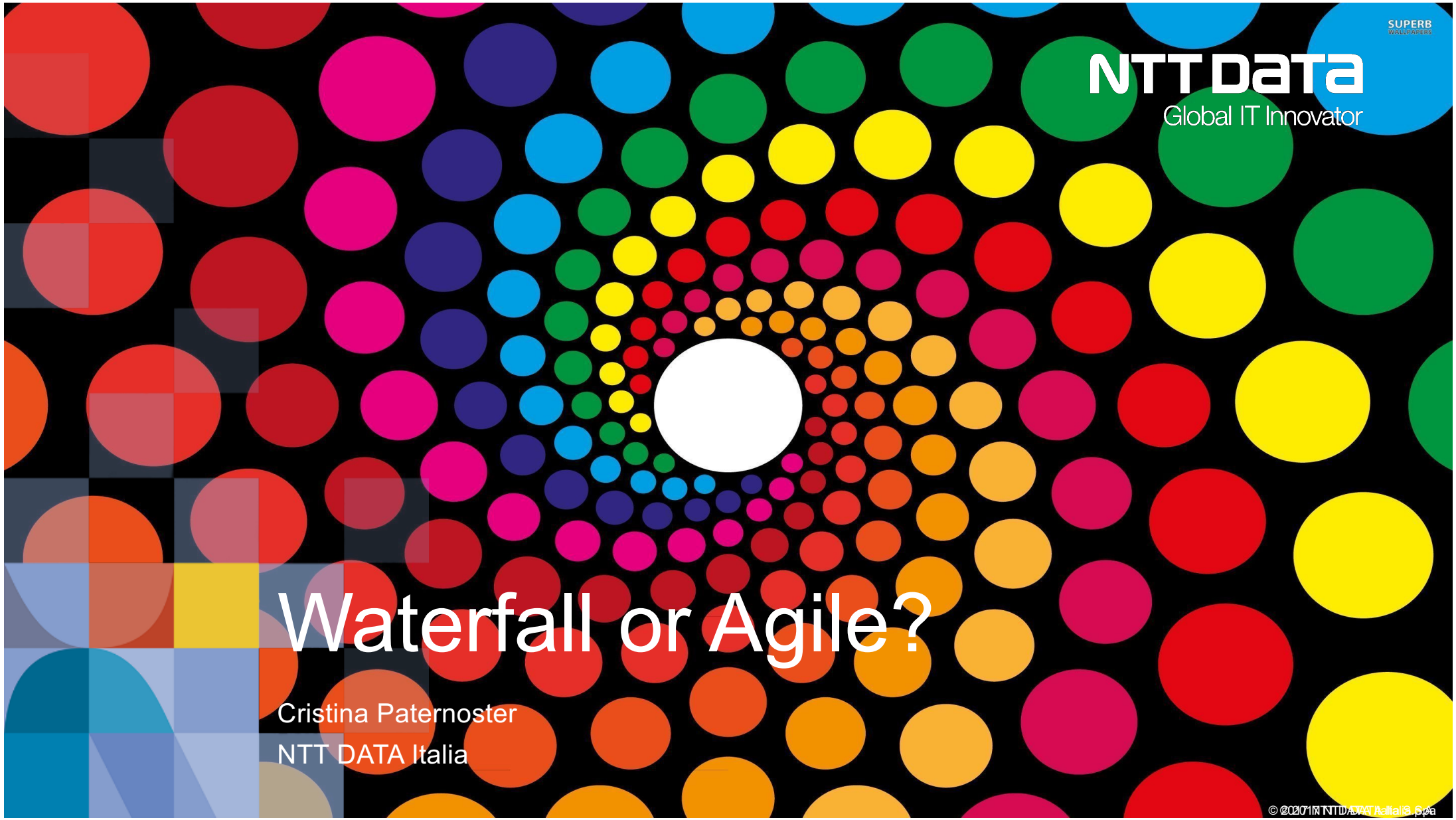
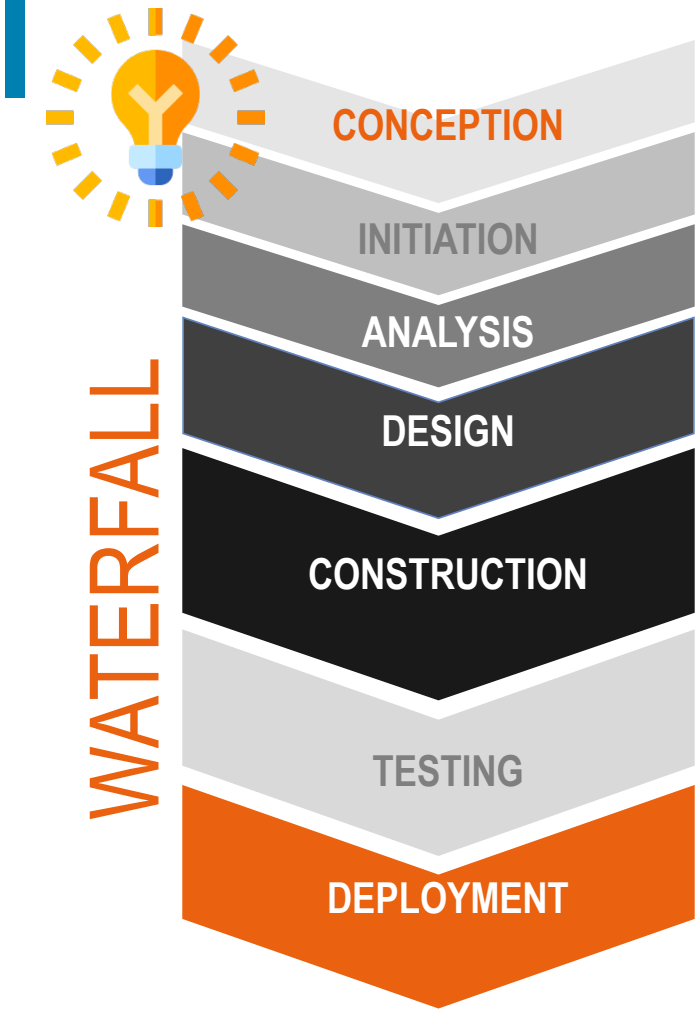

SUPERB
WALLPAPERS
NTT DATA
Global IT Innovator
Waterfall or Agile?
Cristina Paternoster
NTT DATA Italia
© 2021 NTT DATA Italia S.p.a

WATERFALL (TRADITIONAL METHODOLOGY)



Waterfall projects go through a number of SEQUENTIAL OR OVERLAPPING PHASES. This defines the life-cycle of the development effort of the project. The principal difference with Agile is that in Waterfall, REQUIREMENTS ARE DEFINED NEAR THE START OF THE PROJECT AND THEN MAY BE SUBJECT TO CHANGE CONTROL THROUGH ALL FOLLOWING PHASES.

A "Waterfall" type of approach is common on major projects especially in large-scale engineering projects where physical design is a critical activity itself and the implications or cost of changing the design will be very significant.

Source:www.pmis-consulting.com/agile-versus-waterfall/

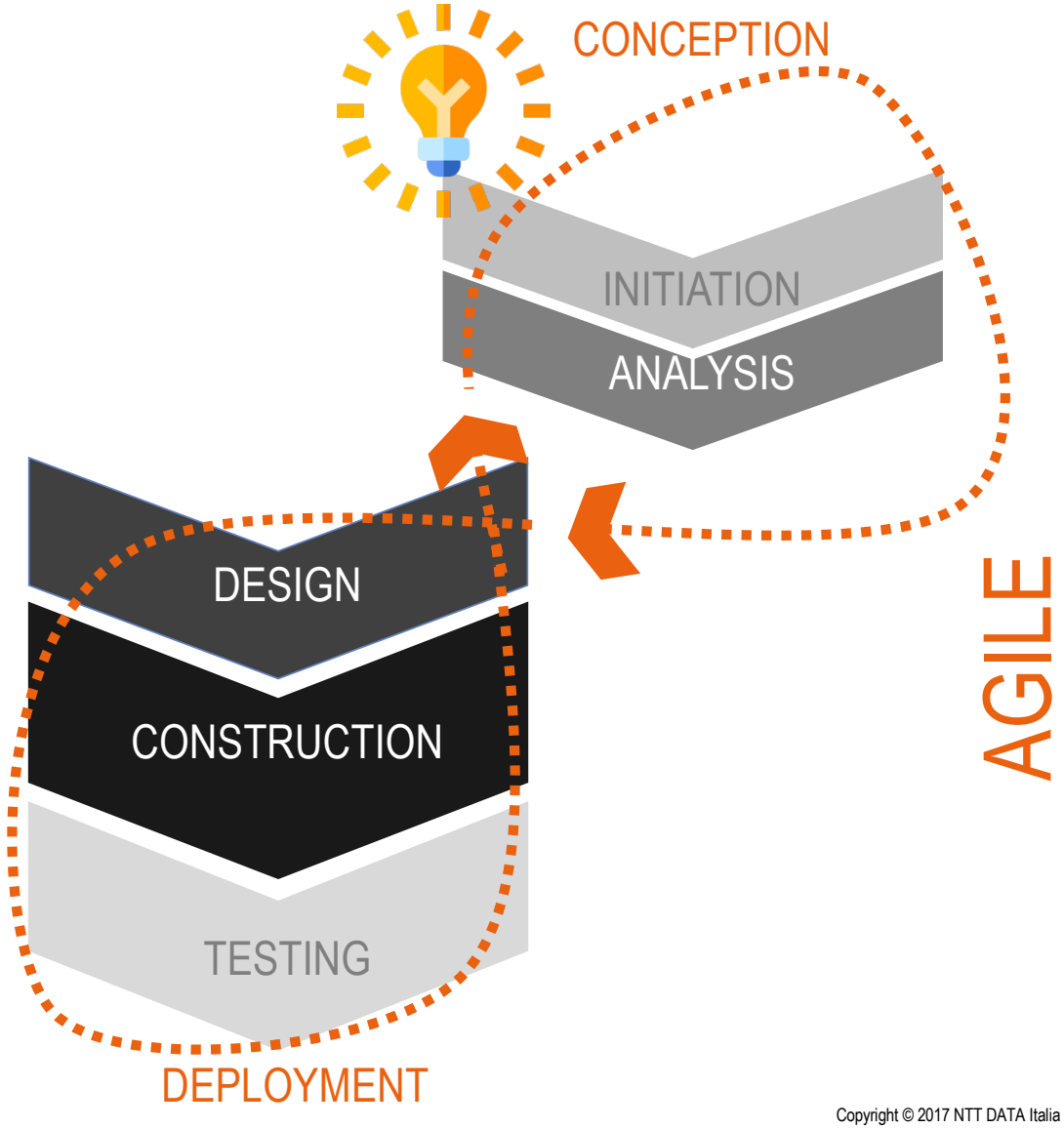
consulting

AGILE METHODOLOGY

AGILE IS NOT A PM METHOD – it is an approach to delivering mainly software projects. The biggest difference between the two is that typically in Agile the DELIVERABLE IS PRODUCED AND ACCEPTED INCREMENTALLY around short iterations.

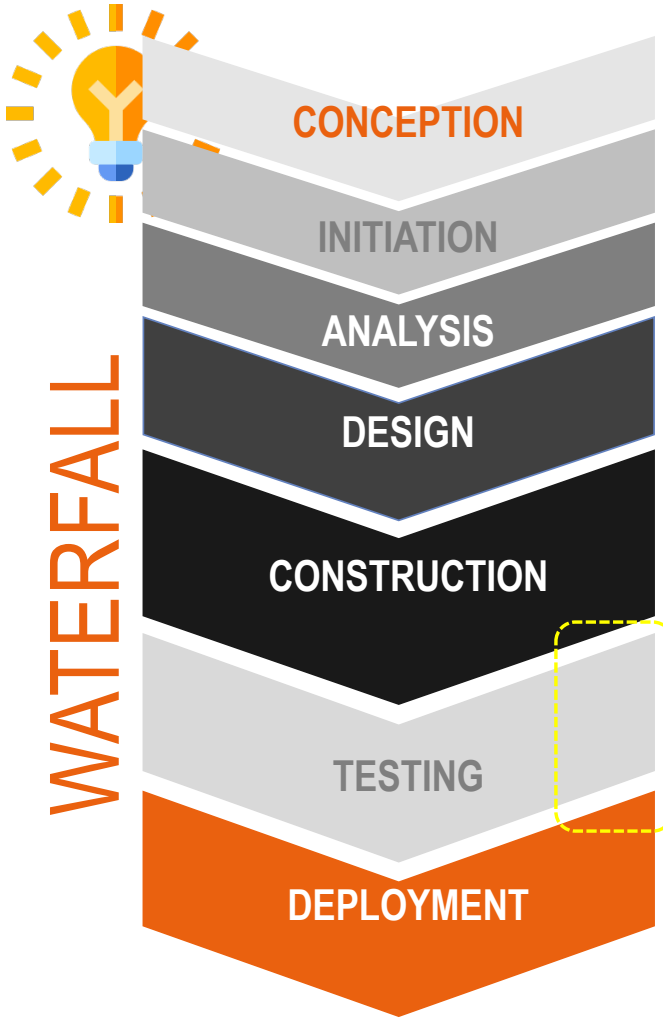
Additionally, REQUIREMENTS ARE NORMALLY DEVELOPED AROUND EACH ITERATION, rather than at the start of the project in a single requirements phase. ONE KEY AIM OF AGILE IS TO RETAIN AS MUCH FLEXIBILITY AS POSSIBLE THROUGHOUT THE DEVELOPMENT CYCLE.

