

10. Agile



a. Agile

- a. The overcoming of traditional methods
- b. Agile
- c. Agile methodologies
- d. Main factors

The overcoming of traditional methods

PROJECT comes from Latin «pro» + «agere» literally «to push beyond» «carry on»

In the old Greek the act of «push beyond», «carry on» is translated into «προβάλλω»* that in the modern language is referring to “PROBLEM”...

Attention: in English we use the term «Design» which often refers to the design itself, so in English both «Design» «Project» are used

And what about «Agile»?



*Provallo

The overcoming of traditional methods

Agile is a Project Management approach that differentiates from the traditional «Waterfall» approach

In the early 90's software development industry faced a **crisis**: developers were frustrated about lead time between the requirements and the solution's development: many projects were cancelled before the release and others were released while requirements was changed

In 2001 a group of developers met in Utah to discuss lightweight development methods about software development and published the «Agile Manifesto»

Agile approach is referring to projects characterized by the use of **short work iterations and incremental development of products**, made possible by focusing on business priorities and customer value

The overcoming of traditional methods

Agile describes an approach to project management specifically created for **software** development, but also valid for **other types of projects**

This approach allows developers and coders to work with greater **autonomy** and an improved focus on quality rather than fulfilling certain function on a team

Agile approach is **iterative** and uses **increments** to ensure quality results

the Manifesto For Agile Software Development drastically changes team dynamics and the approach to project management

The Agile approach redirected the power dynamic inside organization because emphasize the values of the developers (who performs the activities) and team member to meet the customer needs in the shortest time rather than focusing on getting the product complete (as for the wishes of managers, but far from the customer needs)

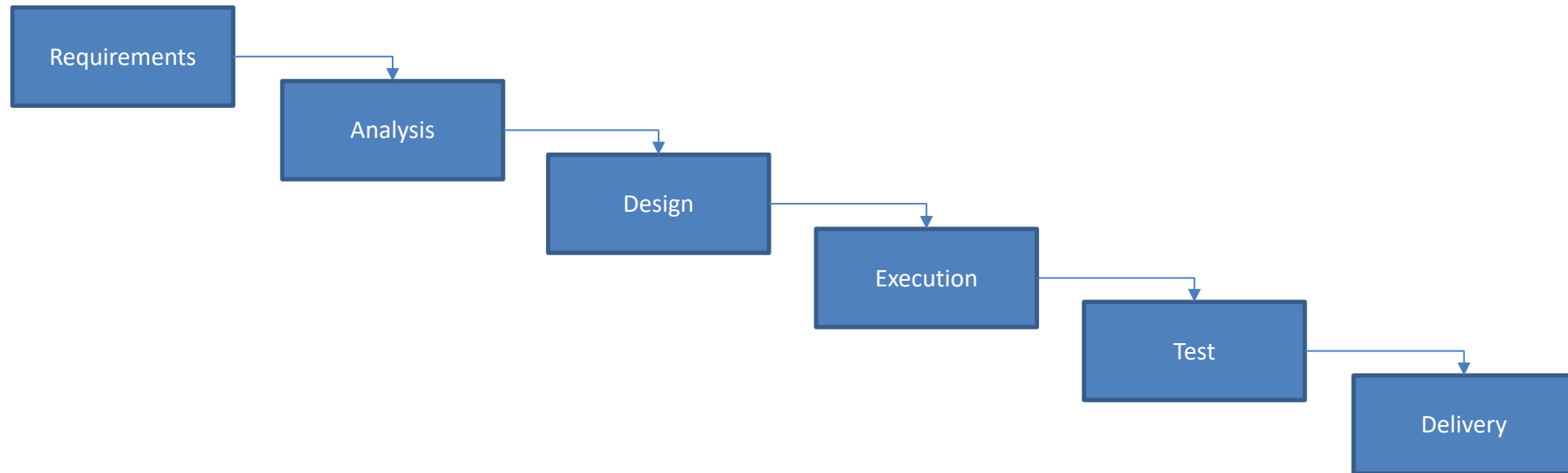
The core values of Agile include **placing individuals and relationships** over the processes an tools

The overcoming of traditional methods

The Agile approach contrasts with heavy project management methodologies (Waterfall and traditional project management approaches)

Waterfall approach

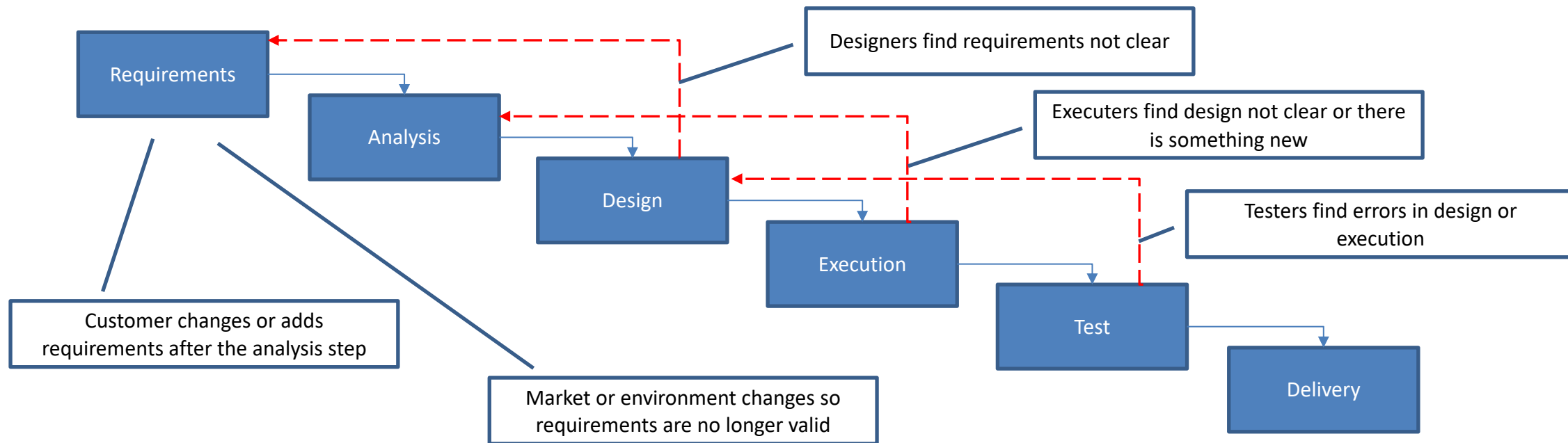
The technical lifecycle of a project **is divided into rigid phases** performed in **sequential mode**. A generic phase begins only when the previous phase is completed



The overcoming of traditional methods

The problem of this method is that in some phases project managers realize that **is necessary to go back to a previous phase** to change something

Example: in the Execution phase it may be necessary to return to the analysis phase to better clarify some aspects of the system, or in case of a modification requested by the customer

