

Agile

1

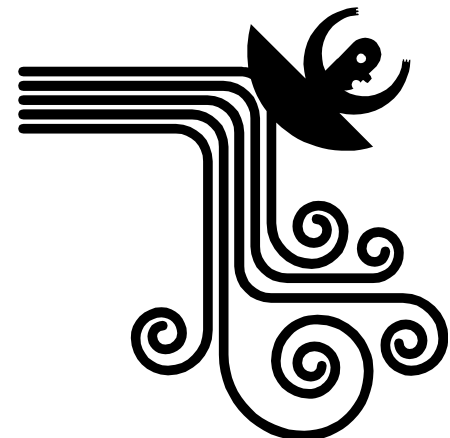
Mohammad Reza Karimi Dastjerdi
Arash Kayhani

TABLE OF CONTENTS

- Earlier Methodologies
- What Happened?
- The Agile Manifesto
- Agile principles
- Why use agile development methodology?
- The Key Components of Agile
- Comparison
- Agile development process
- Human Factors
- The Big Paradigm Shift
- When is Agile best?
- Criticism
- References

EARLIER METHODOLOGIES

- Requirements & Tasks are expected to be well defined at outset.
- Methodology is highly document driven.
- Project roles are highly structured and well defined.
- Communication is through PM and Sponsor.
- Typically long cycle



WHAT HAPPENED?



So *lightweight* agile software development methods evolved in the mid-1990s as a reaction against the *heavyweight* waterfall-oriented methods

THE AGILE MANIFESTO (2001) - A STATEMENT OF VALUES

AGILE PRACTICE FAVORS:

Individuals and
interactions

over

Process and tools

Working Product

over

Comprehensive
documentation

Customer
collaboration

over

Contract negotiation

Responding to
change

over

Following a plan

AGILE PRINCIPLES

- 1) Customer Satisfaction
- 2) Welcome changing
- 3) Frequently delivery
- 4) Measure of progress
- 5) Sustainable development
- 6) Cooperation between business people and developers
- 7) Face-to-Face Communication
- 8) Motivation
- 9) Technical excellence
- 10) Simplicity
- 11) Self-organization
- 12) Adjust behavior

WHY USE AGILE DEVELOPMENT METHODOLOGY?

- Produce visible results quickly
- Create a proof-of-concept and early rapid prototype
- Get feedback from the target audience in order to guide future development
- Test different types of functionality and user experiences for best results
- Limit the risk and investment
- Start using the product as soon as possible
- Quickly iterate and evolve the product based on demonstrated user needs

