
Upfront engagement with stakeholders to ensure business and technical alignment.
The project/product vision is explicitly aligned to strategy.
Post-implementation reviews confirm strategy alignment.
Occasional engagement with stakeholders throughout delivery cycle to review business and technical alignment.
Backlog items are created to deal with strategy alignment issues.
Frequent engagement with stakeholders to review business and technical alignment.
Team presents product to achitecture and design teams with the intent of picking up alignment issues.
Metrics to measure strategy elements defined and tracked.
5.
All major strategy alignment backlog items have been resolved.
All requirements are attached to a business metric.
Metrics are tracked over time and improvement targets set and achieved.
Team independently innovates by creating new features or optimisations in support of or extending the strategy.

Upfront engagement with sta
The project/product vision is e
Post-implementation review t

Mark which attributes are attained

The derived maturity level together with the lowest and highest scores is used to facilitate discussion and agree as a team on team's maturity per dimension.

HOW DOES IT WORK?

2. Team decides on priority skills to level up

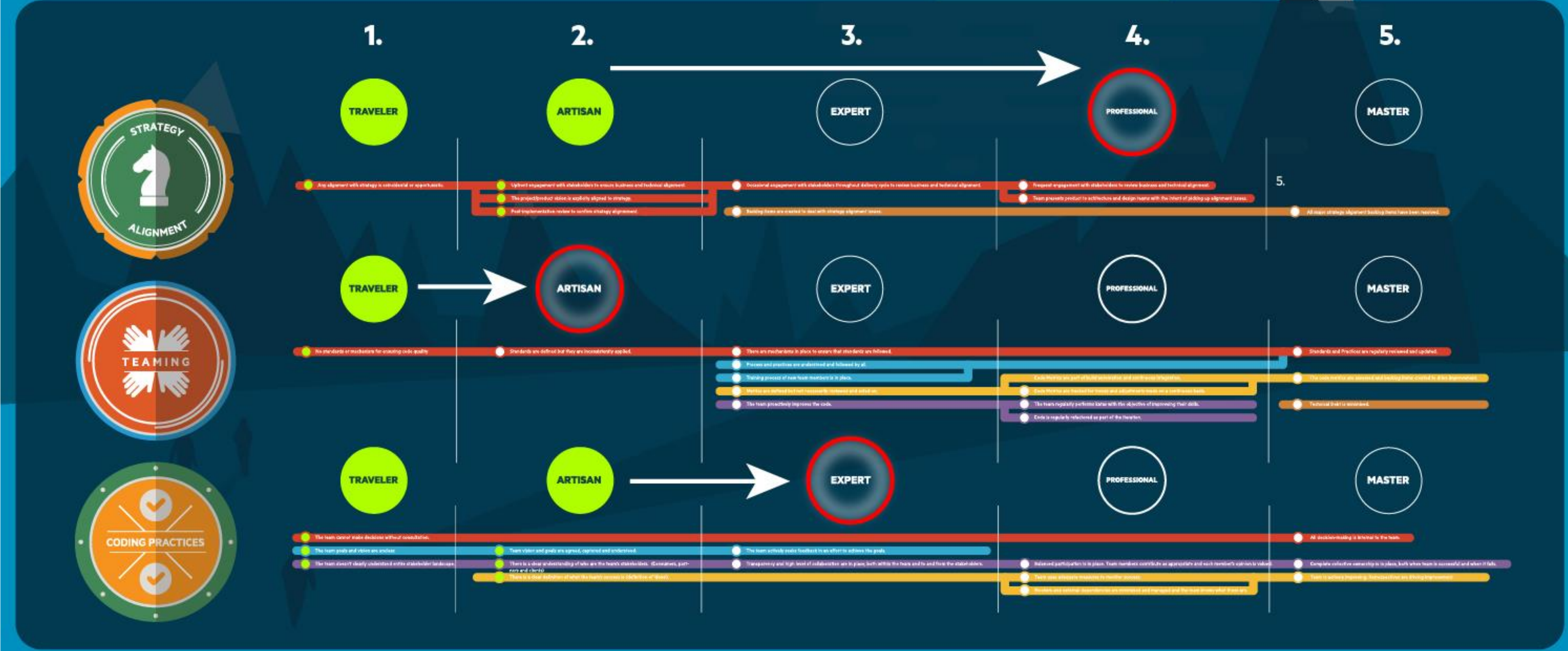
TEAMING

STRATEGY ALIGNMENT

CODING PRACTICES

HOW DOES IT WORK?