Source:www.pmis-consulting.com/agile-versus-waterfall/



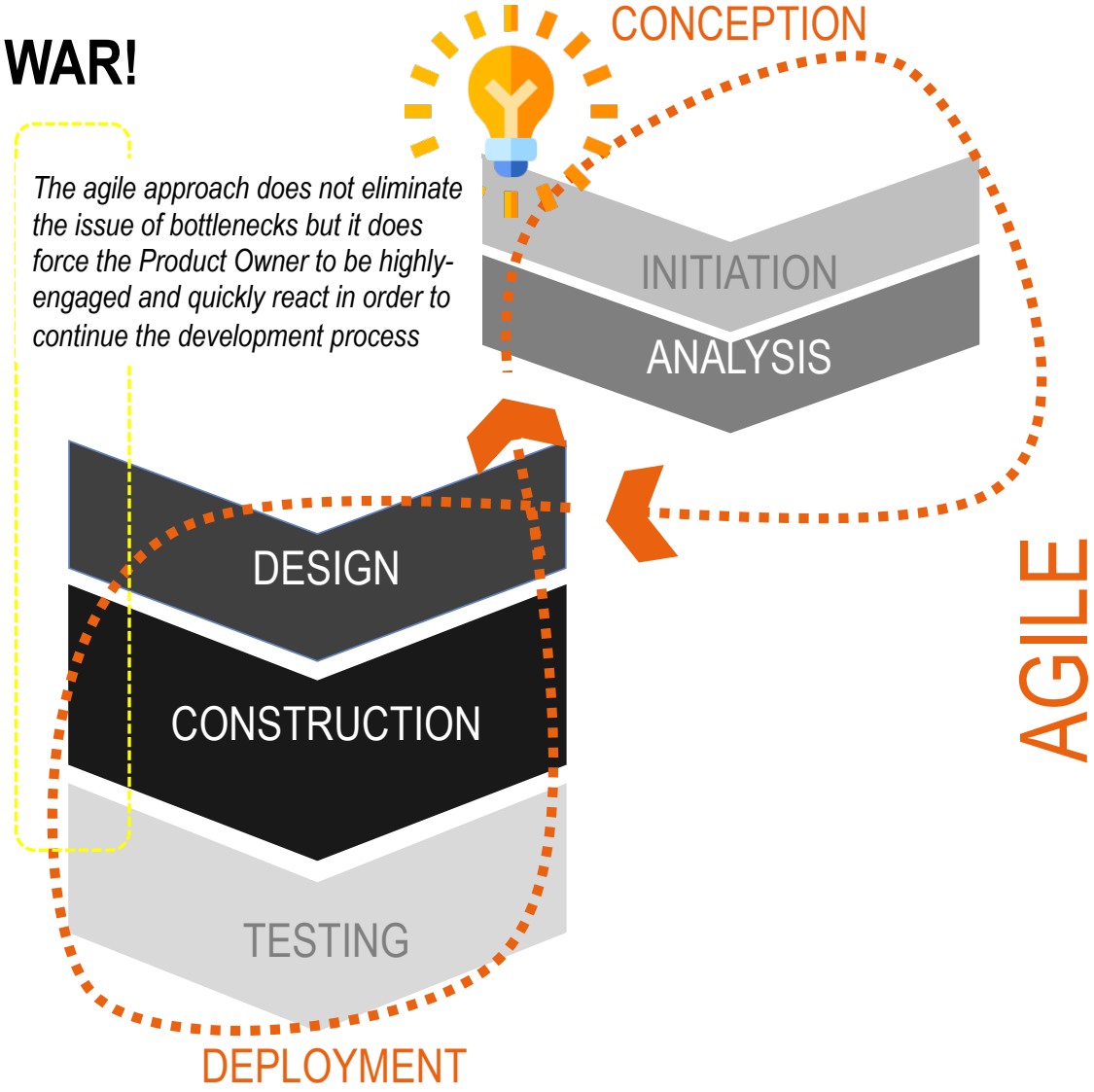
consulting

AGILE VS WATERFALL: IT'S NOT A WAR!



The number of change request may be large and is typically directly related to the length of time since last review

The agile approach does not eliminate the issue of bottlenecks but it does force the Product Owner to be highly-engaged and quickly react in order to continue the development process



consulting

AGILE ENVIRONMENT

The «Agile» environment complexity

The AGILE “world” is very complex in terms of frameworks, methods and practices. And, as a result, it's not always that easy to guide the best decision with the solution that could fully meet your business needs.

AGILE methodologies, considered by many to be the domain of start-ups for a long time, are actually now in use even by large companies.

FRAMEWORK – LARGE SCALE METHODS

Enterprise Scrum
LeSS, *L*arge *S*cale *S*crum
Nexus, Scaled professional Scrum
SAFe, *S*caled *A*gile *F*ramework
Scrum @ Scale
Scrum of Scrum
Setchu, Scrum-based lightweight framework
Xscale

METHODS

ASD, *A*daptive *S*oftware *D*evelopment
Agile Modeling
AUP, *A*gile *U*nified *P*rocess
BADM, *B*usiness *A*nalyst *D*esigner *M*ethod
Crystal Clear Methods
DAD, *D*isciplined *A*gile *D*elivery
DSDM, *D*ynamic *S*ystem *D*evelopment *M*ethod
XP, e*X*treme *P*rogramming
FDD, *F*eature *D*riven *D*evelopment
Kanban
Scrum
Scrumban

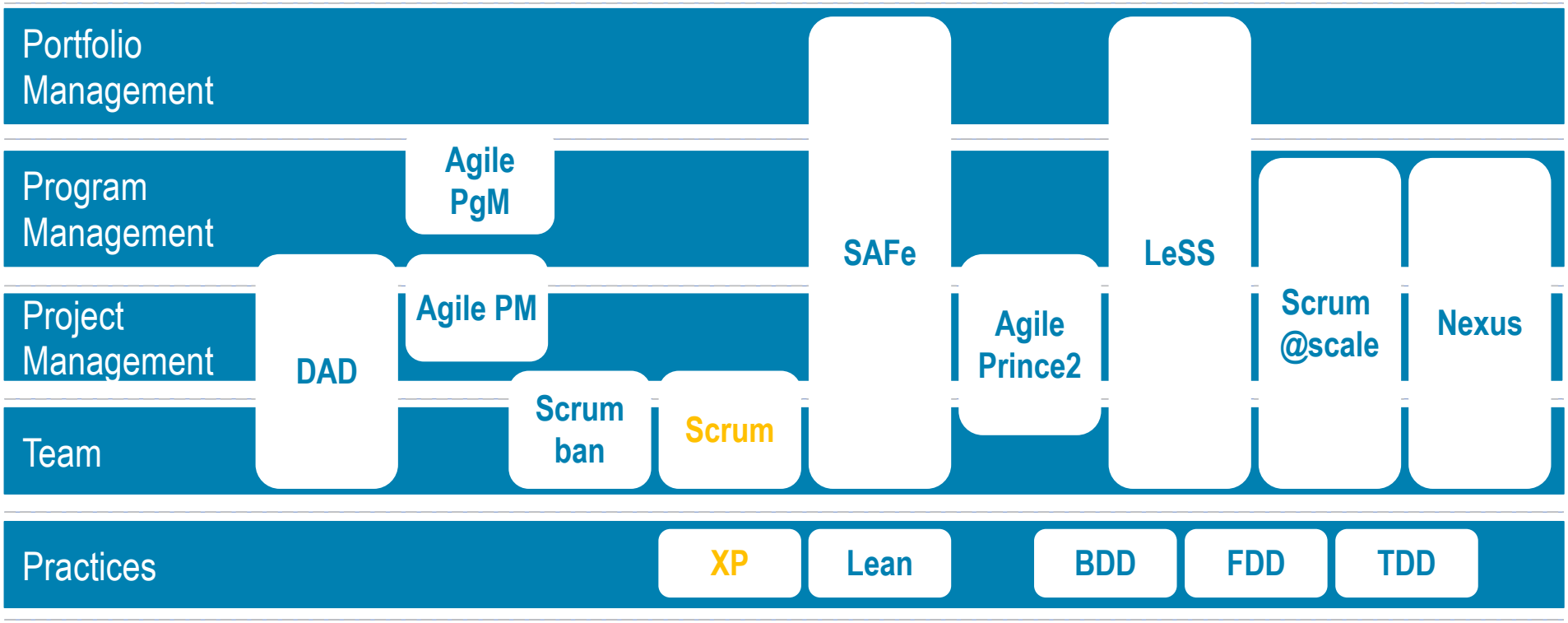
PRACTICES

ATDD, *A*cceptance *T*est *D*riven Development
BDD, *B*ehavior *D*riven *D*evelopment
CD, *C*ontinuous *D*evelopment
CI, *C*ontinuous *I*ntegration
CT, *C*ontinuous *T*esting
DDD, *D*omain *D*riven *D*evelopment
IID, *I*terative and *I*ncremental *D*evelopment
TDD, *T*est *D*riven *D*evelopment

consulting

AGILE BLUEPRINT

In the AGILE world, different practices, methods and frameworks can be used in different contexts

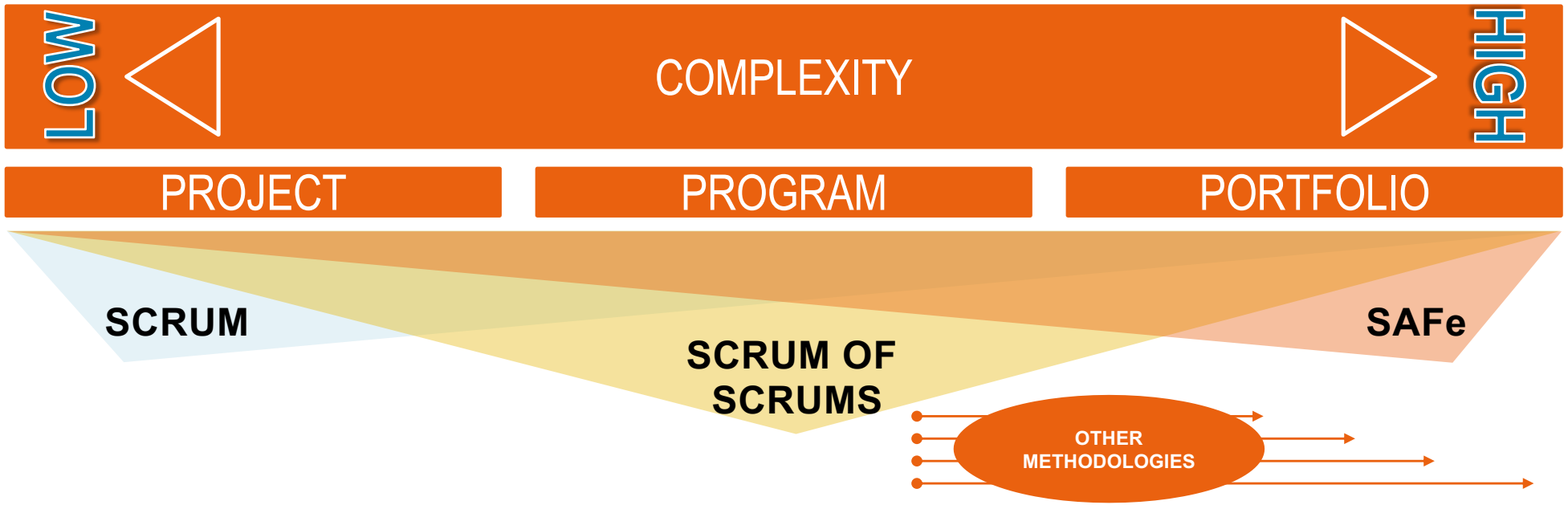


The main methodologies used: in a recent survey it emerged that the most used methodologies are Scrum in 58% of cases and Scrum / XP hybrid in the remaining 10%.

consulting

HOW TO MANAGE COMPLEXITY IN AN AGILE WORLD

AGILE METHODOLOGIES CAN BE ADOPTED, COMBINED AND USED IN COMPLEX PROJECTS. Of course, it's necessary to tailored project, program or the whole porfolio with the righ agile methodologies-framework

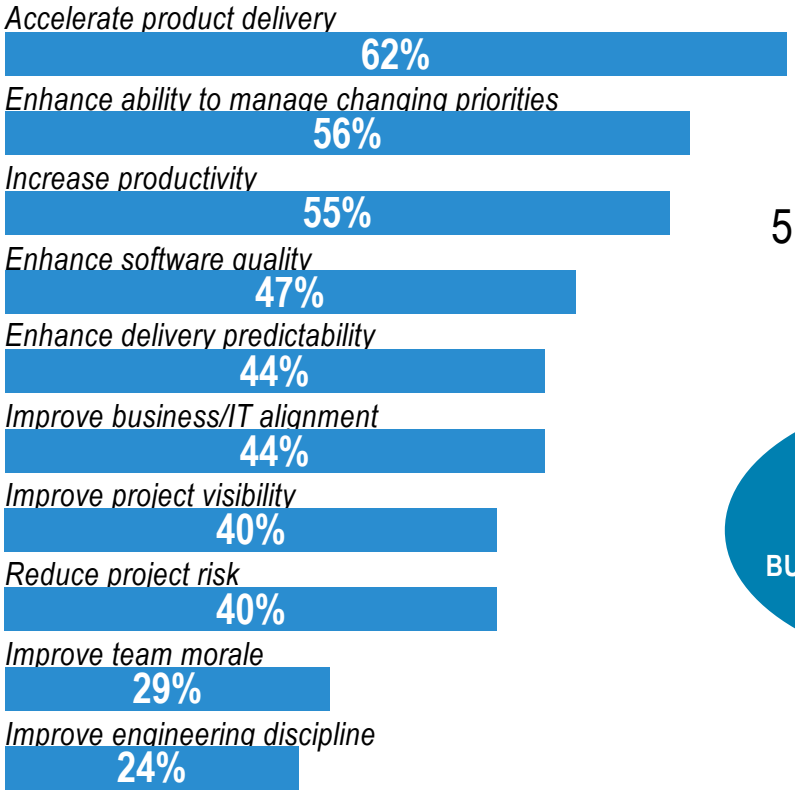


consulting

WHY COMPANIES CHOOS AGILE AND WHAT BARRIERS THEY ENCOUNTER

In Italia ci sono le stesse aspettative e gli stessi problemi nell'uso di una metodologia Agile

WHY CHOOSE AGILE



5 DRIVER FOR AGILE SUCCESS

ON-TIME DELIVERY
PRODUCT QUALITY
CUSTOMER SATISFACTION
BUSINESS VALUE IMPROVEMENT
REQUIREMENTS MEETING

WHY PROJECTS FAILS

LOW COMMITMENT

Source: 10 Annual State of Agile Survey by VersionOne (2016).

consulting


Manifesto for
AGILE SOFTWARE
DEVELOPMENT

THE AGILE MANIFESTO

«We are uncovering better way of developing software by doing it and helping other do it»

CUSTOMER COLLABORATION over contract negotiation	**INDIVIDUALS AND** INTERACTIONS over processes and tools
RESPONDING TO CHANGE over following a plan	**WORKING** SOFTWARE over full documentation

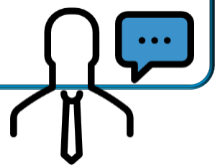
Il termine **OVER** potrebbe essere frainteso: non è da sottovalutare infatti l'importanza della negoziazione, processi, piani e documentazione che nei framework AGILI sono in ogni caso presenti, ma assumono connotazioni proprie.

consulting

VEDIAMO INSIEME I PRINCIPI ESPRESSI DALL'AGILE MANIFESTO

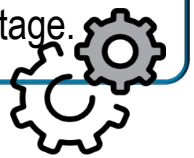
1

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



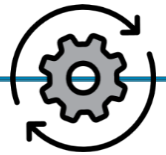
2

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.

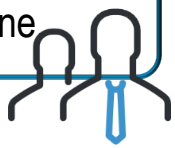


4

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

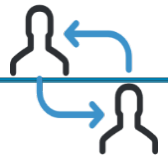
5

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



6

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



consulting

VEDIAMO INSIEME I PRINCIPI ESPRESSI DALL'AGILE MANIFESTO

7

Working software is the primary measure of progress.

