SOFTWARE ENGINEERING

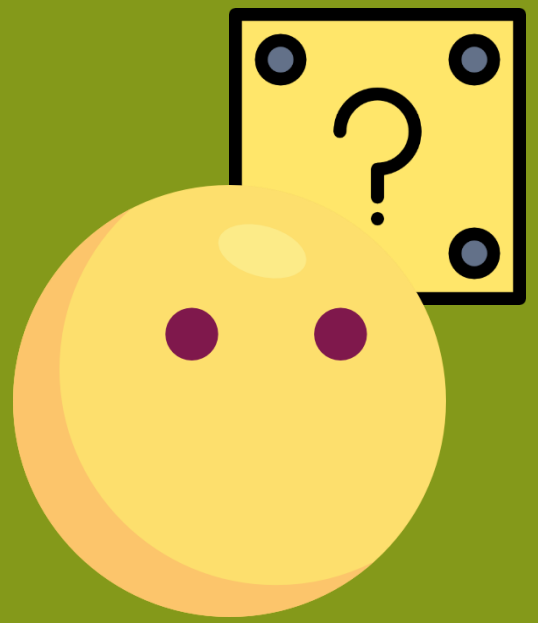
Chapter 4.1: Agile Methodologies

Part I

MOTIVATION...

“Agility within and of itself is a strategy.” [Pearl Zhu]

Can I tell you an Story...?



Agile Manifesto A

Let’s get back to 2001!!

Another kind of software crisis was arising!

Agile Manifesto B

Representations of SCRUM, XP, DSDM, Crystal, Feature Driven had a meeting at the Snowbird ski resort

It actually was winter/cold

Agile Manifesto C

Why do you dare to call us lightweight methodology?
They argued!!

Agile Manifesto C

Software development doesn’t need those heavy corporative structures and hierarchies!

Why do we have to develop and create software projects with all of those constraints and Document-Driven approaches?

Agile Manifesto D

We want to survive!!!

e/commerce

e/business

What do we want?

Individuals and interactions

Over processes and tools

Working software

Over comprehensive Documentation

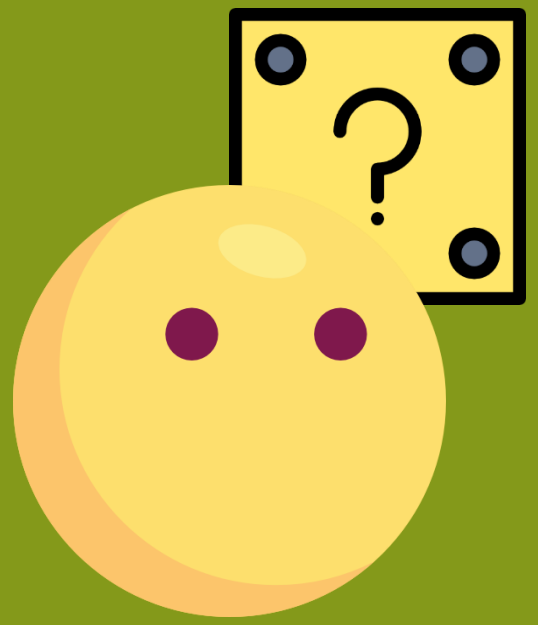
Customer collaboration

Over Contract Negotiation

Responding to change

Over Following a Plan

What about the twelve principles?



Principle #1

Our highest priority is to satisfy the customer
through early and continuous delivery
of valuable software.

Principle #2

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Principle #3

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Principle #4

Business people and developers must work together daily throughout the project.

Principle #5

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

Principle #6

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Principle #7

Working software is the primary measure of progress.

Principle #8

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Principle #9

Continuous attention to technical excellence
and good design enhances agility.

Principle #10

Simplicity--the art of maximizing the amount of work not done--is essential.

Principle #11

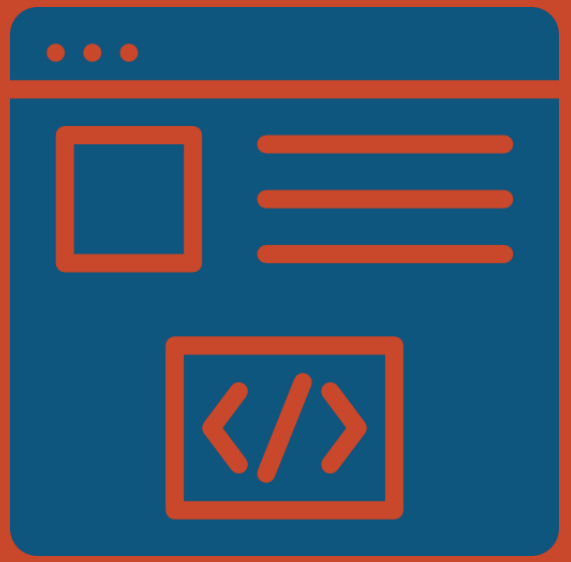
The best architectures, requirements, and designs emerge from self-organizing teams.