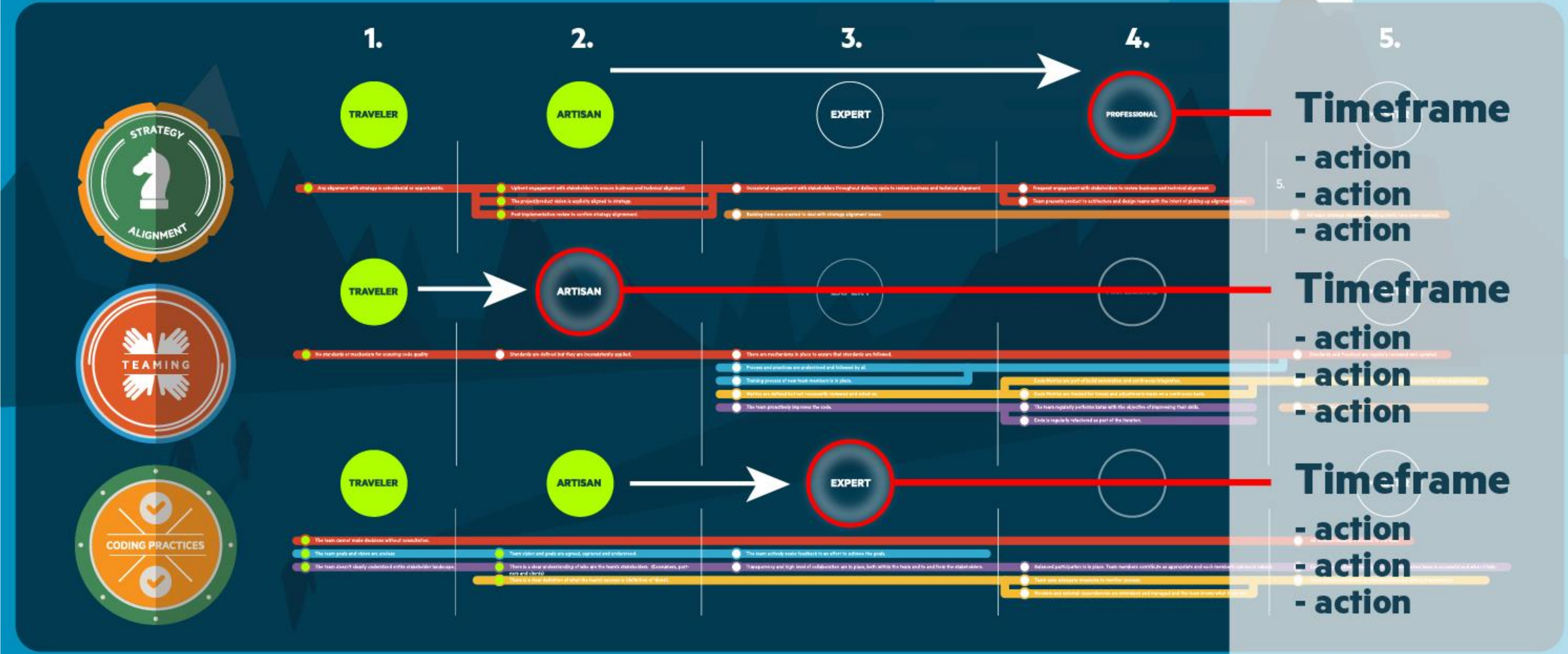
3. Agree on target state for each skill



	1.	2.	3.	4.	5.
STRATEGY ALIGNMENT	TRAVELER	ARTISAN	EXPERT	PROFESSIONAL	MASTER
	Any alignment with strategy is coincidental or opportunistic.	Upfront engagement with stakeholders to ensure business and technical alignment. The project/product vision is explicitly aligned to strategy. Post-implementation review to confirm strategy alignment.	Occasional engagement with stakeholders throughout delivery cycle to review business and technical alignment. Backlog items are created to deal with strategy alignment issues.	Frequent engagement with stakeholders to review business and technical alignment. Team presents product to architecture and design teams with the intent of picking up alignment issues.	5. All major strategy alignment backlog items have been resolved.
TEAMING	TRAVELER	ARTISAN	EXPERT	PROFESSIONAL	MASTER
	No standards or mechanism for ensuring code quality	Standards are defined but they are inconsistently applied.	There are mechanisms in place to ensure that standards are followed. Process and practices are understood and followed by all. Training process of new team members is in place. Metrics are defined but not necessarily reviewed and acted on. The team proactively improves the code.	Code Metrics are part of build automation and continuous integration. Code Metrics are tracked for trends and adjustments made on a continuous basis. The team regularly performs katas with the objective of improving their skills. Code is regularly refactored as part of the iteration.	Standards and Practices are regularly reviewed and updated. The code metrics are assessed and backlog items created to drive improvement. Technical Debt is minimized.
CODING PRACTICES	TRAVELER	ARTISAN	EXPERT	PROFESSIONAL	MASTER
	The team cannot make decisions without consultation. The team goals and vision are unclear. The team doesn't clearly understand entire stakeholder landscape.	Team vision and goals are agreed, captured and understood. There is a clear understanding of who are the team's stakeholders. (Consumers, partners and clients) There is a clear definition of what the team's success is (definition of 'done').	The team actively seeks feedback in an effort to achieve the goals. Transparency and high level of collaboration are in place, both within the team and to and from the stakeholders.	