1927 USTP Southern
UNIVERSITY OF
SCIENCE AND TECHNOLOGY
OF SOUTHERN PHILIPPINES

COLLEGE OF
INFORMATION
TECHNOLOGY
AND COMPUTING

ELECTIVE 4 (IT 415)

Junar A. Landicho

junarlandicho@ustp.edu.ph

“

Develop a passion for learning. If you do, you will never cease to grow.”

Anthony J. D’Angelo

Topic 3: Agile Development

Objectives

By the end of this topic, students will be able to:

- Understand the rationale for agile software development methods, the agile manifesto, and the differences between agile and plan-driven development.
- Know about important agile development practices such as user stories, refactoring, pair programming and test-first development.
- Apply the Scrum approach to agile project management.
- Understand the issues of scaling agile development methods and combining agile approaches with plan-driven approaches in the development of large software systems.

Overview

1. Current Situation in SW Development
2. Plan-driven and Agile Development
3. Agile Development Methodologies
4. Issues and Factors

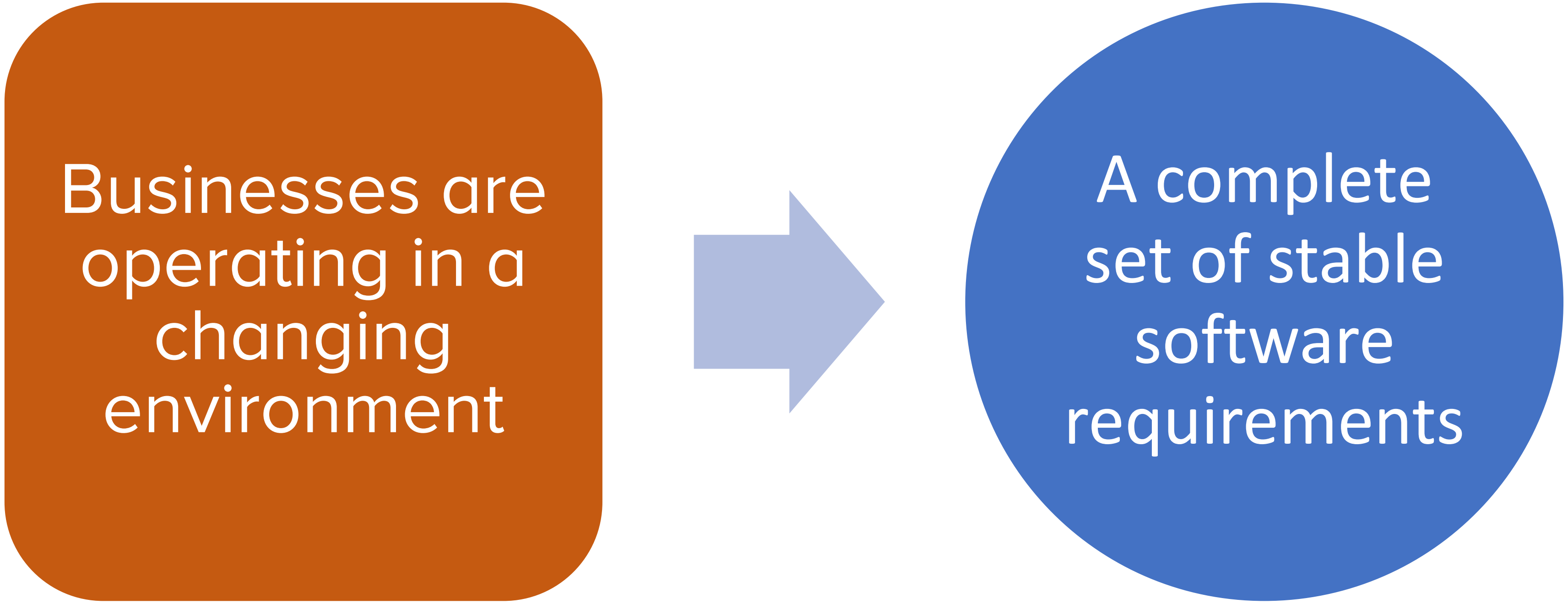
Current Situation in SW Development

```
    mirror_mod.use_y = True
    mirror_mod.use_z = False
elif _operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

    #selection at the end -add back the deselected mirror modifier object
mirror_ob.select= 1
modifier_ob.select=1
bpy.context.scene.objects.active = modifier_ob
print("Selected" + str(modifier_ob)) # modifier ob is the active ob
    #mirror_ob.select = 0
```

Current Situation in SW Development

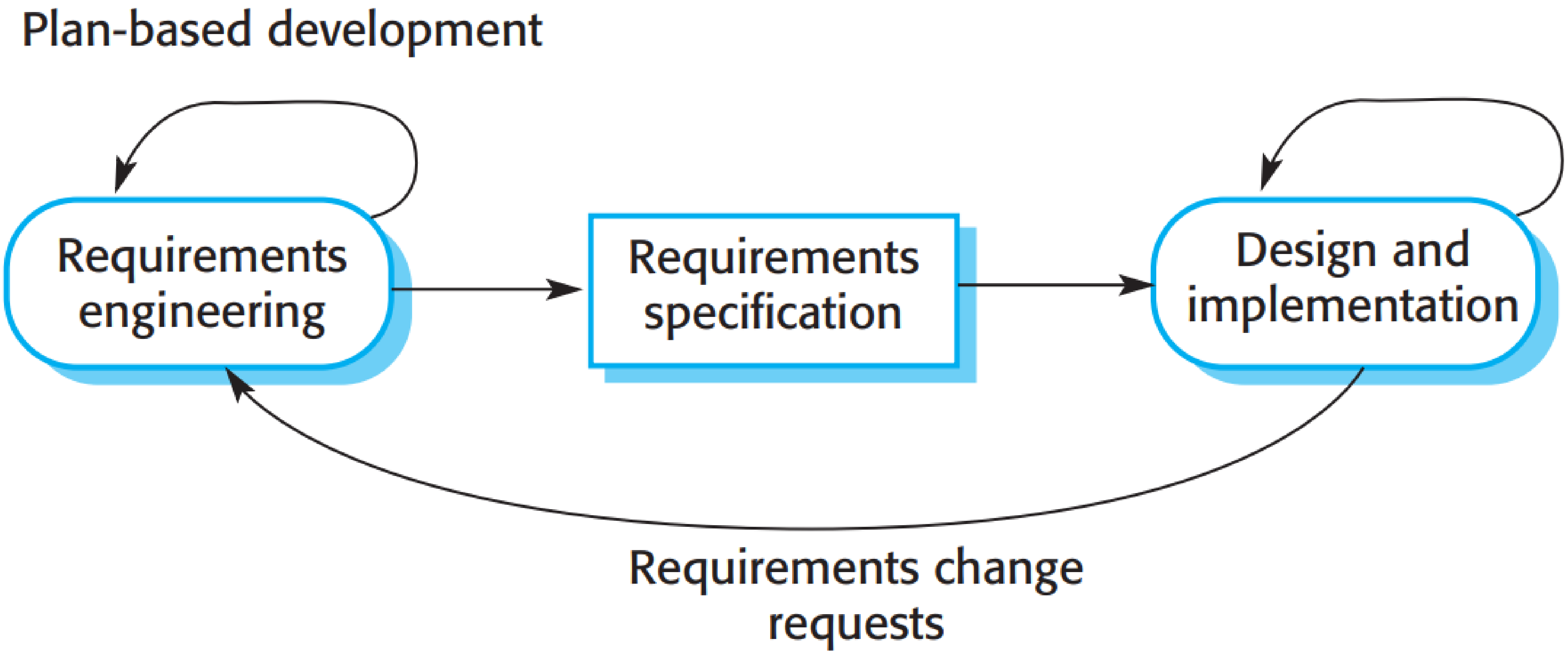
- Rapid software development and delivery



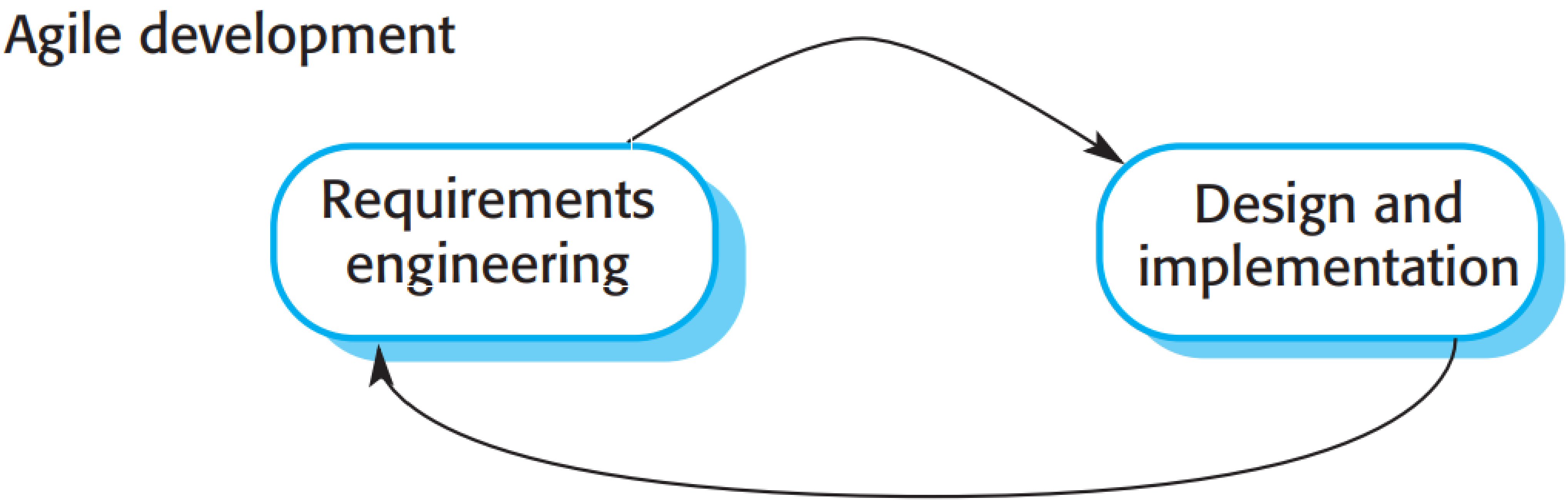
Businesses are operating in a changing environment

A complete set of stable software requirements

Plan-Driven Development


Plan-based development
Requirements engineering
Requirements specification
Design and implementation
Requirements change requests

