WHY AGILE?
WHAT TO CONSIDER: OVERVIEW, BENEFITS, DISADVANTAGES

FOREWORD

The intention of this presentation is to give an overview of what the agile methodologies entail and what the impact of a transformation to agile could have on an organisation.

It does not:

A. Intend to train the reader in any of the agile methodologies

B. Teach agile methodologies

C. Impose a view of which approach (waterfall or agile) is better suited

The reader is expected to be able to determine what the best option for his/her organisation or initiative is, by using the considerations provided, and if necessary, use professional transformation services or utilise the existing knowledge within their organisation .

WHAT IS AGILE?

1. A time-boxed, iterative approach to software delivery that allows for incremental target deliverables

2. Its main characteristic is the breaking of *features or processes* down into small sized batches of functionality

3. It grants the ability to set and change priorities for each deliverable, and providing these in short cycles referred to as "interactions"

WHAT AGILE IS NOT!

Agile Metrics: Velocity is NOT The Goal !!!

1. **The silver bullet**. It does not guarantee the success of the project, as initiatives that run on agile principles also face the same project issues (e.g. missed deadlines, over budgeting, etc.)

2. A specific way to develop software that a company can only learn if they buy the services of a consultancy firm

3. A way or running software projects with no documentation, planning or discipline

4. A one "size fits all" solution based on strict rituals

5. Scrum! Although Scrum is one of several agile methodologies

AGILE & WATERFALL PROJECT ENVIRONMENT

The Agile Manifesto (**by The Agile Alliance**) published in 2001, acknowledged by the software community that requirements evolve, and cannot always be fully pre-defined

WATERFALL APPROACH

Initially everything is fixed, but once the **scope matures or evolves**, then **time, quality** and **costs** are **impacted** by the **changes in scope**
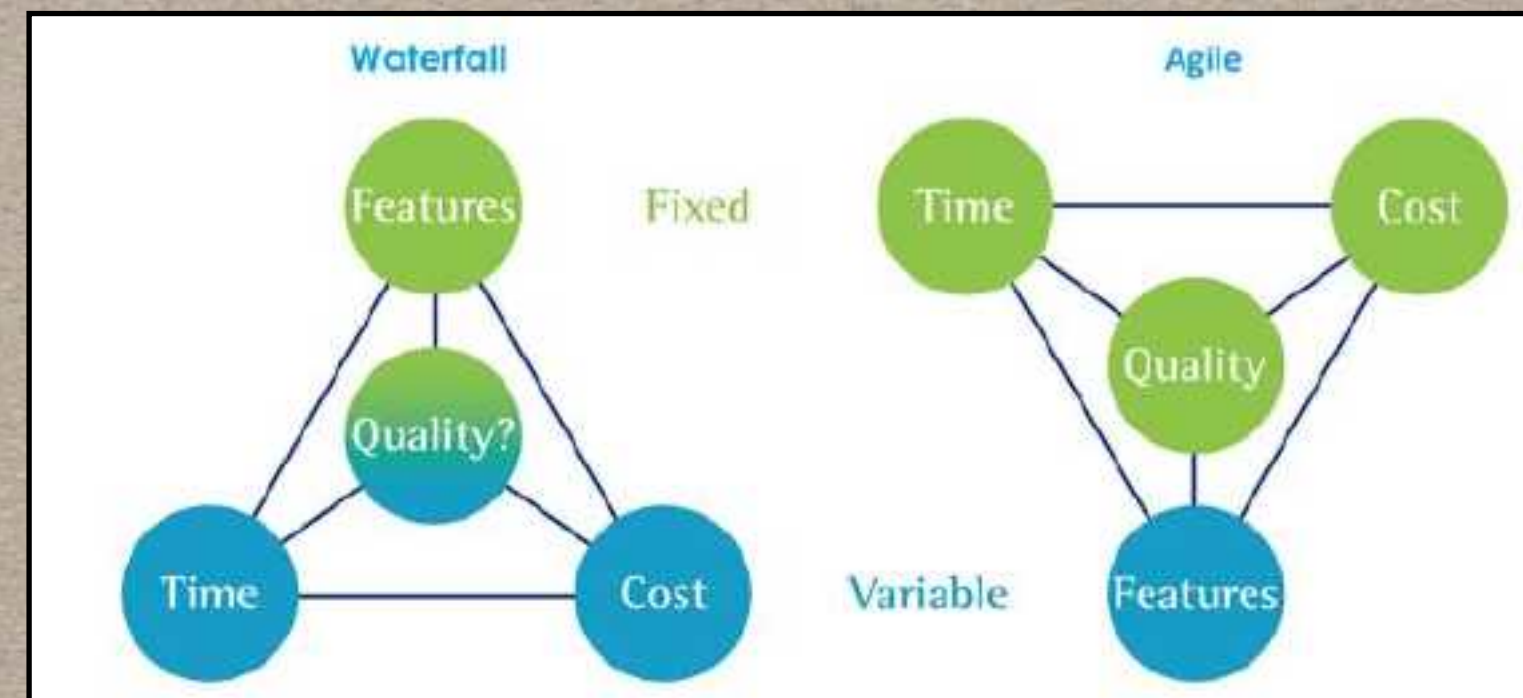
AGILE APPROACH

