



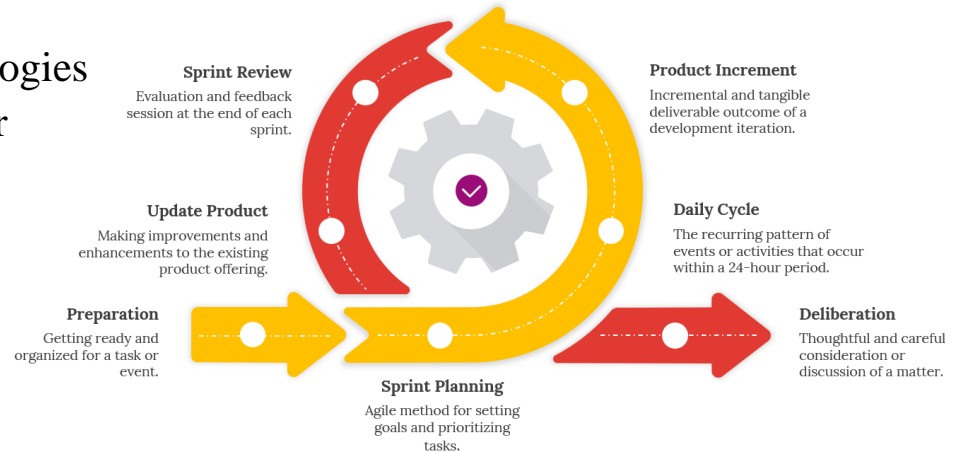
Agile Software Development

Best Practices and Implementation

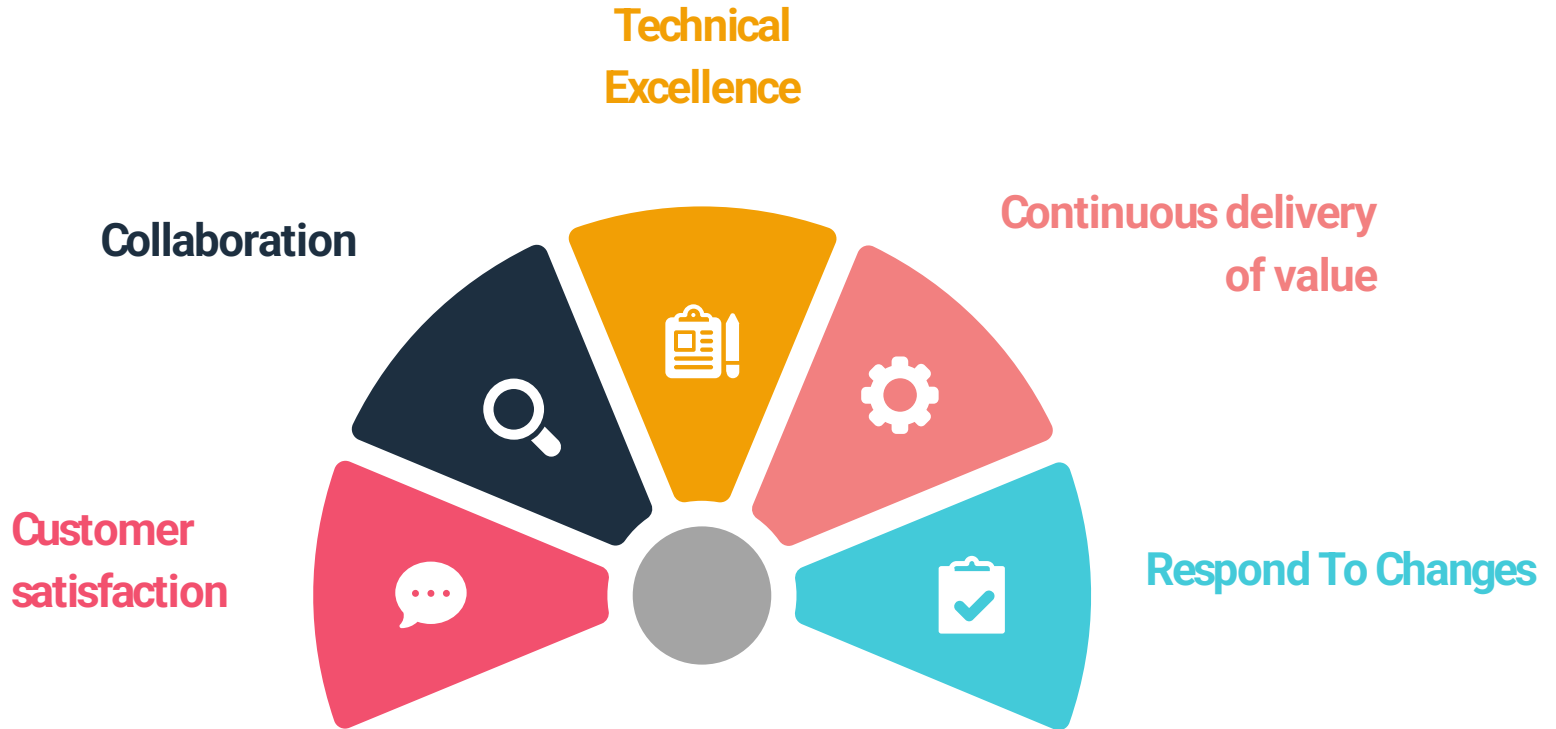
Introduction

- Agile is a software development methodology that emphasizes flexibility, collaboration, and iterative progress.
- Origin:

Response to rigid waterfall methodologies
Focuses on adaptability and customer satisfaction.



Agile Manifesto



Advantages

01

Iterative Progress

Frequent
Delivery of
Product

02

Collaboration

Face to face
communication
with stakeholder

03

Flexibility

Change
according to
customer need

04

Time

Less
documentation

Disadvantages

01

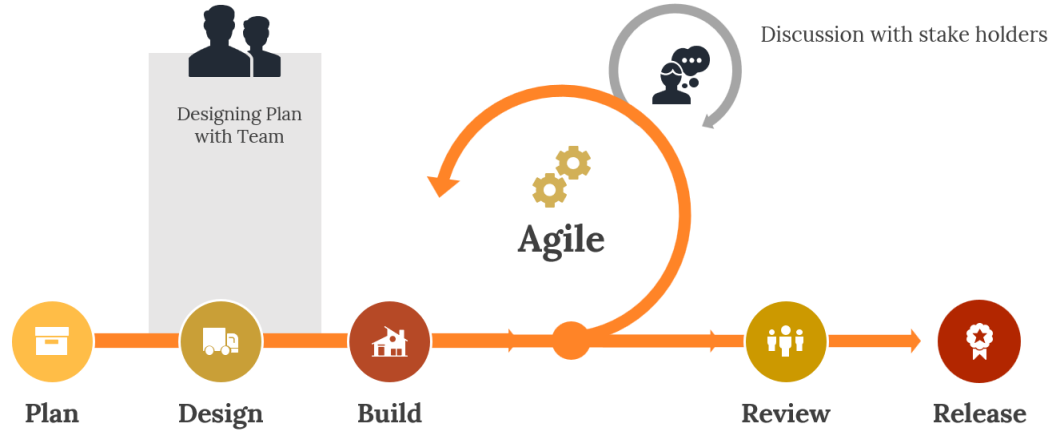
Documentation

Due to Frequent releases

02

Maintainance

Due to lack of documentation



Agile Principles



Principle 1

Satisfy Customers Through Early and Continuous Delivery

Principle 2

Welcome Changing Requirements Even Late in the Project

Principle 3

Deliver Value Frequently

Principle 4

Break the Silos of Your Project

Agile Principles



Principle 5

Build Projects Around Motivated Individuals

Principle 6

The Most Effective Way of Communication is Face-to-face

Principle 7

Working Software is the Primary Measure of Progress

Principle 8

Maintain a Sustainable Working Pace

Agile Principles



Principle 9

Continuous Excellence Enhances Agility

Principle 10

Simplicity is Essential

Principle 11

Self-organizing Teams Generate Most Value

Principle 12

Regularly Reflect and Adjust Your Way of Work to Boost Effectiveness

Agile Frameworks



Kanban

Focuses on visualizing workflow and limiting work in progress (WIP)