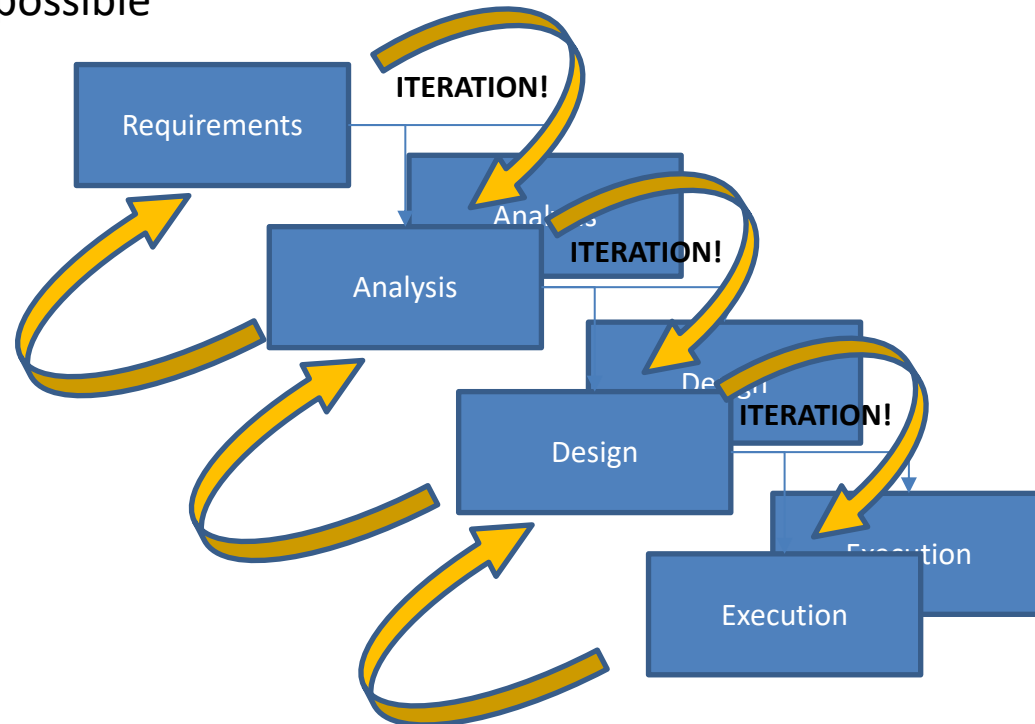


The waterfall approach is good for projects with a **low rate of change**, but is not suitable for fast changing environments and short time to achieve the target

The overcoming of traditional methods

Agile approach shows the same phases as the Waterfall and PM model, but the phases durations are **faster** and **it is not necessary to complete a phase to start the next one** because the phases take place with a **iterative** logic

Example 1: the Analysis phase can start even if the Requirements are not 100% completed. In an iterative way of working, the Analysis phase will be updated with the last requirements when available the shortest time as possible

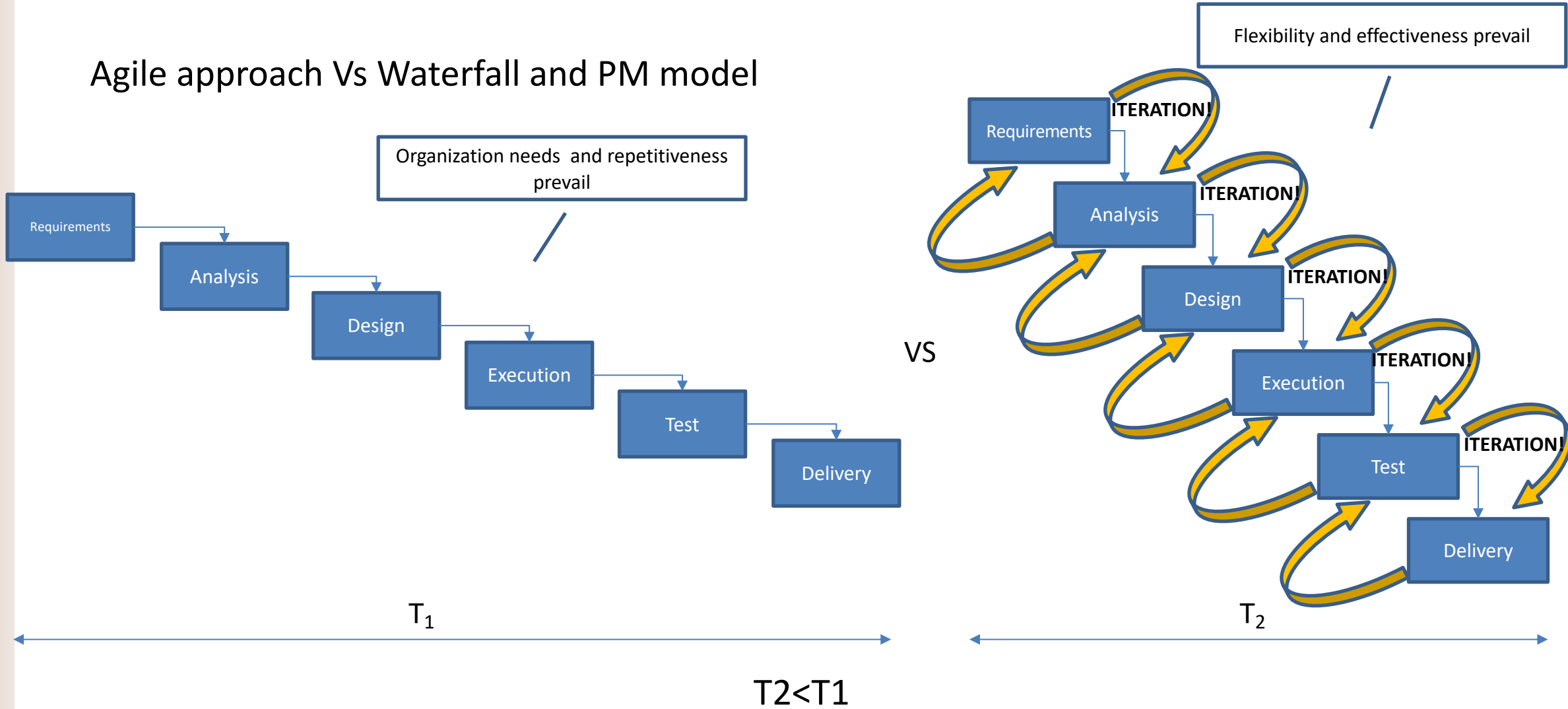


At the same time, the Design activity can start even if the Analysis is not 100% complete... A close coordination between the managers of the two phases and an iterative logic can allow to complete the Analysis phase when the Design phase is already in an advanced state and can be completed with the latest information deriving from the analysis

The same thing can happen for the Execution phase...

