SER516
Software Agility:
Project and Process Management

Lecture 02. Introduction

Javier Gonzalez-Sanchez
javiergs@asu.edu
javiergs.engineering.asu.edu

Office Hours: By appointment

Definition

SER516 – Software Agility: Project and Process Management

Organized set of **steps**

Definition

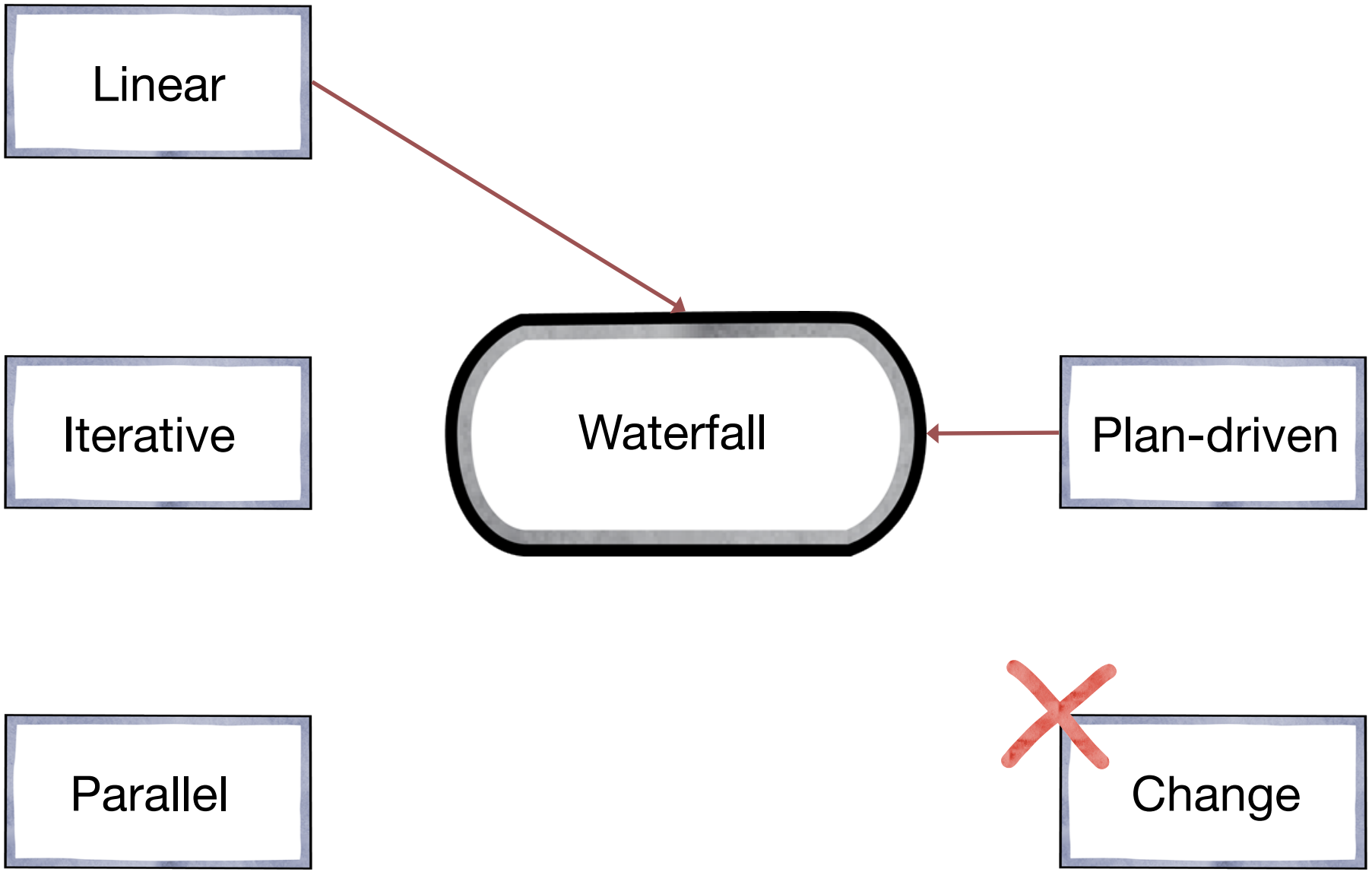
SER516 –

Software **Agility**:

