

AGILE ESSENTIALS

Understanding agile product management

PROJECTS & PRODUCTS



Projects vs products?

Projects	Products
Led by a Project Manager	Led by a Product Manager (Product Owner)
Fixed, short term	Ongoing, long term
Deliverable focussed	Value focussed
Assumed certainty through upfront planning	Acceptance of change/uncertainty
Predictive planning and risk management	Adaptive planning and continuous risk management
Stage gates and tasks measure progress	Continuous delivery measures progress
Change management through contracts	Change embedded into regular stakeholder feedback loops
Time, budget, scope metrics	Impact and outcome metrics
Value handed over at the end	Value delivered continuously
Project work not BAU work (ad-hoc delivery)	Projects, BAU and all work for the team
Distribute people across the project(s)	Take work to the (long-lived) team

Agile product management

Agile's focus is product management



The project organisation

- Internally focussed
- Fixed mindset
- Defend existing processes
- Deliver functionality for shareholders

The project team

- Top-down
- Individual goals and responsibilities
- Little interaction
- Group work

The project team

- Committee driven
- Management hierarchies
- Capability silos
- Groupwork



The agile organisation

- Externally focussed
- Growth mindset
- Create new practices
- Deliver value to citizens

The agile team

- Self-managing
- Cross-functional
- Much interaction
- Team goals & responsibilities

The agile team

- Product led
- Servant leadership
- Networks of teams
- Team work

Law of the Customer

Law of the Small Team

Law of the Network

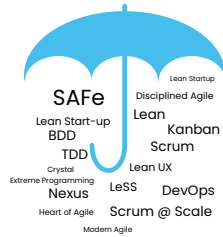
Adapted from: Steve Denning (2016)

Agile product management

Agile is an umbrella term.
Agile is **not** a methodology!

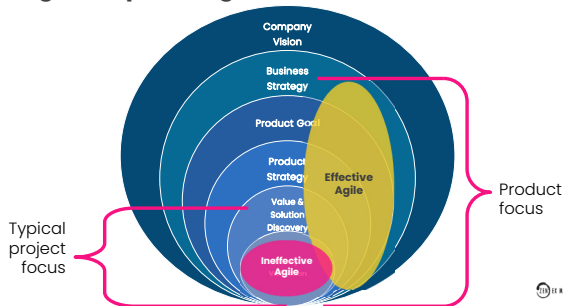
Agile is a different way of thinking that seeks alternatives to traditional project management by focussing on **products**, outcomes and value.

Agile approaches are enacted to help teams respond to unpredictability through incremental, iterative work cadences, known as "Sprints". Agile frameworks are an alternative to waterfall, or traditional sequential work processes.



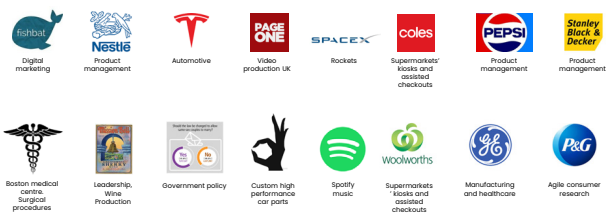
TE BACHTER

Agile's operating model

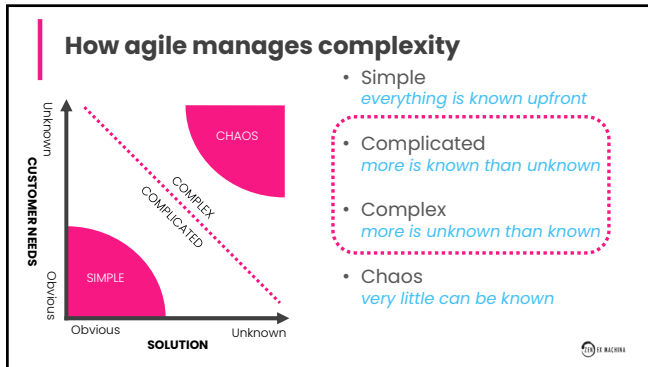


TE BACHTER

Agile manages complexity



TE BACHTER



Complexity and leadership

Environment	Characteristics	Leader's Job
Chaos	<ul style="list-style-type: none"> High turbulence No clear cause-and-effect Unknowables Many decisions and no time 	<ul style="list-style-type: none"> Immediate action to re-establish order Prioritise and select actionable work Look for what works rather than perfection Act first, examine the environment, respond
Complex	<ul style="list-style-type: none"> More unpredictability than predictability Emergent answers Many competing ideas 	<ul style="list-style-type: none"> Create guardrails for action Increase levels of interaction and communication Servant leadership Generate ideas Experiment, assess the results, decide what to do next based on the results
Complicated	<ul style="list-style-type: none"> More predictability than unpredictability Fact-based management Experts work out wrinkles 	<ul style="list-style-type: none"> Utilise experts to gain insights Use impact metrics to gain control Assess what good practice to use, analyse the results, decide what to do next
Simple	<ul style="list-style-type: none"> Repeating patterns and consistent events Clear cause-and-effect Well established knowns Fact-based management 	<ul style="list-style-type: none"> Use best practice Extensive communication not necessary Establish patterns and optimise to them Command and control