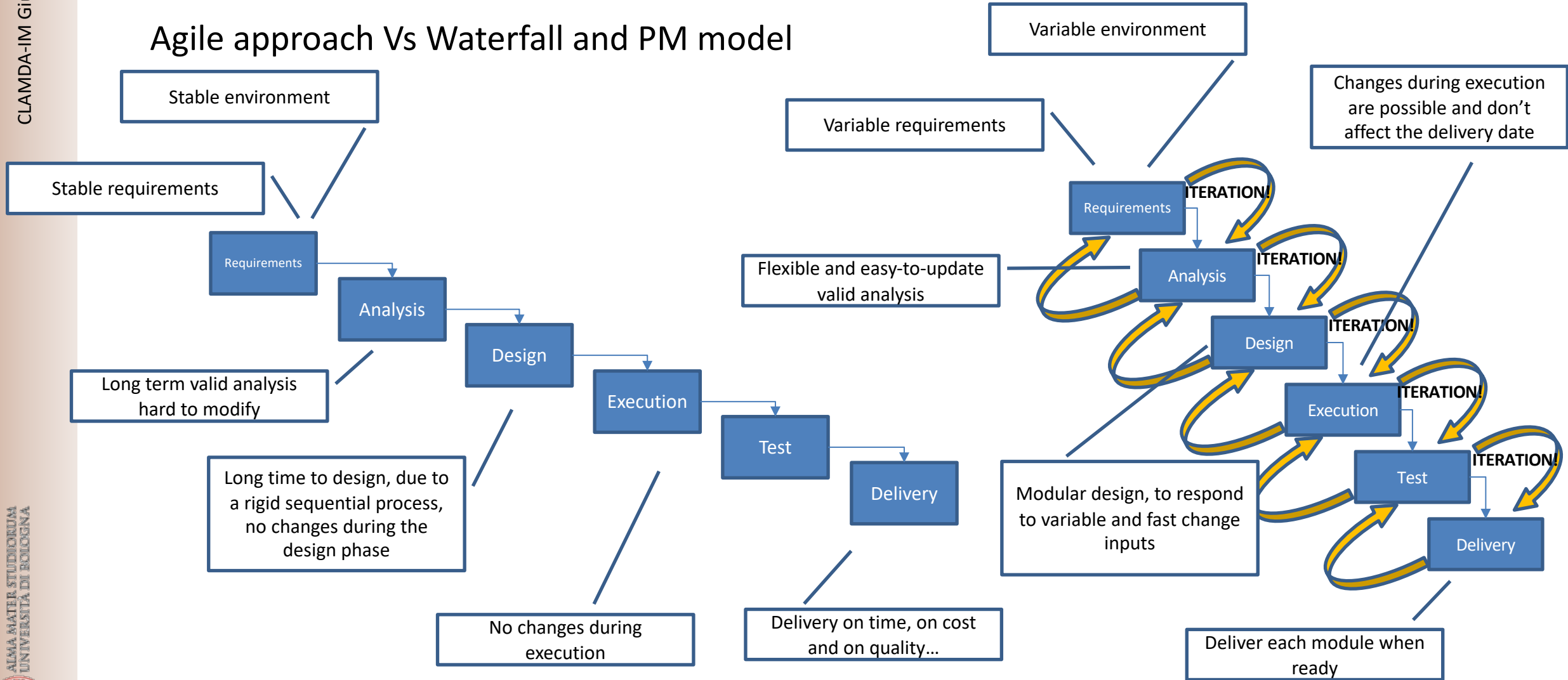
The overcoming of traditional methods

Agile approach Vs Waterfall and PM model



The overcoming of traditional methods

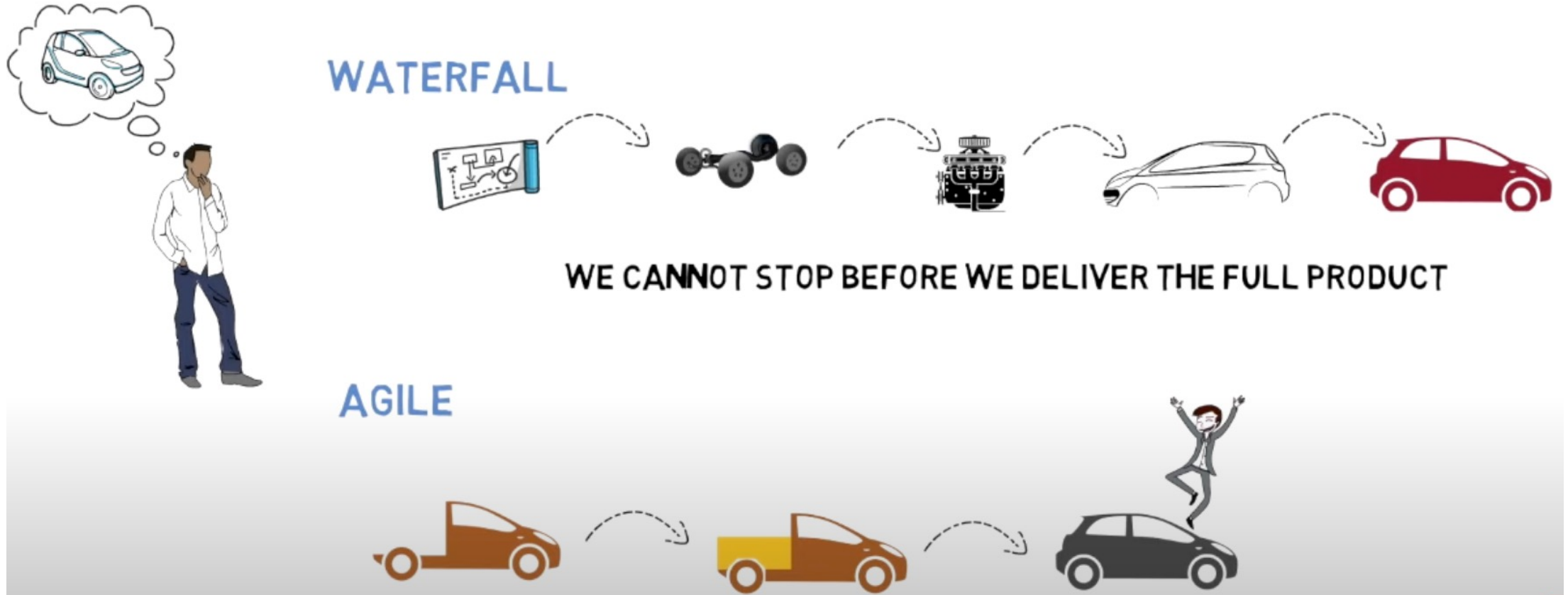
Agile approach Vs Waterfall and PM model



The overcoming of traditional methods

Agile approach Vs Waterfall and PM model: focus on the product

LET'S TAKE AN EXAMPLE....



The overcoming of traditional methods

Waterfall or traditional PM

Project is executed along well-defined stages aiming at completion with expected deliverables to stakeholders

Agile

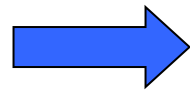
Iterative development methodology that values and encourages **communication** and **feedbacks, fast adjustment to changes** and focuses on **incremental working results** along the way

Project Management

10. Agile

a. Agile

a. The overcoming of traditional methods



b. Agile

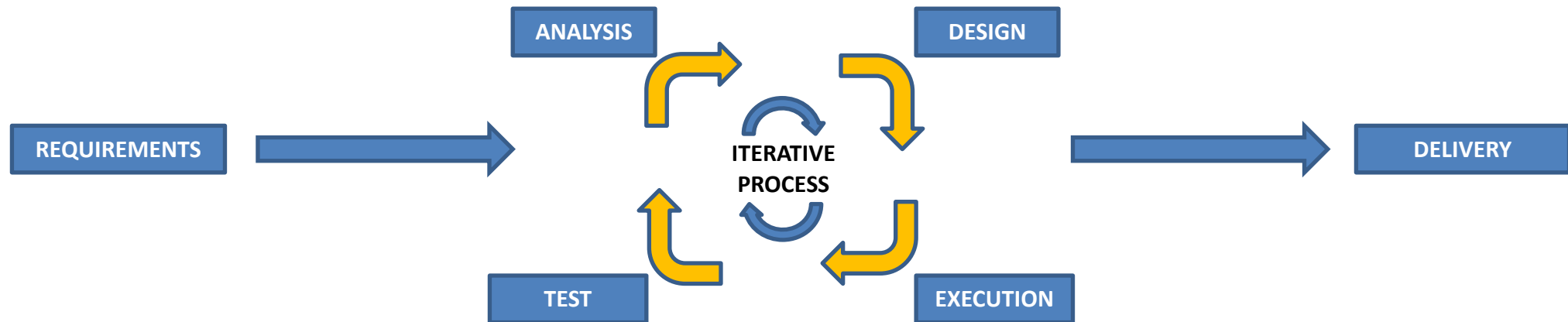
c. Agile methodologies

d. Main factors

Agile

The Agile's main keypoints may be described as:

- Constant effort to achieve the result;
- Constant focus on the team;
- Progressive small and well defined results;
- Building the complete project result (product) through an **iterative process**, rather than a pure sequence of planning-design-development of the project result;



Agile

The Agile method is based on **four declared values**:

1. **Individuals and iterations are more important than formalized processes and tools;**
2. **Working product is preferable to exhaustive documentation;**
3. **Collaboration with the customer is more important than contracts;**
4. **Responding to change rather than executing the plan.**

The method consists in putting **people at the center**, facilitating autonomy, communication and collaboration; **brings the customer to touch the product** that has been requested, creating a context suitable for responding to change

By applying Agile correctly, project times are reduced and delays due to the passage of work between different company departments are overcome

Agile

Value 1: Individuals and interactions

In the past, a lot of software teams would concentrate on having the best possible tools or processes with which to build their software

The Agile suggests that while those things are important, the **people behind the processes are even more so**

Having the right group of individuals on your software team is vital to success

The best possible tools in the wrong hands are worthless

Perhaps even more important is **how these individuals communicate** with each other

The interactions between team members are what helps them to collaborate and solve any problems that arise.