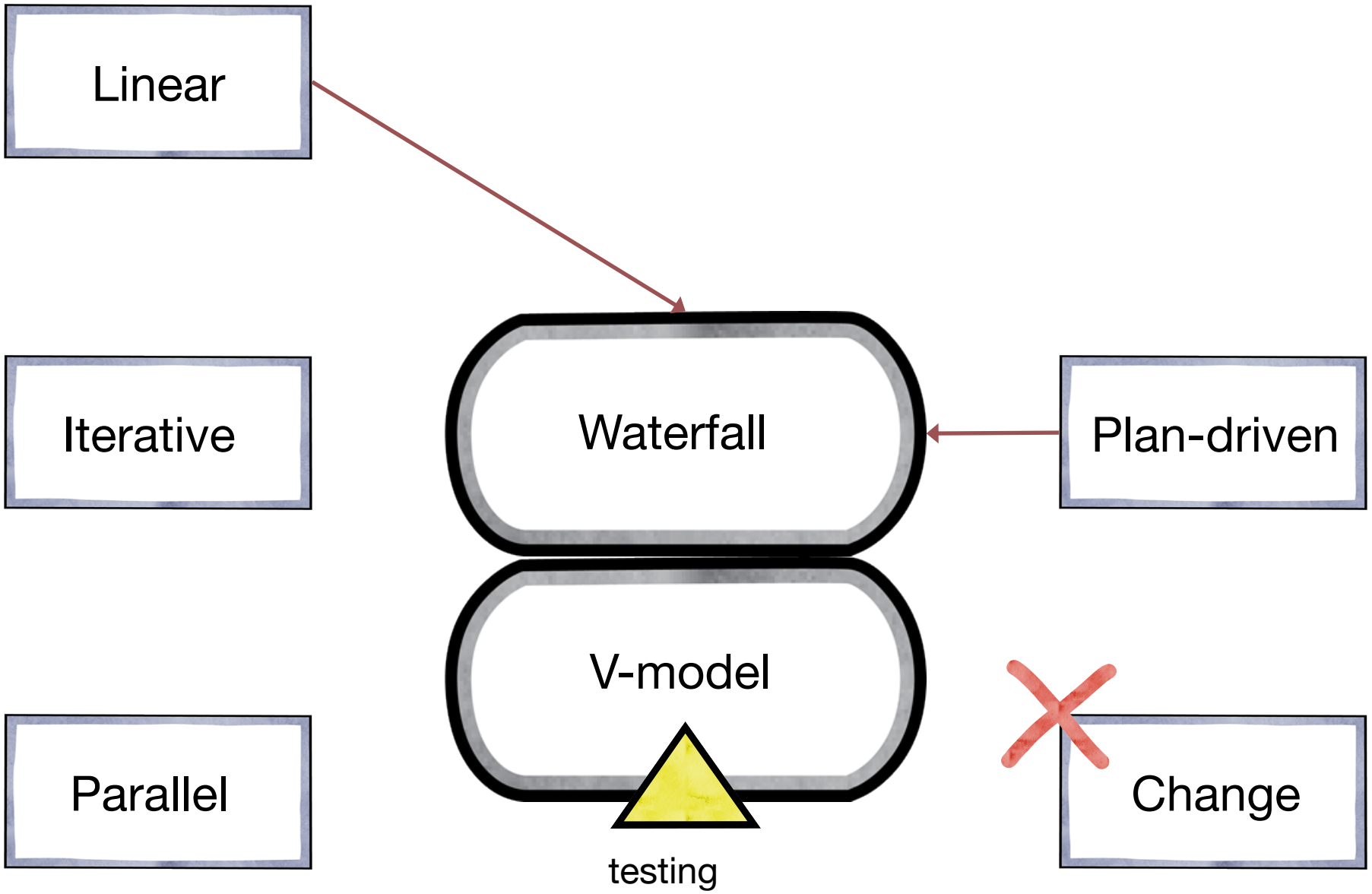
Project and Process Management

Organized set of **steps**

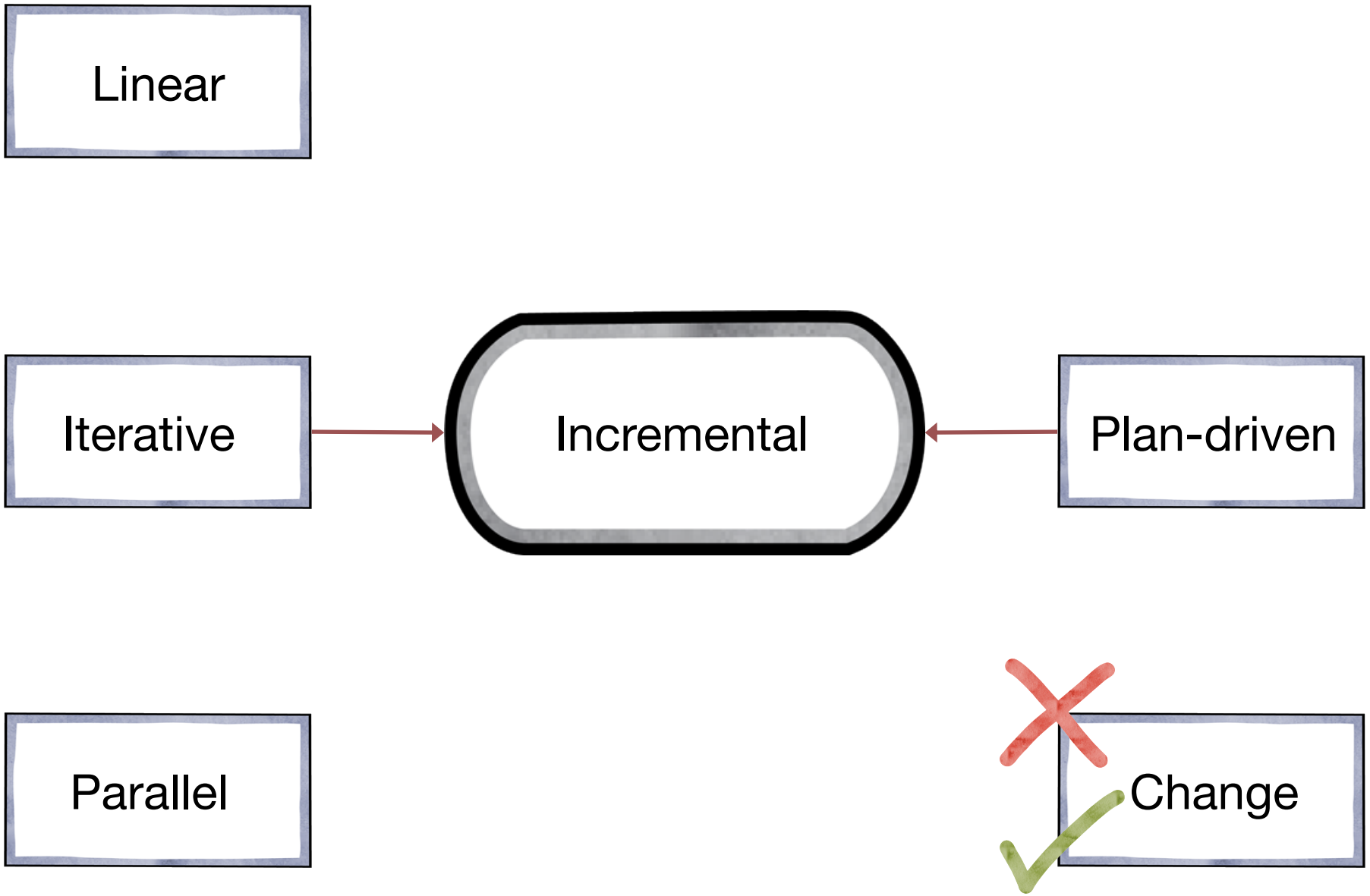
Process Models



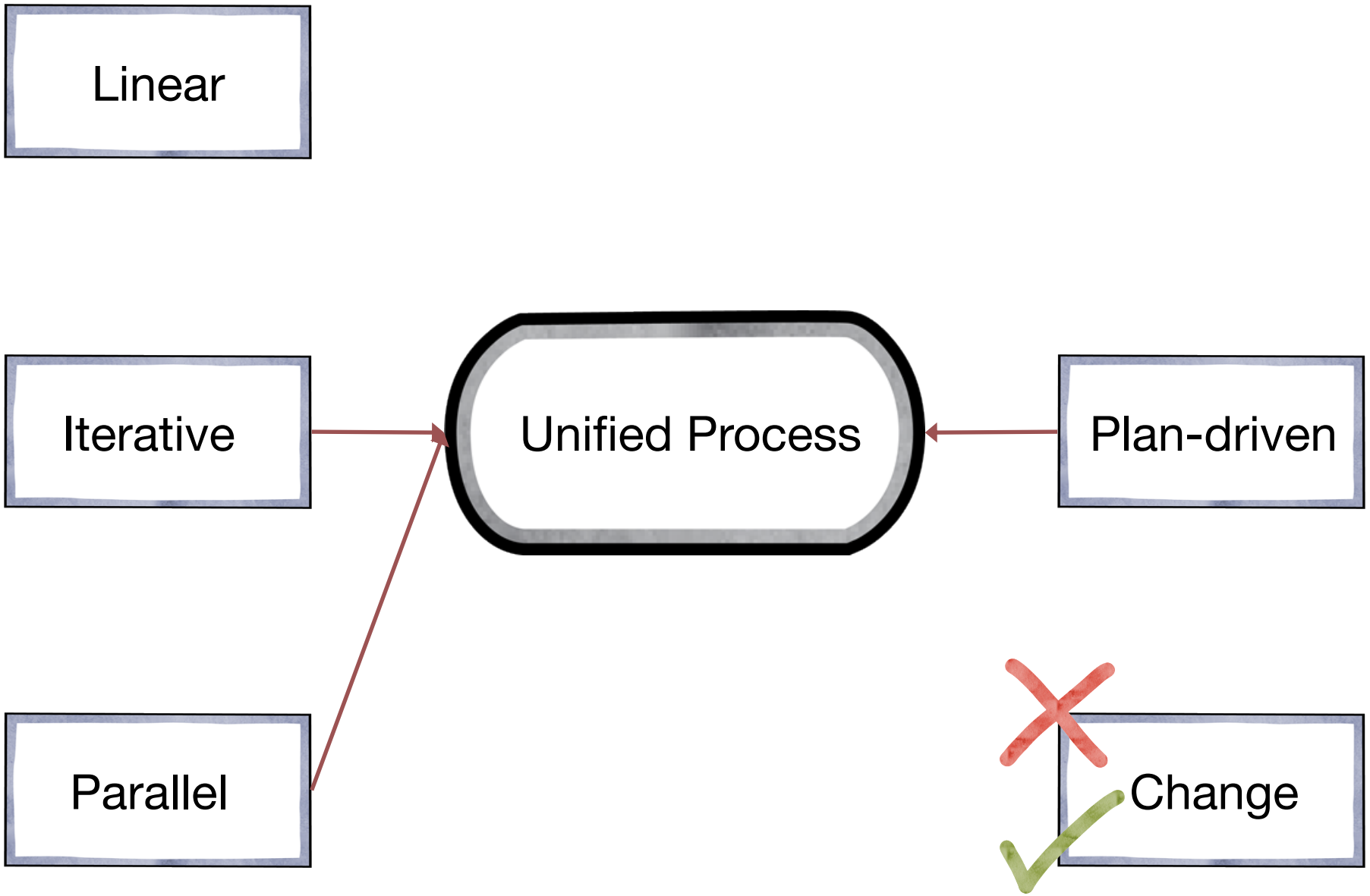
Process Models



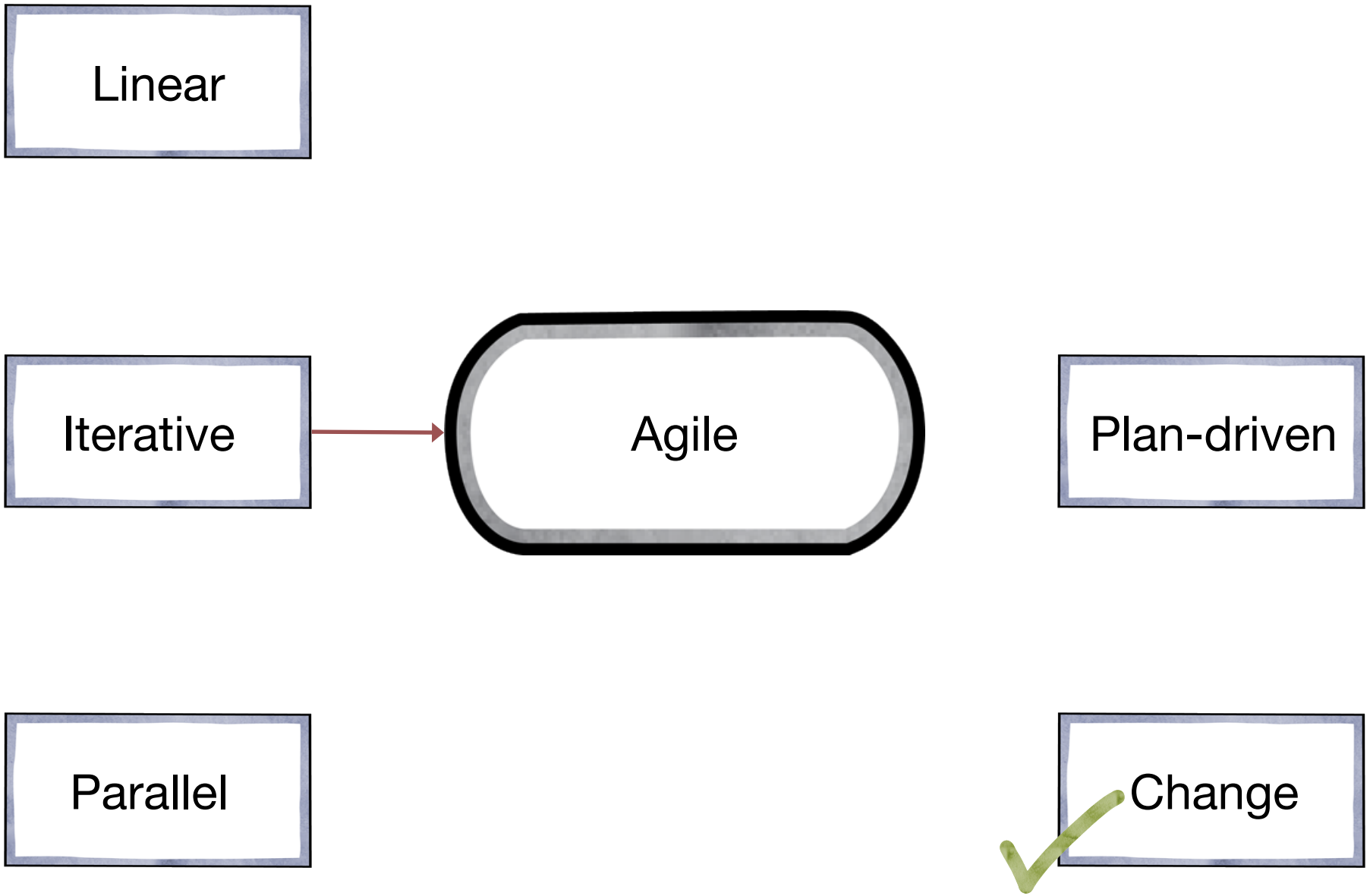
Process Models



Process Models



Process Models

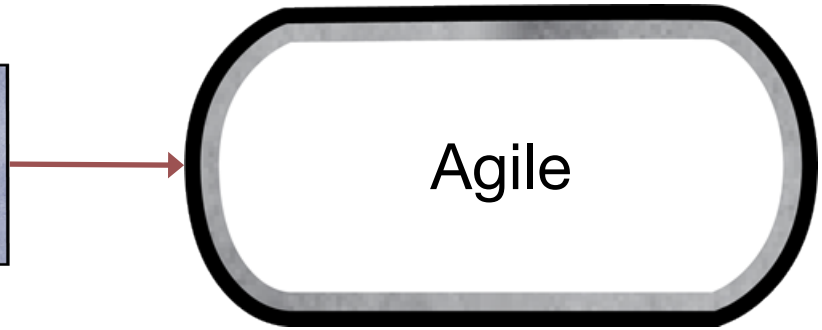