8

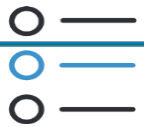
Agile processes promote sustainable development. The sponsor, developers, and users should be able to **maintain a constant pace indefinitely.**

9

Continuous attention to technical excellence and good design enhances agility

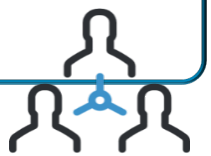
10

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



11

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



12

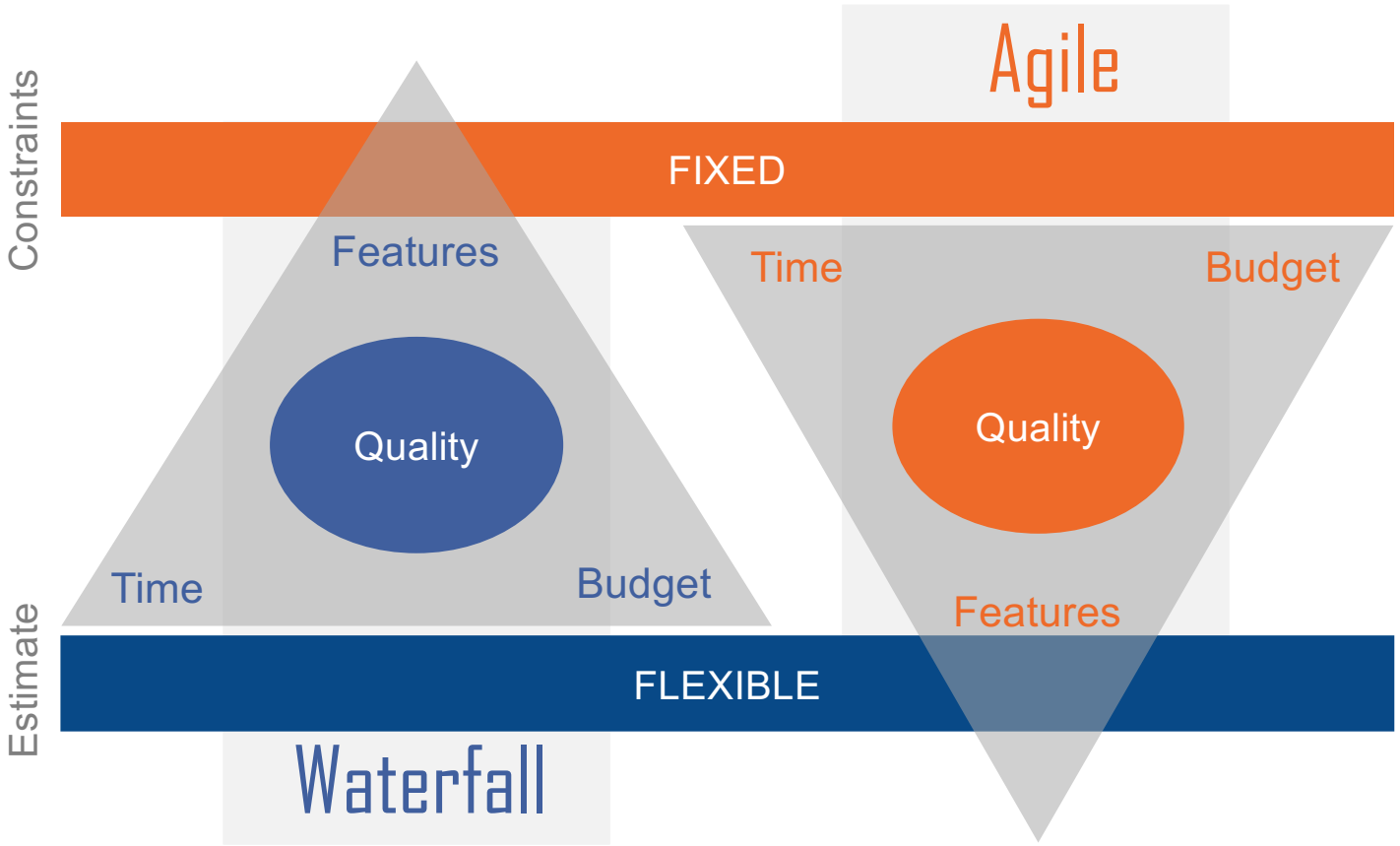
At regular intervals, **the team reflects** on how to become more effective, then tunes and adjusts its behavior accordingly.

consulting

AGILE CHANGES THE PARADIGM

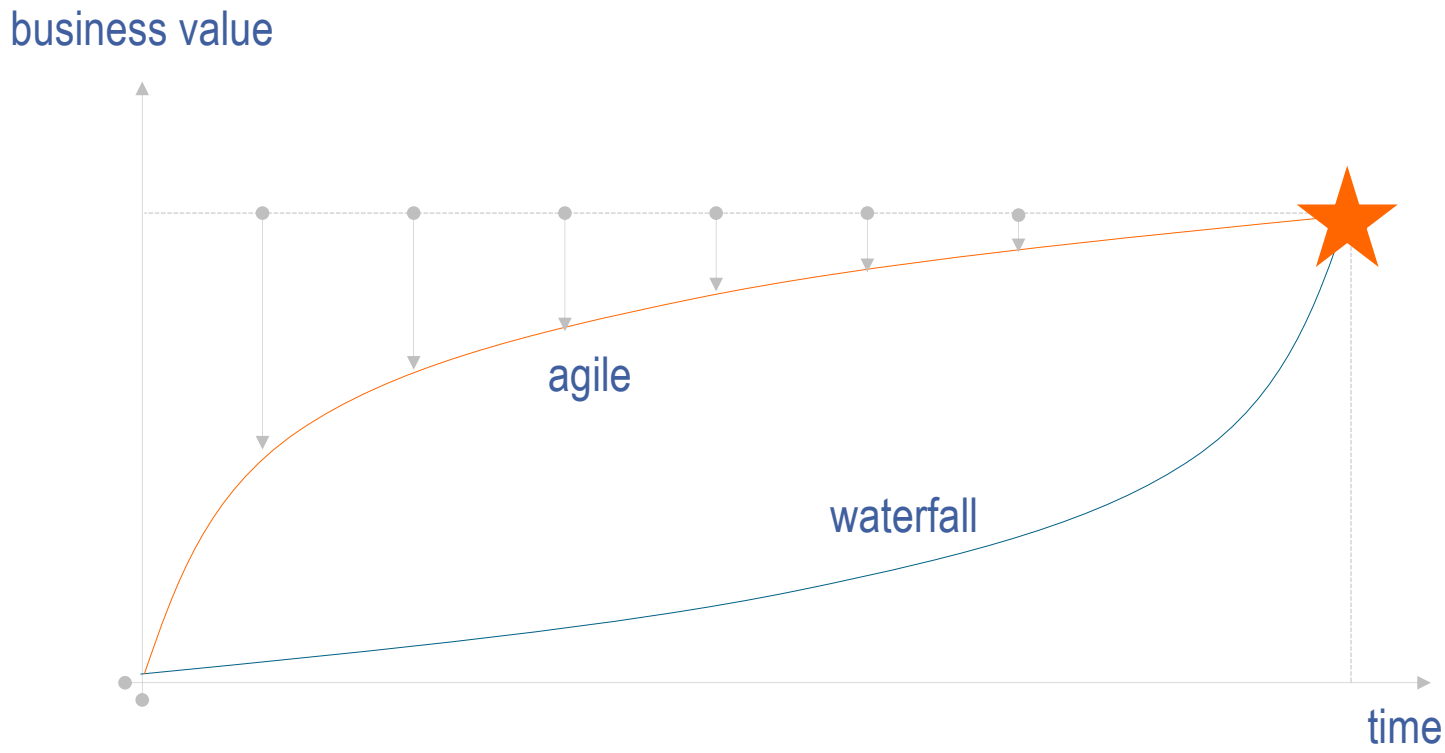
AGILE focuses on TIMES AND COSTS, while the scope of the Project, in the light of empiricism, can change.

CHANGE IS INTENDED IN ITS FULLY POSITIVE ACCEPTANCE, as a continuous search for value and reduction of pockets of inefficiency.



AGILE VERSO WATERFALL

Agile changes the paradigm, even in the estimates



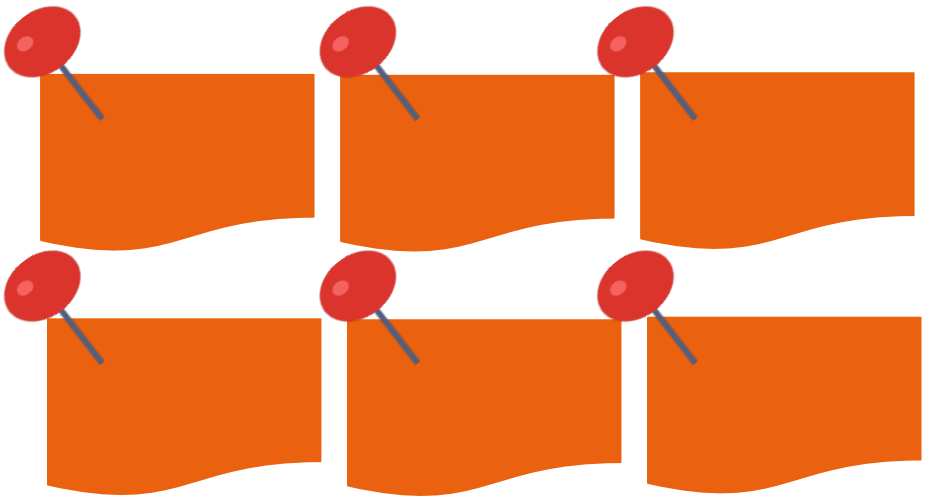
consulting


Just a little
practice before start

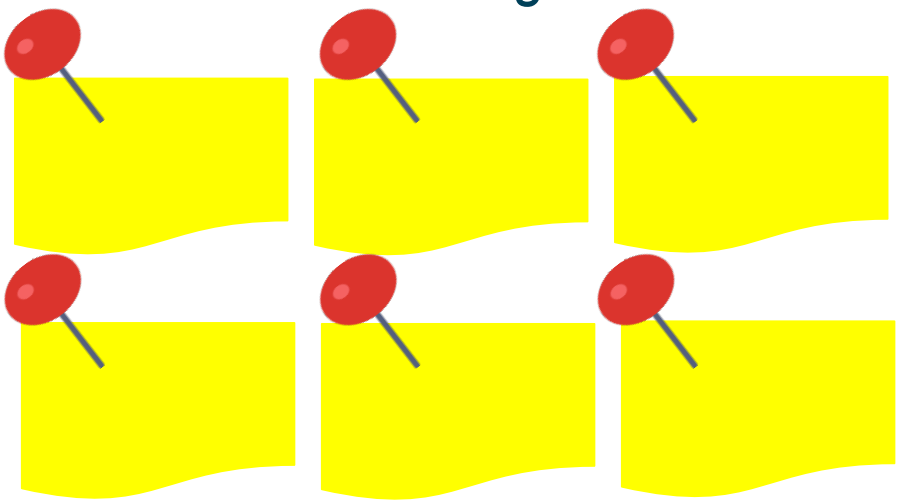
A LITTLE WORKSHOP

AGILE VS WATERFALL

advantages



disadvantages



consulting

ADVANTAGES & DISADVANTAGES OF WATERFALL METHODOLOGIES (1/2)

ADVANTAGES

When there is a CLEAR PICTURE of what the final product should be

When clients WON'T HAVE THE ABILITY TO CHANGE THE SCOPE of the project once it has begun

When DEFINITION, not speed, IS KEY TO SUCCESS

- Really EASY TO USE and manage
- PHASES are NOT OVERLAP
- Customers understand COSTS, SIZE AND TIMELINE for their projects
- Every phase contains SPECIFIC DELIVERABLES

consulting

ADVANTAGES & DISADVANTAGES OF WATERFALL METHODOLOGIES (2/2)

DISADVANTAGES

Once a step has been completed, DEVELOPERS CAN'T GO BACK to a previous stage and make changes

Waterfall methodology relies HEAVILY ON INITIAL REQUIREMENTS. However, if these requirements are faulty in any manner, the project is doomed

THE WHOLE PRODUCT IS ONLY TESTED AT THE END. If bugs are written early, but discovered late, their existence may have affected how other code was written

THE PLAN DOESN'T TAKE INTO ACCOUNT A CLIENT'S EVOLVING NEEDS. If the client realizes that they need more than they initially thought, and demand change, the project will come in late and impact budget

consulting

ADVANTAGES & DISADVANTAGES OF AGILE METHODOLOGIES (1/2)

ADVANTAGES

Agile promotes some of the BEST PRACTICES FOUND IN DEVELOPMENT ENVIRONMENTS. Some of the risk in a project should be reduced as the output of developers is reviewed early and constantly during development

When CLIENTS WILL BE ABLE TO CHANGE THE SCOPE of the project

When the product is intended for an industry WITH RAPIDLY CHANGING STANDARDS

When you have SKILLED DEVELOPERS who are ADAPTABLE and ABLE TO THINK INDEPENDENTLY

Really EASY TO UNDERSTAND | FLEXIBLE to CHANGE | Continuous FEEDBACK | TEAM MOTIVATION

consulting

ADVANTAGES & DISADVANTAGES OF AGILE METHODOLOGIES (2/2)

DISADVANTAGES

Agile is simple to understand in principle but HARD TO DO WELL IN PRACTICE. It requires real commitment and first attempts are not likely to go very well