Agile

Value 2: Working software

Previously, software developers would spend ages creating detailed documentation

That was before they even started writing a single line of code

And while documentation isn't a bad thing, there comes a point when you should focus on providing your customers with working software

The Agile Manifesto places shipping software to your customers as one of the highest priorities. You can then **gather feedback to help you improve future releases**

Agile

Value 3: Customer collaboration

Once upon a time, contracts were king. You would draw up contracts with your customers who would then detail the finished product

As a result, there was often a contrast between what the contract said, what the product did, and what the customer actually required

According to the Agile Manifesto, the focus should be on **continuous development**

You need to build a feedback loop with your customers so that you can constantly ensure that your product works for them

Agile

Value 4: Responding to change

Can you imagine a time where you would draw up a roadmap and it would never change?

Well, in the past that's exactly what happened

The trouble with static roadmaps is that we don't live in a static world

Needs and requirements are always shifting, and priorities are always changing

That static roadmap will soon grow outdated

That's why the Agile Manifesto suggests that a software team should have the **ability to pivot and change direction whenever they need to**, with a flexible roadmap that reflects that.

A dynamic roadmap can change from quarter to quarter, sometimes even month to month, and agile teams are able to keep up with those changes.

Agile

The Agile twelve principles

1. **Our highest priority is to satisfy the customer** through early and continuous delivery of valuable software.
2. **Welcome changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.
3. **Deliver working software frequently**, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. **Business people and developers must work together** daily throughout the project.
5. **Build projects around motivated individuals**. Give them the environment and support they need, and trust them to get the job done.

Agile

6. **Face-to-face conversation** is the most efficient and effective method of conveying information to and within a development team
7. **Working software (product)** is the primary measure of progress
8. **Agile processes promote sustainable development.** The sponsors, developers, and users should be able to maintain a constant pace indefinitely
9. **Continuous attention to technical excellence** and good design enhances agility
10. **Simplicity** – the art of maximizing the amount of work not done – is essential
11. **Self-organizing teams** is the best way to produce best architectures, requirements, and designs
12. **At regular intervals, the team reflects on how to become more effective,** then tunes and adjusts its behavior accordingly

Agile

The key points to be addressed in order to achieve effective results are:

1. Replacement of cascade processes and predictive models in favor of iterative and simple ones
2. Elimination of "compartmentalized" departments
3. Evolution of team to components / modules in favor of cross functional teams
4. Increase in product quality
5. Realization of a product that meets the actual needs of the customer

Agile

At the base of the success of Agile there is the **need to increase the effectiveness of response to customer needs**, adapting to the ever shorter times for the realization of the product.

Design autonomy is brought to the highest levels and takes the form of the creation of new teams, made up of people with different skills and roles, who deal with product development.

This means that the traditional division by departments and hierarchies also ceases

Individuals are transformed into a close-knit work team that works to achieve a goal that already has **immediate value for the customer**

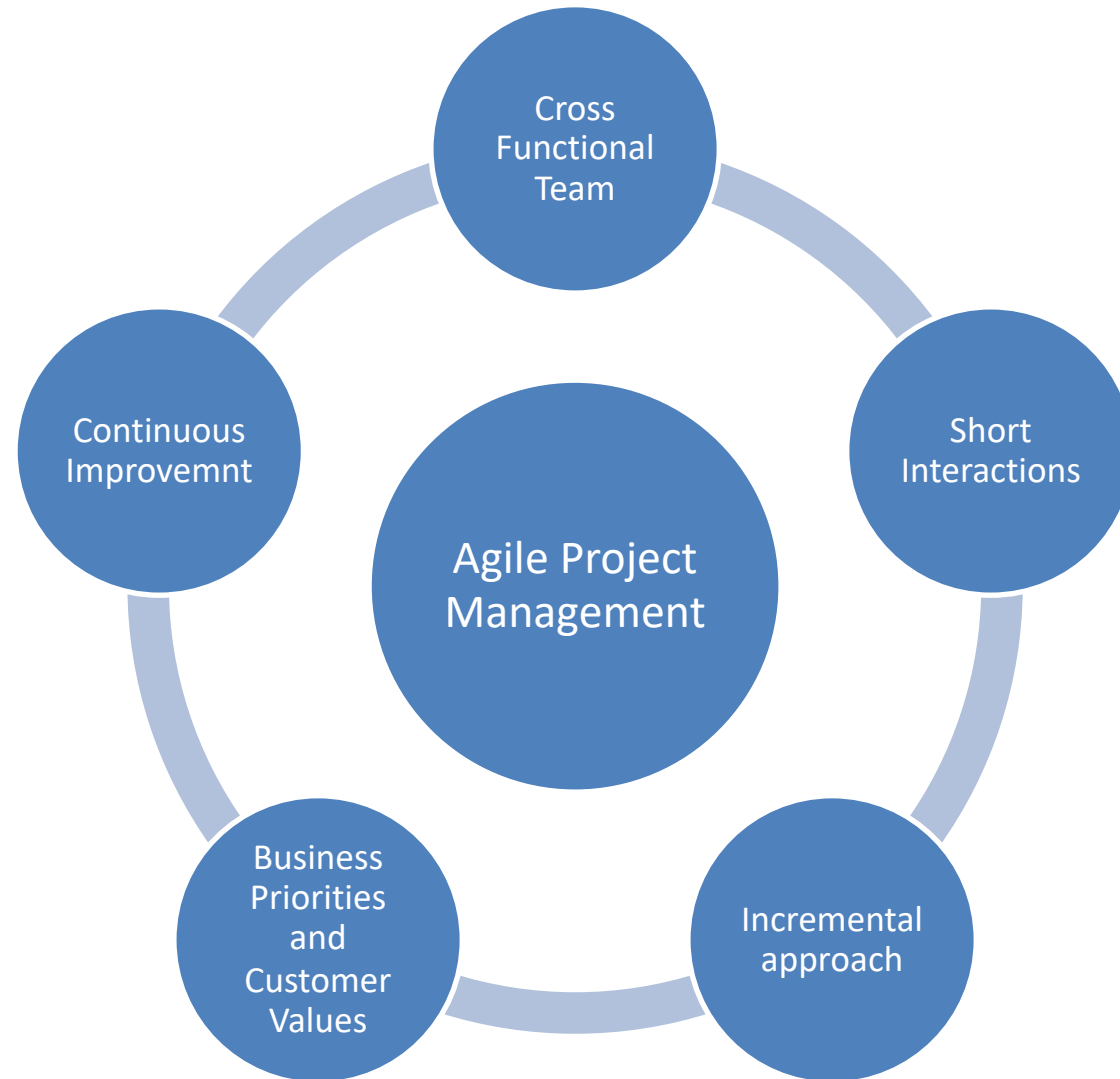
Agile

The activities are conducted through **short-term iterations** through which the team will provide the customer with a **functional and usable product or response**