In this approach, **time, quality** and **costs are fixed**, and therefore the **scope** and **features are variable**

The Agile Manifesto does not prescribe any methodology; it provides a set of **values and principles on which agile methodologies are based**

After gaining experience with a methodology through it's use, it should be **individually customised to fit the organization** (people and culture) **as well as the projects** (defined success criteria, existing technologies and tools, processes)

AGILE VALUES & PRINCIPLES

Individuals & Interactions   Vs.   Processes & Tools

Working Software   Vs.   Documentation

Customer Collaboration   Vs.   Contract Negotiation

Responding to Change   Vs.   Following a Plan

Agile Manifesto: "There is **value** in the **items** on the **right**, although we **value** the **items** on the **left more**"

6

THE ECONOMIC CASE FOR AGILE SOFTWARE DEVELOPMENT

The Standish Group Chaos Report (2002) and a Cutter Consortium Report (2007) studied the use of completed features in over 1,000 projects, producing the following results:

- ☐ 45% Never Used

- ☐ 19% Rarely Used

- ☑ 16% Sometimes Used

- ☑ 13% Often Used

- ☑ 7% Always Used

Software Features Used

Legend: Never Used — Rarely Used — Sometimes used — Often Used — Always Used

7
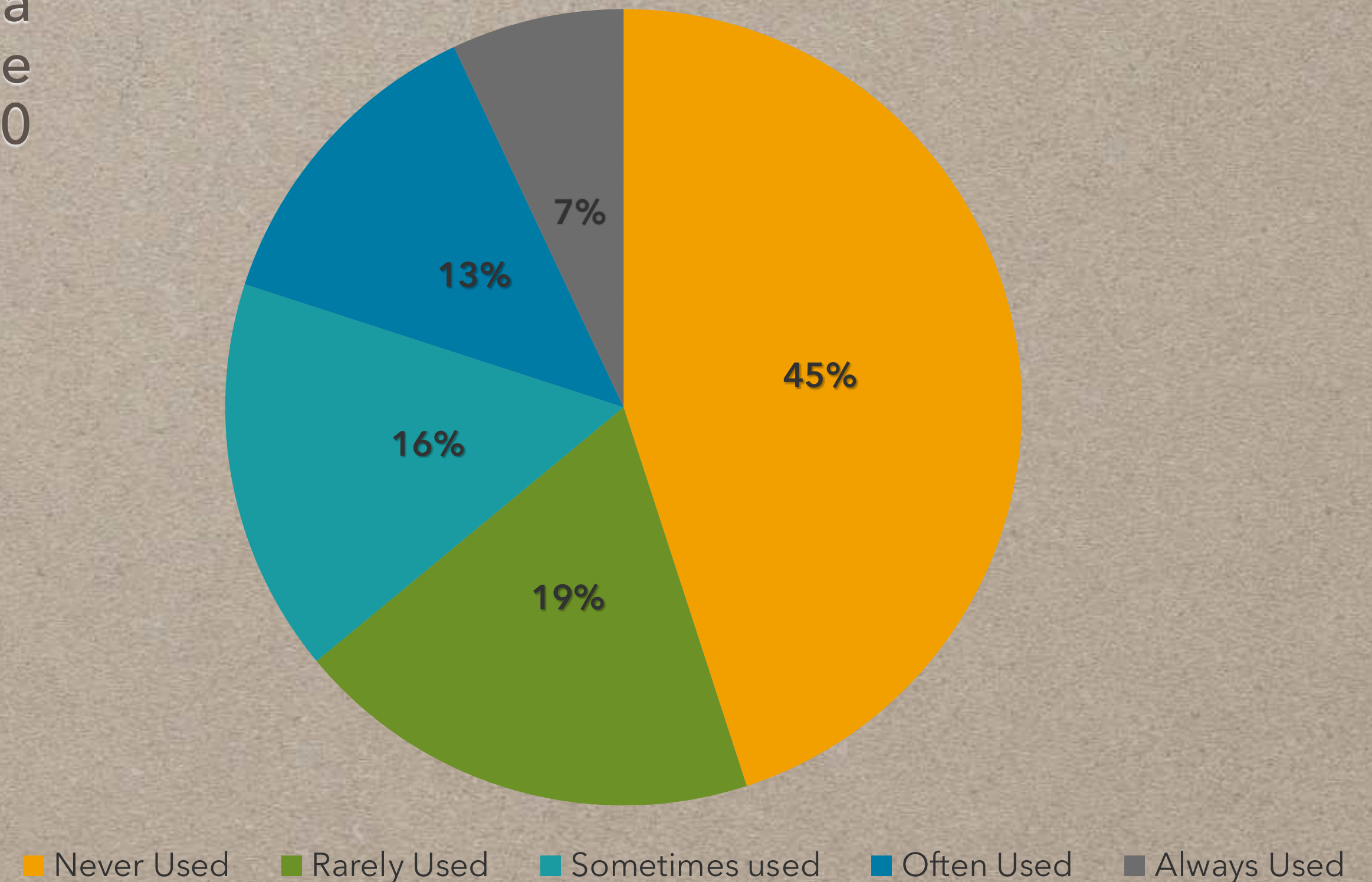
WHEN (IN PRINCIPLE) DOES AGILE WORK BEST?

