How Agile is Your Organization? The Agile Maturity Model

D. Vanderbist 13/06/2018

(cf. model by Mark J. Balbes)

Agile Principles

Deliver Value

Focus on continuously delivery value to the customer, key-users …

Embrace Change

Change is good and inevitable. Change avoids waste by adapting before it is too late.

Business + ICT

Avoid Chinese walls between teams. Work closely together every day.

Simplicity

Focusing on what is good enough to avoid gold-plating. Work smarter not harder.

Frequent Delivery

Delivery version frequently to have short feedback cycles.

Self-Organizing Teams

Motivated individuals will be able to identify how to organize the work and what they need.

Communication

Transparent, open and face-to-face communication helps insights and clear understanding.

Self-Emerging

Avoid analysis-paralysis and big design up-front. Design for what is needed now and adapt.

Progress Monitoring

Measure progress as delivered software i.e. potential shippable product increments.

Constant Pace

The team should work according at a pace they can keep up without feeling pressured.

Technical Excellence

Focus on the quality of artefacts and development process.

Continuous Improvement

Be self-reflective and make incremental improvements to the development process.

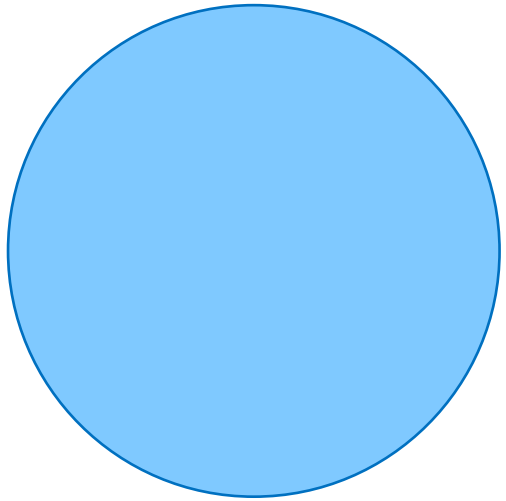
VDB

Maturity Levels

Maturity Measured on 6 Axes

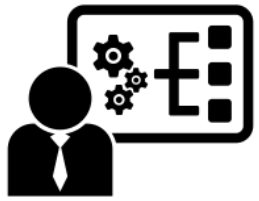
User Experience

Technical Environment



Quality Management

Team Dynamics



Project Management

Ownership Management

\+

Eco-System

Technical Environment

Automated Unit Tests	Unit Test part of automated build?
Technical Testing	Performance, Scale, Stress, Load testing … ?
Test Driven Development	Write test before code is written?
Continuous Integration	Check-Ins trigger builds?
Pair Programming	Programmers work together all the time?
Spiking	R&D is timeboxed to a separate spike?
Source Control / Branching	Branching strategy is linked with Continuous Integration?
Release Management	Release and promotions are automated?
Coding Standards	Coding are automatically measured and enforced?
Development Process	The process is known and automated?
Shared Code Ownership	People are empowered to change anything?
Software Changeability	Software is not brittle to change?

Principles of Extreme Programming

Quality Management

Code Quality
Process Definition
Internal Quality
External Quality
Team Ownership of Quality
Defect Management
User Acceptance Testing
Exploratory Testing

Static and dynamic code assessment tools are in place?
Quality assurance is part of the development process?
Quality is tracked and measured against KPI's?
Bugs result is tests?
Quality is part of the Definition of Done?
Defects are followed up by root-cause-analysis?
Automated UI testing is in place?
Testing is following fixed as well as add-hoc test scripts?

Quality Assurance in favor of Quality Measurement

User Experience

Graphic Elements
UX
Usability Testing

Designers select components to user?
UX experience is at the bases of the user stories?
Usability feedback is part of the acceptance testing?

Developers are not Designers. Multi-skilled Teams.

Team Dynamics

Team Structure
Retrospectives
Stand-Ups
Team Discussions
Information Radiators
Continuous Improvement
Charter
Self-Empowered

Team Area
Conflict Resolution
Accepting Changes

Team has accepted the agile way of working?
Process feed-back is gathered, implemented and measured?
Stand-Up are efficient focusing on tasks, status and impediments?
Discussions are always timeboxed?
Charts and KPI's on progress are visible in the team's space?
Team focusses every sprint to improve on the previous one?
The team created a Definition of Done and code of conduct?
Team quickly aligns on goals and the approach to complete assigned tasks?
Team is co-located physically or technically?
Team have a prescribed process to solve differences in opinions?
The team has a high-tolerance to change?

Self-Reflective Mature Teams

Ownership Management

Identified Stakeholders
User Stories
Acceptance Criteria
Story Sizing

Delivering Value
Demos

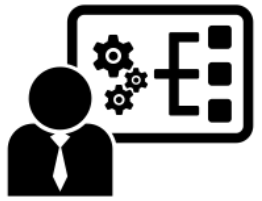
User Feedback
Prioritization

Release Cadence
Backlog
Product Owner Accepts Changes

There is a RACI explaining everyone's role?
User stories are applying INVEST criteria?
Acceptance criteria are SMART?
The task breakdown is granular fine enough for transparent progress monitoring?
Effort is spend on activities that add value to the business?
New features are demoed regular and early to maximize feedback?
Feedback is hunted and gathered?
Frequently the team evaluates the open tasks and these tasks still deserve their attention?
Release happen every x weeks without any exception?
The backlog is groomed?
The product owner allows the business to change and focusses on the impact of the changes?

Product Owner bridges between IT and Business to maximize the business values of a project.

Project Management



Kanban
Reporting
Meeting Minutes

Milestone Review
Staffing

A Kanban board is used to track task's states and progress?
Regular reporting happens to the stakeholders?
All decisions are captured in meeting minutes and shared between the participant of the meting?
Milestones are reviewed at the beginning of every sprint?
The team is a multi-disciplined team or comprises of multi-disciplined team members?

Scrum Master enforces principles to enhance collaboration and transparency.

Eco-System MISC

Risk Identification	Impediments are monitored for potential risks?
Risk Monitoring	Risks are checked at the beginning of each sprint?
Risk Mitigation	Predefined set of strategies to mitigate risks is available?
Learning Culture	The organization is open for changes communicated bottom-up?
Champions	The organization supports knowledge transfer form champions across different teams?
Governance	IT and Business understand the Agile concepts: the flexibility and the limitations of the approach?
Internal Change Management	Internal leaderships enforces and guides teams to the Agile way of working?
External Change Management	IT is a mentor and evangelist of Agile to be applied in the full organization, across all departments?

The Organizational Eco-System