The customer will then have the opportunity to **concretely verify the result** of their requests and in turn provide feedback, as well as adapt new requests based on experience

In addition to interactions within the team, the **direct iteration between team and customer** becomes fundamental

Agile: Key Characteristics



Agile: Benefits

Reduces Risks:

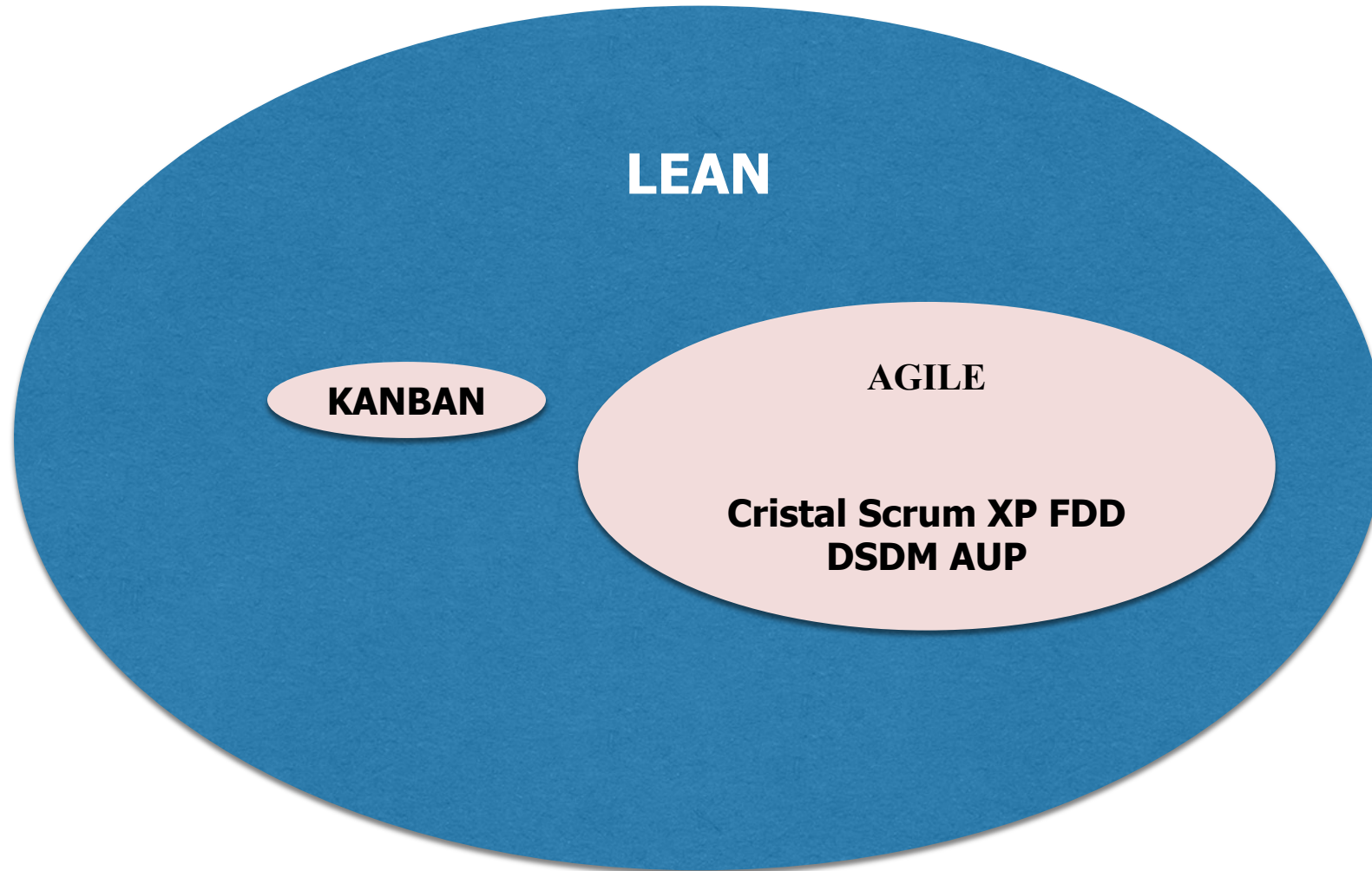
- Developing the wrong thing
- Building poor quality
- Not meeting the budget and schedule → Cost Increase!!!!

Speeds up delivery

Generates Value

Reduces cost of changes

Agile: Context



Agile: is it the end of the Project Plan?

What a project Plan includes? Scope, Cost, Schedule, Activities, Deliverables, Milestones, Resources

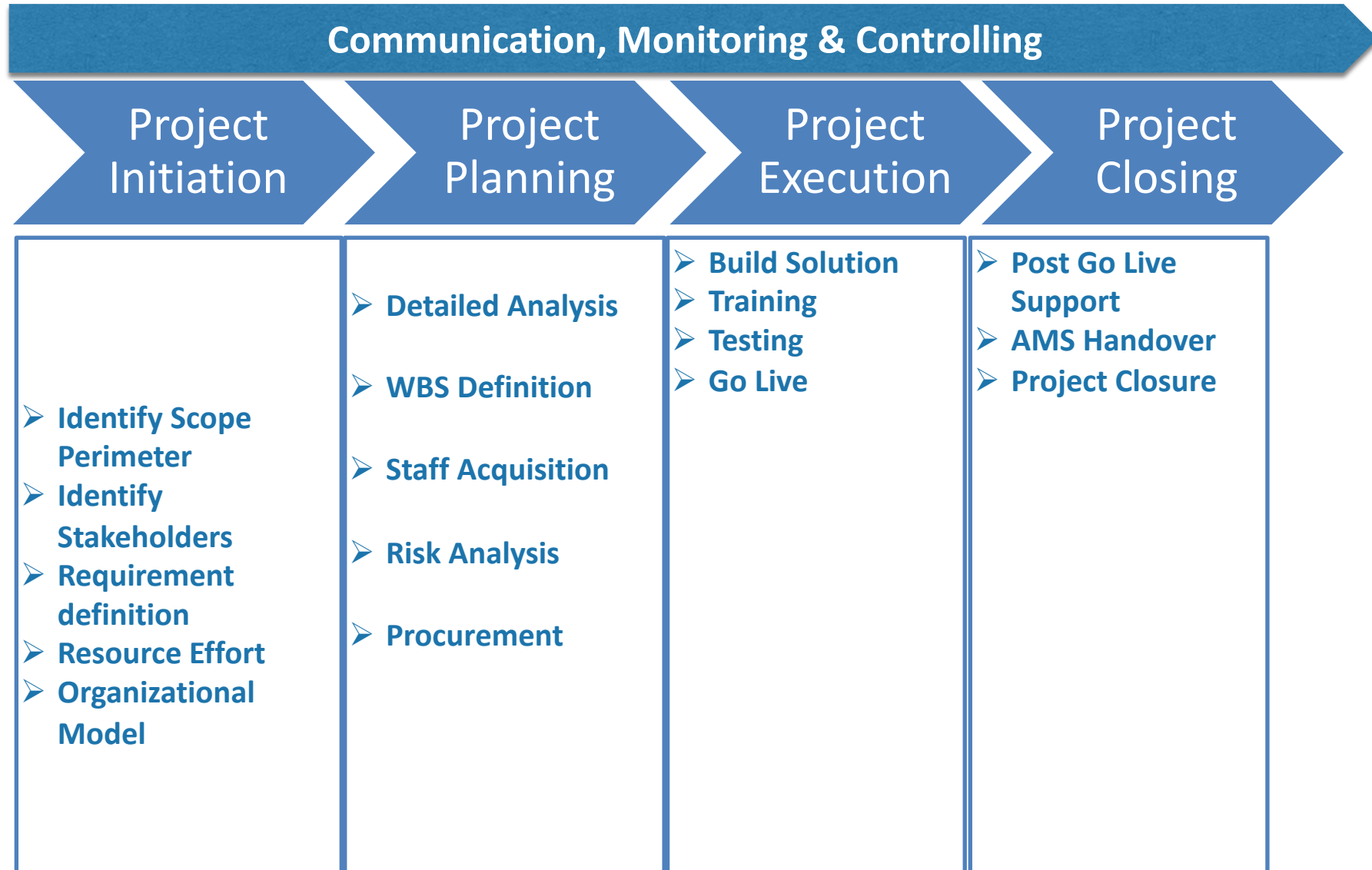
Defined Approach	Empirical approach
<p>A project plan is the document that defines in detail all the specs of the project. It's the most important document as it's used to be approved and review from all the stakeholders.</p> <p>When a change is requested a project leader and a project manager analyze the impacts on the plan and a «Change Request» will be formalized.</p>	<p>The plan is detailed enough only to start the activities and the first interaction.</p> <p>The plan is focused on which project or product should be released without prescribing the detailed activities to accomplish it.</p>