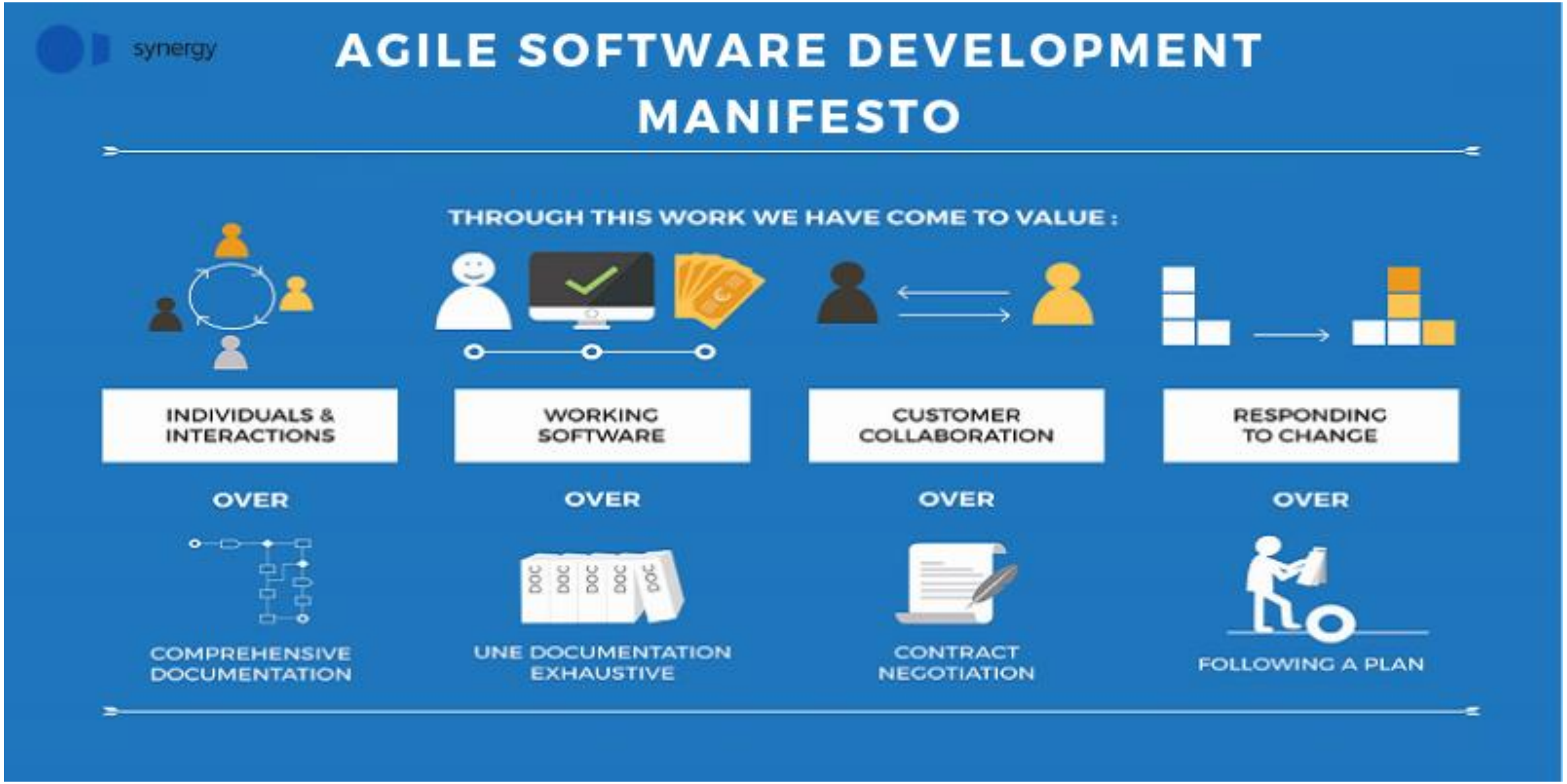
Agile Development



Agile Methods

- Dissatisfaction with the overheads involved in software design methods of the 1980s and 1990s led to the creation of agile methods. These methods:
  - Focus on the code rather than the design
  - Are based on an iterative approach to software development
  - Are intended to deliver working software quickly and evolve this quickly to meet changing requirements.
- The aim of agile methods is to reduce overheads in the software process (e.g. by limiting documentation) and to be able to respond quickly to changing requirements without excessive rework.

Agile Methods



synergy

AGILE SOFTWARE DEVELOPMENT MANIFESTO

THROUGH THIS WORK WE HAVE COME TO VALUE :

INDIVIDUALS & INTERACTIONS

OVER

COMPREHENSIVE DOCUMENTATION

WORKING SOFTWARE

OVER

UNE DOCUMENTATION EXHAUSTIVE

CUSTOMER COLLABORATION

OVER

CONTRACT NEGOTIATION

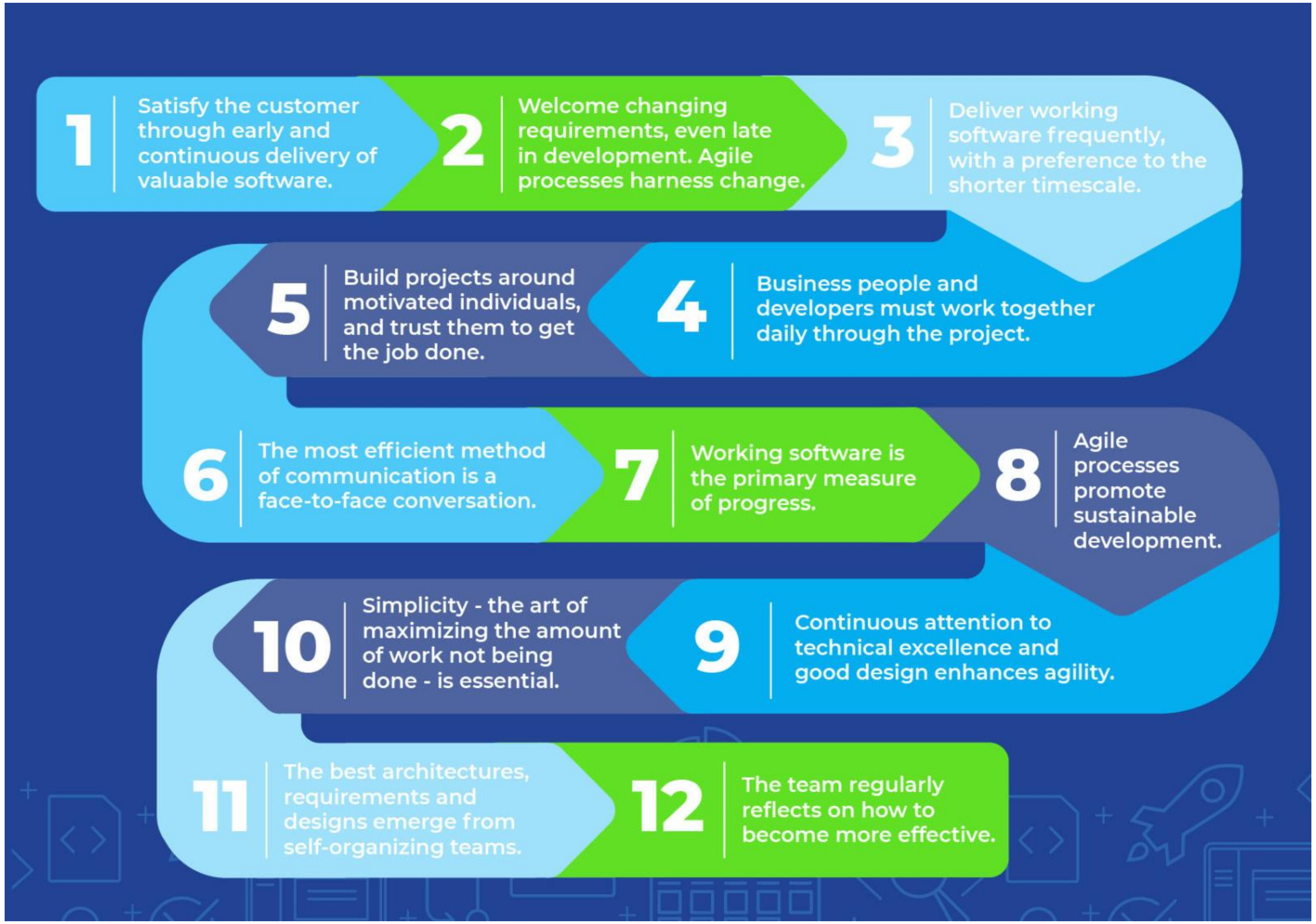
RESPONDING TO CHANGE

OVER

FOLLOWING A PLAN

Agile Methods

- Agile Principles



1 Satisfy the customer through early and continuous delivery of valuable software.

2 Welcome changing requirements, even late in development. Agile processes harness change.

3 Deliver working software frequently, with a preference to the shorter timescale.

4 Business people and developers must work together daily through the project.

5 Build projects around motivated individuals, and trust them to get the job done.

6 The most efficient method of communication is a face-to-face conversation.

7 Working software is the primary measure of progress.

8 Agile processes promote sustainable development.

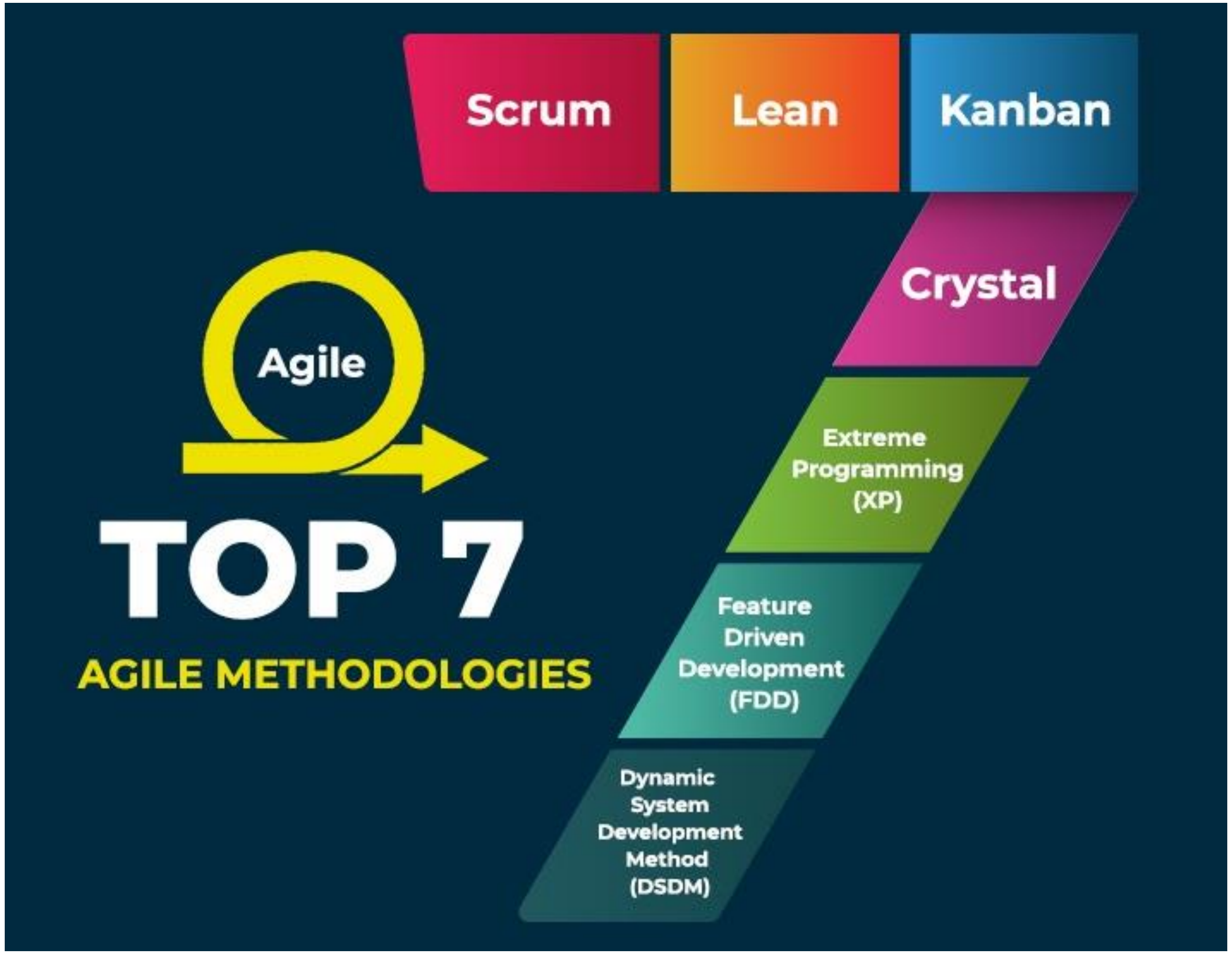
9 Continuous attention to technical excellence and good design enhances agility.