☑ The agile team is able to adapt to new challenges and work independently

☑ Can afford to iterate and don't need to deliver a fully functional software at once on a specific date

☑ Can't map out the business needs, so the design needs to emerge through trial and error

☑ Unlimited access to the customer / stakeholder who is ready for extensive involvement

☑ No limit to timing, or the estimation of schedule is not possible

WHEN (IN PRINCIPLE) AGILE MIGHT NOT THE BEST APPROACH

☐ Need to capture the market before there is any competition

☐ There is strict schedule (e.g. regulatory mandate)

☐ Unable to change the project's size and content when the team begins work on it

8

AGILE DEVELOPMENT: BENEFITS & DISADVANTAGES

Benefits of Agile	Disadvantages of Agile
Early Delivery: If performed correctly, organisations could often have solutions (partial) delivered earlier, getting sole level of value sooner	**Non-effective long-term planning:** At the onset of the project, it is difficult to accurately determine the amount of time and money that will be needed to complete, due to constantly changing requirements and **no vision of what the final product will look like** or **how much it will cost**
Customer Satisfaction: Usually results in higher degree of customer satisfaction, with better adaptation to changes and faster responses	**Lack of Clear Project Boundaries:** High risk that a project could lead to an uncontrolled expansion of scope for unexperienced product owners, which can cause the initiative to never reach completion
Incremental Value: Because of the constant releases, the product can gain value with every deployment, and provide returns before reaching the projected end	**Requires High Level Skillsets:** The teams need to have a solid foundation and compatible skill level in their areas of expertise and within the market / business goals of the product / project
Stakeholder Involvement: Continued involvement of business stakeholders increases transparency and collaboration	**High stakeholders' expectations management:** A high level of interaction between the business and the developers is required, which could take time if the company's culture and structure are set up differently. Stakeholders might feel inconsequential if their feature or proposal gets re-prioritised downwards
Higher Value Delivered Early: Opportunity to change and reprioritise requirements based on market or business conditions and feedback	**More suited for small to medium size organisations:** The fewer people there are in a team, or set of teams, the easier it is to make a decision and respond to change where the "fail fast" is the dominant mantra

ADOPTION / TRANSFORMATION TO AGILE

	Keeping Existing Methodology	Inserting* a New Agile Framework
Benefits	• Experience and knowledge of the existing practice allowing it to adapt as it matures	• Should empower technology-oriented groups to produce **solutions guided by the business vision** • **Great for Product Experimentation**
Risks	• Models **could be** expensive and / or inefficient	• Complex and conceptual at insertion • Resistance to change • Non applicable for non-IT projects • Regulatory projects with hard deadlines require a centralised decision-making structure and fixed schedule
Challenges	• **Scope modifications** to adapt to changing market or client conditions without impacting the project's performance	• **Non-effective resource/capital planning** due to the teams' lack of knowledge of what the end product will look like • Changes in working and reporting structure for agile teams' autonomous teams
Impediments	• No comparison to different approaches and possibly no improvement in project delivery	• "**See as you go**" cycles make it **difficult to set KPI's** at the start of the projects making tracking and reporting subjective and arbitrary

* For a cross-enterprise (all teams) transformation, when an organisation is already using an established project delivery methodology (including Agile)

EPILOGUE

Many organisations focus on following a prescribed methodology to become "agile", which many times undermines the real reason for the transformation, loosing focus on the grounds behind a project or initiative, and focusing only on team performance and "agility" metrics.

The structure, support, advantages, and disadvantages of the different agile methodologies need to be weighed against one another to **select the best option** given the **characteristics of the organisation, its culture, stakeholders, customers, and most importantly the nature of the projects to be delivered.**

Organisations should focus on the business results of the initiatives and reason for change. Business results should always be the basis of the transformation and the projects it delivers.

Revolutionary deliveries do not result from thinking that one size fits all, or by projecting it will work because someone else is doing it. They arise from asking the question: ***What's the best way to solve a problem?***

"There is nothing quite so useless, as doing with great efficiency, something that should not be done at all ..."

Peter F. Drucker

DISCLAIMER AND DISCLOSURE

- **Third-party Content:** This document references frameworks, tools and software that are publicly available and proprietary to their respective creators. The author does not claim ownership or affiliation with these third-party products.

- **Transparency Notice:** While every effort has been made to ensure accuracy, readers should verify information independently and consult official sources or documentation for the mentioned tools and software.