Process Models

Linear

Agile emerged in the **late 1990s** and it aim was to **radically reduce the delivery time** for working software systems

Iterative



Plan-driven

Parallel

Change

Process Models

Linear

Iterative

Parallel

Agile

An Adaptive Plan (Informal)

+ daily work tracking

vs

A Contract

Plan-driven

Change

Scenario 1

Case Study

3.1 Software Process Model

Agile software process model will be followed for this project. During its development, the work will be divided in the form of incremental deliverables. Each requirement will be created as a user story on the storyboard. Each sprint will have a duration of two weeks. Any issues or new feature will be added as a new user story in the open source project management tool, Taiga.

Case Study

Sprint 1 (20 Feb 2017 - 26 Mar 2017)

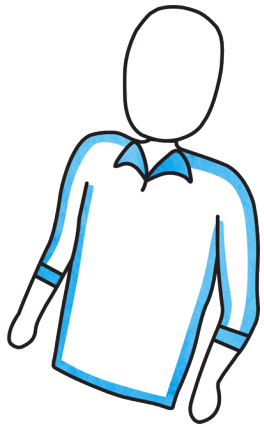
a) Requirements elicitation and analysis
i) Elicit the requirements
ii) Perform requirements analysis
iii) Conduct requirements verification
iv) Elaborate each requirement