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

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

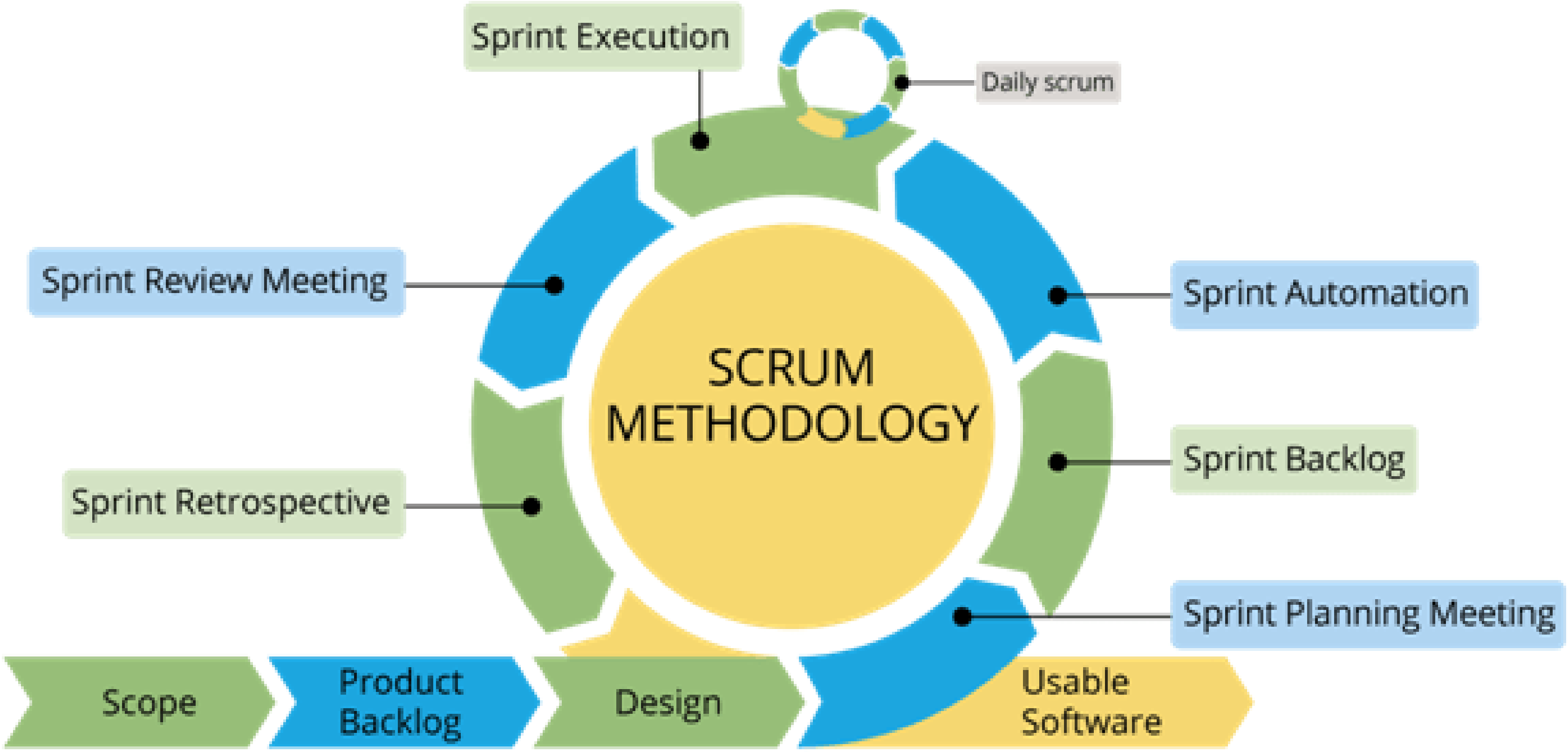
12 The team regularly reflects on how to become more effective.

Agile Methods



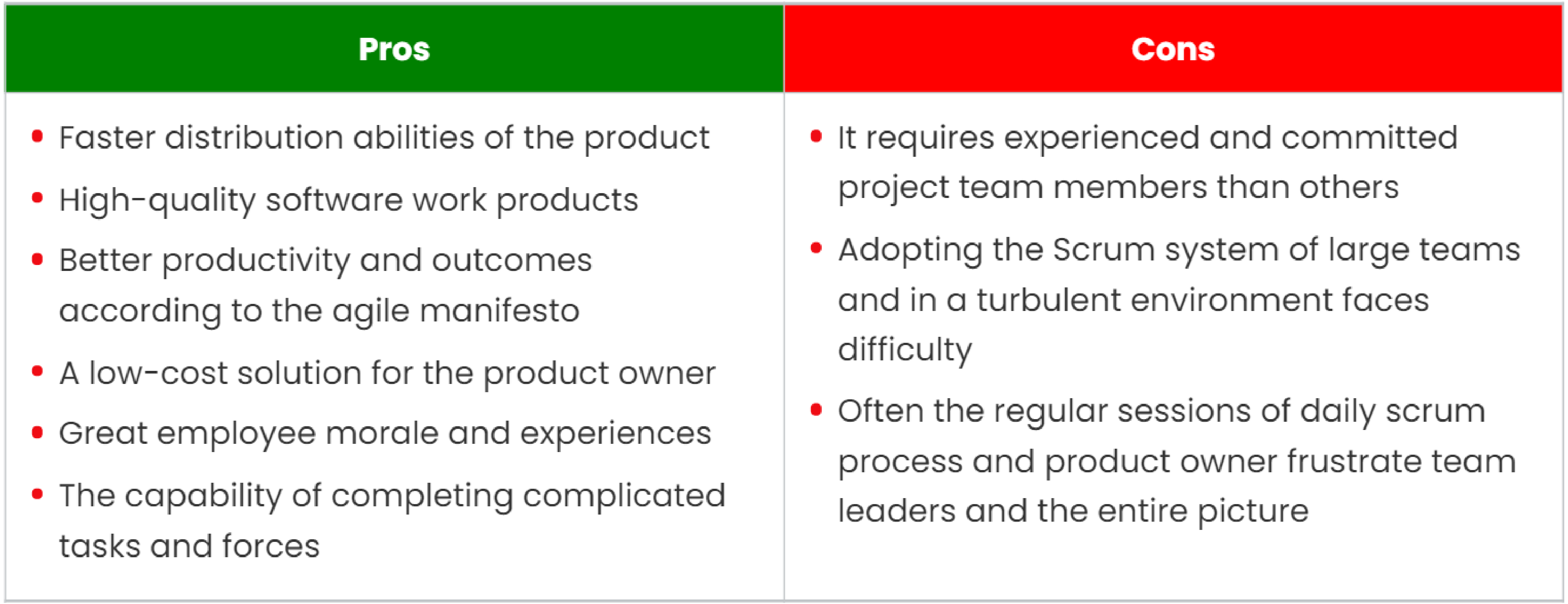
Agile Methods

- ▶Scrum Methodology



Agile Methods

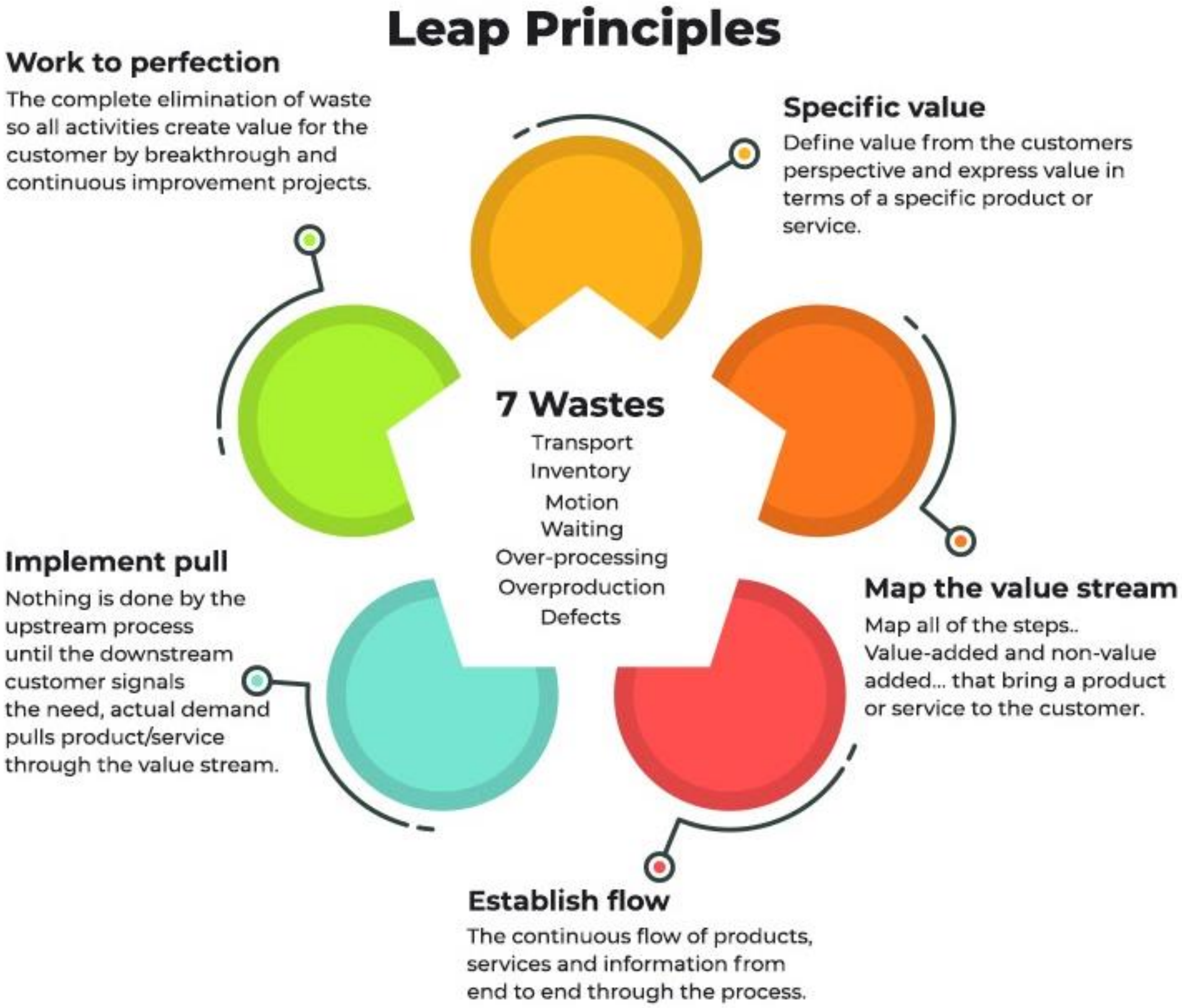
▶Scrum Methodology



Pros	Cons
• Faster distribution abilities of the product • High-quality software work products • Better productivity and outcomes according to the agile manifesto • A low-cost solution for the product owner • Great employee morale and experiences • The capability of completing complicated tasks and forces	• It requires experienced and committed project team members than others • Adopting the Scrum system of large teams and in a turbulent environment faces difficulty • Often the regular sessions of daily scrum process and product owner frustrate team leaders and the entire picture

Agile Methods

- ▶Lean Software Development



Leap Principles

Work to perfection

The complete elimination of waste so all activities create value for the customer by breakthrough and continuous improvement projects.

Specific value

Define value from the customers perspective and express value in terms of a specific product or service.

7 Wastes

Transport
Inventory
Motion
Waiting
Over-processing
Overproduction
Defects

Implement pull

Nothing is done by the upstream process until the downstream customer signals the need, actual demand pulls product/service through the value stream.

Map the value stream

Map all of the steps.. Value-added and non-value added... that bring a product or service to the customer.

Establish flow

The continuous flow of products, services and information from end to end through the process.