Phases of the Agile



- 1. Envisioning:** determine vision, scope, schedule, team
- 2. Speculating:** Develop Estimate Interactions, Release Project's plan, project's risk mitigation plan
- 3. Exploring:** Implementation and Test features
- 4. Adapting:** review of the interactions and plan the next (not only with the customer but also internally with the team)
- 5. Closing:** release and documentation

Phases of the Agile



Project Team roles

In Agile approach the team should be responsible and proactive to accomplish the requirements.

The approach suggest to have different kinds of team members:

- Designer: is the one to design a solution for each requisite and she/he is the responsible of simplification
- Developer: is the one who produce an output from a design (normally is a team of programmer)
- Tester: is the one who tested the output produced by a developer on a ongoing basis

Key roles: Project Manager, Project Leader, Product Manager

Project Management

10. Agile

a. Agile

a. The overcoming of traditional methods

b. Agile

 c. Agile methodologies

d. Main factors

Differences between Approach and Methodology

“What is the difference between Approach and Methodology?”

- The overall style that guides you when trying to overcome a problem is called **the approach** of solving the problem;
- Approach becomes methodology when it has been time tested and proved its efficacy again and again;
- **Methodology is specific and has a step by step procedure to solve a problem.** On the other hand, **approach is generalized** and tells one how to go about a problem;
- A beginner is helped greatly by methodology while a seasoned person is comfortable with just approach;
- Approach is casual while methodology is organized, scientific, and well researched”.

www.differencebetween.com

Agile methodologies

Agile approach is translated into different methodologies although even if every agile methodologies has different characteristics they all maintain essential agile principles.

The most used agile methodologies are:

- Scrum
- XP (Extreme Programming)
- Lean

Agile methodologies: Scrum

Scrum is widely used in software development project and it's defined as a "Lightweight management Framework". It is based on:

Daily Meeting (about 10-15 min) "Stand Up Meeting" where the team share the status of the development and discuss the interactive phases

- What work have I completed since the last scrum, and why?
- What do I plan on completing between now and the next scrum?
- Do I have any roadblocks or problems that the team can help overcome?

The interactive phase or the time range used to develop work items (or tasks) is defined «SPRINT»

Sprint duration is typically from two weeks to a month (max)

Agile methodologies: Scrum - Backlogs

Product backlog

“A product backlog evolves over the course of a project to list every possible project requirement, including all the features, functions, enhancements, and fixes that must be made to the next iteration. The product backlog is the only document used to list all possible requirements during a project, and because it's constantly updated and reordered, it's considered to be complete.”

Sprint backlog

“A sprint backlog details the work items that team members have agreed to complete in a given fixed-length sprint. Once a sprint starts, no changes should be allowed to interfere with its goal

www.pmi.org

Agile methodologies: Scrum – Terminology 1/2

Time is measured in «**STORY POINTS**»

Time as we know is represented with day, month, year and this is a convention. In agile time is converted in story points: 0, 0.5,1,2,3,100.... But there's no a conversion table!

Everything is decided during the estimation of the real effort. For example: prepare the user manual is about 2days or 16hours that in story point is translated into 20p!

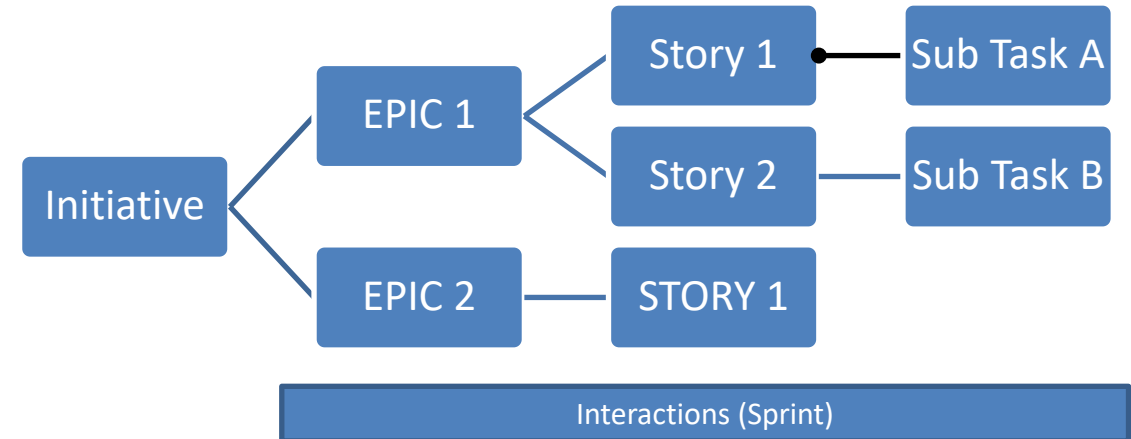
Why this:

- Days months and year don't count how much time we take to read and reply to email (daily activities)
- Dates are defined as «emotional items» and emotion (like anxiety should be removed from the estimation)
- The time used by a team to produce an effective output can be different from another one
- During the estimation of a phase, each team/ role can play at a «poker» game proposing the Story point for each task

Agile methodologies: Scrum – Terminology 2/2

In Scrum Methodology, the project is defined as:

- **Initiative:** normally defines the project (or the requirement)
- **Epic:** large work items broken by different stories
- **User Story/Task:** user stories is the most simple unit of a task and normally are defined as «Person + needs + purpose»
- **Sub Task:** is the small activities to be done in a task



Agile methodologies: Scrum – Example of Backlogs

«Jira» by Atlassian is one of the most used software for Scrum methodology

The screenshot shows the Jira interface with a sidebar on the left containing 'Overview', 'Board', 'Summary', 'Users', and 'Settings'. The main area is divided into 'Features' and 'Tasks'. The 'Tasks' column shows a list of user stories with their IDs, assignees, and status labels like 'Launch', 'Onboarding', and 'Search'.