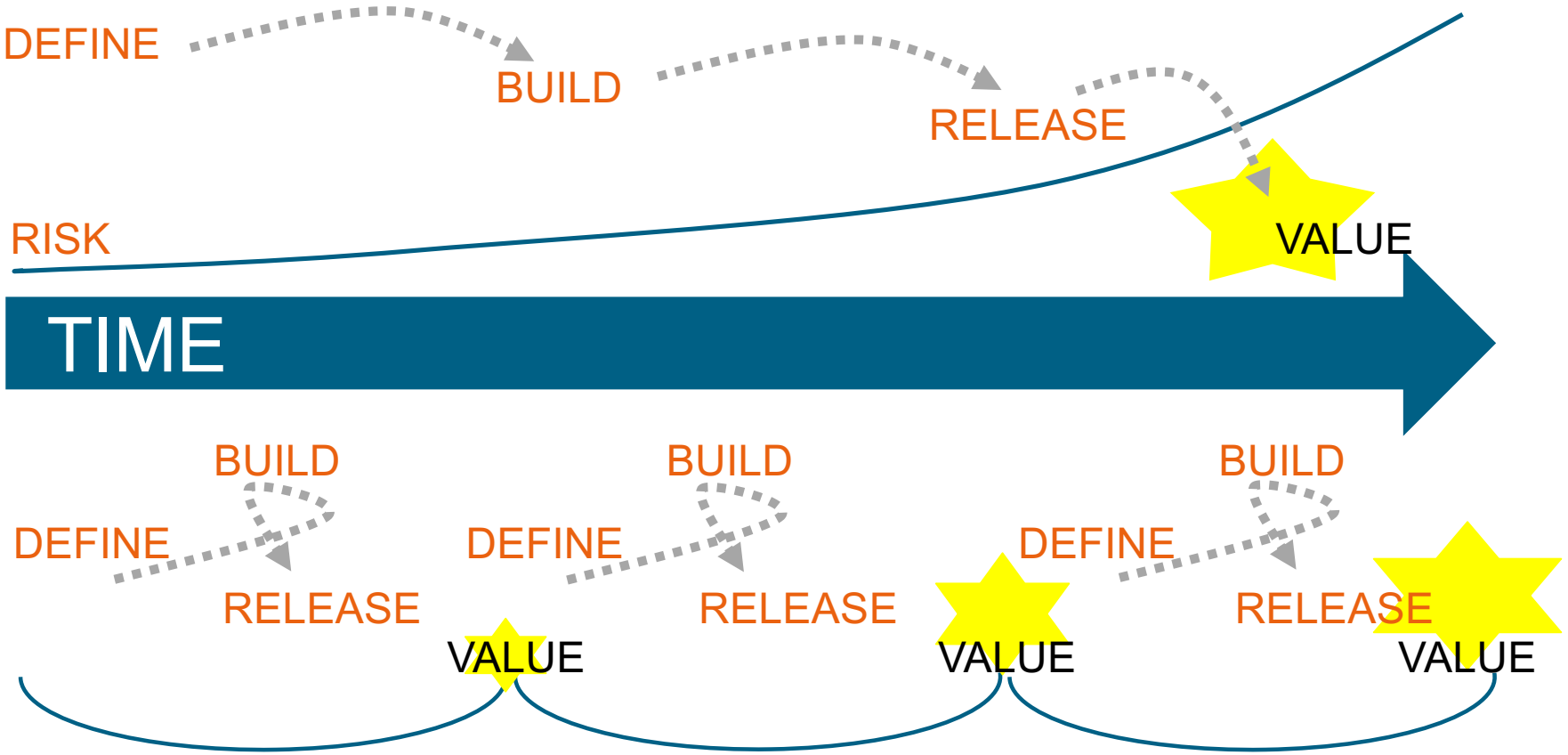
AGILE IS VERY INTENSIVE for both developers and users

AGILE CAN BE VERY CHALLENGING on much larger projects or where co-location is not possible (between developers and the business)

In Agile there can be a GREAT RELUCTANCE (BY SOME) TO ADOPT OR ACCEPT DEADLINES. Projects don't exist in isolation so when this happens it can be a real issue

consulting

AGILE VS WATERFALL



consulting


SCRUM

HISTORY OF SCRUM

Ever since its first publication in 1995 up to now, Scrum has been adopted by a vast amount of software development companies. It is today recognized as the **most applied framework for agile** software

1990 ..... ○ Jeff Sutherland and Ken Schwaber Codified Scrum

1996 ..... ○ Introduction of Scrum at OOPSLA conference and published a paper "Scrum Software Development Process"

2001 ..... ○ Jeff and Ken were amongst the 17 software development leaders creating the Manifesto for Agile Software Development. Shortly after publication "Agile Software Development with Scrum" by K. Schwaber & M. Beedle

2002 ..... ○ Ken Schwaber founded the Scrum Alliance with Mike Cohn and Esther Derby

2006 ..... ○ J. Sutherland created his own company, Scrum.inc

2009 ..... ○ K. Schwaber founded the Scrum.org

2010 ..... ○ Publication of the Scrum Guide in 2010 (updates in 2011/2013)

consulting

SCRUM EMPIRICISM

Empiricism means working in a fact-based, experience-based, and evidence-based manner.

Scrum implements an empirical process where progress is based on observations of reality, not fictitious plans. Scrum also places great emphasis on mind-set and cultural shift to achieve business and organizational Agility.

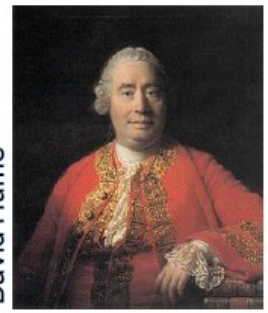
Scrum empiricism

- Scrum is founded on **empirical** process control theory, or **empiricism**
- Empiricism asserts that **knowledge comes from experience** and **making decisions based on what is known**
- Scrum employs an **iterative, incremental** approach to optimize predictability and control risk

https://www.scrum.org/resources/empiricism-essential-element-scrum


John Locke


George Berkeley


David Hume

consulting

I 3 PILLARS DI SCRUM

- **Transparency** to share a vision and create visibility
  - Aspects of the process must be **visible** to those responsible for the outcome
  - Transparency requires those aspects be defined by a **common standard** so observers share a common understanding of what is being seen
    - A **common language** referring to the process must be shared by all participants
    - Those performing the work and those accepting the work product must share a common definition of "**Done**"

consulting

I 3 PILLARS DI SCRUM

- **Inspection** to react rapidly
  - Scrum users must frequently inspect **artifacts** and **progress** toward a **Sprint Goal** to detect undesirable variances. Their inspection should not be so frequent that inspection gets in the way of the work

- **Adaptation** to respond accurately to the needs
  - If an inspector determines that some aspects of a **process** deviate outside acceptable limits, and that the resulting **product** will be unacceptable, the process or the material being processed must be adjusted asap to minimize deviation

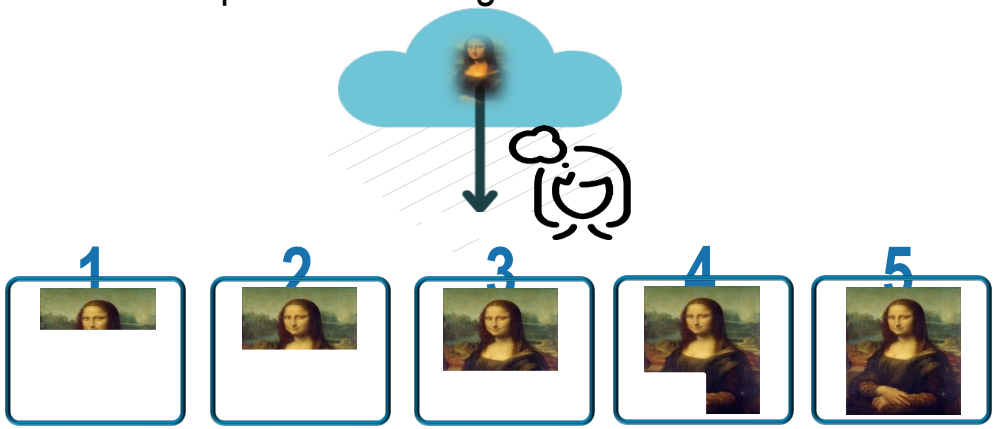
- **4** events for **I&A**: Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective

consulting

INCREMENTAL APPROACH VERSUS ITERATIVE APPROACH

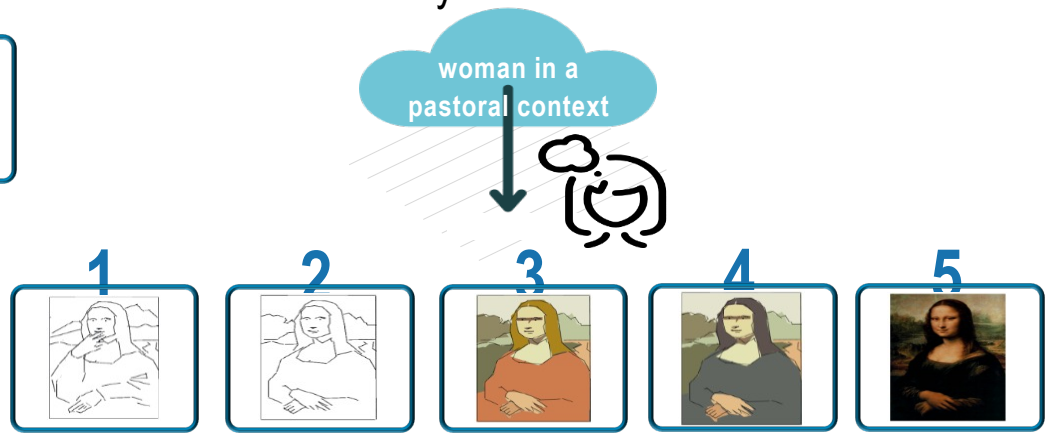
APPROCCIO INCREMENTALE

The incremental approach requires a complete and stable conception of the target to be achieved.



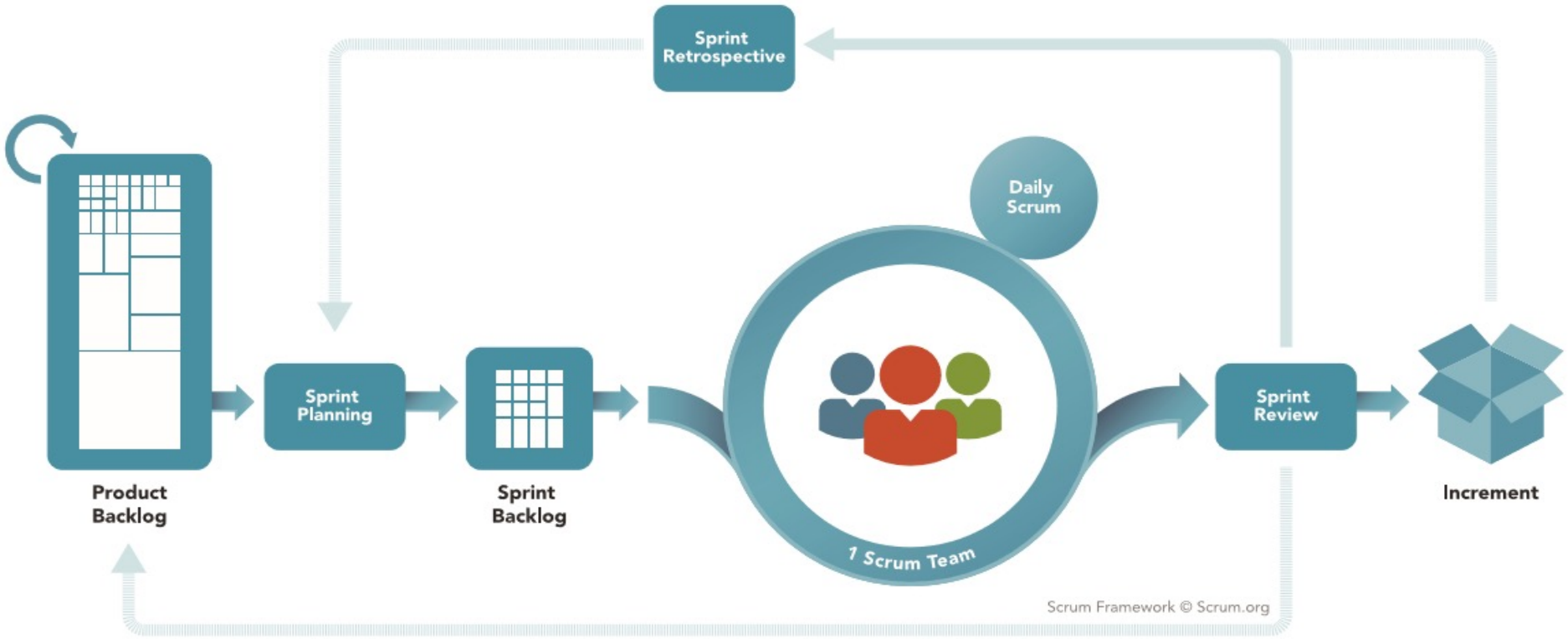
APPROCCIO ITERATIVO

The iterative approach involves the "construction" through successive versions, their validation up to the construction of the final result, introducing quality. It allows you to work around an idea that is not completely defined, gradually including changes until the result is what you want.



consulting

SCRUM FRAMEWORK

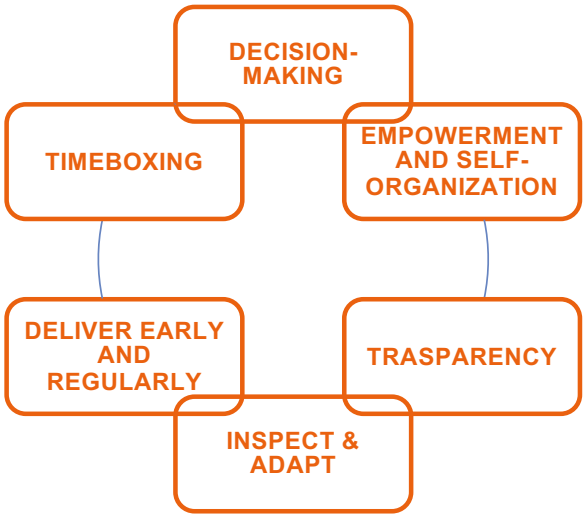


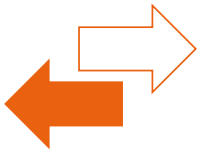
Scrum.org

consulting

SCRUM

SCRUM is an AGILE FRAMEWORK, not a methodology or body of knowledge, for developing and sustaining COMPLEX ADAPTIVE PRODUCTS at the HIGHEST VALUE. Scrum is more about project delivery rather than project management. The Scrum Framework should be used entirely (we do not tailor it) and consists of ROLES, EVENTS, ARTIFACTS, and the RULES that bind them together. Scum is LIGHTWEIGHT, SIMPLE TO UNDERSTAND, DIFFICULT TO MASTER.





ROLES	Product Owner Scrum Master Team
CEREMONIES	Sprint planning Sprint review Sprint retrospective Daily stand up meeting
ARTIFACTS	Product backlog Sprint backlog Burndown charts

consulting

SCRUM TEAM

Product Owner, Scrum Master and Development Team

Product Owner

1 person
Full-time or part-time
Business oriented

Scrum Master

1 person
Full-time or part-time
Scrum coach and facilitator

Development Team

3 to 9 people
Full-time (recommended)
Specialist

The term “Scrum Team” refers to all the project team members: everyone internal to the project. Scrum Team members usually have only one of the three standard roles of Scrum: Product Owner, Scrum Master, or Development Team member.

It is possible for a single person to be assigned to more than one of the standard roles, but it is not recommended.

consulting

AGILE TEAM CHARACTERISTICS: SELF ORGANIZING & CROSS FUNCTIONAL

Product Owner, Scrum Master and Development Team

They are cohesive

Agile teams put an emphasis on collaboration. They are constantly learning from one another and making use of their shared set skills to work harmoniously together. In addition, their working relationships are clearly established so they know that they can work towards a common goal.