Case Study

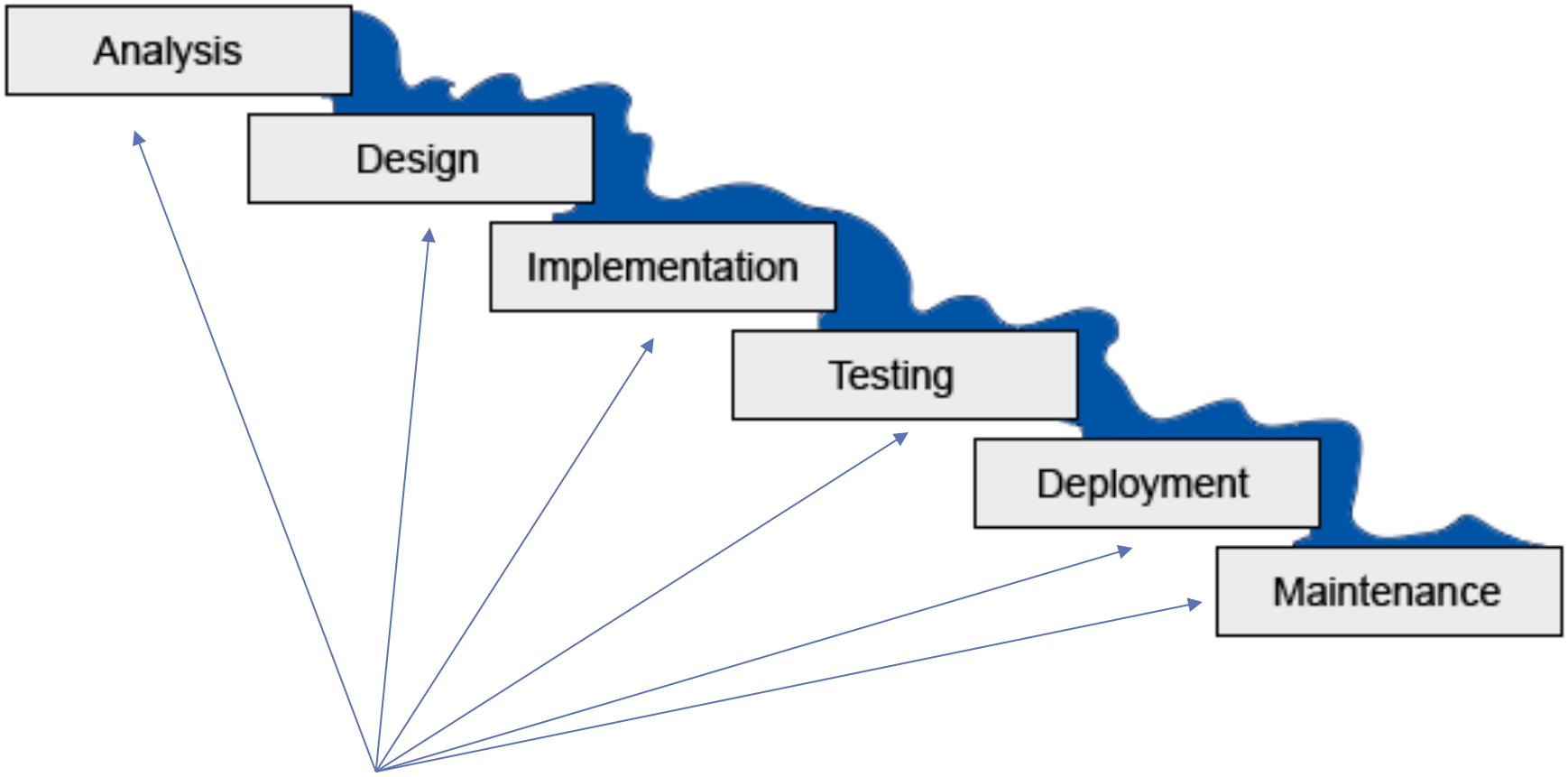
Sprint 2 (27 Mar 2017 - 02 Apr 2017)
a) Implement the functionality to read
b) Implement the functionality to visualize

Case Study

Sprint 8 (16 Oct 2017 - 15 Nov 2017)
c) Perform verification, validation & Testing of the overall desktop application.
d) **Demo**



Error



This are not sprints!

Scenario 2

Just Programming ...

Agile Manifesto

- Through this work we have come to value:

*"**Individuals and interactions** over **processes** and tools,*

Working software** over **comprehensive documentation,

Customer collaboration** over **contract negotiation,

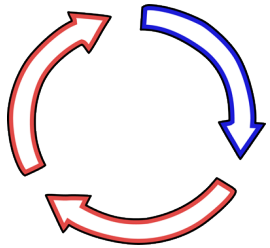
*Responding to **change** over following a **plan.** "*

- That is, while there is value in the items on the right, we value the items on the left more.

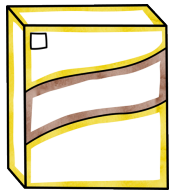
Agile Manifesto

Human-Human Interaction

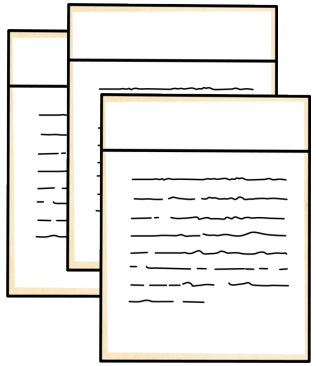
while there is value
in the items on the right,
we **value**
the items on the left
more.