Framework with defined roles, ceremonies and artifacts

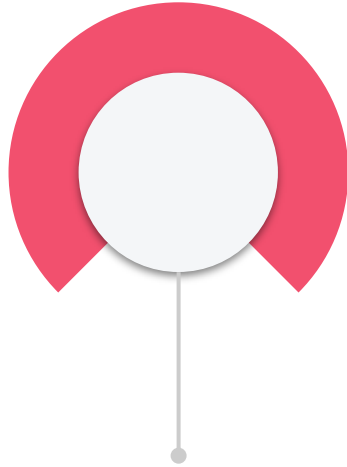
Scrum



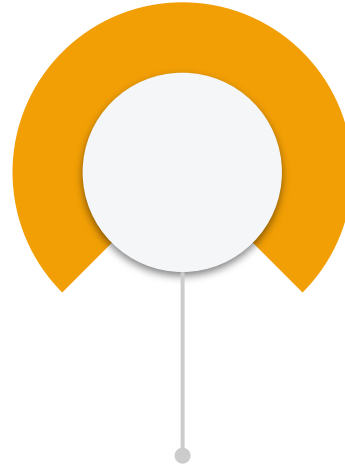
Agile Practices



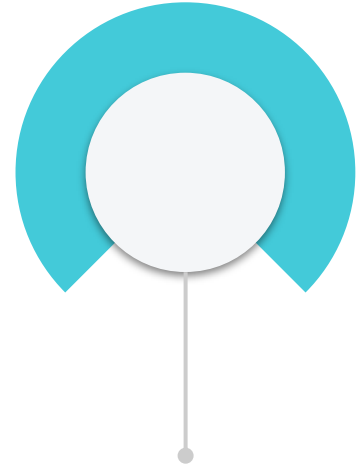
**Sprint
Planning**



**Daily stand-
ups**

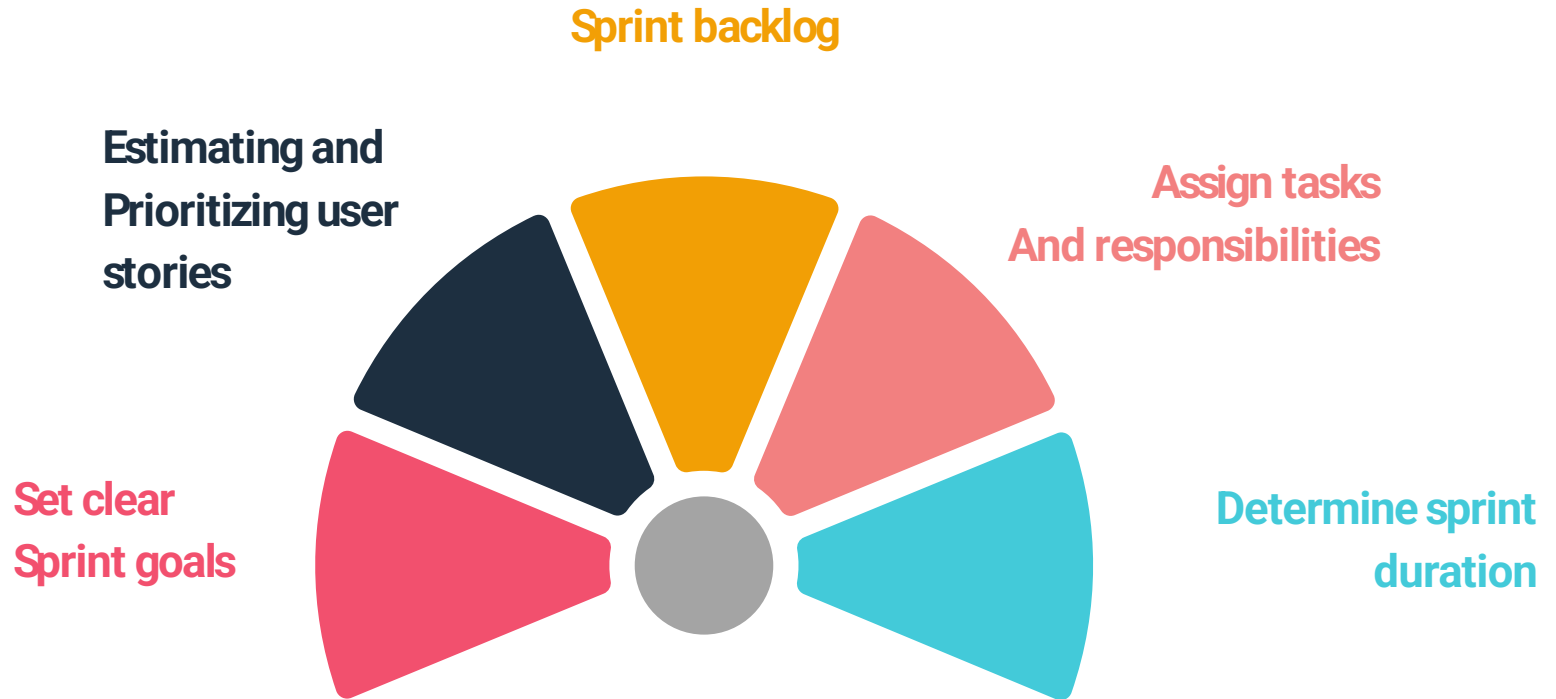


**Backlog
grooming**



Retrospectives

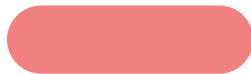
S print Planning



Daily Stand-ups