They can function without outside help

Typically, an agile team have sufficient skills within the team so that they don't need to outsource help from an outside group. Agile members have the advantage of being versatile enough to specialize in one thing but also dabble in other fields. Not only does this help save time but also allows members to expand their expertise

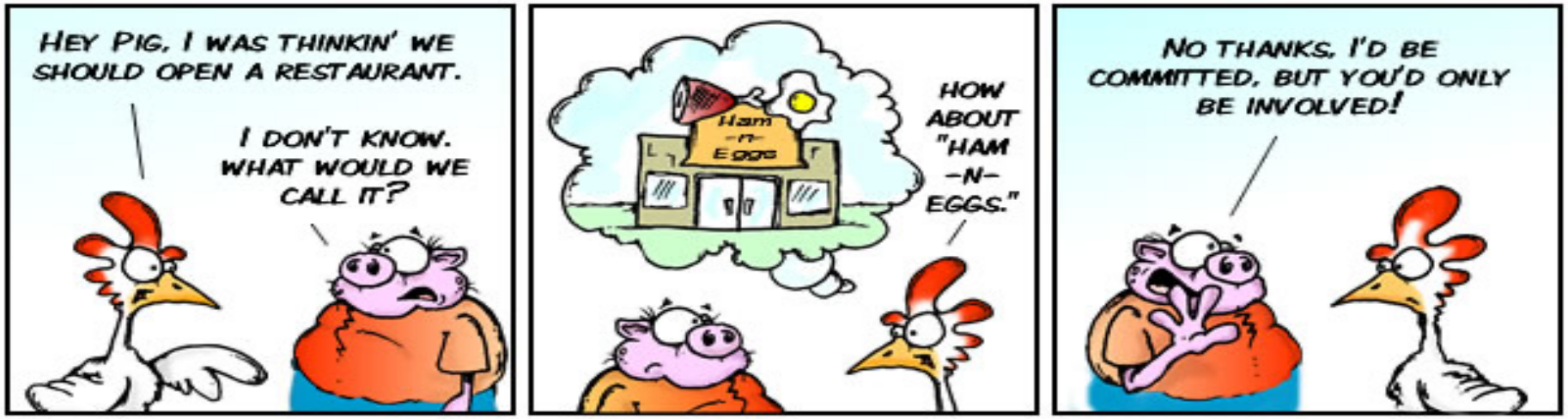
They are stable

With Agile all members are trained to collaborate, cooperate, and deliver results in a timely manner.

The Agile method stresses and is, therefore, a great way to ensure that they deliver timely and relevant outputs. so there's no confusion about who does what..

consulting

PIGS & CHICKEN



PIGS → SCRUM TEAM

- SCRUM MASTER
- PRODUCT OWNER
- TEAM

CHICKENS → STAKEHOLDERS

- MANAGEMENT
- USERS
- CUSTOMER

consulting

LO SCRUM TEAM

- **Development Team (Team), Scrum Master, Product Owner**

- Are **self-organizing** and **cross-functional**
  - **Self-organizing:** choose how best to accomplish their work, rather than being directed by others
  - **Cross-functional:** have all competencies needed to accomplish the work without depending on others not part of the team

- Deliver products **iteratively** and **incrementally**, maximizing opportunities for **feedback**

- Incremental deliveries of **"Done"** product ensure a potentially useful version of working product is always available

consulting

IL PRODUCT OWNER

The Product Owner

The Product Owner is responsible for maximizing the value of the product resulting from work of the Development Team. How this is done may vary widely across organizations, Scrum Teams, and individuals.

The Product Owner is the sole person responsible for managing the Product Backlog. Product Backlog management includes:

- Clearly expressing Product Backlog items;
- Ordering the items in the Product Backlog to best achieve goals and missions;
- Optimizing the value of the work the Development Team performs;
- Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Scrum Team will work on next; and,
- Ensuring the Development Team understands items in the Product Backlog to the level needed.

The Product Owner may do the above work, or have the Development Team do it. However, the Product Owner remains accountable.

The Product Owner is one person, not a committee. The Product Owner may represent the desires of a committee in the Product Backlog, but those wanting to change a Product Backlog item's priority must address the Product Owner.

For the Product Owner to succeed, the entire organization must respect his or her decisions. The Product Owner's decisions are visible in the content and ordering of the Product Backlog. No one can force the Development Team to work from a different set of requirements.

https://www.youtube.com/watch?v=iBBnVGXw60Y

consulting

IL PRODUCT OWNER

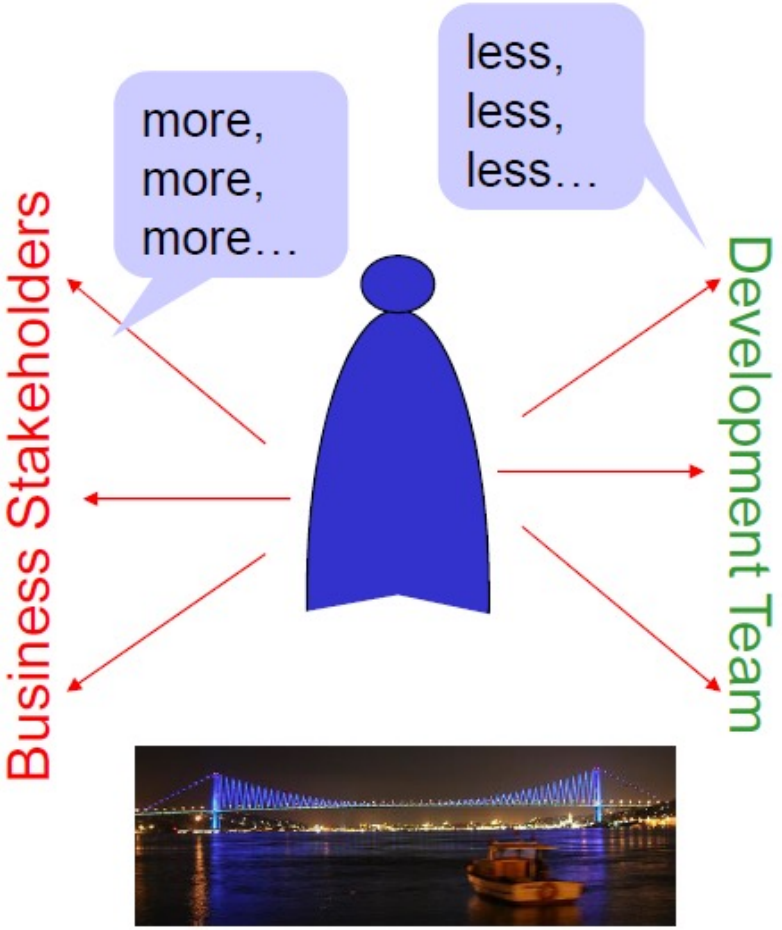
- Is responsible for **maximizing the value of the product** and the **work of the Team**

- The PO is the **sole person responsible for managing the Product Backlog**
  1. **Expressing** the items
  2. **Ordering** (prioritize) the items to best achieve goals and missions
  3. **Optimizing** the value of the work the Team performs
  4. Ensuring that the Product Backlog is **visible, transparent, and clear to all**, and shows what the Scrum Team will work on **next**
  5. Ensuring the Team **understands** items to the level needed

consulting

IL PRODUCT OWNER

- The PO act as a Business Analyst
- Is accountable for the value the Project will generate
- Accepts or reject the work
- Helps define "Done"
- Knowledgeable, empowered, engaged
- Co-located with team as much as possible
- Manages stakeholder expectations
- Motivates team, celebrates success



Don't be an impatient Product Owner

consulting

IL PRODUCT OWNER

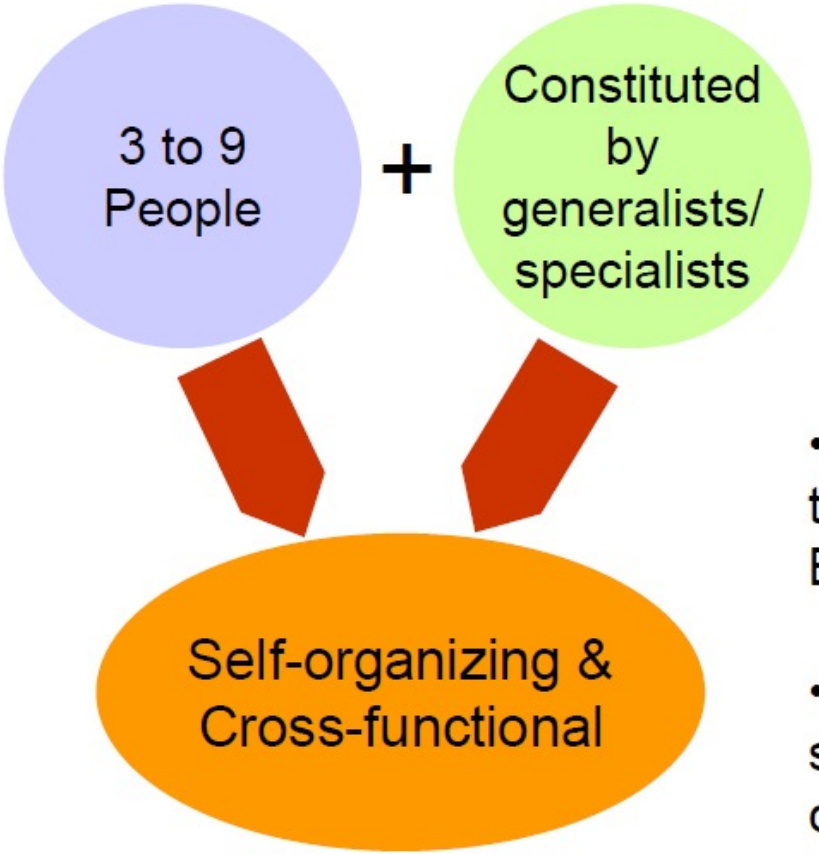
- The PO may do the above work, or have the Team do it. However, the **PO remains accountable**

- **The PO is one person**, not a committee. He may represent the desires of a committee in the Product Backlog, but **those wanting to change a Product Backlog item's priority must address the PO**

- The entire organization must respect the PO decisions. **No one is allowed to tell the Team to work on a different set of requirements, and the Team isn't allowed to act on what anyone else says**

consulting

THE DEVELOPMENT TEAM



- Self-organizing: no one tells them how to turn Product Backlog into product Increments
- Cross-functional: have all of the skills as a team necessary to create a product Increment

http://en.wikipedia.org/wiki/The_Magical_Number_Seven,_Plus_or_Minus_Two, Miller's Law

consulting

THE DEVELOPMENT TEAM

- Does the work of **delivering the product increment**
- **Has all the necessary skills** to deliver each increment of the product
- **Forecasts** how much they can do in one sprint
- **Self-organizes** to get the work done, **deciding among themselves who does what**
- All team members should be available to the project **full time**

THE DEVELOPMENT TEAM

- Scrum recognizes **no titles** for Development Team members **other than Developer**

- Scrum recognizes **no sub-teams** in the Development Team, regardless of particular domains that need to be addressed like testing or business analysis

- Individual Development Team members may have specialized skills and areas of focus, but **accountability belongs to the Development Team as a whole**

consulting

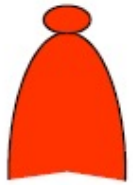
SCRUM MASTER

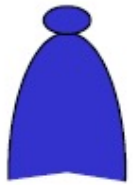
- Is responsible for **ensuring Scrum is understood and enacted**
- Is a **servant-leader** for the Scrum Team
- Helps those **outside the Scrum Team** understand which interactions with the Scrum Team are helpful and which aren't

consulting

THE SCRUM MASTER AT THE SERVICE OF THE PRODUCT OWNER

- Finding **techniques** to help Product Backlog management
- Helping the Scrum Team understand the need for **clear and concise Product Backlog items**
- Understanding **product planning** in an empirical environment
- Ensuring the PO knows how to **arrange the Product Backlog** to maximize value
- **Facilitating** Scrum events as requested





consulting

THE SCRUM MASTER AT THE SERVICE OF THE DEVELOPMENT TEAM

- **Coaching** the Team in self-organization and cross-functionality
- Helping the Team to create **high-value** products
- **Removing impediments**
- **Facilitating** Scrum events as requested
- **Coaching** the Team in organizational environments in which Scrum is not yet fully adopted and understood

consulting

THE SCRUM MASTER AT THE SERVICE OF THE DEVELOPMENT TEAM

- **Leading** and coaching the organization in Scrum adoption
- **Planning Scrum implementations** within the organization
- Helping stakeholders **understand and enact Scrum**
- Causing **change** to increase the productivity of the Scrum Team
- Working with **other Scrum Masters** to increase the effectiveness of the application of Scrum in the **organization**

consulting

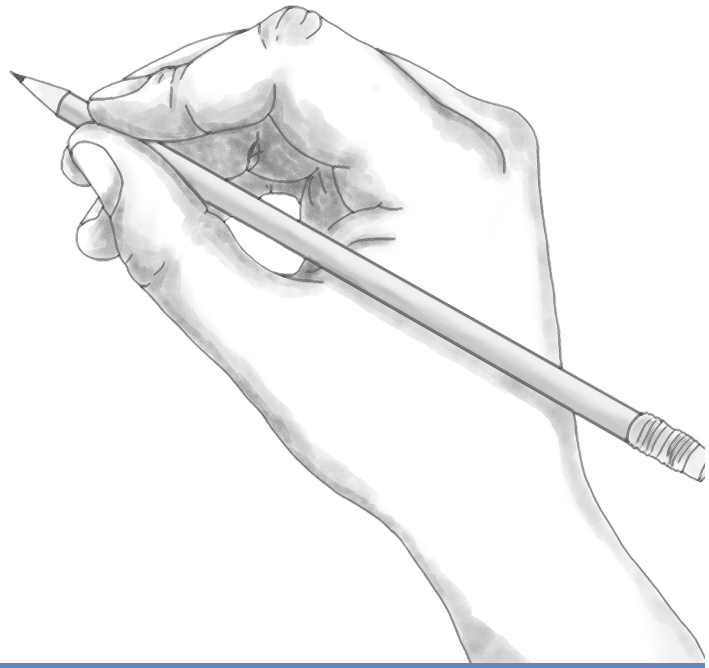
WHY WE SPEAK ABOUT BUSINESS ANALYSIS

Incomplete requirements	13,1%
Stakeholders not involved in the analysis	12,4%
Lack of resources	10,6%
Unrealizable expectations	9,9%
Lack of Governance	9,3%
Change requests	8,7%
Lack of planning	8,1%
No longer necessary	7,5%

Fonte: Standish Group (sulla base di 8000 progetti analizzati)