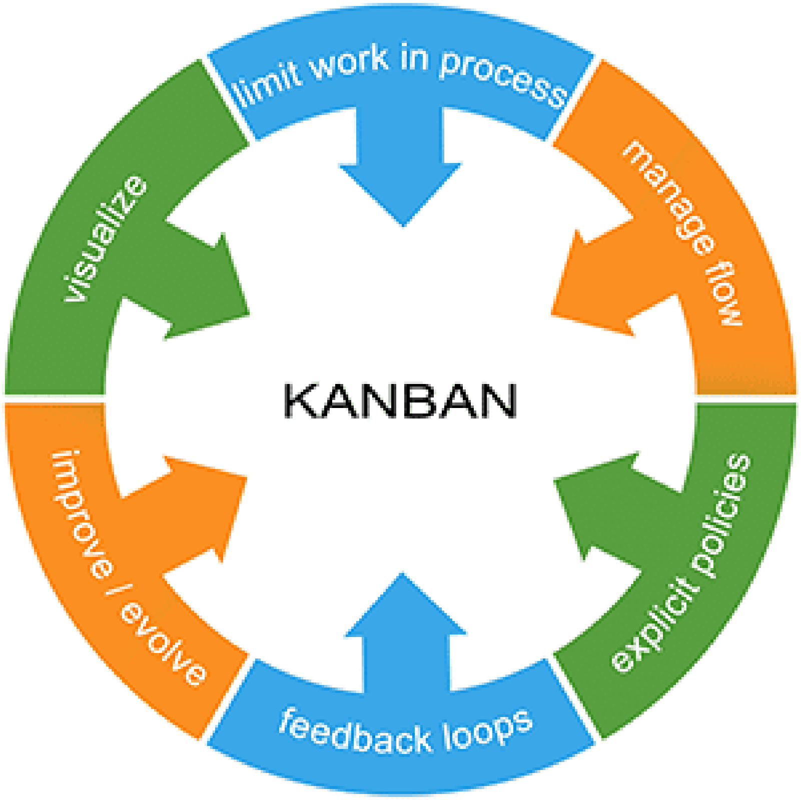
Agile Methods

►Lean Software Development

Pros	Cons
• Eliminates the superfluous activity • Takes less time to deliver working functionalities, even in the agile manifesto • Easily scalable • Better predictability • High adaptability by development teams • Solutions evolve through collaboration	• Dependent on the team's ability • Teams find difficulties to concentrate as tasks are divided into various elements and often indulges in distractions • Need effective documentation

Agile Methods

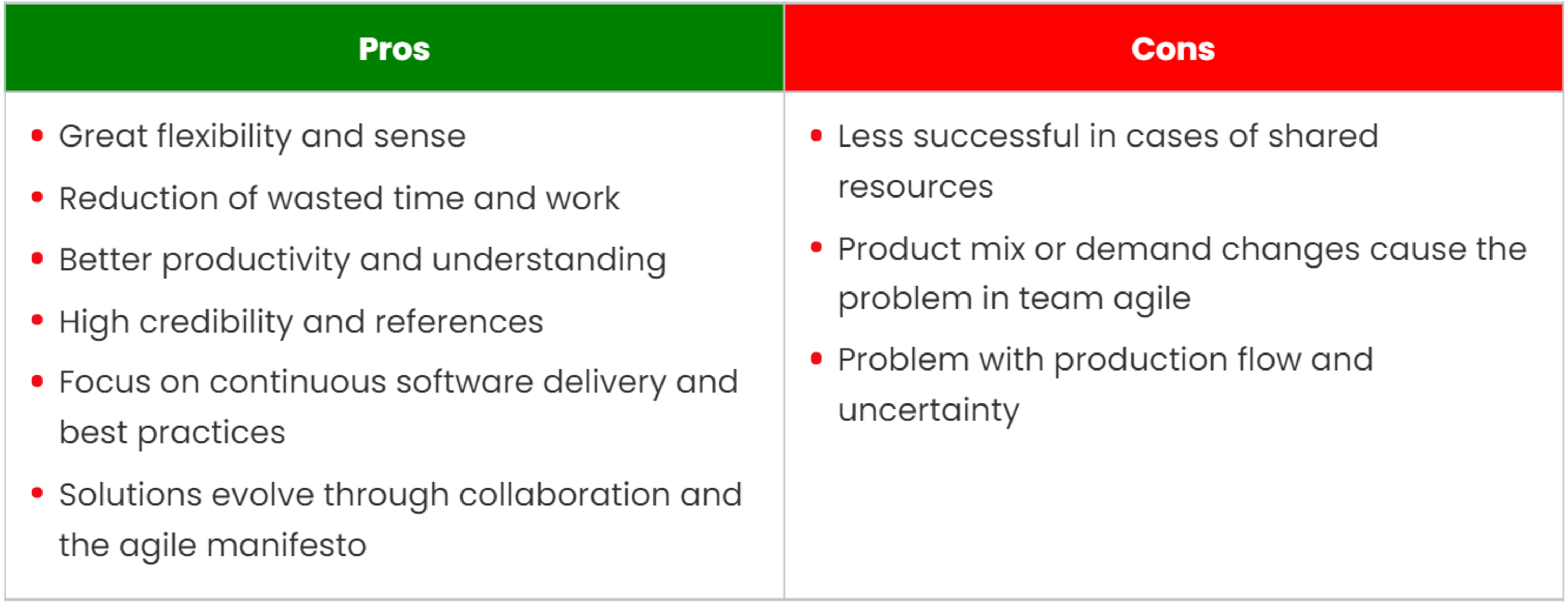
- Kanban



Kanban
To Do
Doing
Done

Agile Methods

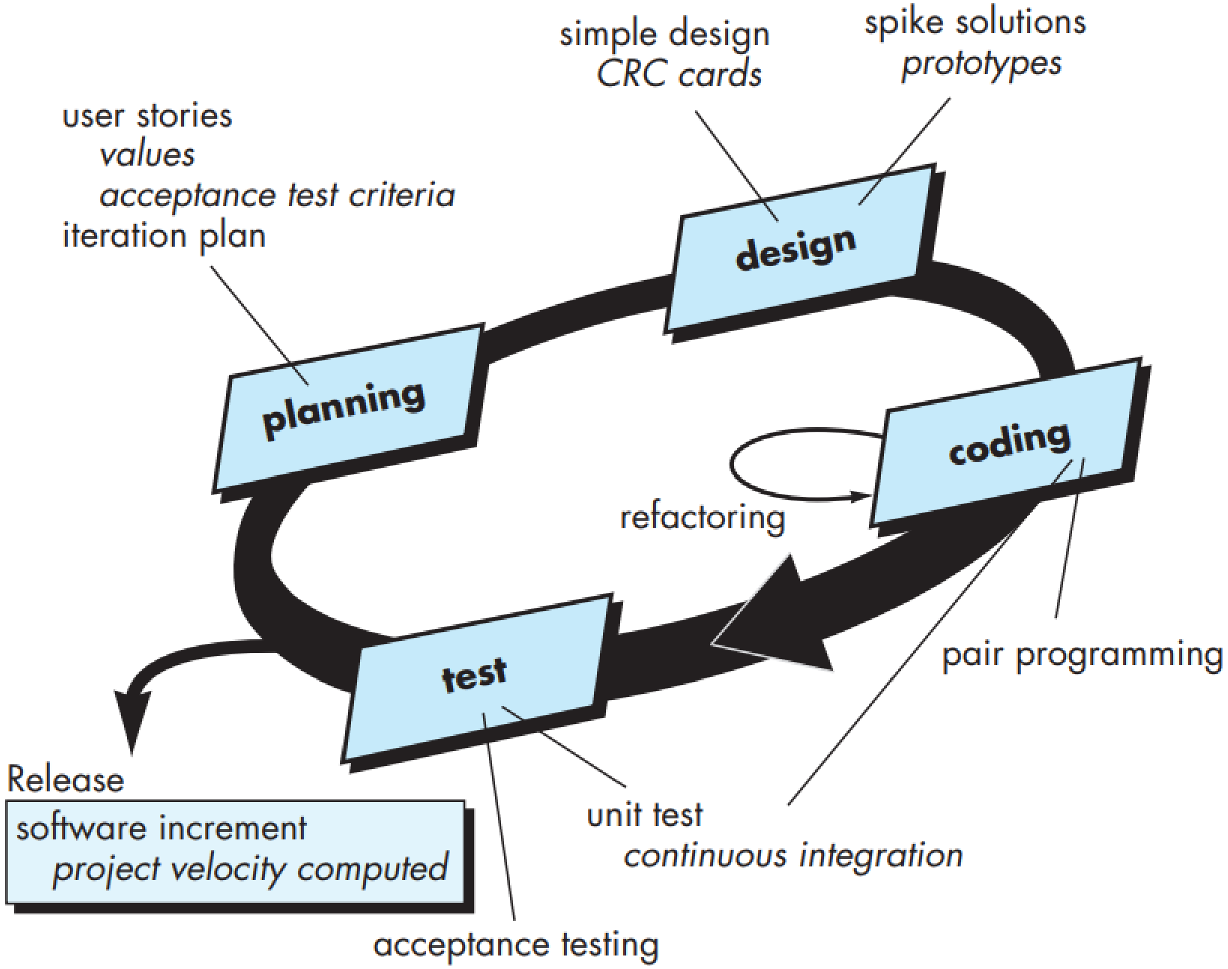
- Kanban



Pros	Cons
• Great flexibility and sense • Reduction of wasted time and work • Better productivity and understanding • High credibility and references • Focus on continuous software delivery and best practices • Solutions evolve through collaboration and the agile manifesto	• Less successful in cases of shared resources • Product mix or demand changes cause the problem in team agile • Problem with production flow and uncertainty

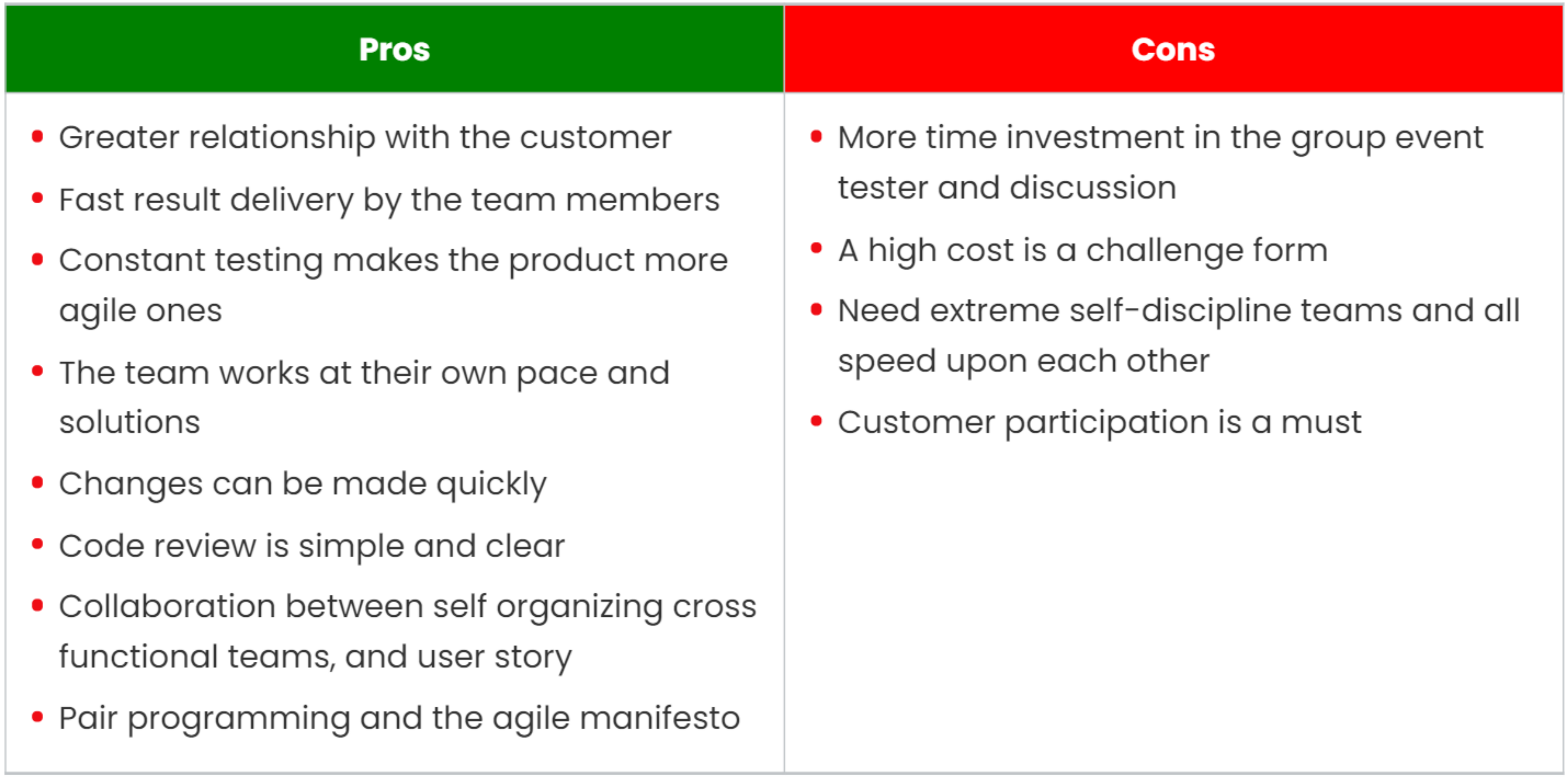
Agile Methods

- ►Extreme Programming (XP)