Principle #12

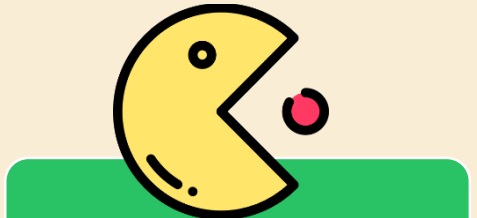
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

XP Methodology

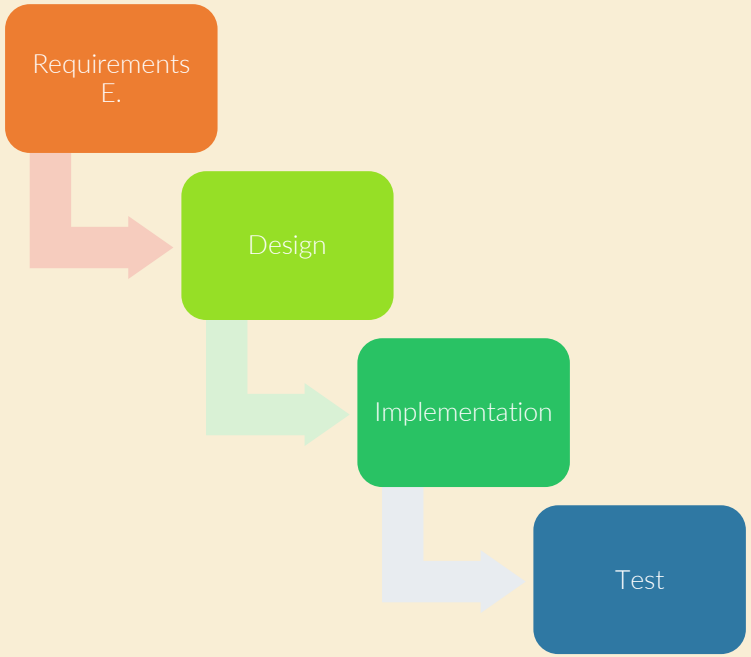
Xtreme Programming!



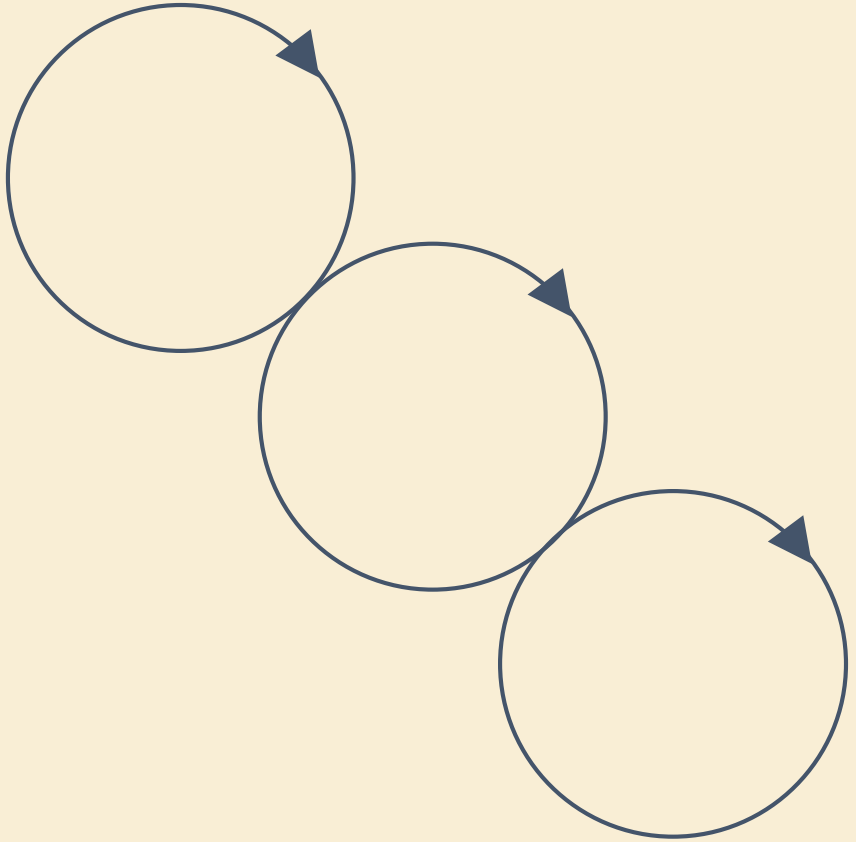
Variables!