Complexity and leadership

Environment	Characteristics	Leader's Job
Chaos	<ul style="list-style-type: none"> High turbulence No clear cause-and-effect Unknowables Many decisions and no time 	<ul style="list-style-type: none"> Immediate action to re-establish order Prioritise and select actionable work Look for what works rather than perfection Act first, examine the environment, respond
Complex	<ul style="list-style-type: none"> More unpredictability than predictability Emergent answers Many competing ideas 	<ul style="list-style-type: none"> Create guardrails for action Increase levels of interaction and communication Servant leadership Generate ideas Experiment, assess the results, decide what to do next based on the results
Complicated	<ul style="list-style-type: none"> More predictability than unpredictability Fact-based management Experts work out wrinkles 	<ul style="list-style-type: none"> Utilise experts to gain insights Use impact metrics to gain control Assess what good practice to use, analyse the results, decide what to do next
Simple	<ul style="list-style-type: none"> Repeating patterns and consistent events Clear cause-and-effect Well established knowns Fact-based management 	<ul style="list-style-type: none"> Use best practice Extensive communication not necessary Establish patterns and optimise to them Command and control

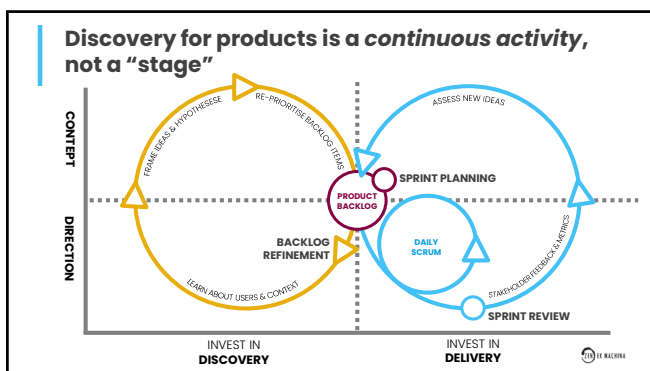
Law of the Customer



Agile is “obsessed with delivering value to customers. The primary importance of the customer is recognised in the first principle of the Agile Manifesto.”

Steve Denning (2016) Forbes.

THE BACCHINI



Fastest path to minimal viable product, release, then iterate

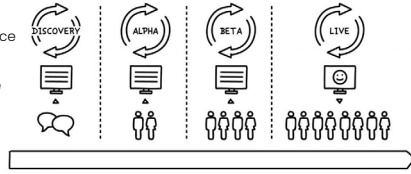
- Work Sprint-to-Sprint
- Ensure work is potentially releasable
- Release
- Get feedback
- Iterate based on feedback
- Get more feedback



c2000 - 2010

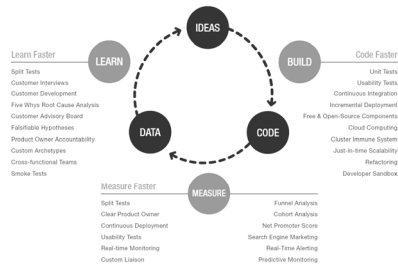
Lean Startup / DTA DSS

Adapted from **gov.uk's** Service Standard, the Lean Startup pattern often ends up being implemented as linear stage gates (e.g. using PRINCE2 project management)



Lean Startup

What is the smallest thing we can get sufficient feedback on to understand whether our product idea is viable, so we can then spend more money on it?



DEFINING WORK FOR PRODUCT TEAMS

Writing product scope

- Generic term: Product Backlog Item (PBI from Scrum)
- Often referred to as a "User Story" (Extreme Programming)
- Define just enough of the problem / scope for the team to start delivering the work
- Not traditional "requirements" to "sign off" (the governance is different)

* Stories, intentional design, ADRs, and a workflow/approval tool are sufficient for ANAO audit processes.



User Stories format

TITLE

Attack on the Death Star

PERSPECTIVE

As Princess Leia

NEED

I want the Death Star destroyed
To end the tyrant of the Empire in this sector

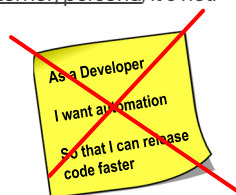
REASON

Author: General Nadine Date: 10.2910. BY Size: S

User Stories format

The "User" is the end user/customer/persona, it's not:

- You.
- Your team.
- The stakeholder.
- Your organisation.
- An IT system.
- An application.



Other ways to write product scope

I	Independent	<ul style="list-style-type: none"> Items can be delivered in any order, based on value.
N	Negotiable	<ul style="list-style-type: none"> Use sequencing to account for dependencies. Don't treat everything as MOSCOW "must have". Work is co-created by the team, the PO and customers.
V	Valuable	<ul style="list-style-type: none"> Items have identifiable value to customers. Work creates an impact and positive outcomes.
E	Estimable	<ul style="list-style-type: none"> The scope boundaries about "what" is needed are known. The team can forecast how large the work is.
S	Small	<ul style="list-style-type: none"> Work fits into a Sprint.
T	Testable	<ul style="list-style-type: none"> There is clear criteria that demonstrates the outcome has been met, i.e. has "acceptance criteria".

What about 'acceptance criteria' and other important things?