Agile Methods

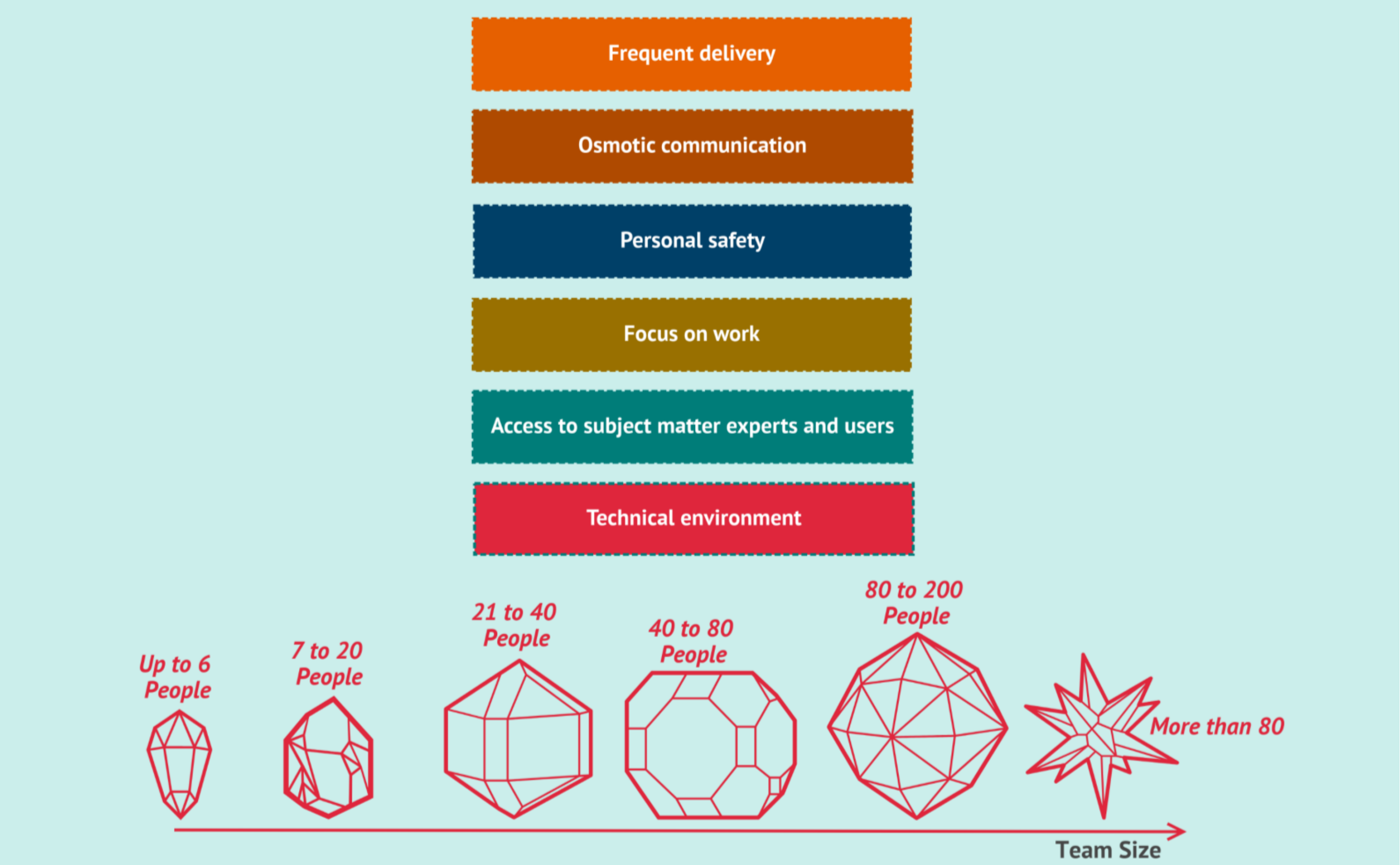
- Extreme Programming (XP)



Pros	Cons
- Greater relationship with the customer - Fast result delivery by the team members - Constant testing makes the product more agile ones - The team works at their own pace and solutions - Changes can be made quickly - Code review is simple and clear - Collaboration between self organizing cross functional teams, and user story - Pair programming and the agile manifesto	- More time investment in the group event tester and discussion - A high cost is a challenge form - Need extreme self-discipline teams and all speed upon each other - Customer participation is a must

Agile Methods

- Crystal



Frequent delivery

Osmotic communication

Personal safety

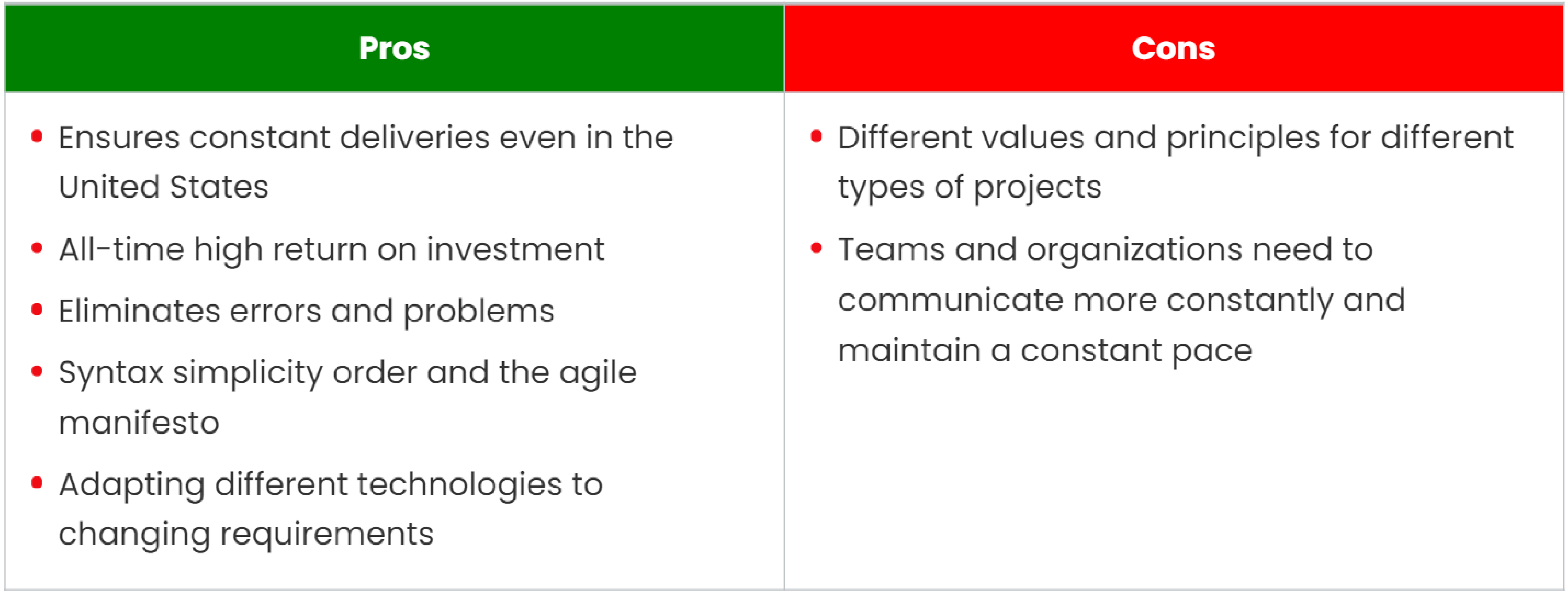
Focus on work

Access to subject matter experts and users

Technical environment

Agile Methods

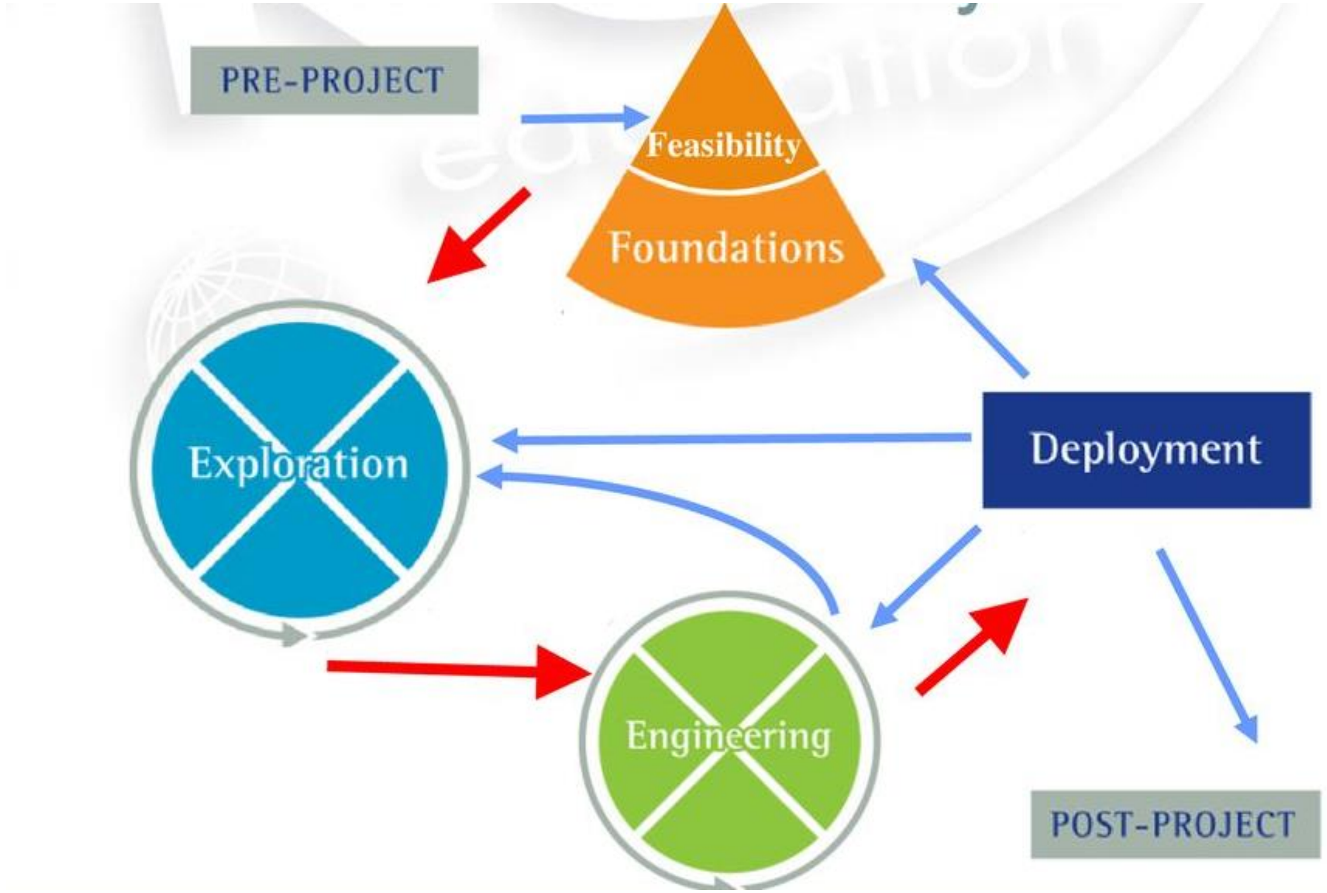
- Crystal



Pros	Cons
• Ensures constant deliveries even in the United States • All-time high return on investment • Eliminates errors and problems • Syntax simplicity order and the agile manifesto • Adapting different technologies to changing requirements	• Different values and principles for different types of projects • Teams and organizations need to communicate more constantly and maintain a constant pace