Balanced participation is in place. Team members contribute as appropriate and each member's opinion is valued. Team uses adequate measures to monitor success. Blockers and external dependencies are minimized and managed and the team knows what these are.	All decision-making is internal to the team. Complete collective ownership is in place, both when team is successful and when it fails. Team is actively improving. Retrospectives are driving improvement.

HOW DOES IT WORK?

4. Prepare a time bound action plan to reach that level



Any alignment with strategy is coincidental or opportunistic.

Upfront engagement with stakeholders to ensure business and technical alignment.

The project/product vision is explicitly aligned to strategy.

Post-implementation review to confirm strategy alignment.

Occasional engagement with stakeholders throughout delivery cycle to review business and technical alignment.

Backlog items are created to deal with strategic alignment issues.

Frequent engagement with stakeholders to review business and technical alignment.

Team presents product to architecture and design teams with the intent of picking up alignment issues.

Timeframe
- action
- action
- action

No standards or mechanism for ensuring code quality.

Standards are defined but they are inconsistently applied.

There are mechanisms in place to ensure that standards are followed.

Process and practices are understood and followed by all.

Training process of new team members is in place.

Metrics are defined but not necessarily reviewed and acted on.

The team proactively improves the code.

Code Metrics are part of build automation and continuous integration.

Code Metrics are tracked for trends and adjustments made on a continuous basis.

The team regularly performs katas with the objective of improving their skills.

Code is regularly refactored as part of the iteration.

Timeframe
- action
- action
- action

The team cannot make decisions without consultation.

The team goals and vision are unclear.

The team doesn't clearly understand entire stakeholder landscape.

Team vision and goals are agreed, captured and understood.

There is a clear understanding of who are the team's stakeholders. (Consumers, partners and clients)

There is a clear definition of what the team's success is (definition of "done").

The team actively seeks feedback in an effort to achieve the goals.

Transparency and high level of collaboration are in place, both within the team and to and from the stakeholders.