WS2 - Business Performance Indicators	BPC Masterdata Test
Created by:	SDG
Tested by:	<Name of the tester>
Date of test (Start):	<Name of the tester>
Date of test (End):	<Name of the tester>

ID	Test Description	Object to Test	Tested?	Expected Results	Results 1st Round
1.7	For one Operating Unit, simulate the creation of a new Business Dimension element. Run the Data Collection Book and Verify that the new element is correctly displayed in the book. [Remind to run activities at 3.1 - Procedures Test before check the results]	Data Collection Book Financial Monthly	NO		
1.8	[This test should be managed accordingly to subsidiaries representatives and repeated three times if it's necessary, one time for each subsidiaries]	Data Collection Book Financial Variable	NO		

Jira tool to document User Testing vs the most «traditional» test library (excel...)

Agile methodologies: Scrum – Pros & Cons

PROS	CONS
Planning Framework	Low customer interaction
Team are self- organized and committed from the planning to the development	Lack of Testing methodology
	Less formality

Agile methodologies: XP

XP is another methodology belong the agile approach and it's a «programmer centric» model focused on the delivery of small releases of a software

It's a very disciplined where **a release should be delivered rapidly with intervals that normally are between 30 to 180 days** (and of course in this time range there are interactions).

The result of an interaction is a «Production Ready Code» well tested before the release

The team is composed by programmers (divided in one or more team) that collaborate not only internally but also closely to the customer. The focus is:

- Design and realize a simple solution
- Ongoing testing
- Improve the performance of a code already released

Agile methodologies: XP - Characteristics

Three are the characteristic of XP method:

Pair Programming: programmers are encourage to switch from each other workstation **to check each other's code** to create innovation (that normally is a value), facilitate resolutions, speed up and improve programming and testing

Sustainable Pace: programmers shouldn't work more than 40 hours per week. The more higher is the life balance the higher is the quality of the solution expected

Ongoing Automated Test: before start implementation programmers are encouraged to write down the script test as testing is considered as part of the development process and it reduces the UAT of the final user before the release.

Agile methodologies: XP - Lean

Lean Approach gets principles from manufacturing companies like Toyota and lean principles are incorporated to the project management into agile approach. Lean approach is based on seven principles:

- 1. Eliminate waste:** this is one of the most important principle better known as KANBAN. Everything that is not necessary to the project must be eliminated (for example: documentation of features that maybe will be implemented). On each step there is an evaluation of what will add values and what don't
- 2. Incorporate continuous learning:** a better development will reduce the risk of issues later. Each interaction should verify where there are bugs to be solved before the release

Agile methodologies: XP - Lean

3. **Delay Decision:** Making decisions about software design and code as late as possible reduces the risk that new information – or a change in business needs – will invalidate those decisions.
4. **Deliver software quickly:** planning smaller release to meet the customer needs faster and efficiently
5. **Empower the programming team:** the team is self organized and participates to decisions and monitoring process
6. **Focus on system integrity:** implement a stable and useful software as higher value to the customer
7. **Focus on the whole system:** ongoing testing, integration rather than specialization of a single software component

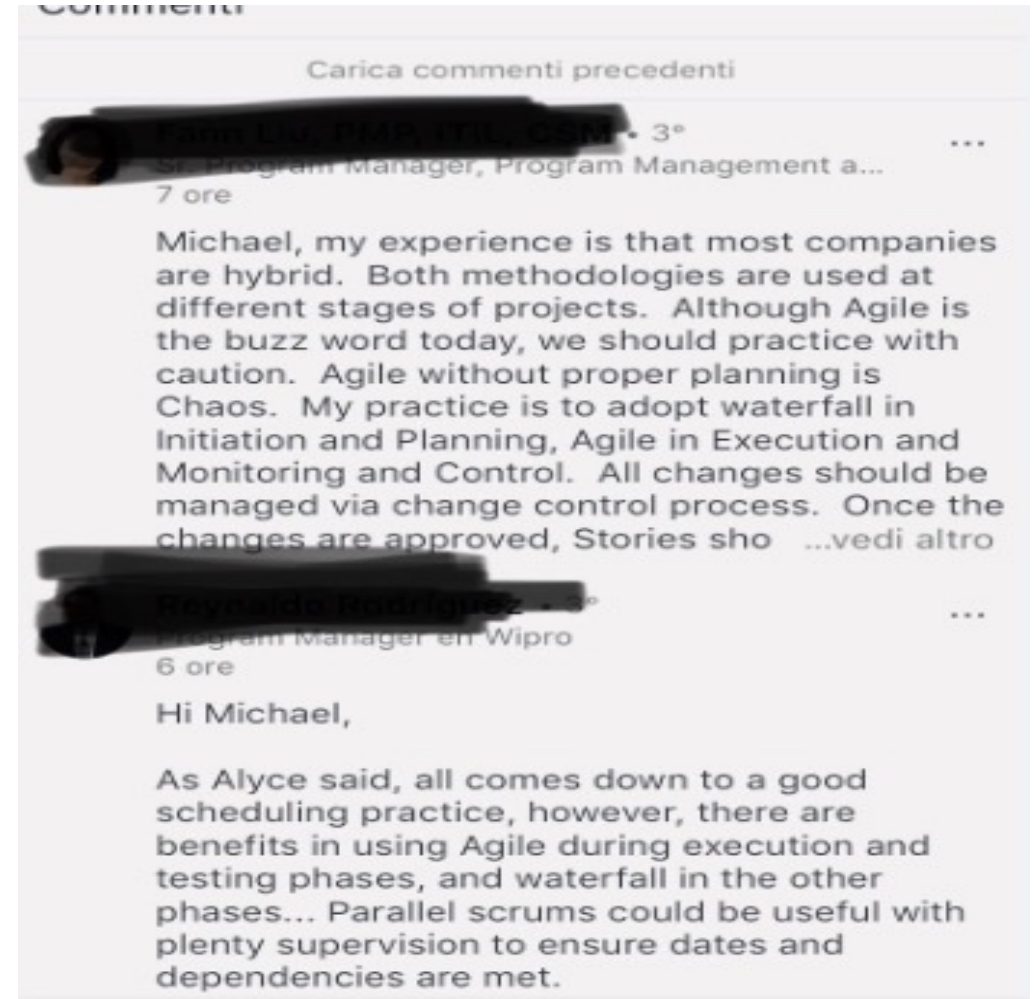
Agile methodologies: Other Methodologies

There are other methodologies that employ the Agile approach like:

- **Crystal** (Clear, Orange, Yellow, Red, Maroon) and each type is referring to the complexity of a project
- **Feature Driven Development (FDD)** is a prescriptive model from development
- **Dynamic Systems Development Method (DSDM)** focused on Business solution rather than a software solution
- **Adaptive Software Development (ASD)**: an approach based on three phases: Speculate, Collaborate, Learn
- **UP (Unified Process), AUP (Agile Unified Process), EssUP (Essential Unified Process), Open Unified Process (Open UP)** are methodologies strictly related to software development

These methodology differentiate from the scope of the project and can be adapted to the specific case

Conversations between PM's



Question to you: Waterfall or Agile

Contract requirement: in case Agile approach is strictly required, customer will take on its own most project risks because scope and business requirements might not be completely clear at project kick-off. Adoption by contract of Waterfall approach moves almost all risks on PM and his/her team (i.e. SDG). Businesses having a mature organizational status do prefer Agile approach.

Risk evaluation: in case of high level of uncertainty (i.e. technology, scope, organizational, ...), Waterfall approach is more suitable.

Type of deliverables: in case deliverables are not well-defined and many interactions and changes might be required, Agile approach should be preferred.

Neither Waterfall nor Agile can be identified as “BEST approach”

In case there are no contract obligations, hybrid approach can be acceptable

Myths about agile

It's not applicable with other methodologies/ Approach: A project management approach is something related from people to people and as people are never black and white also this approach can have different shades! Decide which approach to use for sure is a matter of an analytical evaluation but it's not a drastic policy.