**Keep it short
15 min or less**

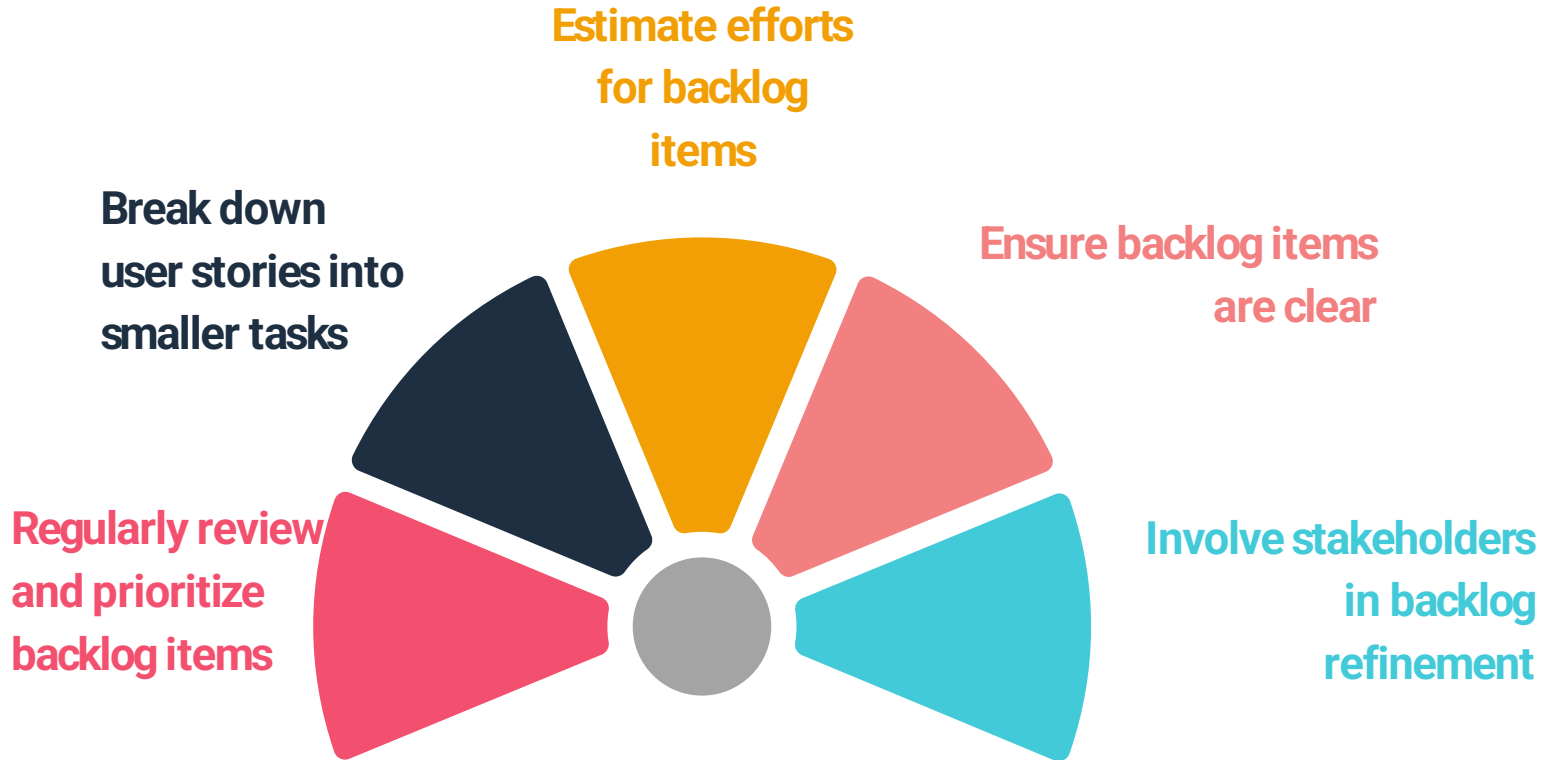


**What did you do yesterday?
What will you do today?
Are there any
impediments?**



**Encourage collaboration
and problem-solving**

Backlog Grooming



R etrospectives



**Schedule regular
retrospective meetings**



**Create a safe and open
environment for team
members to share feedback**



**Focus on continuous
improvement**



**Identify what went well,
what didn't, and actionable