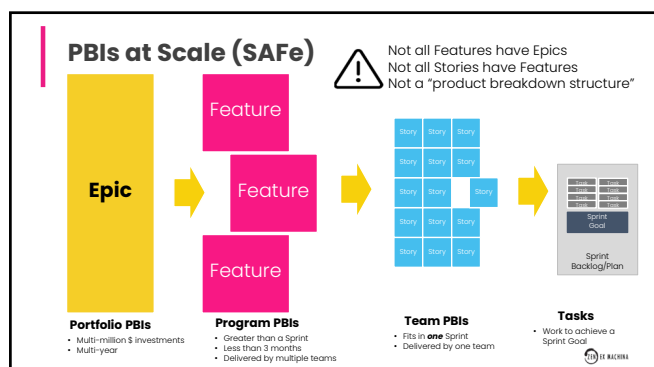
Additional things you could add:

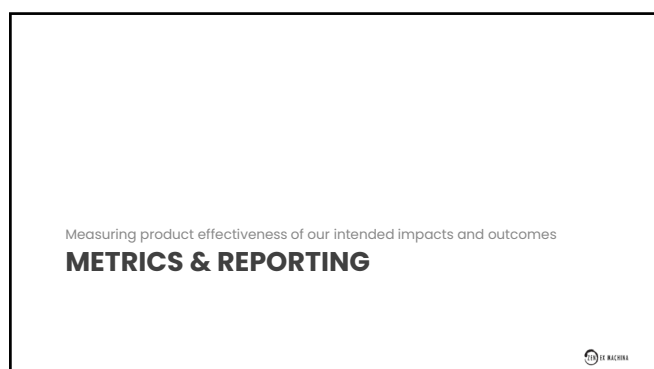
- Story size and estimates
- Business rules and process steps
- Acceptance criteria
- Intentional design or high level architecture advice
- Wireframes
- Team delivery notes

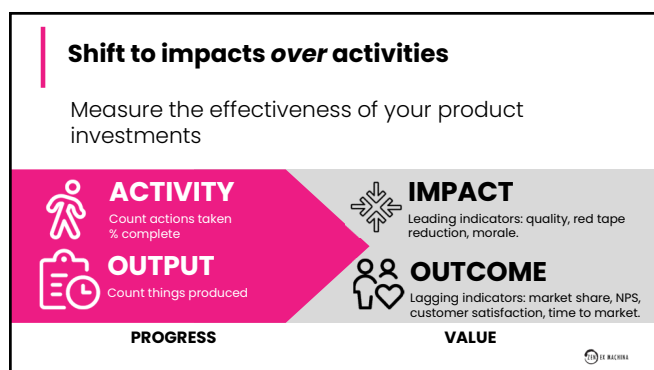
Handy hints



- Focus on defining the problem for the team to solve.
- Avoid creating and delegating pre-determined tasks.
- Empower the team to come up with their own solution and plan to solve the problem.
- Then, assess whether the plan and solution is worth investing in (the Product Owner's concern).
- Allow the plan to adapt as the team learn new things about the viability of the solution. Inspect and adapt!



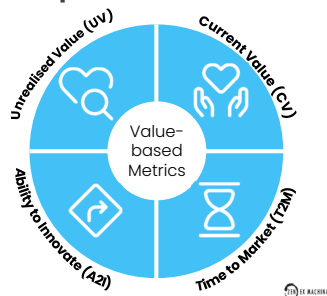




Agile OKRs measure impact and outcome

What area of your customer's needs are you investing in?

What metrics will change if that investment makes a difference?



Agile OKR areas



Unrealised value

- Market share
- Satisfaction gap



Current value

- Product cost ratio
- Customer sentiment
- Usage index



Ability to innovate

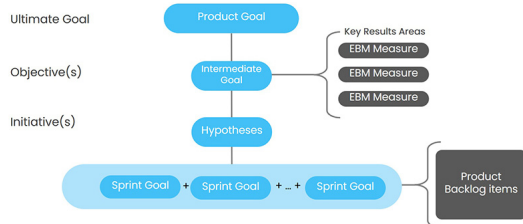
- Lead time
- Cycle time
- Time to learn
- Capability maturity



Time to market

- On product index
- Time spent context switching
- Innovation index

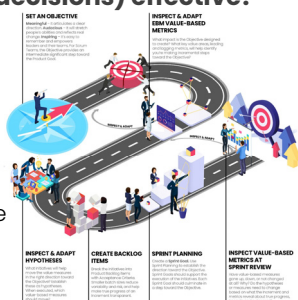
Agile OKRs apply to features, not teams or individuals



Is our product (and decisions) effective?

Use **Sprint Review** to inspect and adapt metrics:

- Are the metrics moving? No?
- Do you need a change of metrics, features, or both?



Definition of Done

BUILDING-IN QUALITY



Building-in Quality

- Teams use a **Definition of Done** to ensure the quality of their product or service.
- Sets expectations about what the team is required to do to make Increments of work **potentially releasable**.
- When a Story meets the Definition of Done it becomes an **Increment**.
- Increments are inspected at Sprint Review to elicit feedback on what to do next.



What's in the Definition of Done

Product quality

- What ever is mandated by the organisation to **make the increment potentially releasable**.
- Quality criteria.
- Compliance.
- Non-functional requirements.
- Organisation's standards and expectations.
- Industry regulations and standards requirements, including **security, performance, usability**.

Quality practices

- Practices to make features fully tested, peer reviewed, and "ready", for release or be handed over to the customer.
- What ever is considered best-practice by industry, the organisation, and capability managers.

THE BACHTEL

Definition of Done vs Acceptance Criteria

Product Backlog Item (PBI)

- Communications for corporate event
- **Tasks:**
 - Identify attendees (primary and secondary), identify RSVP timings, write copy.
- **Acceptance Criteria:**
 - Send out invitations

Applies to just this PBI

Definition of Done

- Plain-English (Australian).
- Peer reviewed.
- Social media calendar populated in Hubspot.
- Social media automated and scheduled.
- Uses corporate template and style guide.
- Originals stored in the Google Drive.