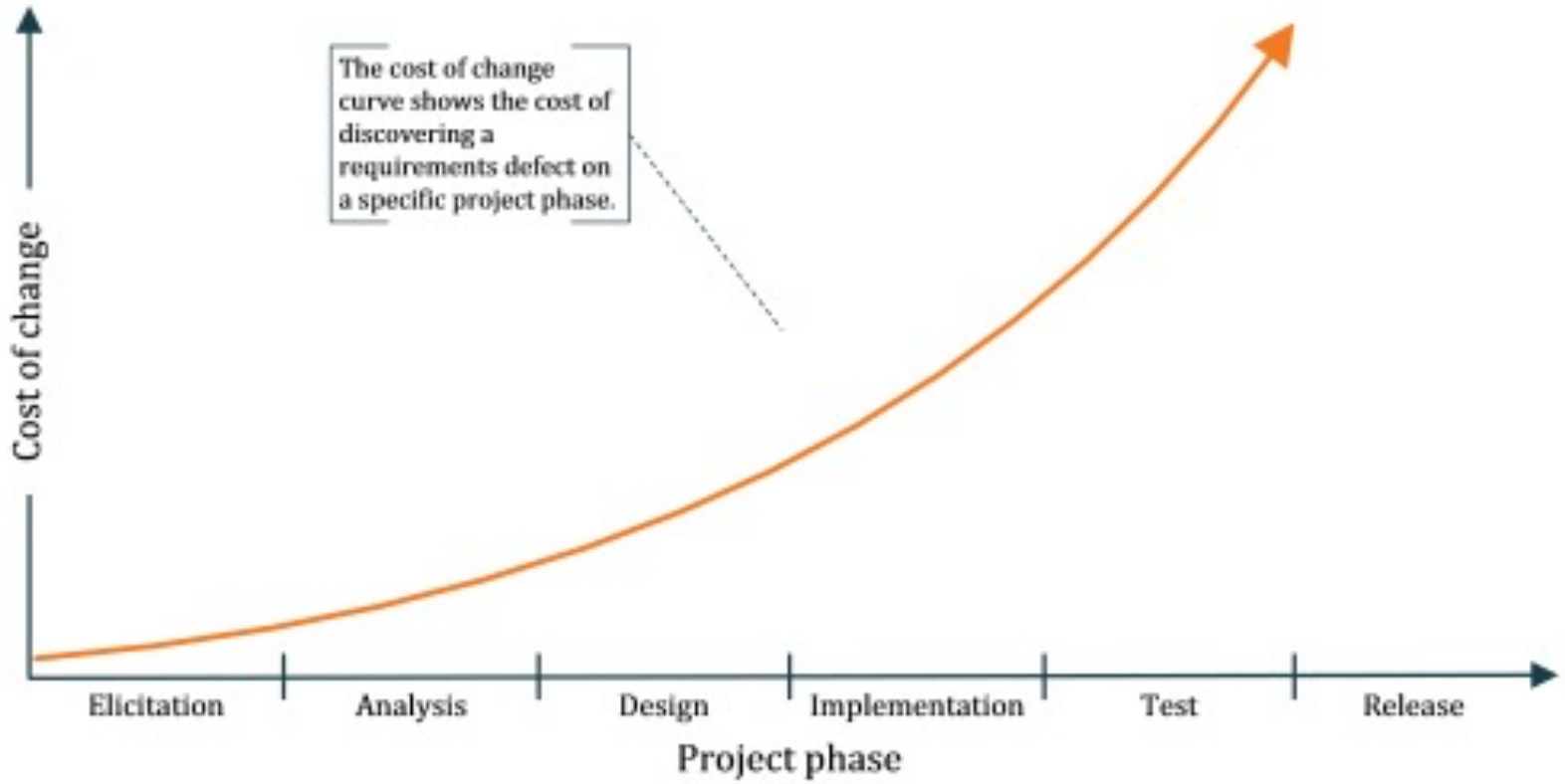
More than 50% of CAUSES of FAILURE of ICT projects can be traced back to LACKS OF BUSINESS ANALYSIS

BUSINESS ANALYSIS supports organizations in defining the expected product / service and in developing the "BETTER BUSINESS OUTCOMES", guiding them in the change, towards the right direction.



WHY WE SPEAK ABOUT BUSINESS ANALYSIS

COST OF CHANGE CURVE



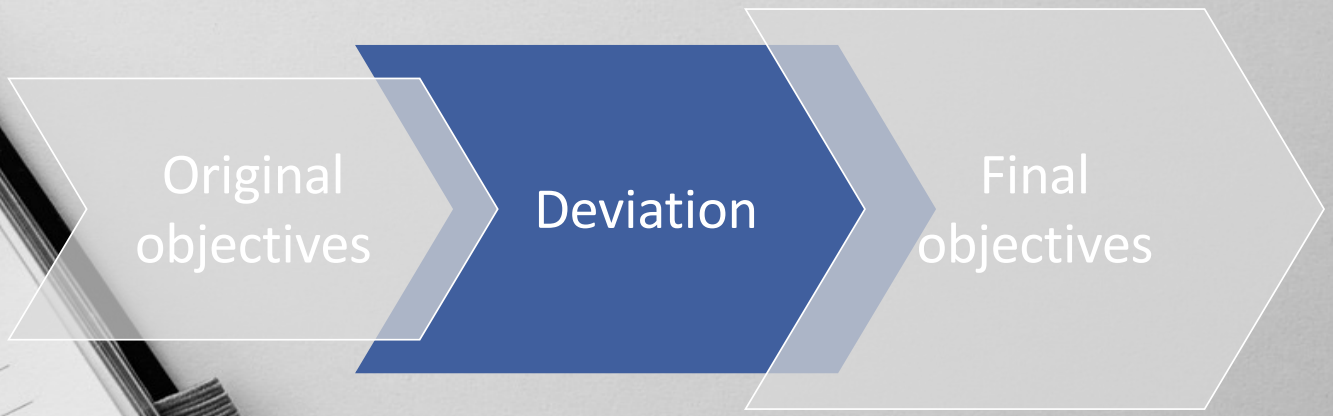
WHY WE SPEAK ABOUT BUSINESS ANALYSIS

SCOPE CREEP FENOMENON

SCOPE CREEP generally refers to a PROJECT that has seen its ORIGINAL OBJECTIVES EXPAND while the project is in progress

As the term "creep: slowly advancing" suggests it is a SUBDLE PROCESS that begins with SMALL ADJUSTMENTS and ends with PROJECTS that EXTEND much longer than expected or even FAIL even before completing

Even if the project is completed, the scope creep can lead to FINAL RESULTS FAR AWAY from the ORIGINAL ones



WHAT BUSINESS ANALYSIS IS

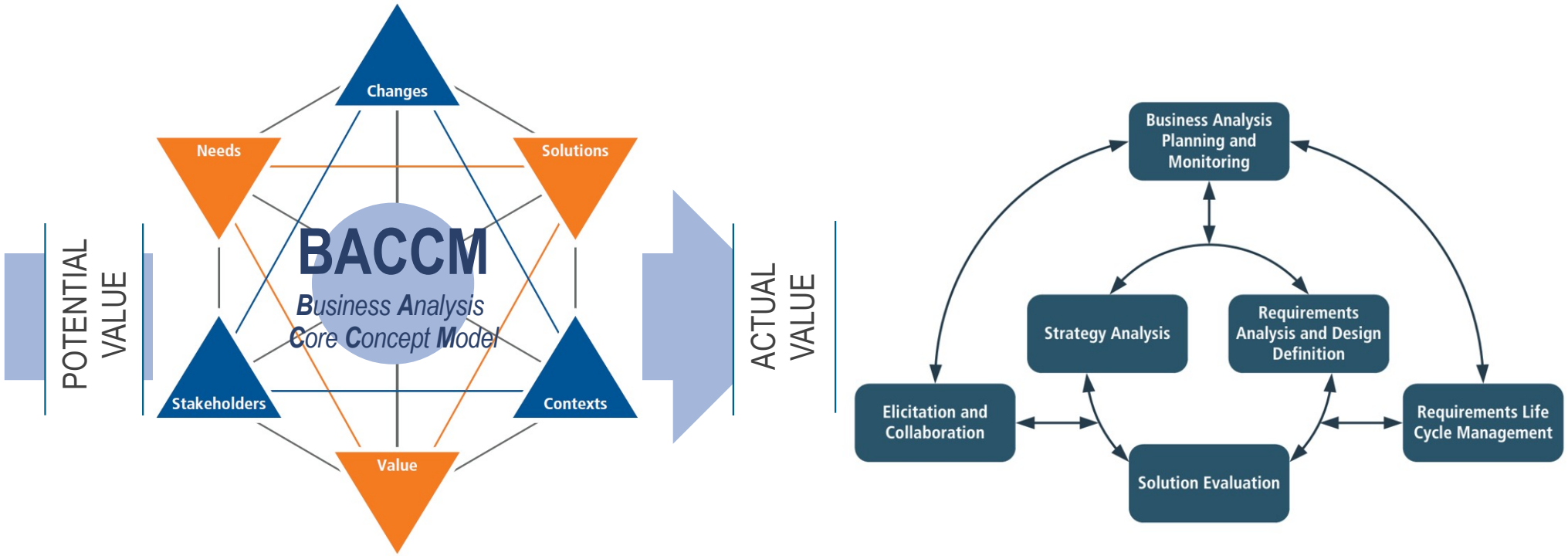
It is the set of KNOWLEDGE and SKILLS necessary to effectively SUPPORT the CHANGE of a company, starting from the definition of business needs up to the proposition of solutions that offer VALUE to the STAKEHOLDERS involved

It is used to understand the CURRENT STATE, define the FUTURE STATE, and determine all the necessary actions to GO from the current state to the future one

Through the application of BEST PRACTICE, skills, knowledge, skills and techniques NOTED and SHARED by a large international community

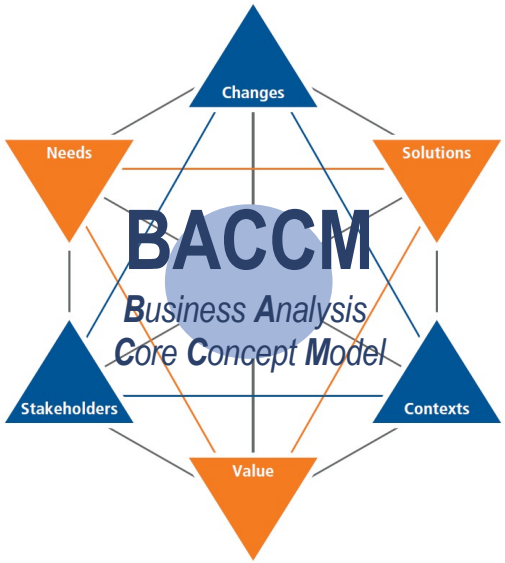
BUSINESS ANALYSIS CORE CONCEPT MODEL

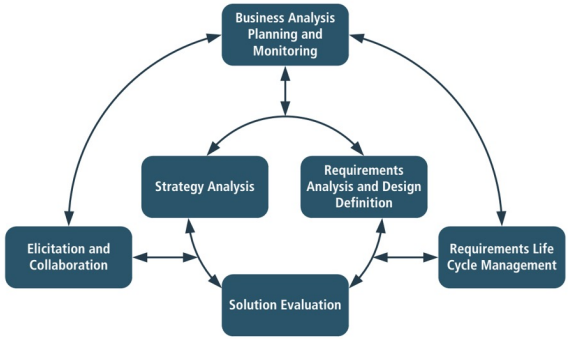
The Business Analysis Core Concept Model ™ is a CONCEPTUAL FRAMEWORK that defines the 6 KEY PRINCIPLES OF BUSINESS ANALYSIS:

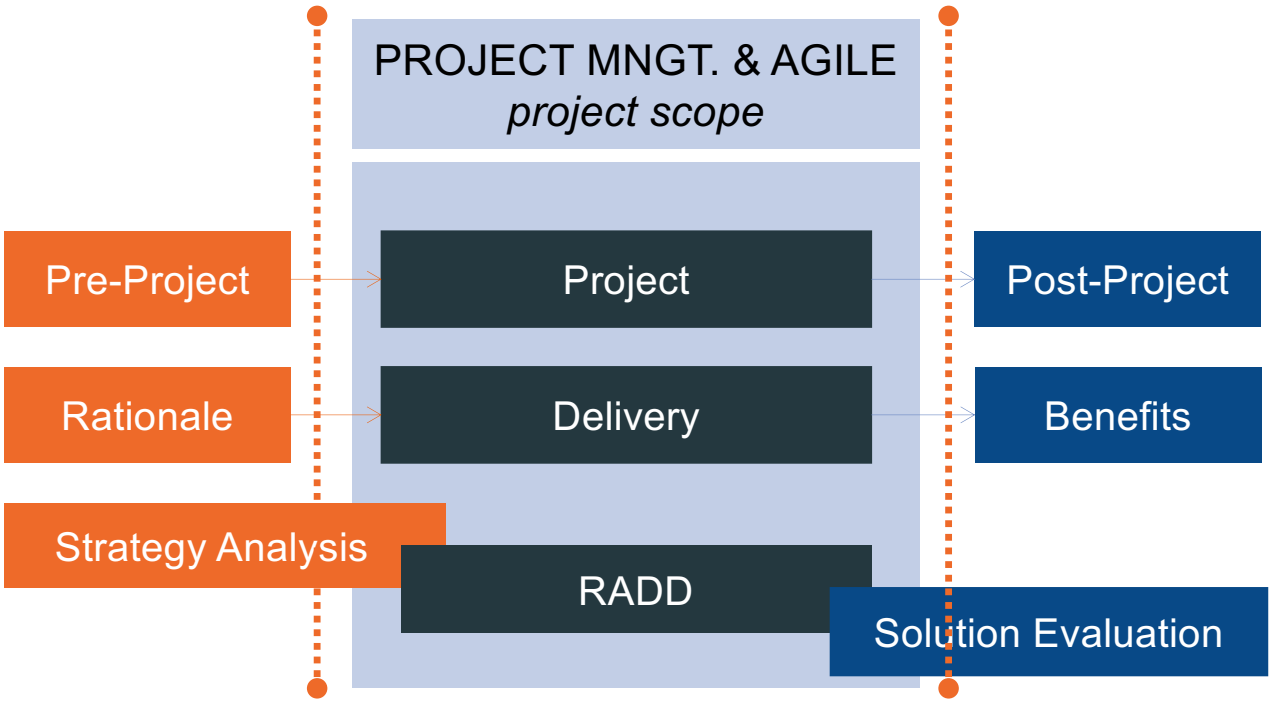


From the knowledge of the Organization's NEEDs, the Business Analyst identifies the SOLUTIONS to obtain the greatest VALUE, analyzing the reference CONTEXT and what the STAKEHOLDERS really need in order to direct the CHANGE