items for improvement**

R e a l W o r l d E x a m p l e s

PlayStation Network

Collaboration

Trouble collaborating between hundreds of members

Dependency

Highly interdependent teams located in different cities

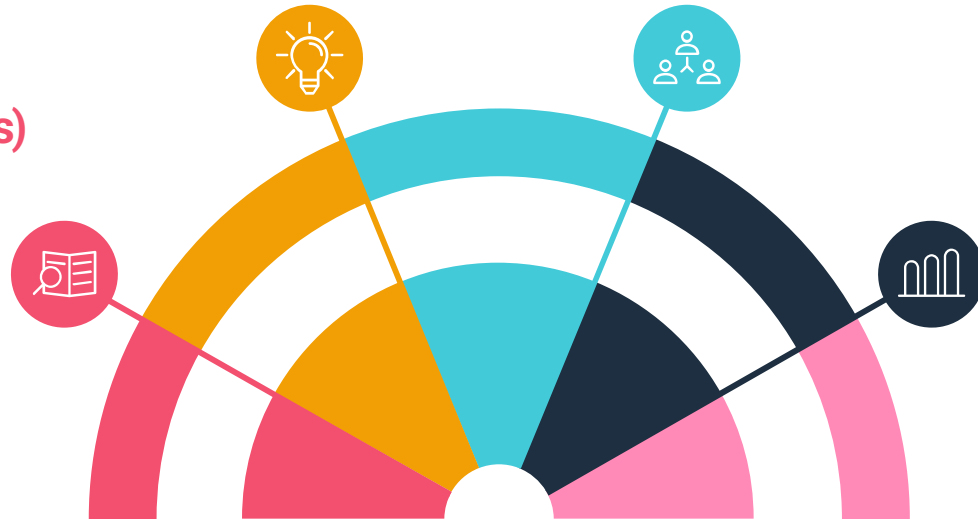
Problems

Planning

Teams could only plan one or two iterations in advance

Initial framework(s)

Initially used Waterfall and Scrum



S olution?