Applies to all PBIs

THE BACHTEL

Definition of Done vs Acceptance Criteria

Product Backlog Item (PBI)

- Town hall event to increase awareness of the strategy for research in the organisation 2020-2025
- **Tasks:**
 - Source the strategy, identify key messages, create PowerPoint, book the hall, book telepresence, invite delegates.
- **Acceptance Criteria:**
 - Handed over to the client by Friday 3 March.

Definition of Done

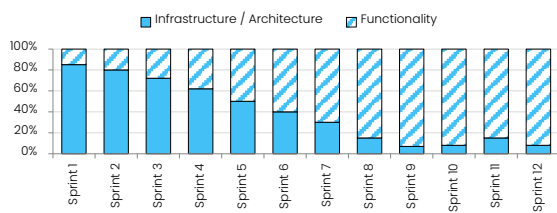
- Plain-English.
- Peer reviewed.
- Uses corporate template and style guide.
- Stored in the shared drive.
- Correspondence answered within 3 days.

THE BACHTEL

PRIORITISING WORK FOR THE TEAM



Prioritise so that every Sprint delivers an Increment of value



By definition, each Sprint delivers value.
There is no such thing as "Sprint Zero" (it's an anti-pattern)

Prioritise work by benefit and outcome

- Product work is best ordered for delivery by its **relative value**.
- At any point in time, one idea, feature, piece of work will deliver more value than others.
- It's the **Product Owner's job** to ensure the most valuable work is delivered.



vs



vs



Which one will deliver most value today?



Smarter ways to prioritise work

- **Priority:** either calculated or relative
- **Development cohesion:** both product and system
- **Business cohesion:** smaller area of business affected
- **Implementation cohesion:** steps in a work flow or customer experience
- **Intentions:** Release grouping
- **Cost (impact) of delay:** What can't afford to do later?

Cohesion
simplifies
solution
development &
implementation



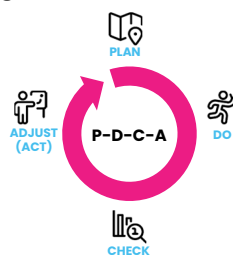
AGILE PRODUCT MANAGEMENT FRAMEWORKS



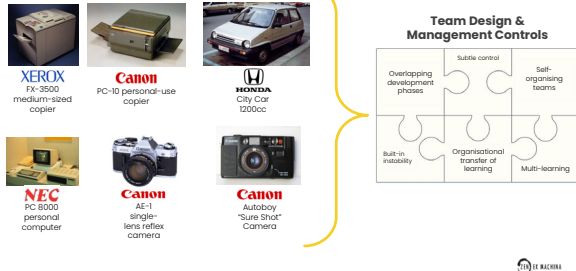
Deming Cycle c1950



W Edwards Deming



Lean & Kanban: Japanese product development (1986)



Agile Mindset: Agile Manifesto (2001)

Focus here

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

But not to the exclusion
of these things
(less focus)

Principles of the Agile Manifesto (2001)

- Our highest priority is to **satisfy the customer** through early and continuous delivery of valuable [outcomes].
- Welcome changing requirements**, even late in [the work process]. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently**, from a couple of months to a couple of weeks, with a preference to the shorter timescale.
- Business people and developers** must work together daily throughout the project.
- Build projects around motivated individuals. Give them the **environment and support** they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a [team] is **face-to-face** conversation.
- Working [product]** is the primary measure of progress.
- Agile processes promote **sustainable [work]**. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to **technical excellence** and good design enhances agility.
- Simplicity** – the art of maximizing the amount of work not done – is essential.
- The best [solutions] emerge from **self-organizing teams**.
- At **regular intervals**, the **team reflects** on how to become more effective, then tunes and adjusts its behaviour accordingly.

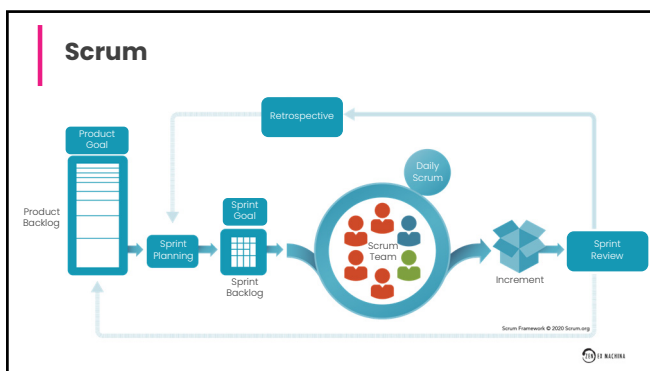
ACTIVITY

Scrum

Pin the items on the Scrum framework

5
MINUTE(s)
TIMEBOX

Scrum Framework © 2020 Scrum.org



Scrum Artefacts

Product Backlog

All the known work for the team to achieve the Product Goal

Sprint Backlog

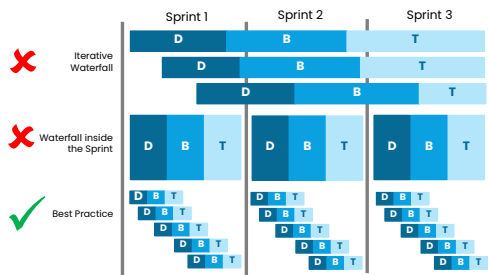
The plan for the Sprint to achieve the Sprint Goal

Increment

Potentially releasable. Work becomes an Increment when it meets the Definition of Done

Scrum Framework © 2020 Scrum.org

Creating an Increment



Scrum commitments



Definition of Done

- Quality, standards, compliance, and practices
- Maximum and minimum standard
- Helps stop rework
- Helps avoid gold plating