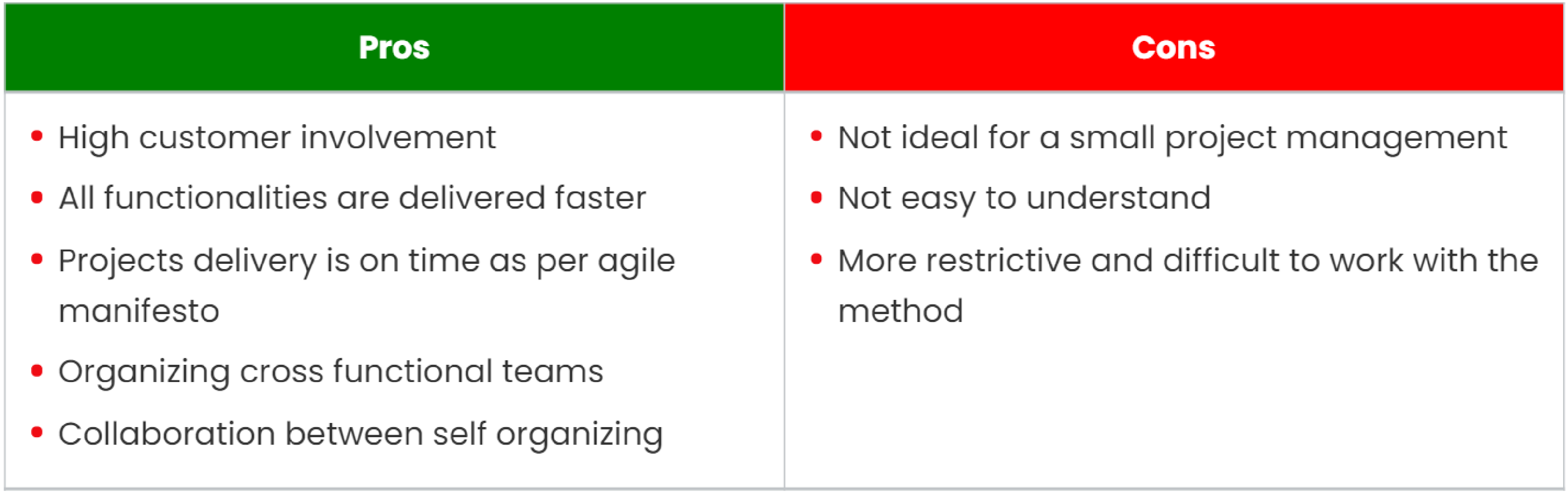
Agile Methods

- ▶Dynamic System Development Method (DSDM)



Agile Methods

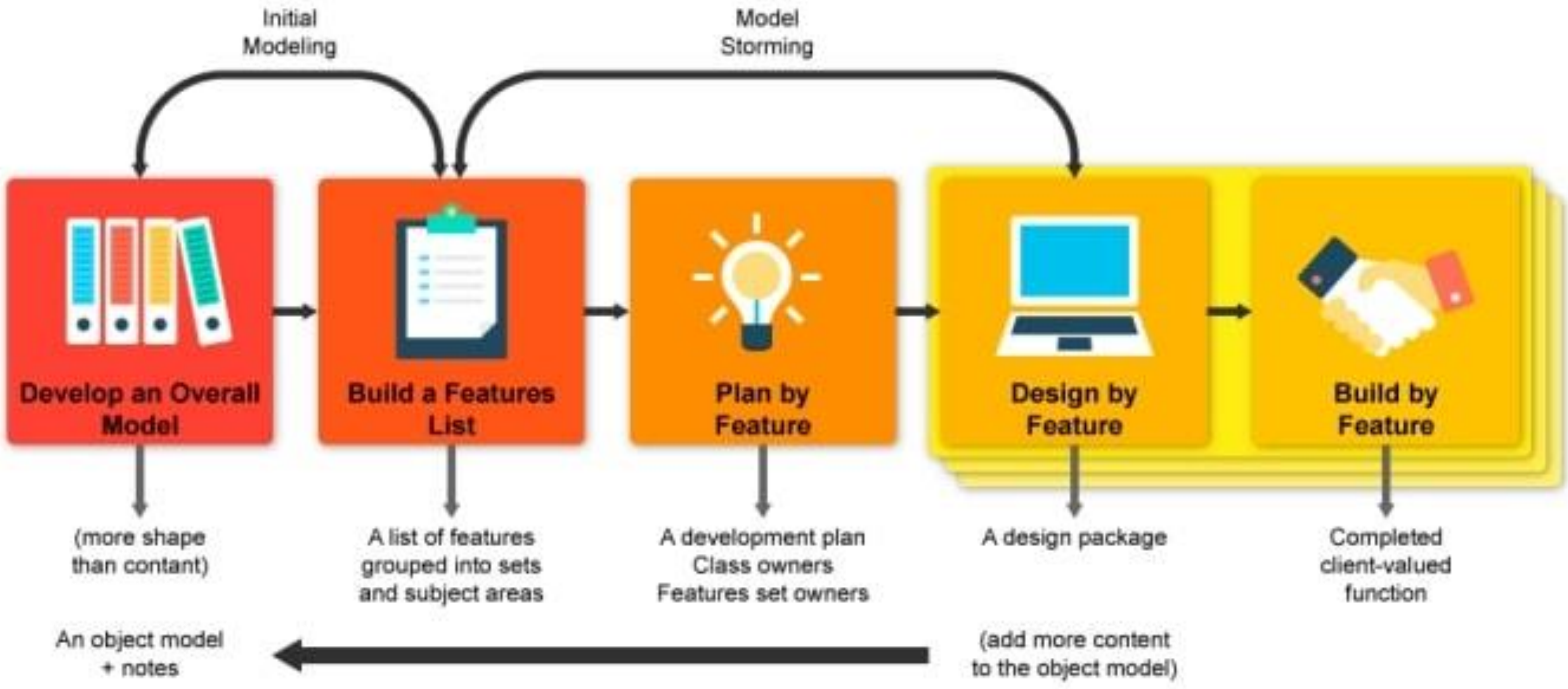
- Dynamic System Development Method (DSDM)



Pros	Cons
• High customer involvement • All functionalities are delivered faster • Projects delivery is on time as per agile manifesto • Organizing cross functional teams • Collaboration between self organizing	• Not ideal for a small project management • Not easy to understand • More restrictive and difficult to work with the method

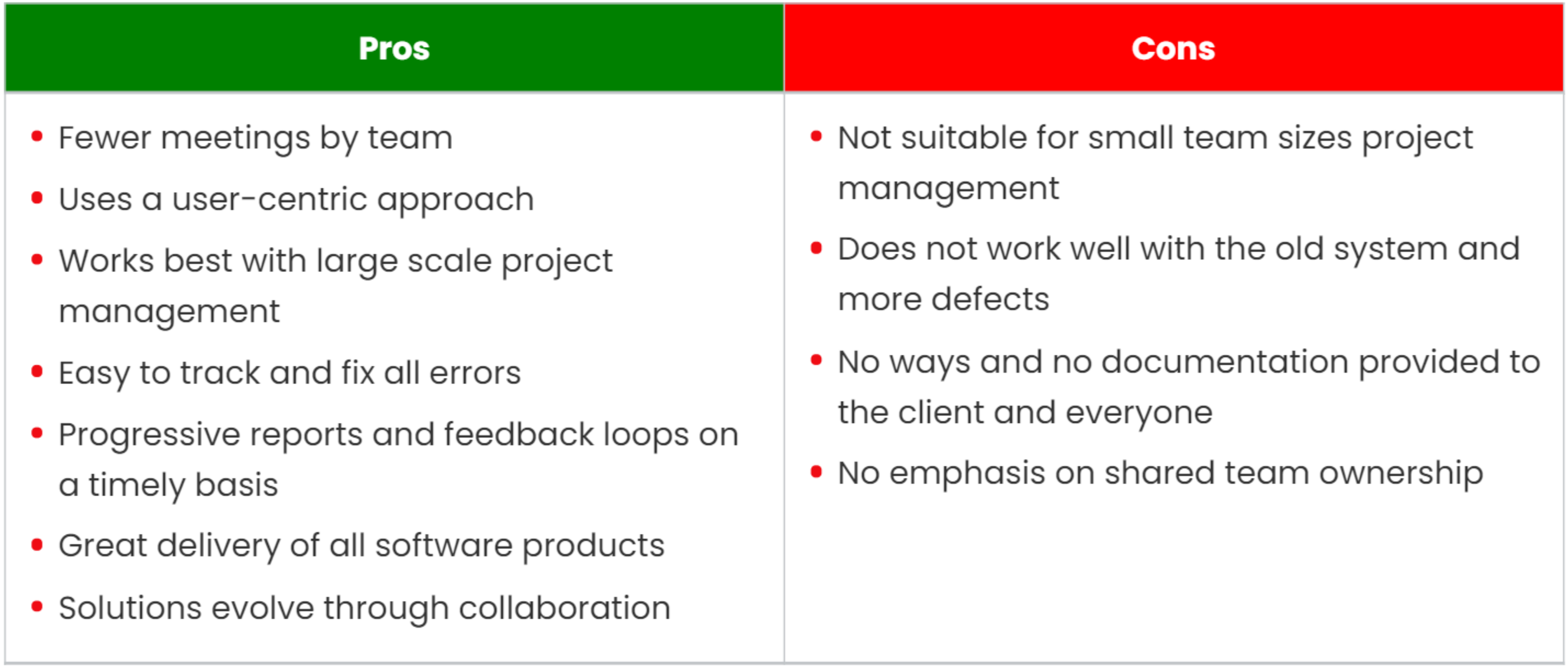
Agile Methods

- ▶Feature Driven Development (FDD)



Agile Methods

▶Feature Driven Development (FDD)



Pros	Cons
• Fewer meetings by team • Uses a user-centric approach • Works best with large scale project management • Easy to track and fix all errors • Progressive reports and feedback loops on a timely basis • Great delivery of all software products • Solutions evolve through collaboration	• Not suitable for small team sizes project management • Does not work well with the old system and more defects • No ways and no documentation provided to the client and everyone • No emphasis on shared team ownership

Issues and Factors

- Scaling agile methods has closely related facets:

1. Scaling up these methods to handle the development of large systems that are too big to be developed by a single small team.

2. Scaling out these methods from specialized development teams to more widespread use in a large company that has many years of software development experience.

Issues and Factors

- When maintenance involves a custom system that must be changed in response to new business requirements, there are three types of problems can arise.