Solution

Switch to SAFe (Scaled Agile Framework)

Value

Managed to deliver double the value compared to before

Planning

Initial planning time cut by 28%

Collaboration

700 members in 60 teams using SAFe; improving collaboration

LEGO Digital Solutions



Expansion

Initially only 5 teams, which expanded to over 20

Collaboration

Trouble collaborating with client

Alignment

Difficult to align teams when they are all disconnected

Release

Release planning more complicated than needed

S olution



Framework

Opted for the
SAFe framework



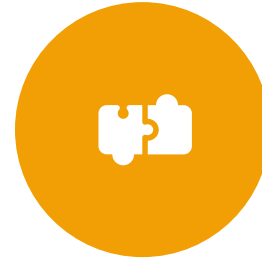
Dependency

Reduction in
dependency
issues



Planning

Improved
planning and
execution



Execution

Gradually trained
the team
managers



Improvement

Innovation is
improved as a result
of issues being
solved

**S u m m a r
y**

Agile

Introduction

Flexibility
Colaboration
Iterative Progress

Advantage

+Time



Frameworks

Kanban
Scrum
Safe(Scaled)

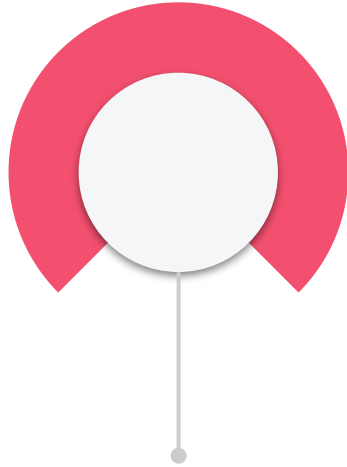
Disadvantage

Documentation
Maintaince

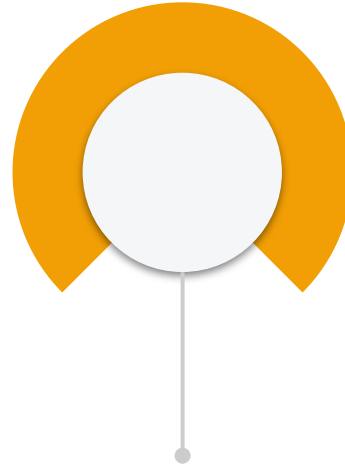
Agile Practices



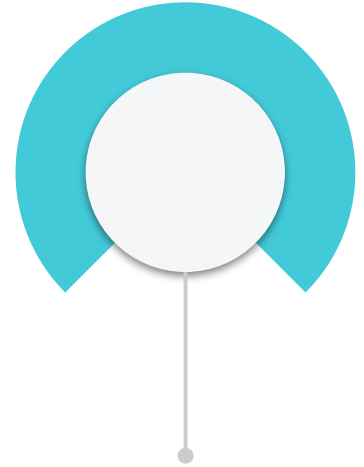
**Sprint
Planning**



**Daily stand-
ups**

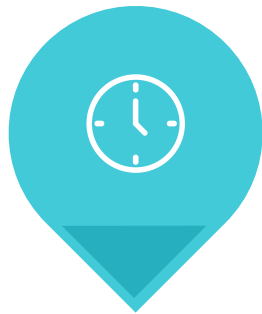


**Backlog
grooming**

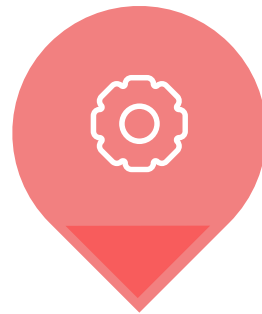


Retrospectives

Real Life Examples



Playstaion



Lego