

Team Manifesto

Introduction

ISE teams work with a new development team in each customer engagement which requires a phase of introduction & knowledge transfer before starting an engagement.

Completion of this phase of ice-breakers and discussions about the standards takes time, but is required to start increasing the learning curve of the new team.

A team manifesto is a light-weight one page agile document among team members which summarizes the basic principles and values of the team and aiming to provide a consensus about technical expectations from each team member in order to deliver high quality output at the end of each engagement.

It aims to reduce the time on setting the right expectations without arranging longer "team document reading" meetings and provide a consensus among team members to answer the question - "How does the new team develop the software?" - by covering all engineering fundamentals and excellence topics such as release process, clean coding, testing.

Another main goal of writing the manifesto is to start a conversation during the "manifesto building session" to detect any differences of opinion around how the team should work.

It also serves in the same way when a new team member joins to the team. New joiners can quickly get up to speed on the agreed standards.

How to Build a Team Manifesto

It can be said that the best time to start building it is at the very early phase of the engagement when teams meet with each other for swarming or during the preparation phase.

It is recommended to keep team manifesto as simple as possible, so preferably, one-page simple document which **doesn't include any references or links** is a nice format for it. If there is a need for providing knowledge on certain topics, the way to do is delivering brown-bag sessions, technical katas, team practices, documentations and others later on.

A few important points about the team manifesto

- The team manifesto is built by the development team itself
- It should cover all required technical engineering points for the excellence as well as behavioral agility mindset items that the team finds relevant
- It aims to give a common understanding about the desired expertise, practices and/or mindset within the team
- Based on the needs of the team and retrospective results, it can be modified during the engagement.

In ISE, we aim for quality over quantity, and well-crafted software as well as to a comfortable/transparent environment where each team member can reach their highest potential.

The difference between the team manifesto and other team documents is that it is used to give a short summary of expectations around the technical way of working and supported mindset in the team, before code-with sprints starts.

Below, you can find some including, but not limited, topics many teams touch during engagements,

Topic	What is it about ?
Collective Ownership	Does team own the code rather than individuals? What is the expectation?
Respect	Any preferred statement about it's a "must-have" team value

Topic	What is it about ?
Collaboration	Any preferred statement about how does team want to collaborate ?
Transparency	A simple statement about it's a "must-have" team value and if preferred, how does this being provided by the team ? meetings, retrospective, feedback mechanisms etc.
Craftspersonship	Which tools such as Git, VS Code LiveShare, etc. are being used ? What is the definition of expected best usage of them?
PR sizing	What does team prefer in PRs ?
Branching	Team's branching strategy and standards
Commit standards	Preferred format in commit messages, rules and more
Clean Code	Does team follow clean code principles ?
Pair/Mob Programming	Will team apply pair/mob programming ? If yes, what programming styles are suitable for the team ?
Release Process	Principles around release process such as quality gates, reviewing process ...etc.
Code Review	Any rule for code reviewing such as min number of reviewers, team rules ...etc.
Action Readiness	How the backlog will be refined? How do we ensure clear Definition of Done and Acceptance Criteria ?

Topic	What is it about ?
TDD	Will the team follow TDD ?
Test Coverage	Is there any expected number, percentage or measurement ?
Dimensions in Testing	Required tests for high quality software, eg : unit, integration, functional, performance, regression, acceptance
Build process	build for all? or not; The clear statement of where code and under what conditions code should work ? eg : OS, DevOps, tool dependency
Bug fix	The rules of bug fixing in the team ? eg: contact people, attaching PR to the issue etc.
Technical debt	How does team manage/follow it?
Refactoring	How does team manage/follow it?
Agile Documentation	Does team want to use diagrams and tables more rather than detailed KB articles ?
Efficient Documentation	When is it necessary ? Is it a prerequisite to complete tasks/PRs etc.?
Definition of Fun	How will we have fun for relaxing/enjoying the team spirit during the engagement?

Tools

Generally team sessions are enough for building a manifesto and having a consensus around it, and if there is a need for improving it in a structured way, there are many blogs and tools online, any retrospective tool can be used.

Resources

[Technical Agility*](#)

Last update: August 22, 2024