Sprint Goal

- Commitment for focus for the Sprint
- Contributes to the Product Goal



Product Goal

- Commitment for focus for team and its delivery of value to the organisation and its customers

Scrum roles



Product Owner

- Accountable for delivery (of value)
- Budget
- Product Backlog management
- Stakeholder engagement



Scrum Master

- Accountable for the effectiveness of Scrum to deliver the product's outcomes
- Coach the PO
- Coach Developers to be cross-functional and self-managing
- Coach stakeholders
- Cause impediments to be removed



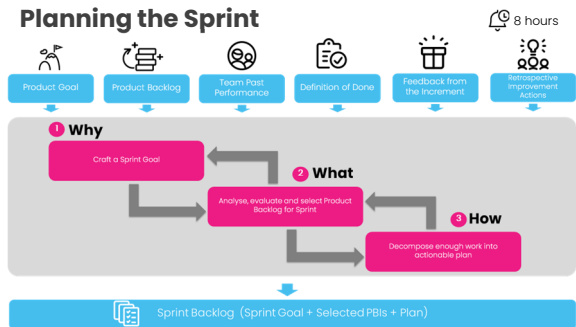
Developers

- People who do the work
- Expected to hold each other accountable to deliver the work as a team

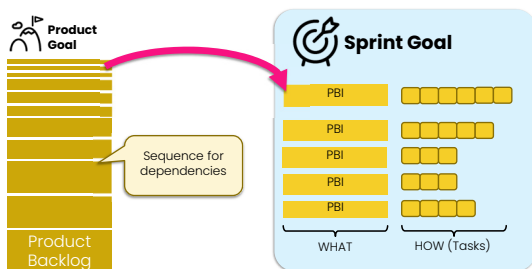
Scrum Events

Event	Purpose		Who Attends	Timebox (1 month)
	Inspect	Adapt		
Sprint Planning	Product Backlog, Product Goal, Definition of Done	Sprint Backlog, Sprint Goal	Scrum Team	8 hours
Daily Scrum	Progress toward Sprint Goal	Sprint Backlog	Developers	15 minutes (always)
Sprint Review	Increment, Sprint, Product Backlog, Progress toward Product Goal	Product Backlog	Scrum Team Stakeholders	4 hours
Sprint Retrospective	Sprint, Definition of Done	Actionable improvements, Definition of Done	Scrum Team	3 hours
Sprint	The container for all events – the heartbeat of Scrum – that turns ideas into value. Continuous refinement. Continuous discovery. Continuous research.			No more than one month

Planning the Sprint



Sprint Planning creates the Sprint Backlog



Sprint Goal examples

This is good:

Our focus is on <Achievement>
We believe it delivers <Benefit> to <Customer>
This will be confirmed when <Event happens>

This is better:

Our focus is on <Outcome>
We believe it delivers <Impact> to <Customer>
This will be confirmed when <Event happens>



Sprint Goal examples

Our focus is to have a tidy garage that we can put our car in.
We believe it decreases the risk of the car being stolen and increases our peace of mind.
This will be confirmed when the car is in the garage and out of sight.

Our focus is to lose a healthy amount of body weight.
We believe it will reduce the strain on my heart.
This will be confirmed when I can run consistently at 5 min/km for 30 minutes.

Our focus is on sending a basic email that contains a link to a spreadsheet.
We believe it delivers confidence in the product to our organisation.
This will be confirmed when we have an email in an inbox.

Our focus is to practice defending corners in Football.
We believe it will enable the team to not concede goals from corners.
This will be confirmed when we defend the goal from corners in a real football match.



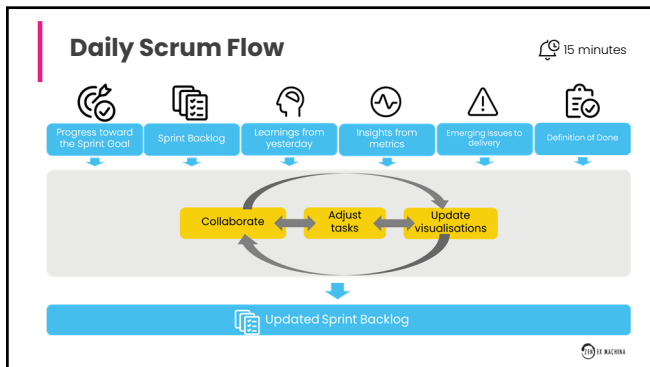
Sprint Goal examples

Poor examples:

Our focus is on **finishing all the items supporting the SAP integration.**
We believe it **delivers satisfaction and closure** for our **project manager.**
This will be confirmed when **Epic SAP-123 is closed in Jira.**

Our focus is on having **SAP integrated into the Corporate system**
We believe it delivers **improved features functionality and increasing productivity** to **Department X**
This will be confirmed when the colleagues in **Department X** are using the **improved features and can see the benefits.**



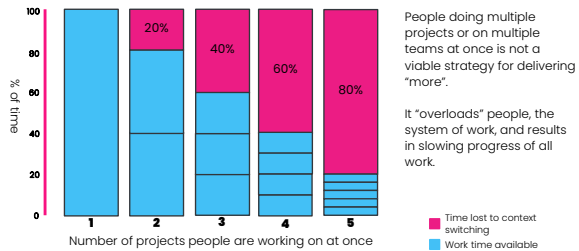


Questions to ask at Daily Scrum

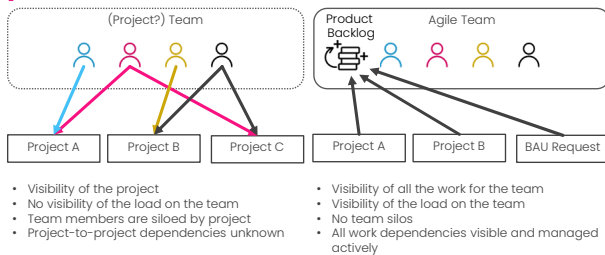
- What did you do yesterday, today, blockers? **NO!**
- How is our progress toward the Sprint Goal?
- Are we where we thought we would be?
- Did anyone learn anything that means we need to change our plan?
- Are we waiting on someone? How will we attack this problem? Does it need escalation?
- What do we need to do **today** to progress the most valuable work before taking on more work?