Documentation is not necessary: Documentation is mostly mandatory. What is not mandatory for agile is the production of unnecessary documentation (that sometimes is related to formality). A user manual for example or a design document are necessary documentation (what is not necessary is a detailed «Change request» or a proliferation of excel file often redundant)

Myths about agile

It fits only on software development: Not at all. It can be adapted in all scenario (project re-engineering or simply... a graduation party where in the first sprint you will compiling the list of guests and in the second sprint you will send the invitation)

Projects don't require planning: Planning is part of the Agile but it looks different from the traditional approach. We don't see WBS or GANTT but it's something like a Just in Time planning where people see what is necessary to plan and the estimation of the interactions.

Project Management

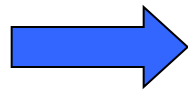
10. Agile

a. Agile

a. The overcoming of traditional methods

b. Agile

c. Agile methodologies



d. Main factors

Main factors for Agile methodology

1. Project TYPE
2. Organizational Structure
3. Organizational Culture
4. Project Management Processes
5. Team Characteristics
6. Industry
7. Customer

Factors: Project Type

Question to you: if you had to build a hotel which approach would you choose?

The Agile approach in a case like this should be very limited because each construction phase should follow a linear way, plus a customer cannot decide (after an interaction) to add 10 rooms more and a swimming pool.....At the same time an Agile approach is not suitable only for software development....

Choosing an agile approach could be the right solutions when:

- The level of uncertainty is high
- The outcome is something completely new
- The project requires creativity and innovation

What we should consider then to adapt agile?

- Criticality: impacts on the organization, costs and risks are too high
- Safety and security requirements: all requirements should come from people that are responsible and the impact of these requirements shouldn't be higher (create a drop down menu in a report is fine, but change technology during the implementation is not...)

Factors: Organization Structure

Agile is not the best solution **when an organization is very hierarchical**, formal and when is resistant to changes

When authority is a high value, giving task and controls are important there's no room to changes and to self-organized teams

Agile can be used for Cooperative organizations, Hybrid (Cooperative – Hierarchical) and it's almost perfect for Project based companies (consultancy companies)

Factors: Organization Culture

- **Trust** [one of the most important specs of agile is that team are SELF ORGANIZED]
- **Openness** [team member should communicate and collaborate with transparency]
- **Responsibility** [as team members are self organized they should be able to deliver an output without any review]
- **Adaptability** [lack of resistances and openness to changes – changes are welcome-]

Are the key factors to help an organization to switch from a Traditional to an Agile approach

Factors: Project Management Processes

An organization that has a PROJECT MANAGEMENT OFFICE can easily **help other business function** to understand if an agile approach can be used.

In structured scenario like this it would be more easier for an organization to understand if agile is the right approach.

But if:

- **Project management processes are very formal**
- **Management requirement are rigid**

Maybe the transition to agile is not the best solution

Factors: TEAM

Characteristic of a project team that can help to agile transition:

- **Skills and experience** not only technical but also personal (collaboration, self-organization, communication are very important)
- **Size of the team:** small team (up to 15) is the best scenario due to the number of interaction with the customer but it's not a strong limitation
- **Team member's location:** as we see during the 12 agile principles people should work together and face to face meeting are encouraged. A co-location is the ideal for an agile transition

Factors: INDUSTRY

Industries that commonly use a lot of procedures are slowly to adapt agile

Communication industries, fashion, web based companies are the most likely scenario to adopt agile as they are opened to rapidly develop technologies

Factors: CUSTOMER

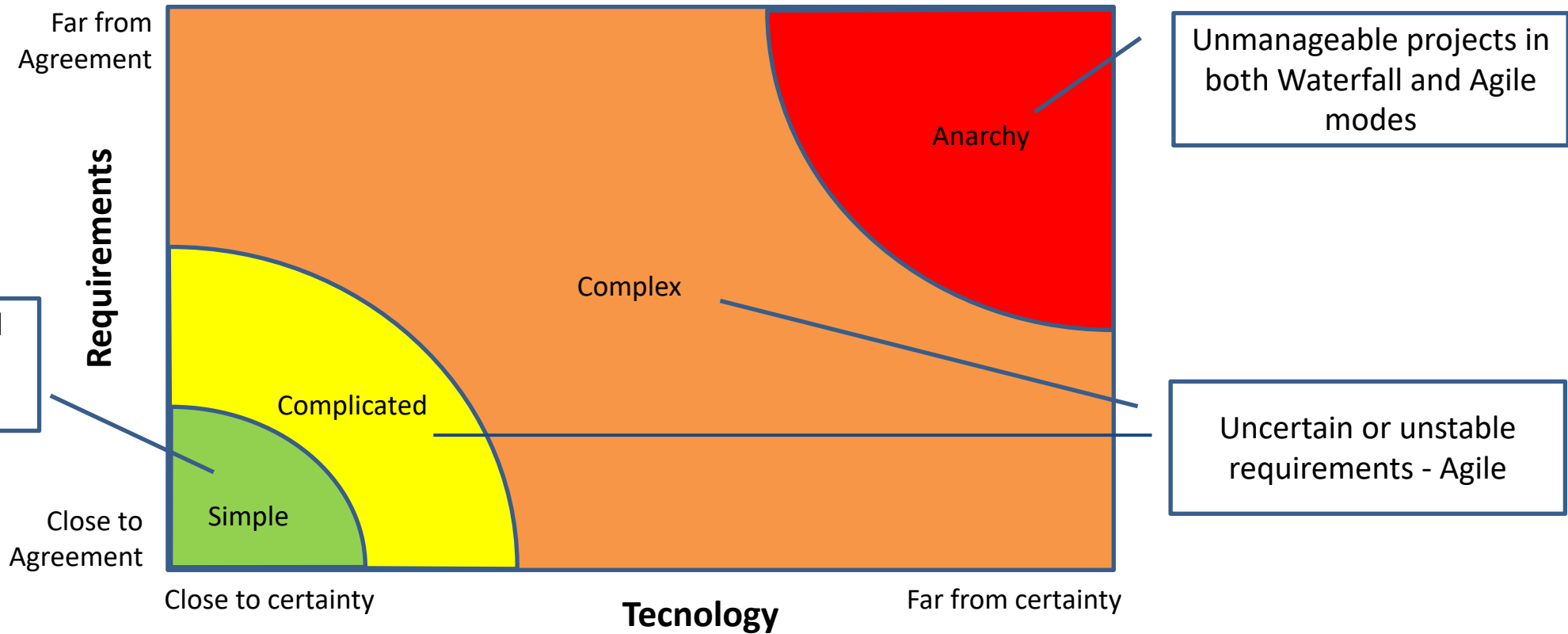
As the number of interaction required **customers should be opened to collaborate** and ideally should be part of the team

Agile requires **huge commitment from the customer side** and if the culture of the customer, internal reasons (sometimes political reasons) and the skills of the people are not so opened then an agile approach cannot be adopted

Another important topic is related to the Budget and the openness from the customer to an agile approach that requires (in terms of cost of the vendor) an **acceptance of the open-ended contract** due to the interaction that can be produced with an agile approach

How to help in the definition of the life cycle

How to define the best project management approach? – The Stacey Model



Conclusion

Independently of the approach that you (or your organization) will use, some factors are fundamental to create values:

Humility, sense of responsibility, vision, commitment, dedication, honesty, ethic and integrity are the most important values that will never make a project fail.

The most important gift that a project can give you is the lessons learned: a project is always a journey from a start to an end and independently that you are working with a linear model or an interactive model you will learn always something.

Dr Valeria Lazzari (my Agile's scout)