BUSINESS ANALYSIS CORE CONCEPT MODEL

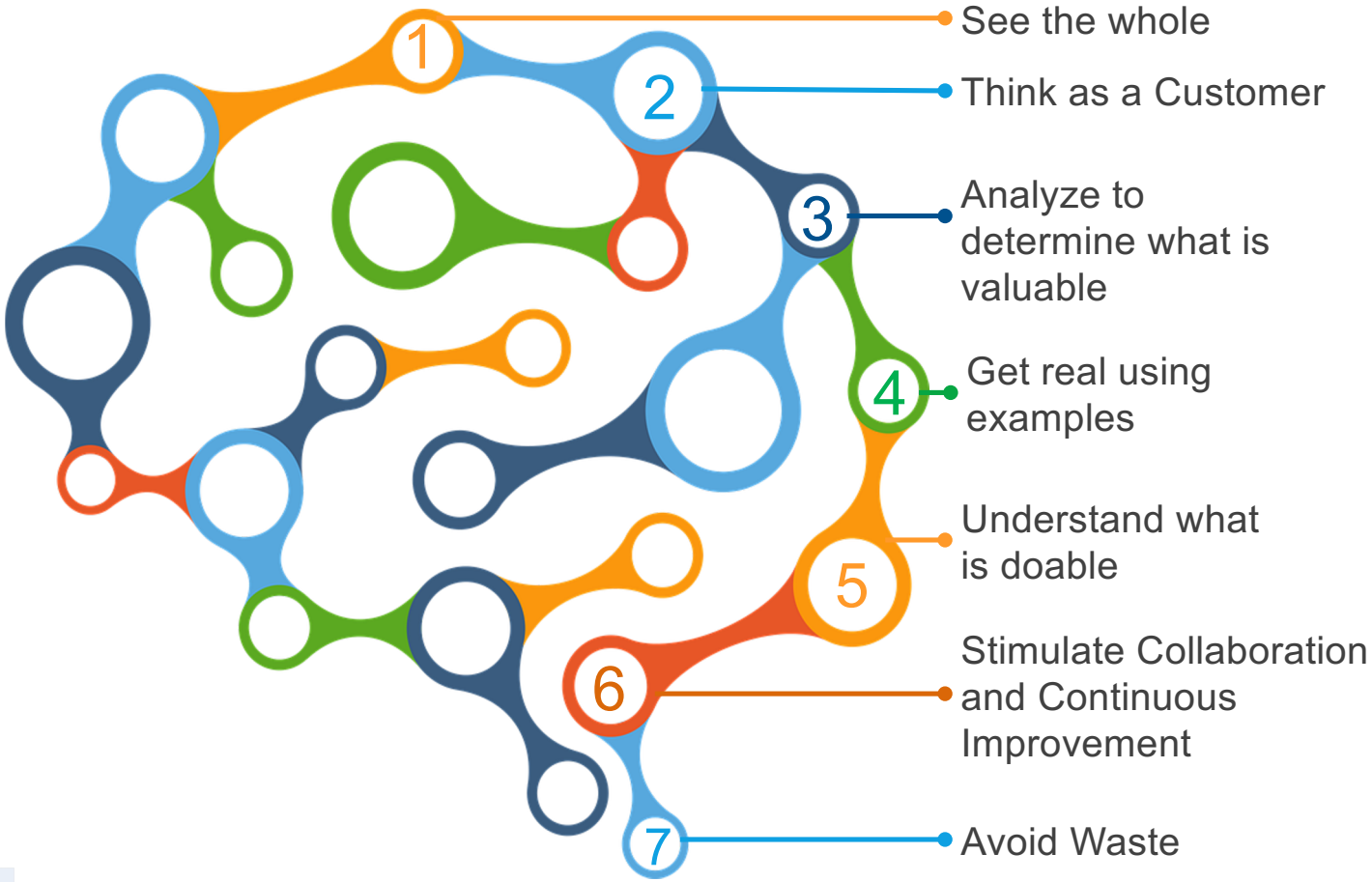






Business Analysis Beyond Projects

ALCUNI PRINCIPI DI AGILE BUSINESS ANALYSIS



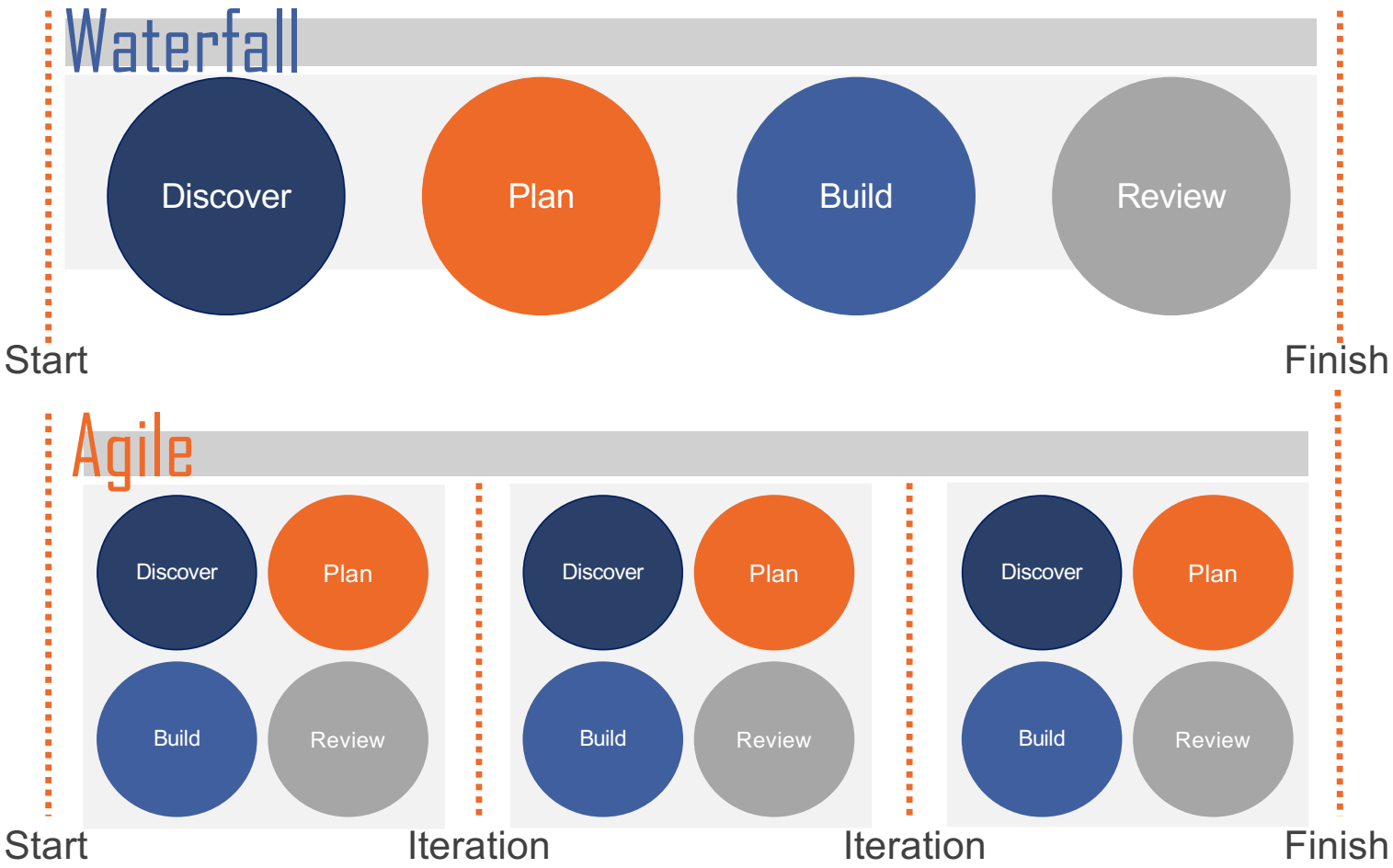
Per guidare il cambiamento, massimizzare il valore, limitare il fenomeno dello SCOPE CREEP, i framework AGILI e le leve fornite dalla BUSINESS ANALYSIS, insieme, possono costituire una valido strumento.

Source: Agile Extension – 2. The Agile Mindset

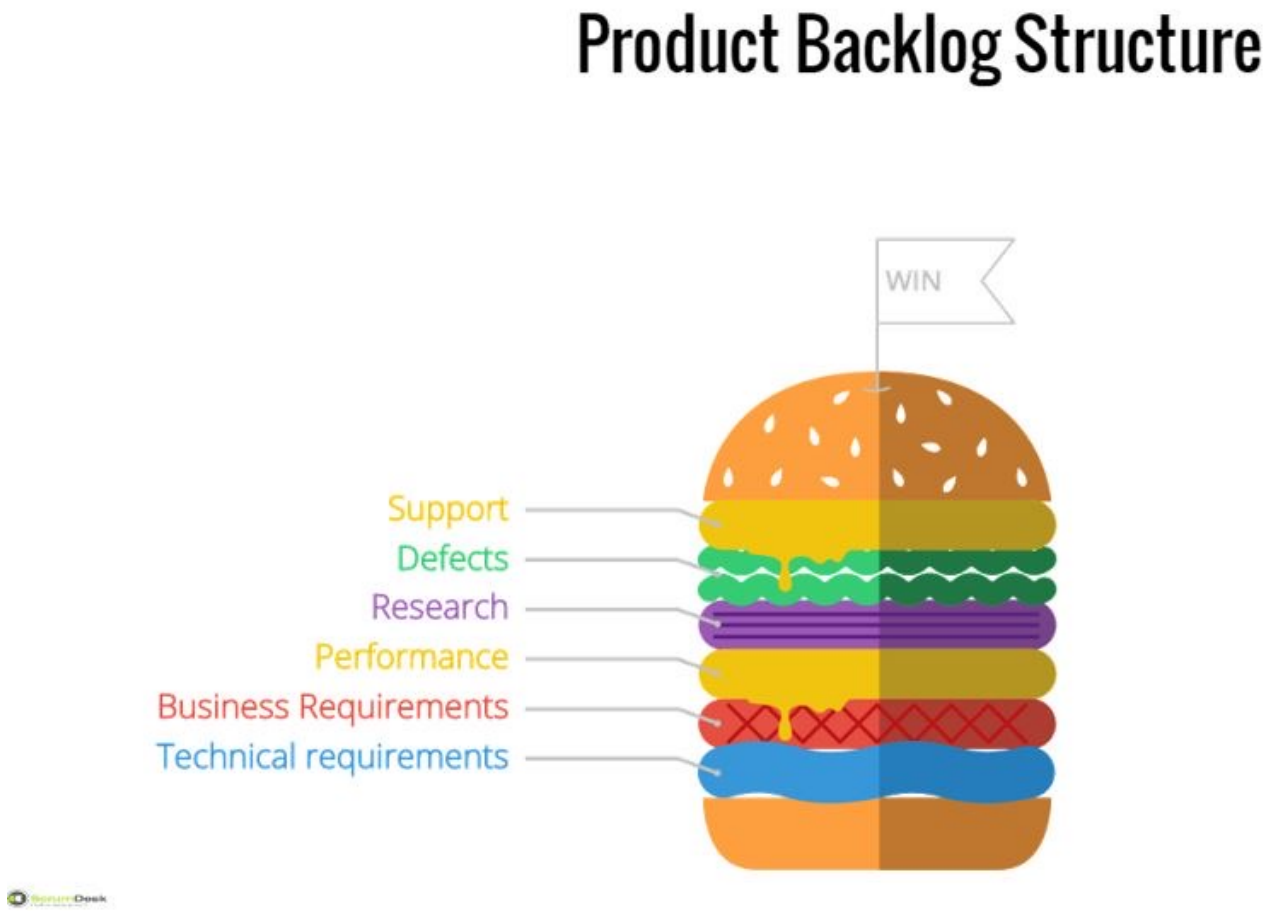
AGILE REQUIRES RISK MANAGEMENT

THE RISK RESIDES IN THE VERY CHANGE OF A simple, rigid and sequential APPROACH such as Waterfall, and in the transition to a more fluid and less immediate model in understanding.

For this reason, THE TRANSITION MUST BE GUIDED.



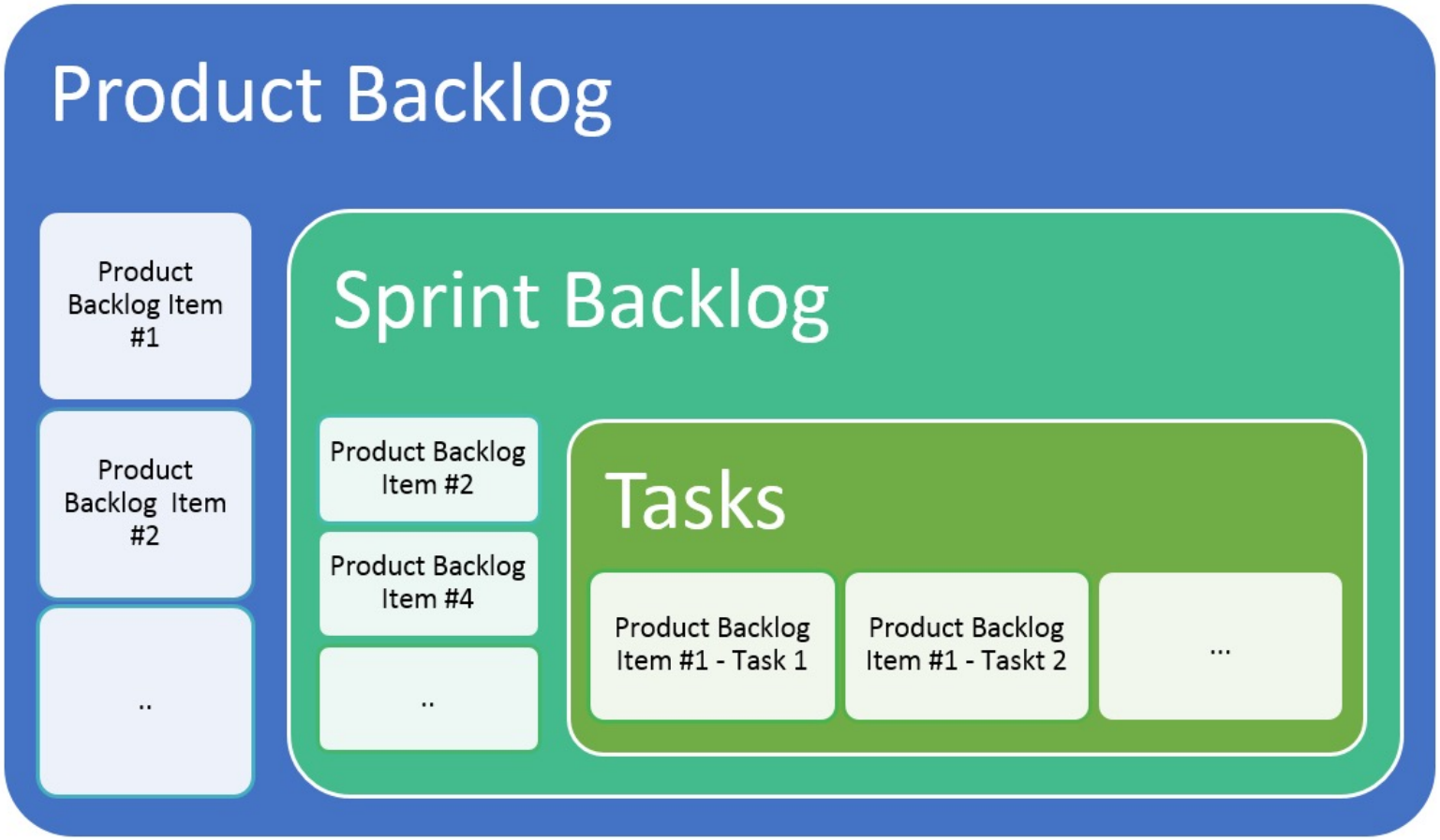
PRODUCT BACKLOG STRUCTURE



PRODUCT BACKLOG CONTENT

- User Stories, Epics, Features, etc.
- Solution and Transition Requirements
- "Spikes" (help the team to gain technical understanding)
- Known bugs
- Other work items