THE LAW OF THE SMALL (AGILE) TEAM

Do more with the same people!?



Team is the focus, not projects



Roles in agile (Scrum) teams



 ZURICH MACHINE

Self organising teams are more effective

Table 2.

Performance	Traditionally managed teams		Self-managed teams		Self-leading teams	
	Mean	SD	Mean	SD	Mean	SD
Productivity	5	.816	6.143*	1.069	6.5*	0.707
Creativity	4.24	.857	5.714	.756	5	1.414
Goal attainment	5	1.826	6****	.577	5	0
Quality	5.475	.957	6.429*	.535	6.5	0.707
Efficiency	4.708	1.294	5.429	.787	5.5	2.121
Output	4	.816	5.143	1.215	4	0
Overall	4.5	1.063	5.810*	.788	5.417	1.240

* Significantly higher than traditionally managed teams at $p < 0.05$ ** Significantly higher than self-managed teams at $p < 0.05$ *** Significantly higher than traditionally managed teams and self-managed teams at $p < 0.05$ **** Significantly higher than self-leading teams at $p < 0.05$

Source: The effectiveness of self-managed teams and self-leading teams measured in performance, quality of work life and satisfaction
 © Laporte, Kater van de Gilt (2005) Online at: <https://www.sciencedirect.com/science/article/pii/S0950080405000000>
 URL: <https://doi.org/10.1016/j.sbspro.2005.03.001>



Team Design

Cross-Functional ("Feature" team)

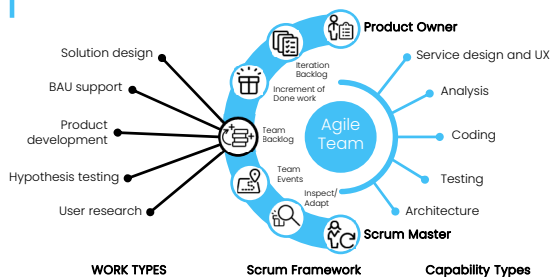
- All the skills it needs to solve problems without having to rely on anyone outside the team
- Work collaboratively
- Solves the problem collectively as a team

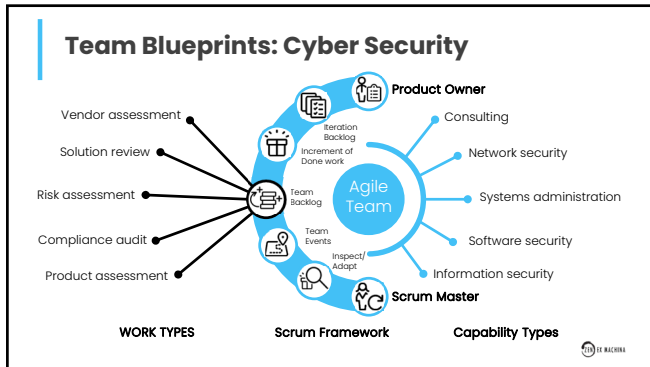
Multi-Disciplinary (MDT)

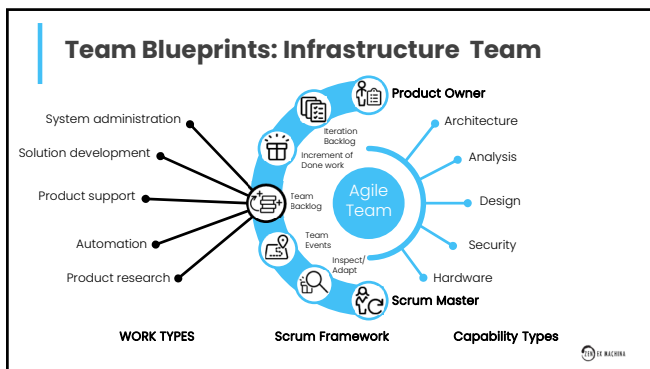
- Attacks a problem from different "disciplines", e.g. gross-motor, psychology, speed pathology, etc.
- Work in silos (disciplines) but coordinate work and communicate openly.
- Problem is investigated and then solved separately by each discipline.