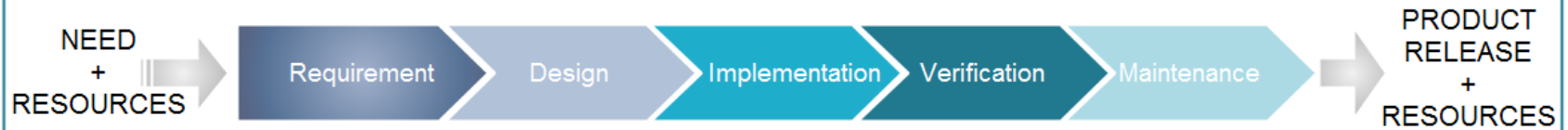
THE KEY COMPONENTS OF AGILE



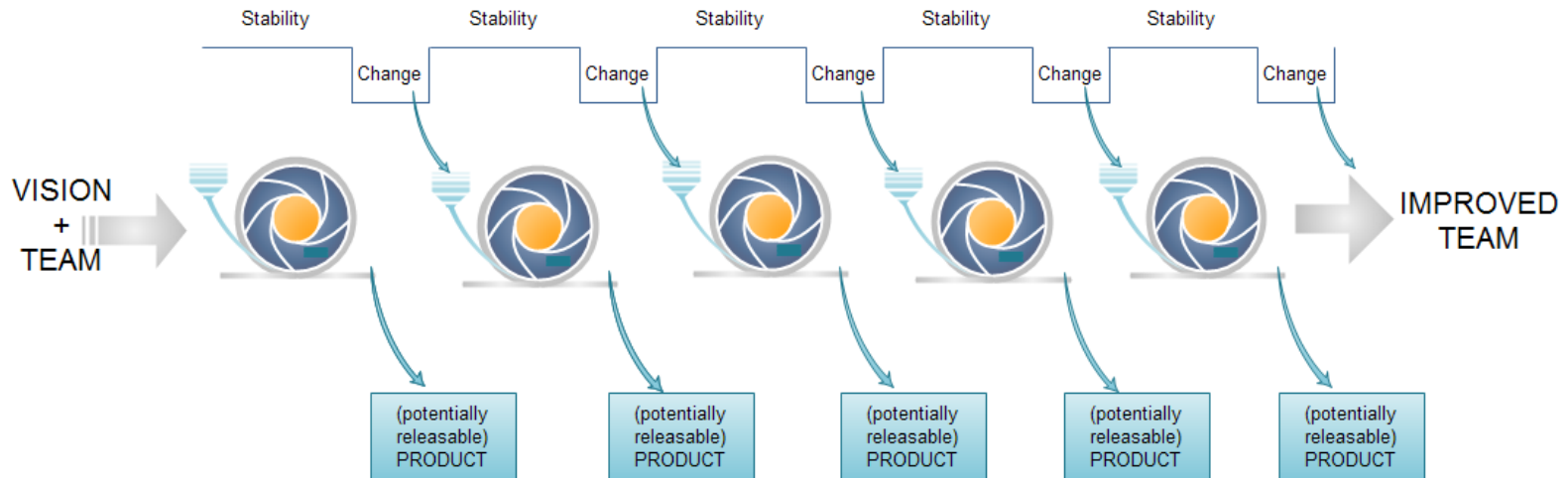
- *User Stories*
- *Product Backlog*
- *Sprint*
- *Sprint Planning Session*
- *Sprint Review Session*