HOW TO USE PRODUCT BACKLOG



PRODUCT BACKLOG

- Is the **single source** of all desired work on project
- The Product Owner is accountable for the **content, availability** (make it public)**, and prioritization**
- **Reprioritized** at start of each sprint (grooming)
- The Product Backlog is never complete: is a **dynamic** artifact

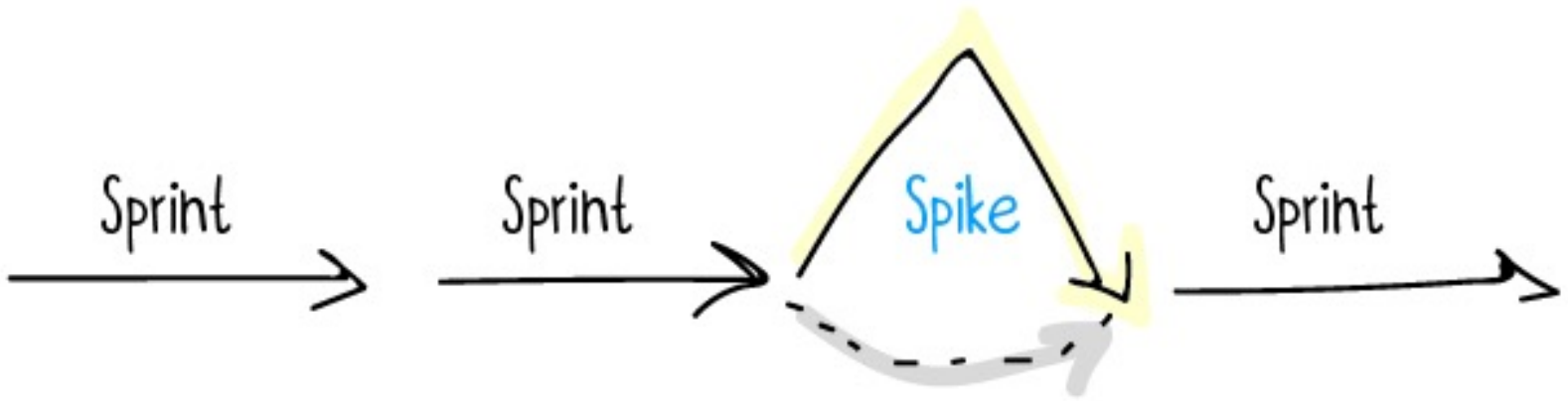
SPIKE

Spikes are an invention of Extreme Programming (XP), are a special type of user story that is used to gain the knowledge necessary to reduce the risk of a technical approach, better understand a requirement, or increase the reliability of a story estimate. A spike has a maximum time-box size as the sprint it is contained in it. At the end of a sprint, the spike will be determined that is done or not-done just like any other ordinary user story. A Spike is a great way to mitigate risks early and allows the team ascertain feedback and develop an understanding on an upcoming PBI's complexity.

Sometime the team unsure if they can complete the story due to some potential blockers and probably can't even estimate the story. Thus, you may consider a spike as an investment for a Product Owner to figure out what needs to be built and how the team is going to build it. The Product Owner allocates a little bit of the team's capacity now, ahead of when the story needs to be delivered, so that when the story comes into the sprint, the team knows what to do.

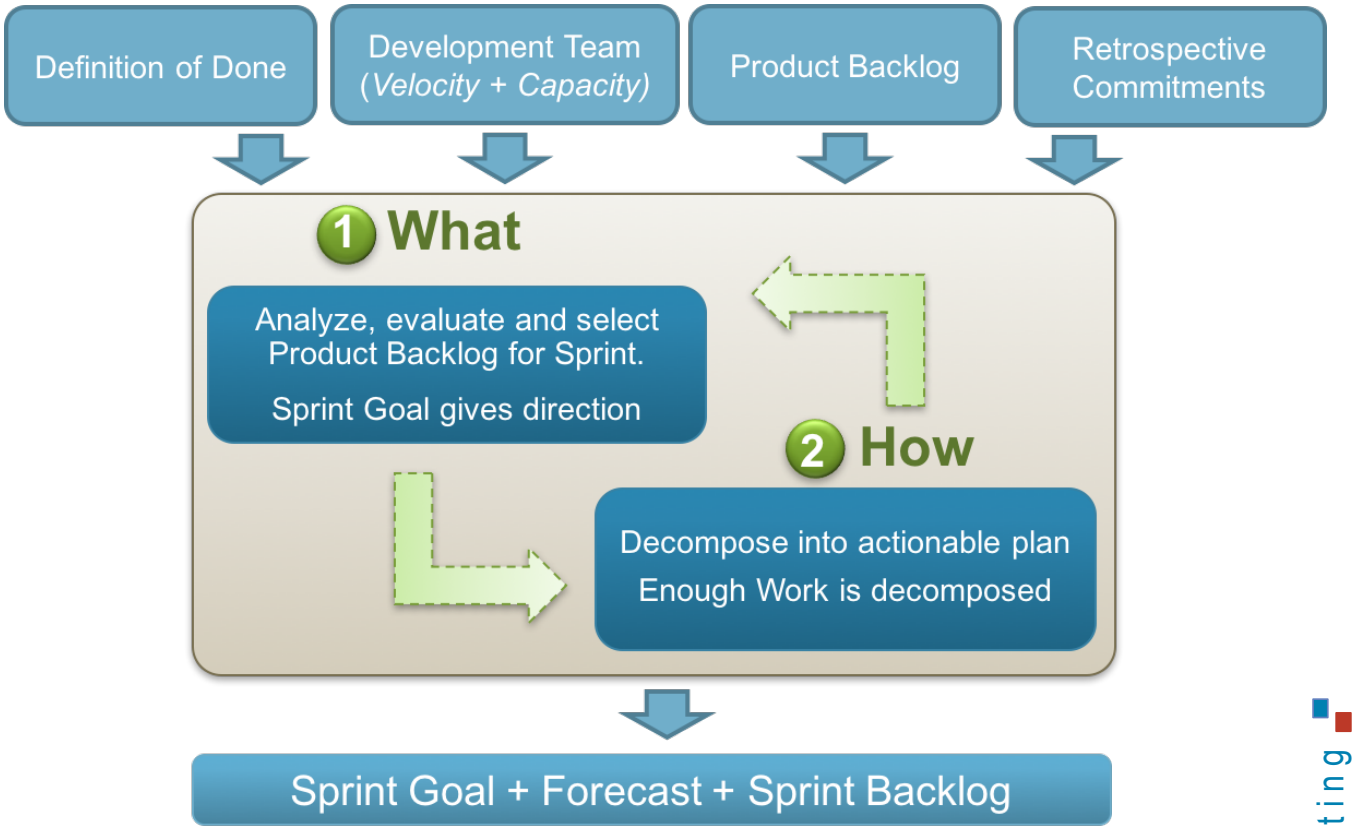
EXAMPLES OF SPIKE

- The team may not have knowledge of a new technology, and spikes may be used for basic research to ensure the feasibility of the new technology (domain or new approach).
- A story requires to be implemented using a 3rd party library with API that is poorly written and documented.
- The story may contain significant technical risk, and the team may have to do some experiments or prototypes to gain confidence in a technological approach that may allow them to commit the user story to some future timebox.



SPIKE, WHEN ESTIMATE IT

Spikes should be estimated as in-Sprint tasks during Sprint Planning. The task's duration should be spent researching and developing some 'thing' that can be delivered. That thing can be a working piece of software, workflow, documentation, etc. Ultimately the value from the spike is a direction or re-direction in the course of the feature. If the team estimated that a Spike takes four hours, then ONLY four hours should be spent researching or developing. Prototypes, Proof of Concepts (PoC), and Wireframes all fall into the classification of Spikes.



consulting

ACCEPTANCE CRITERIA FOR SPIKE STORIES

Just like any other ordinary user story, they need fulfil some certain criteria to obtain the status of done by making sure that the “Spike Story” estimable, demonstrable, and acceptable:

Estimable

Like other stories, spikes are put in the backlog, estimable and sized to fit in an iteration. Spike results are different from a story, as they generally produce information, rather than working code. A spike may result in a decision, a prototype, storyboard, proof of concept, or some other partial solution to help drive the final results.

In any case, the spike should develop just the information sufficient to resolve the uncertainty in being able to identify and size the stories hidden beneath the spike.

Demonstrable

The output of a spike is demonstrable to the team. This brings visibility to the research and architectural efforts and also helps build collective ownership and shared responsibility for the key decisions that are being taken.

Acceptable

And like any other story, spikes are accepted by the product owner when the acceptance criteria for the spike have been fulfilled.

USER STORY

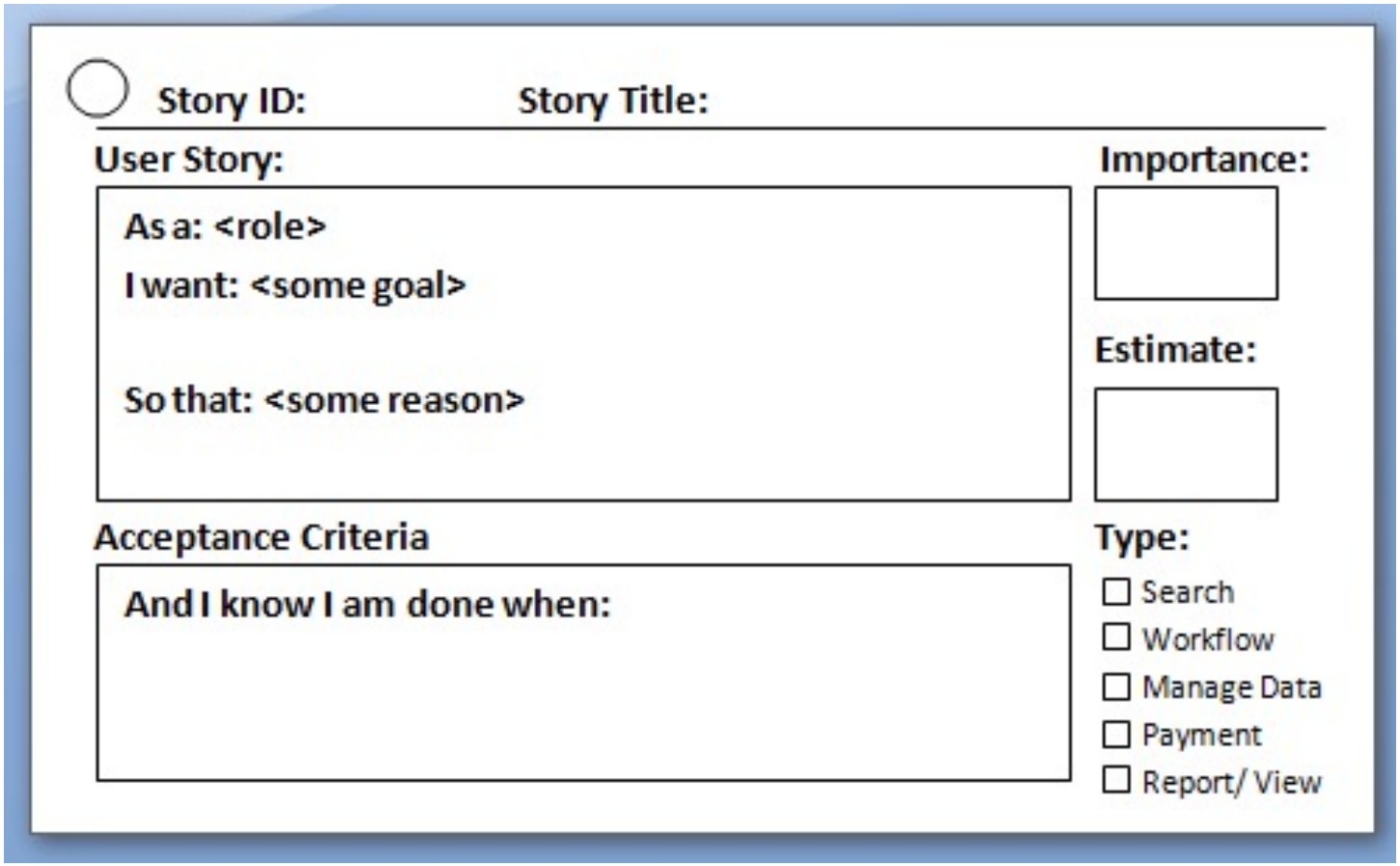
A user story represents a small, concise statement of functionality needed to deliver value to a specific stakeholder

- As a <type of user> - **WHO**
- I want to <feature> - **WHAT**
- So that <benefit / purpose> - **WHY**

Example: as a Security Officer, I need to only allow authorized users to access the *xyz* functionality so I can ensure we enforce *abc* security directive

consulting

USER STORY



consulting

INVEST GUIDELINES

Follow the INVEST guidelines for good user stories!

INVEST GUIDELINES (2/2)

Bill Wake in his article from 2003, introduced a framework that helps everyone create good user stories. It's called **INVEST**.

- **Independent**: Each story should be independent (no overlapping) so it can be developed and delivered separately
- **Negotiable**: Details will be clarified by the cooperation of the developers and customers
- **Valuable**: It needs to be valuable for the users
- **Estimable**: The story should be estimated. It doesn't have to set exact time frame just a good estimate to schedule it in the project
- **Small**: The stories need to be small enough to accurately estimate work it requires
- **Testable**: It needs to be tested easily. Which also indicates that the requirements are well-defined.

PRODUCT BACKLOG REFINEMENT