Life Cycle!

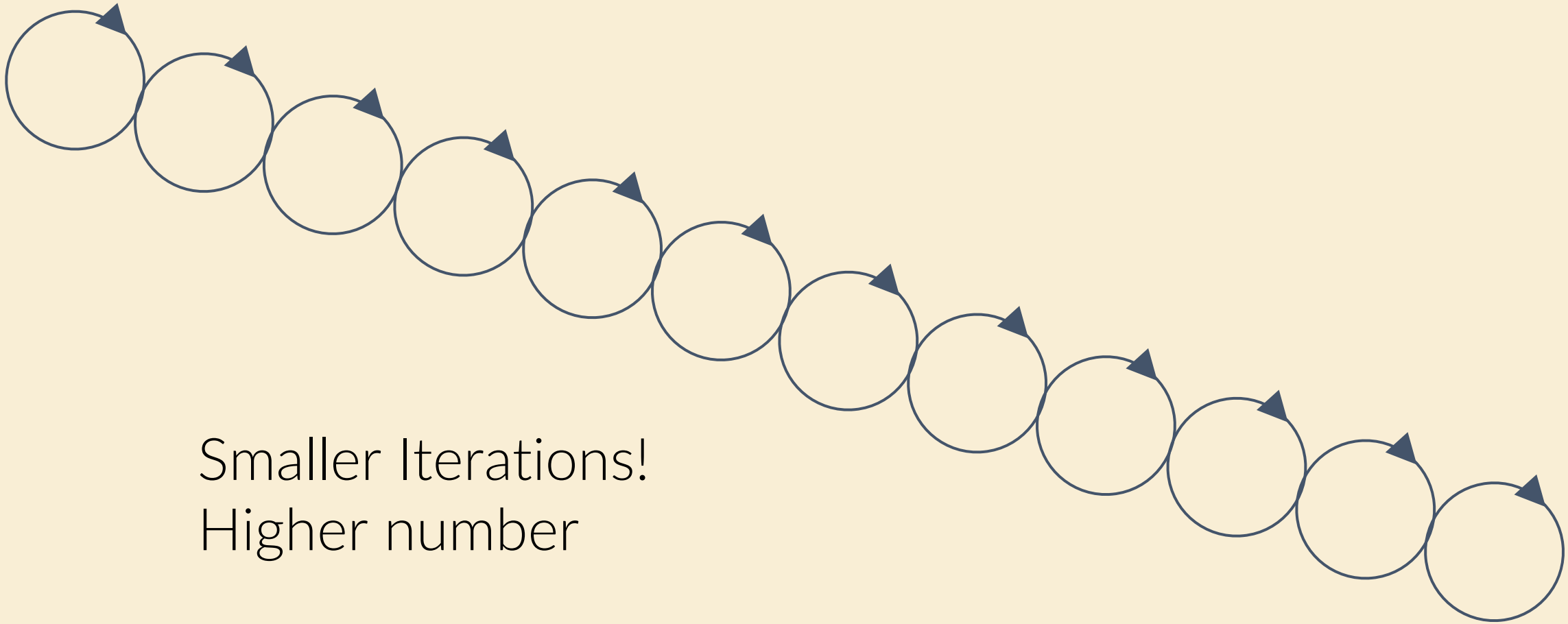


Waterfall



Iterative

XP Life Cycle

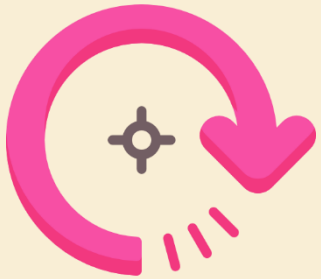


Smaller Iterations!
Higher number

Phases

Exploring

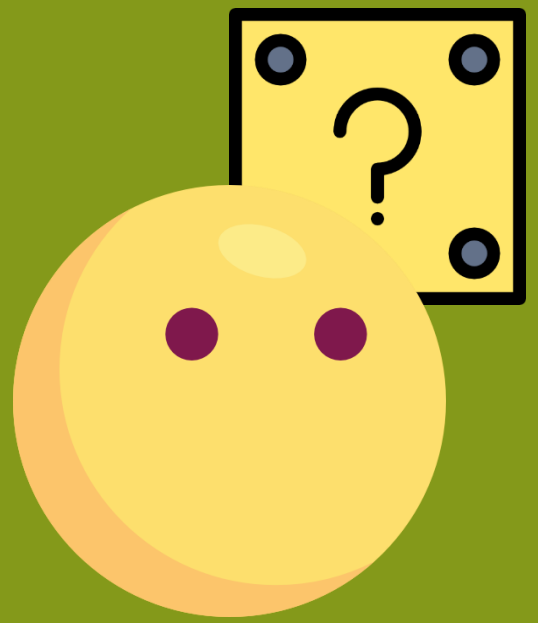
Planning



Iterating

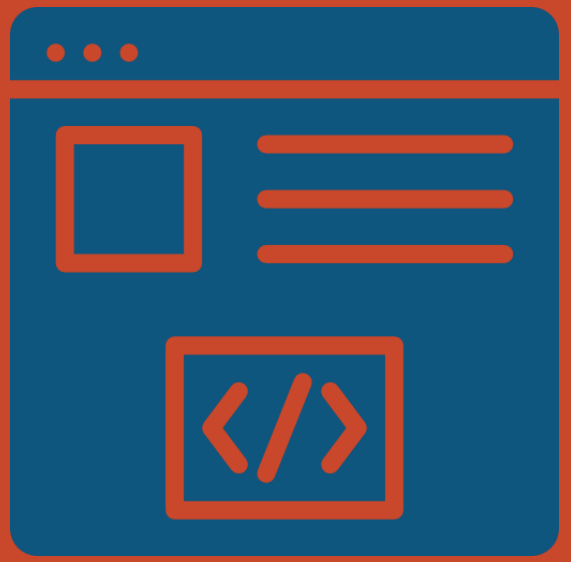
Deployment

Is that it?



Some rules on

Xtreme Programming!



1) Planning

- Create User Stories instead of Use Cases!
- Include ALL The possible stakeholders
- Keep it simple!

User Stories

- They need to be implemented in less than 3 weeks!
- They don’t need all that level of specification!
- Details will be elicitated during development!

Plan!

- The user stories are organized!
- The estimation of development for each user story gives the approximation of the schedule!

Iteration Plan!

- Each iteration needs to be planned
- User stories are translated into activities!!