Balanced participation is in place. Team members contribute as appropriate and each member's [illegible]

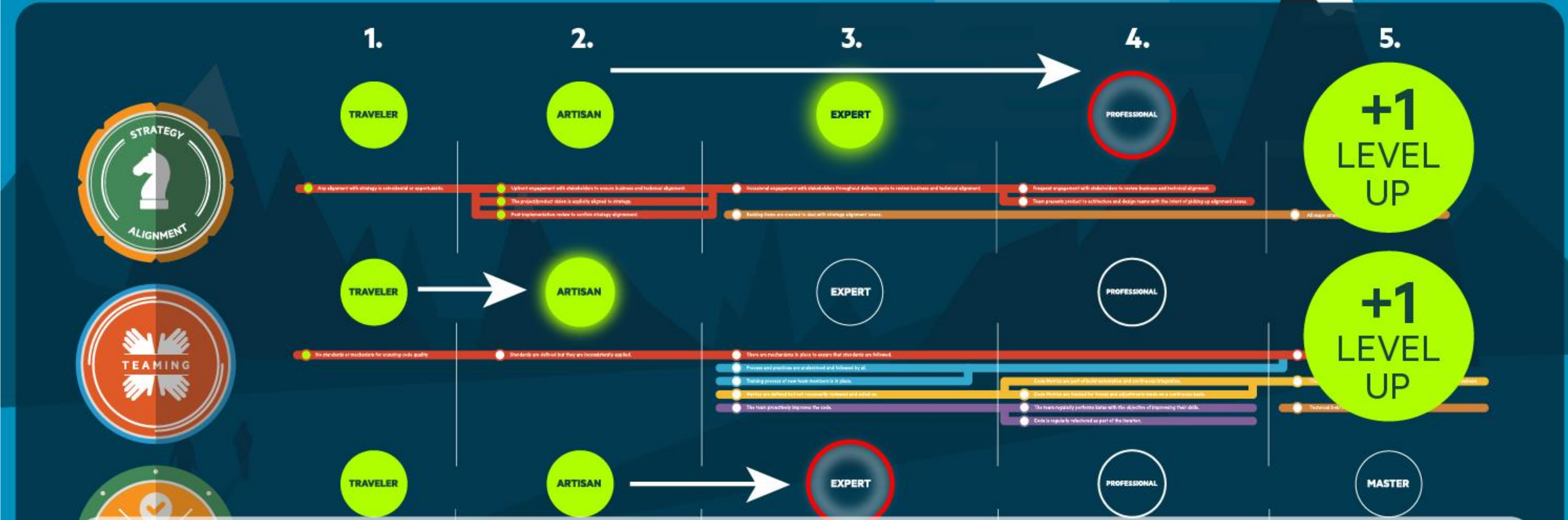
Team uses adequate measures to monitor success.

Vendors and external dependencies are minimised and managed and the team knows what [illegible]

Timeframe
- action
- action
- action

HOW DOES IT WORK?

4. Assess on all skills and progress towards states periodically


1.
2.
3.
4.
5.
STRATEGY ALIGNMENT
TRAVELER
ARTISAN
EXPERT
PROFESSIONAL
+1 LEVEL UP
Any alignment with strategy is coincidental or opportunistic.
Upfront engagement with stakeholders to ensure business and technical alignment.
The project/product vision is explicitly aligned to strategy.
Post-implementation review to confirm strategy alignment.
Occasional engagement with stakeholders throughout delivery cycle to review business and technical alignment.
Frequent engagement with stakeholders to review business and technical alignment.
Team presents product to architecture and design teams with the intent of picking up alignment issues.
Backlog items are created to deal with strategy alignment issues.
All major stra
TEAMING
TRAVELER
ARTISAN
EXPERT
PROFESSIONAL
+1 LEVEL UP
No standards or mechanism for ensuring code quality
Standards are defined but they are inconsistently applied.
There are mechanisms in place to ensure that standards are followed.
Process and practices are understood and followed by all.
Training process of new team members is in place.
Metrics are defined but not necessarily reviewed and acted on.
The team proactively improves the code.
Code Metrics are part of build automation and continuous integration.
Code Metrics are tracked for trends and adjustments made on a continuous basis.
The team regularly performs katas with the objective of improving their skills.
Code is regularly refactored as part of the iteration.
Technical Debt
CODING PRACTICES
TRAVELER
ARTISAN
EXPERT
PROFESSIONAL
MASTER

- Cannot jump the levels -

Must have attained all attributes to level up in a skill

WHAT YOU GET

STAY A LEVEL 5 ENGINEERING TEAM

BY CONTINUOUSLY IMPROVING YOUR SKILLS