COMPARISON

PLAN-DRIVEN, WATERFALL METHOD

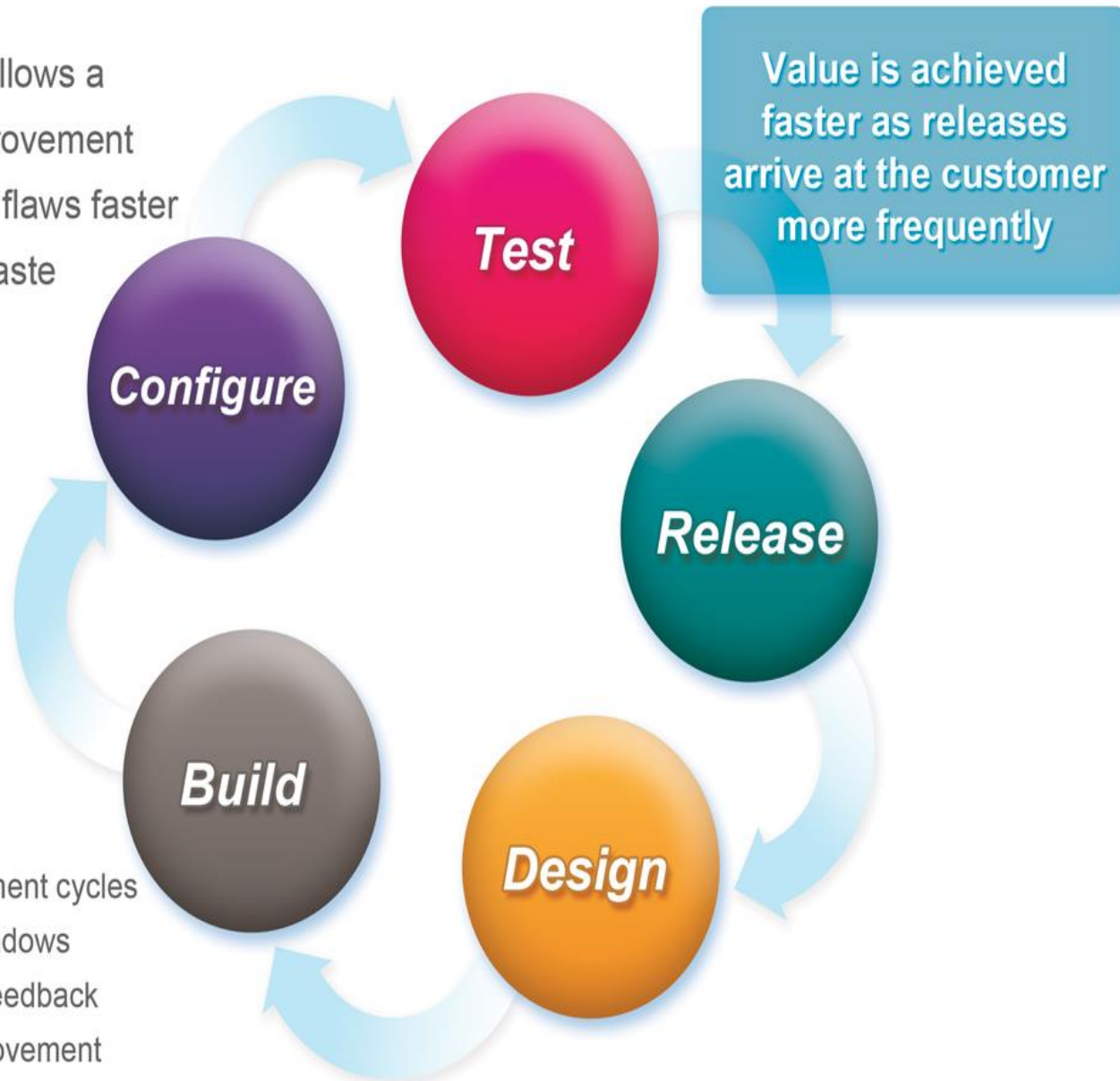


AGILE METHOD



Agile Development Process

Development follows a continuous improvement cycle, exposing flaws faster and reducing waste



Advantage:

- Shorter development cycles
- Wider market windows
- Early customer feedback
- Continuous improvement

AGILE DEVELOPMENT PROCESS

- Iterative, incremental and evolutionary
- Efficient and face-to-face communication
- Very short feedback loop and adaptation cycle
- Quality focus
- ...



HUMAN FACTORS

- Competence
- Common focus
- Collaboration
- Decision-making ability
- Fuzzy problem-solving ability
- Mutual trust and respect
- Self-organization

THE BIG PARADIGM SHIFT – SOME REALITY

We're used to	Agile Wants	What Works
Time Lines	We're done when it's done.	Time Boxes – not more than x time
Project Managers	Disciplined self managing teams.	Collaboration between Coach and Product Owner
Fixed Budgets	Assumed change means no fixed cost.	Cost Boxes – not more than x to spend.
Predictable, all at once deliverables	Incremental deliverables driven by value and constant learning.	Pre-project user story sessions (Release Planning)
Team spread out	Co-location.	Core time in room or on phone/virtual presence
Communication by Document	Information Radiators & Conversations	Information Radiators captured electronically and posted. Daily Meetings
Customer is removed	Customer is in room as part of team.	Core time in room

WHEN IS AGILE BEST?

- Creative Projects
- New Technology Introductions
- New Process Designs
- Projects driven by critical business timing.
- Projects with poorly defined needs



CRITICISM

“The agile movement is in some ways a bit like a teenager: very self-conscious, checking constantly its appearance in a mirror, accepting few criticisms, only interested in being with its peers, rejecting en bloc all wisdom from the past, just because it is from the past, adopting fads and new jargon, at times cocky and arrogant. But I have no doubts that it will mature further, become more open to the outside world, more reflective, and also therefore more effective.”

Philippe Kruchten

REFERENCES

- Software Engineering_ A Practitioner's Approach - Roger Pressman 7th 2010
- Agile in a Flash: Speed-Learning Agile Software Development by Jeff Langr and Tim Ottinger
- Software Engineering For Students: A Programming Approach by Douglas Bell
- www.agileinnovation.eu
- <http://www.inqbation.com/agile-methodology-of-web-development/>
- http://en.wikipedia.org/wiki/Agile_software_development
- <http://agilemanifesto.org/>