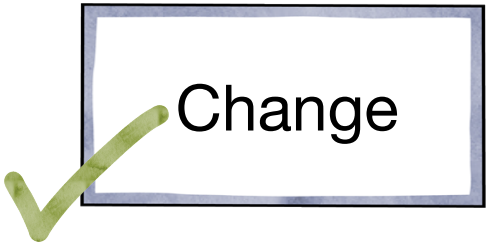
Process



Working Software



Documents



Applicability

- Product development where a software company is developing a **small or medium-sized** product for sale.

  Several software products and apps nowadays are **small or medium-sized;** therefore, they are developed using an agile approach.

- Custom system development within an organization, where there is a clear **commitment from the customer** to become involved in the development process and where there are few external rules and regulations that affect the software.

Pros

- The product is broken down into a set of manageable and **understandable chunks**.
- Unstable requirements do not hold up progress.
- The whole team have visibility of everything and consequently **team communication** is improved.
- Customers see **on-time delivery of increments** and gain feedback on how the product works.

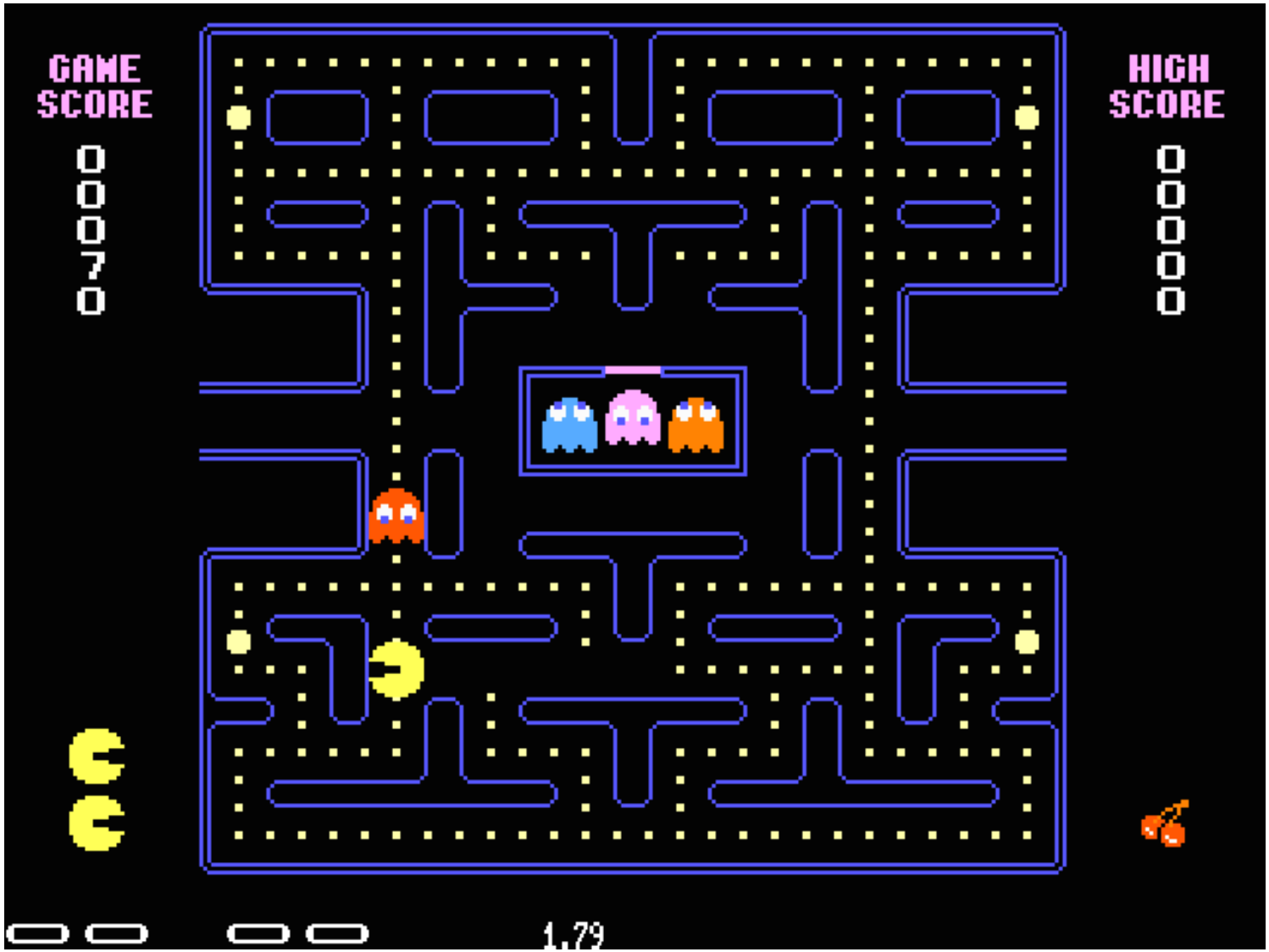
Cons

- The informality of agile development is incompatible with the legal approach to **contract definition (requirement specification)** that is commonly used in large companies.