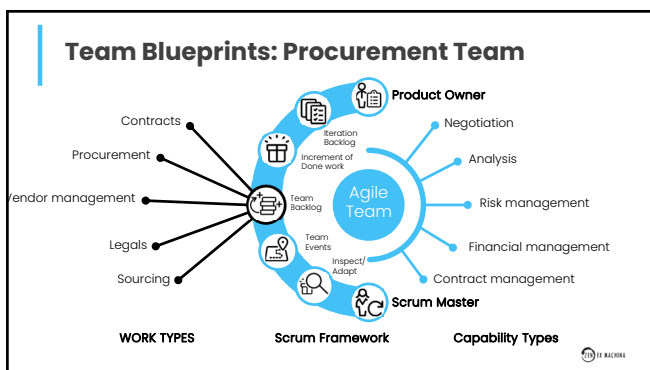


Team Blueprints: IT Team

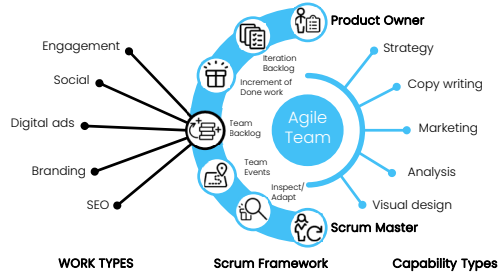




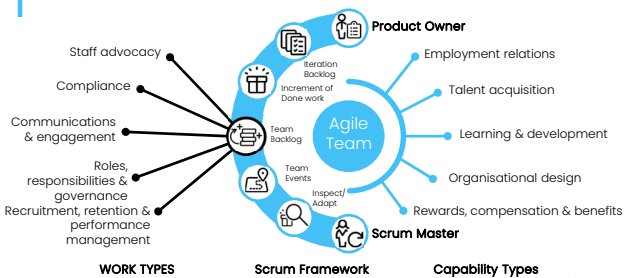




Team Blueprints: Marketing Team



Team Blueprints: HR Team



Roles within the Team?

- No sub-teams:
 - No solution > draft > approve.
 - No UX Team > BA Team > Dev Team > Test Team.
 - Increases handovers and dependencies, increasing the time it takes to create an Increment.
- No sub-roles:
 - Sub-roles tends to reduce "equal voice".
 - We want everyone to work as a single team.
 - Creates functional hierarchy within the team, which leads to sub-teams.
 - "I'm a Dev Lead, so I'll make the decision and delegate that to the junior person".

Roles within the Team?



Ian Mitchell
Professional Scrum Trainer (PST), Scrum.org

In Scrum, no one Team member would be responsible for "leading" the entire team's work, even in one area. The team is self-organising without sub-roles such as team lead, technical lead, or UI expert. Responsibility is not inferred by any specialisms a team member may have ... the Scrum Guide makes it clear that there are no exceptions to this rule.



Optimising flow with Kanban

TRACKING SPRINT DELIVERY



What is Kanban?

Tool for Lean and Scrum
(complementary practice).

Not just a "visual" board.

No such thing as a "Kanban Team".

Encompasses:

- 4 metrics
- 4 practices



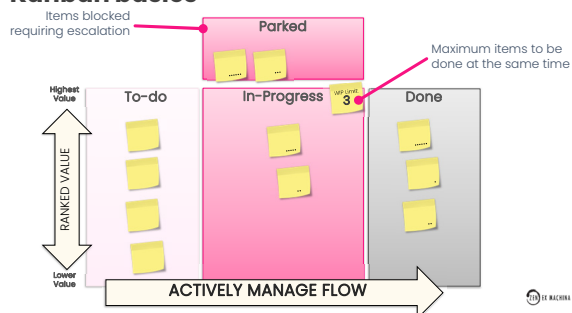
Kanban adds on to Scrum

Helps teams understand:

- Bottlenecks.
- How long it truly takes to deliver work to Done.
- Work item age.
- How to progress work without it coming back for rework, hence improve their Definition of Done.