- Approval tests are provided

Stand Up Meetings

- Everyone shares problems and solutions!
- They are done every single day, and they can last as short as 5 minutes
- Nobody takes a seat in this meeting!

2) Designing

Simplicity

Spike solutions

Refactoring

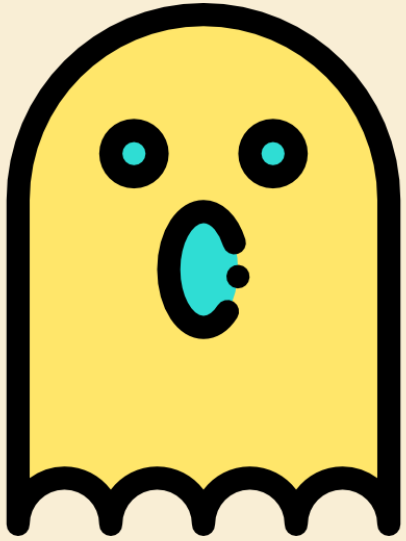
Metaphors

3) Development

- Client!!
- Standards!!
- Test driven programming
- ‘Couples’

- Permanent integrations
- Collective Code
- Sustained Pace

Testing



- Unit Test
- Error Correction
- Approval Tests!

References

- [SOMMERVILLE] Ian Sommervile. *Software Engineering 9th Edition*
- [SCHMIDT] Richard Schmidt. *Software Development Architecture-Driven Software Development*
- [STEPHENS] Beginning Software Engineering. 2015
- [CROOKSHANKS] Software Development Techniques. 2015
- https://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf
- http://agilemanifesto.org/

Class has died… for today!