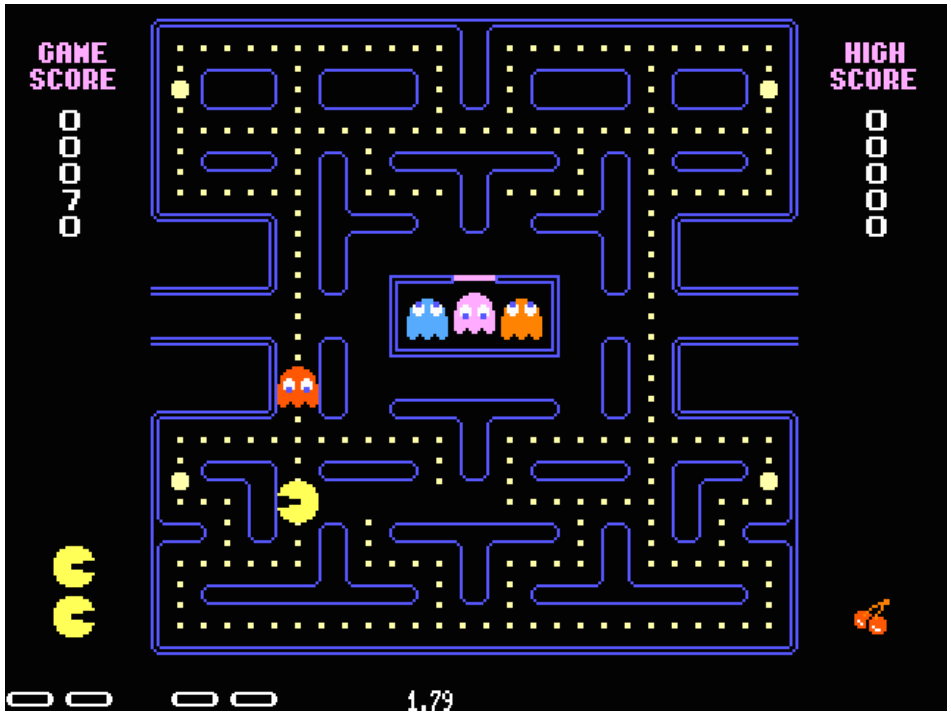
- Agile methods are most appropriate for new software development rather than software **maintenance (documentation)**. Yet most software costs in large companies come from maintaining their existing software systems.

Scenario 3

Pac-Man



Pac-Man



- Files: 11
- Methods: 75
- Lines: 1,257
- Effort: 3.8 Person-months

Some random project

How to select a Process Model?

Agile and Plan-based factors: System

- How **large** is the system being developed?
  - Agile methods are most effective a relatively small co-located team who can communicate informally.
- What **type** of system is being developed?
  - Systems that require a lot of analysis before implementation need a fairly detailed design to carry out this analysis.
- What is the expected system **lifetime**?
  - Long-lifetime systems require documentation to communicate the intentions of the system developers to the support team.
- Is the system subject to external **regulation**?
  - If a system is regulated you will probably be required to produce detailed documentation as part of the system safety case.

Agile and Plan-based factors: Team

- **How good are** the designers and programmers in the development team?
  - It is sometimes argued that agile methods require higher skill levels than plan-based approaches in which programmers simply translate a detailed design into code.

- How is the development **team organized**?
  - Design documents may be required if the team is distributed.

- What **support technologies** are available?
  - IDE support for visualisation and program analysis is essential if design documentation is not available.

Agile and Plan-based factors: Organization

- Traditional engineering organizations have a culture of plan-based development, as this is the norm in engineering.
- Is it standard organizational practice to develop a detailed system specification?
- Will **customer representatives be available to provide feedback** of system increments?
- Can informal agile development fit into the organizational culture of detailed documentation?

References

- Process Models – Somerville Chapter 2
- Introduction to Agile –Somerville Chapter 3
- Introduction to Software Engineering (CSE360) slides:

  http://javiergs.com/teaching/cse360/

SER516 – Software Agility

Javier Gonzalez-Sanchez

javiergs@asu.edu

Spring 2020