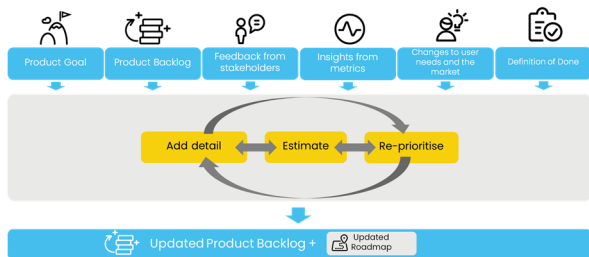
TE BACHT

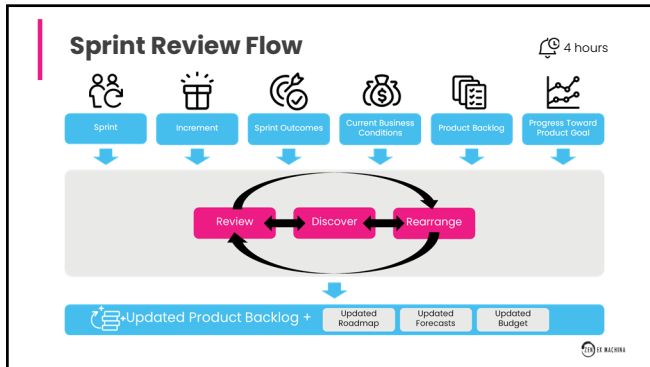
Kanban basics

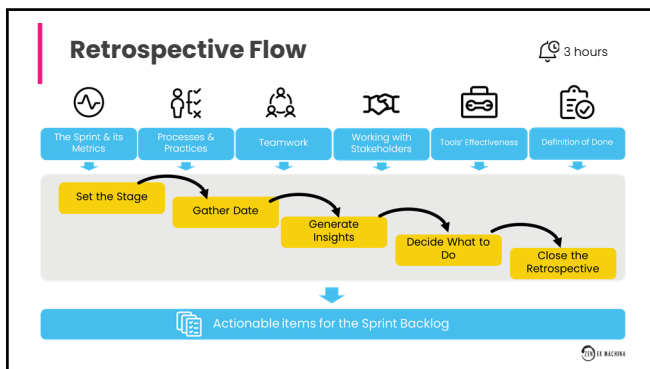


Backlog Refinement Flow

~10% of the Sprint

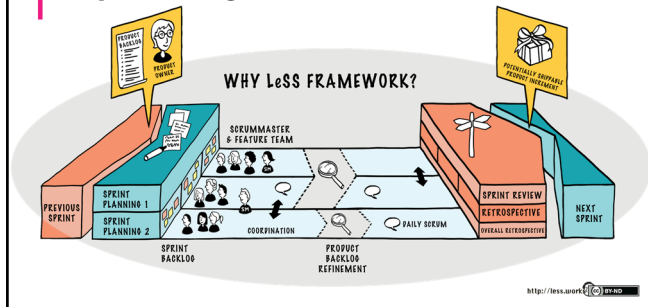




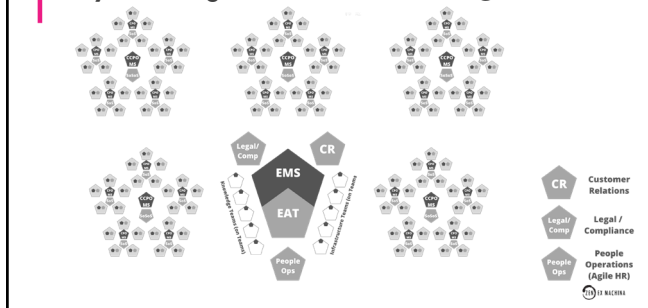




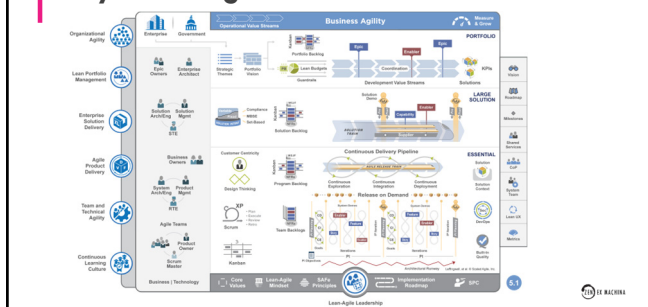
Key scaled agile frameworks: LeSS

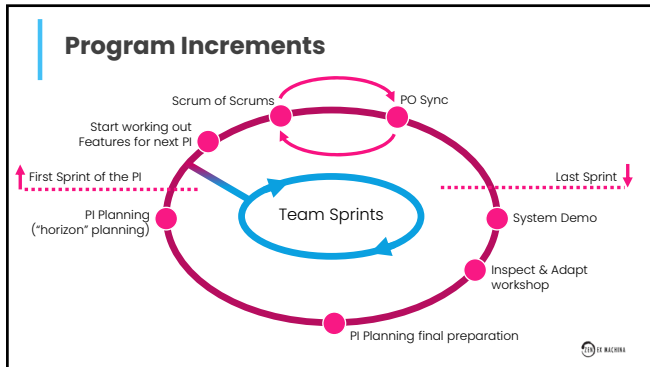


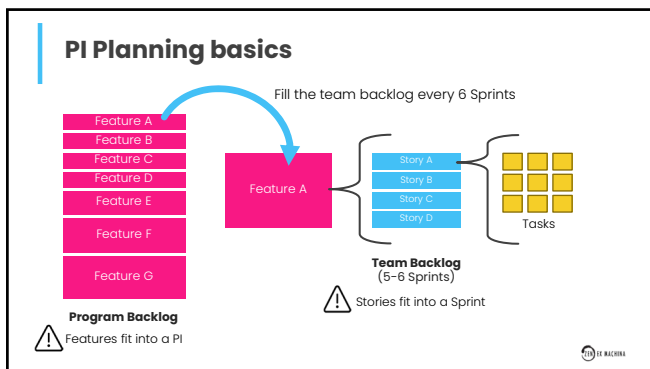
Key scaled agile frameworks: Scrum @ Scale

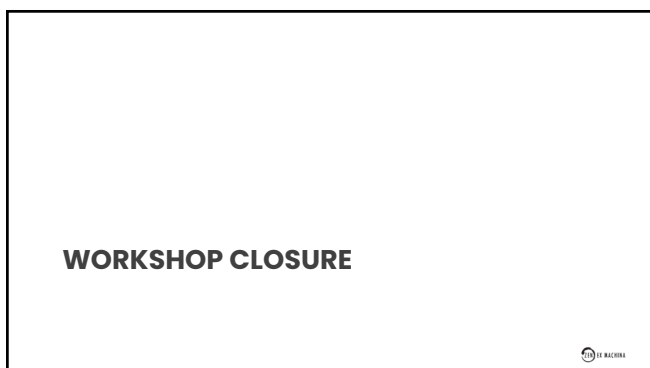


Key scaled agile frameworks: SAFe®









Old skills to new skills

IT'S HARDER THAN YOU THINK IT IS



<https://www.youtube.com/watch?v=MFzDaBzBIL0>



The risks to manage

Over 20 years, agile initiatives continue to fail because of insufficient **leadership participation** and **lack of experience**.

Failure typically boils down to a lack of approach to human change with agile just becoming something "teams" do.



Source: Digital.ai. (2022) State of Agile Survey.

Thank you

Contact us

For more information, please visit
zenexmachina.com

Matthew Hodgson (PST)
CEO and Enterprise Agile Coach

e matthew.hodgson@zenexmachina.com
m +61 404 00 66 95
w zenexmachina.com
f facebook.com/zenexmachina