Lack of product documentation

Keeping customers involved

Development team continuity

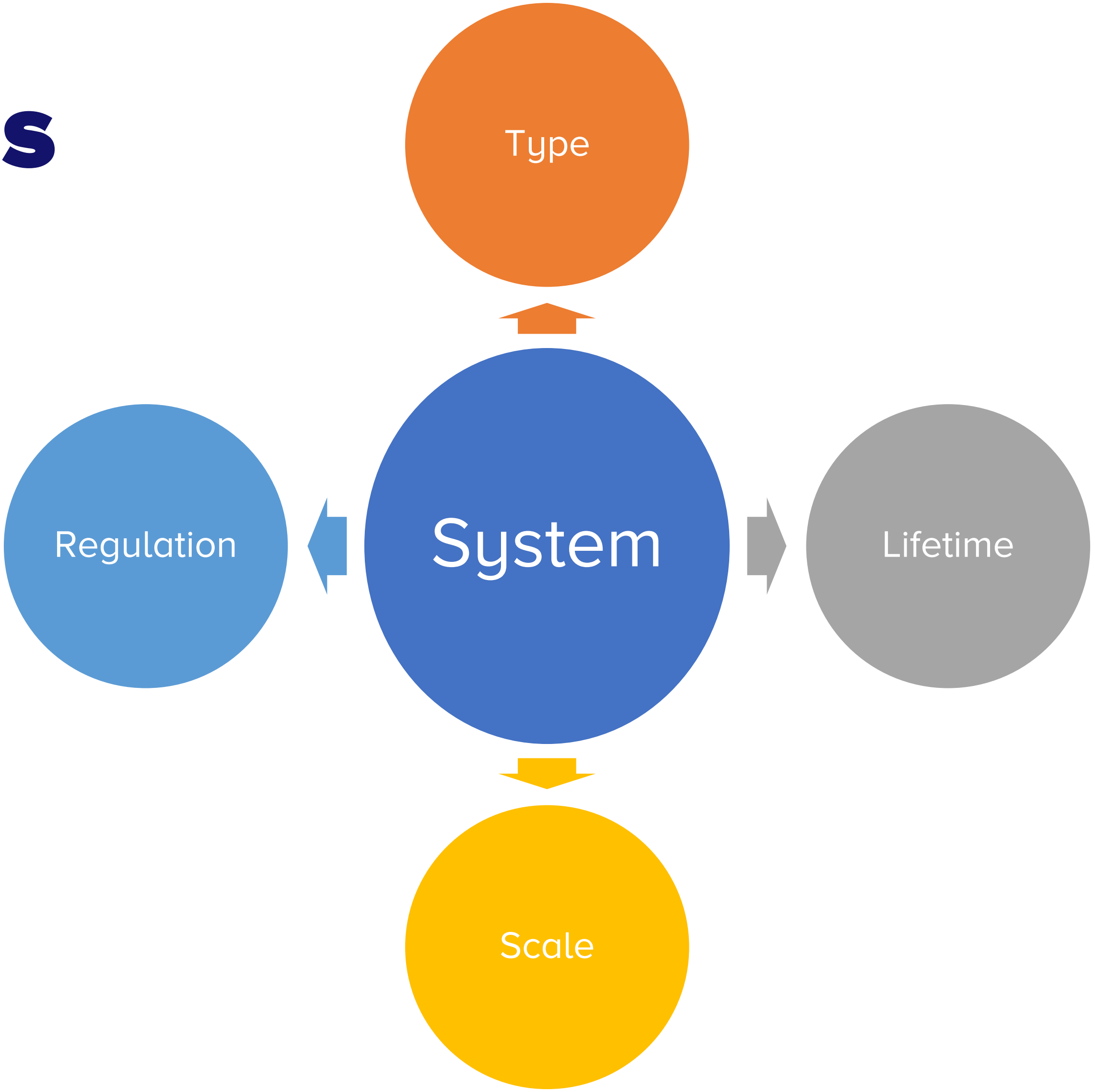
Issues and Factors

- Agile and plan-driven methods

Issues and Factors

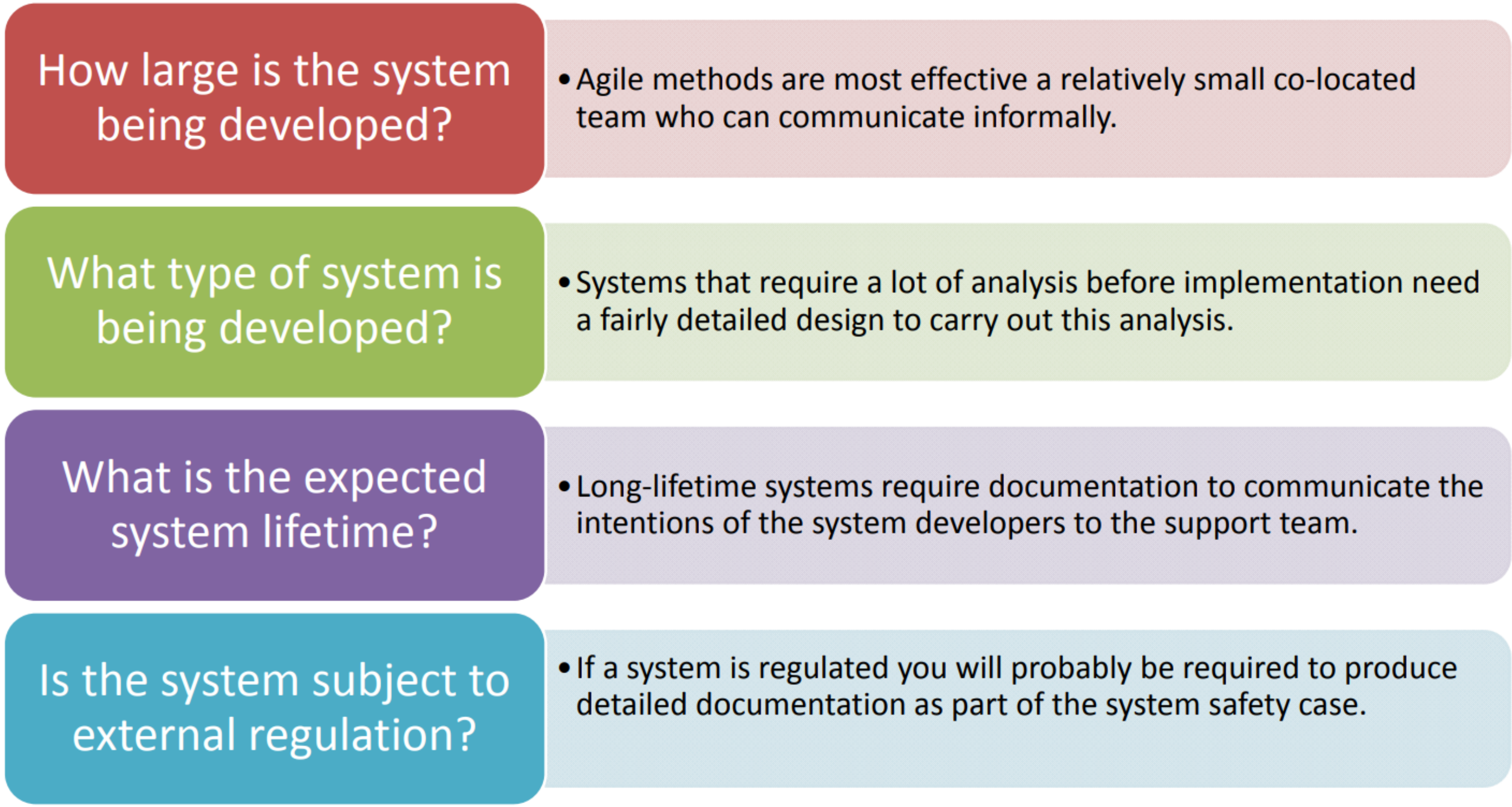
- **System Issues**

Factors influencing the choice of plan-based or agile development



Issues and Factors

▶System Issues



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.

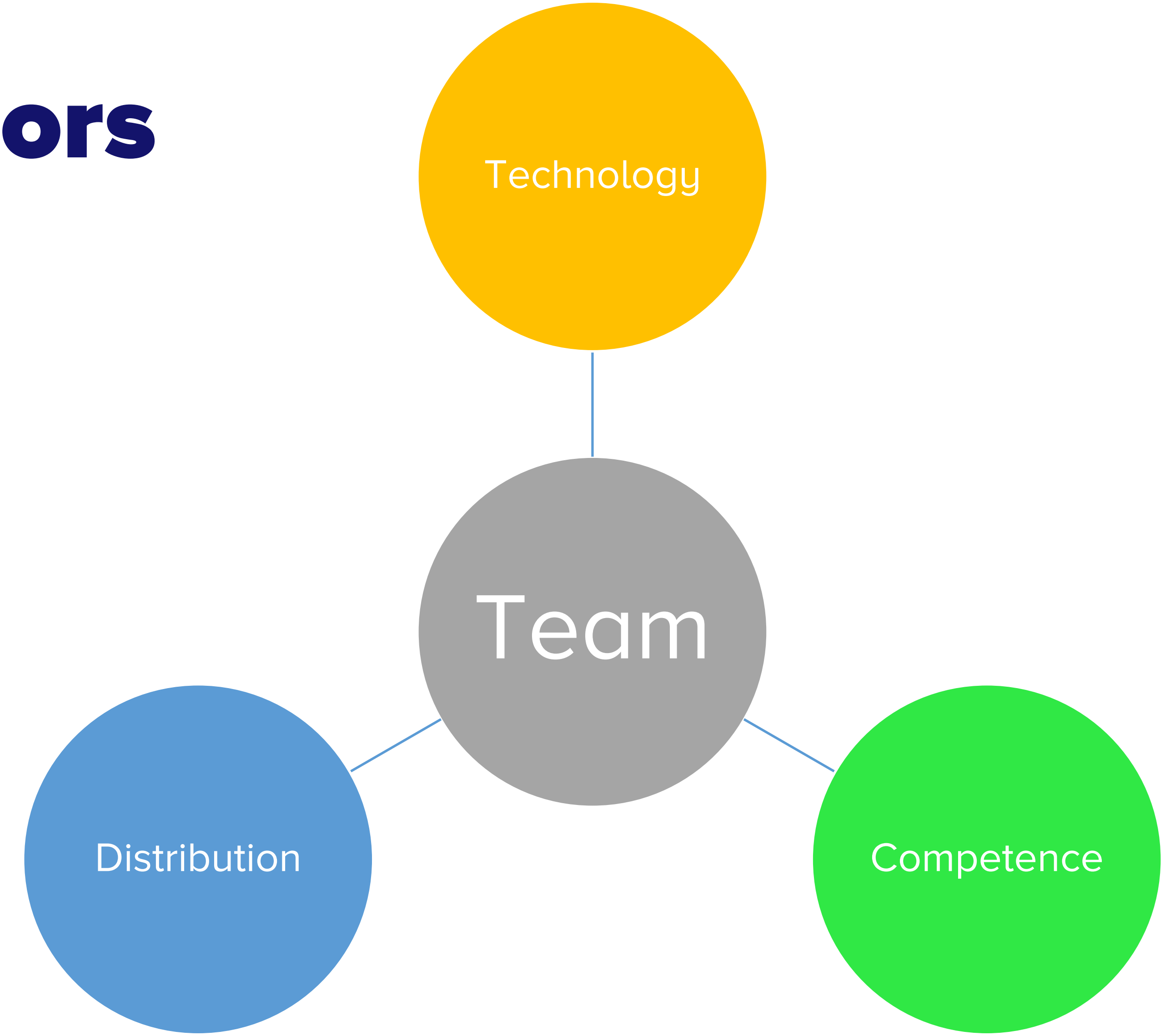
Is the system subject to external regulation?

- If a system is regulated you will probably be required to produce detailed documentation as part of the system safety case.

Issues and Factors

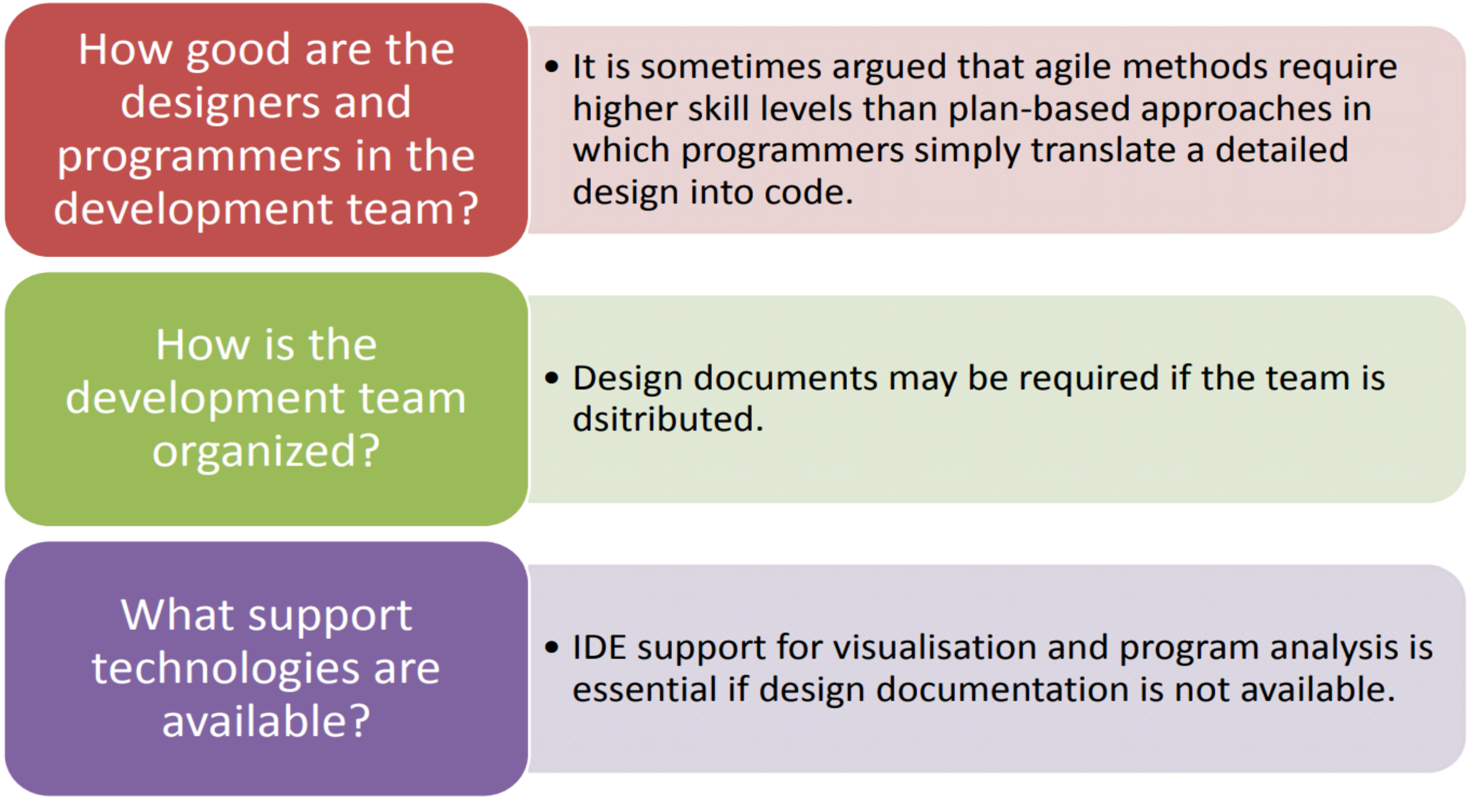
- **People and Teams**

Factors influencing the choice of plan-based or agile development



Issues and Factors

▶People and Teams



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 dsitributed.

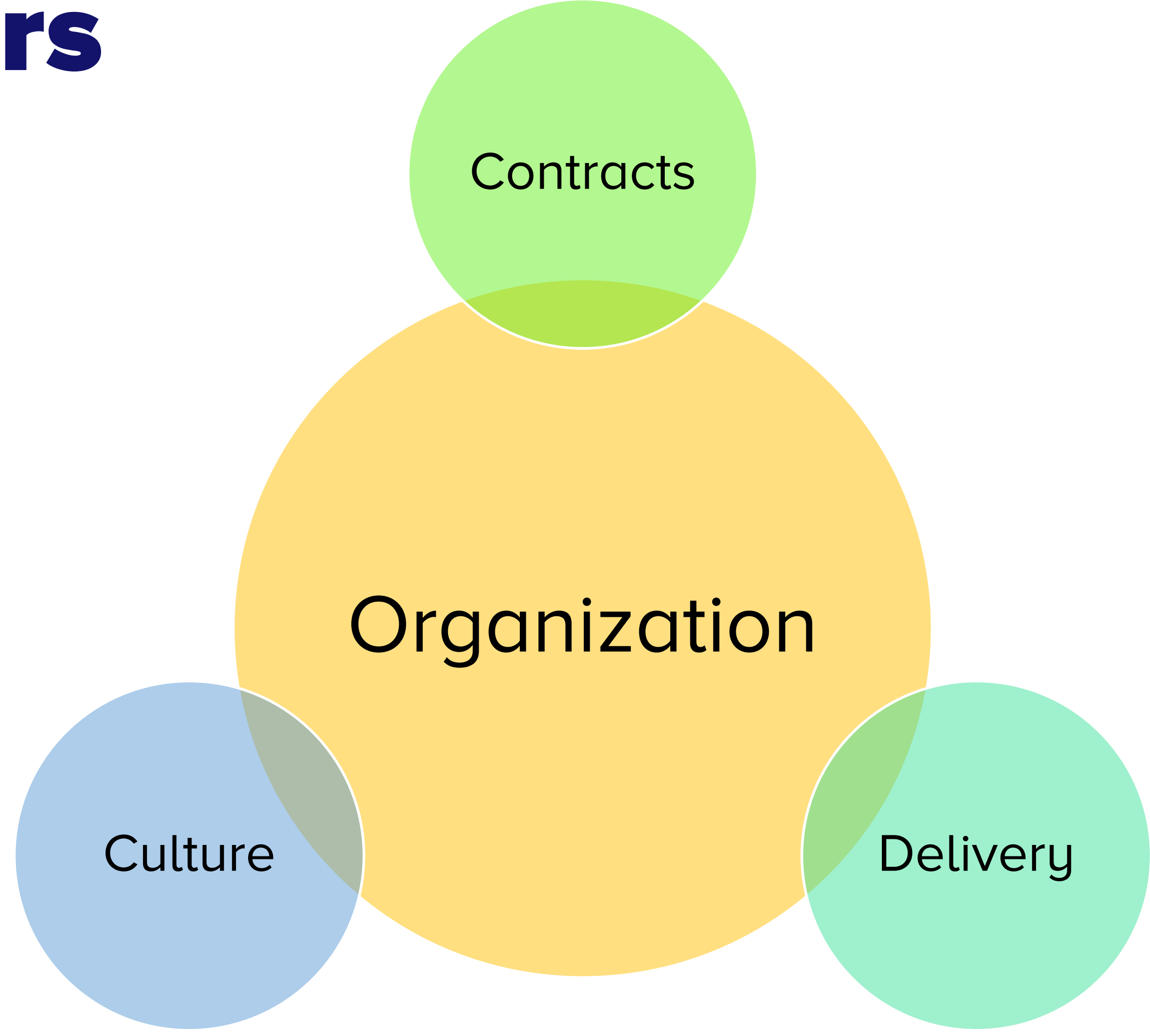
What support technologies are available?

- IDE support for visualisation and program analysis is essential if design documentation is not available.

Issues and Factors

- **Organizational Issues**

Factors influencing the choice of plan-based or agile development



Issues and Factors

- Organizational Issues

Traditional engineering organizations have a culture of plan-based development, as this is the norm in engineering.

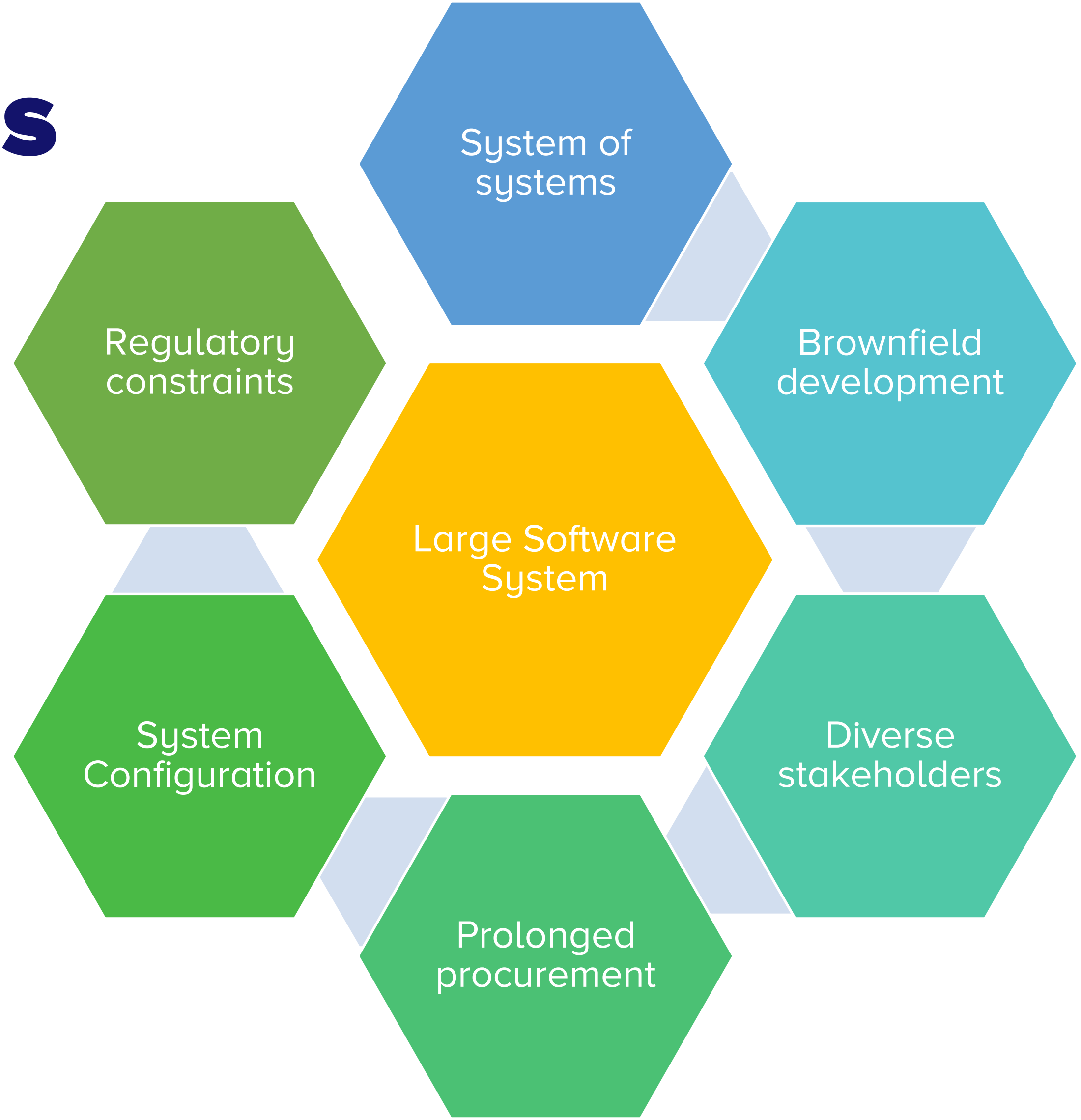
Is it standard organizational practice to develop a detailed system specification?

Will customer representatives be available to provide feedback of system increments?

Can informal agile development fit into the organizational culture of detailed documentation?

Issues and Factors

- ▶Large System Development Issues



Q&A

Lab Exercise 1

Write your answers on A4 paper (doc/docx format), with a font size of 12 pts and any serif fonts (i.e. Times New Roman). There is a 1inch margin on all sides.

Conduct a search for an agile project management and software development tool. Write a review about the tool including functionality and usability. Justify why this tool should be used.

Deadline: **October 1, 2021**

</End>

1927
USTP
Southern
UNIVERSITY OF
SCIENCE AND TECHNOLOGY
OF SOUTHERN PHILIPPINES

COLLEGE OF
INFORMATION
TECHNOLOGY
AND COMPUTING