BITS Pilani
Pilani Campus

BITS Pilani presentation

K.Anantharaman
Faculty CS Department
kanantharaman@wilp.bits-pilani.ac.in

BITS Pilani

Pilani Campus

SE ZG544 , Agile Software Process Lecture No. 2 – Module-2 : Agile Software Development

Module-2 Topics

1. Project Development Life Cycle
   a) Predictive life cycle
   b) Iterative life cycle
   c) Incremental life cycle
   d) Agile/Adaptive life cycle
2. Spiral Model
3. Rational Unified Process
4. Early Agile Methods
5. Agile Mindset

BITS Pilani

Pilani Campus

Project Life cycle models

Life Cycle

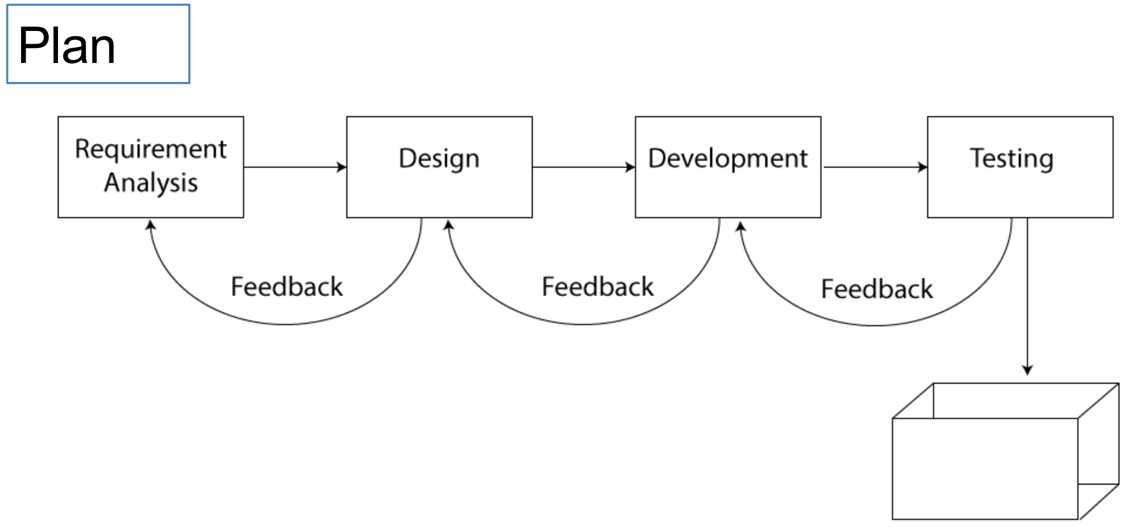
- The **sequence of actions** that must be **performed** in order to **build a software** system

- **Ideally** thought to be a **linear** sequence: **plan, design, build, test, deliver**
  - ➢ This is the waterfall model

- **Realistically** an **iterative process**
  - ➢ Iterative, Incremental, Agile Process

Predictive Project Development Life Cycle (Fully Plan-Driven aka Waterfall)

- A more **traditional approach**, with the bulk of **planning** occurring **upfront**, then executing in a **single pass**; a **sequential** process

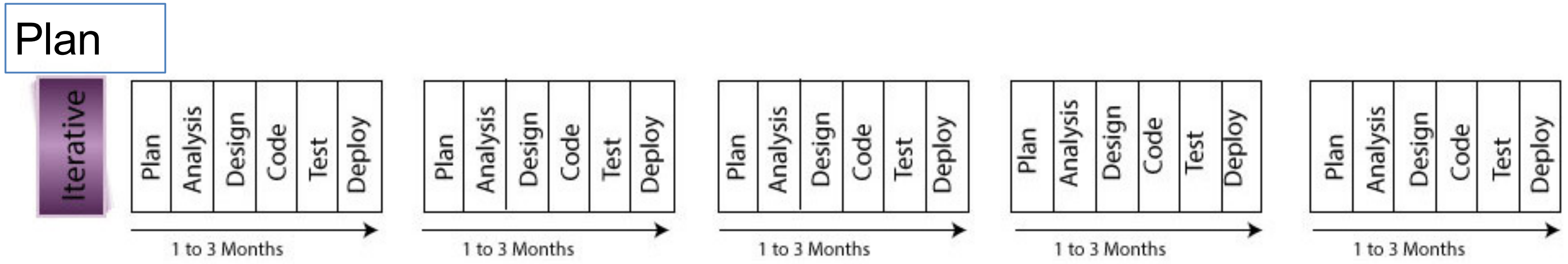


- Requirements/Scope is fixed
- Single delivery
- Goal: Manage Cost
- Minimal feedback changes
- Team is matured in estimation, technology etc..
- Project governance model exists
- **Don't expect long feedback cycle**, If this happens, this lifecycle not suitable for the project

Source : https://www.izenbridge.com/blog/project-management-life-cycle-iterative-adaptive/

Iterative Project Development Life Cycle

- Iterative development is when an attempt is made to **develop a product with basic features**, which then goes through a **refinement process** successively to **add to the richness** in features.



- Goal: **Correctness** of Solution
- **Repeat** until Correct
- **Show** and **receive feedback**
- **Add richness or features**
- **Single Final Delivery**

- **Deliver result** at the end of each iteration.
- Result may **not** be **usable**
- E.g. 1 year project divided into 3 to 4 iterations

Ref: https://www.izenbridge.com//

Examples of Iterative Development

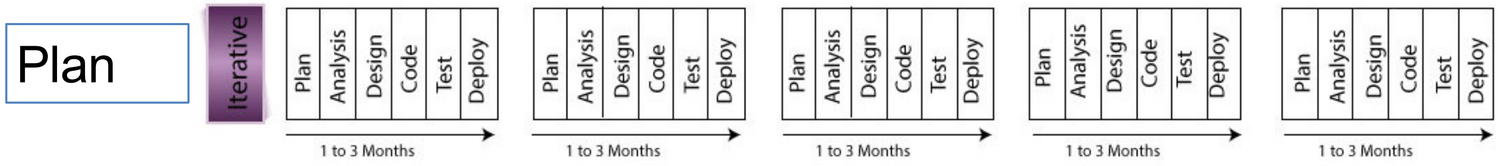
- When you are getting a customized coat made
  - You may be required to go for a trail to check for the fitting.
  - Even though the you may find the coat fitting well, you may not be able to use it as it has not been finished.
  - The fitting test was to give you an idea of the final product, which may not be ready for your consumption.
  - This is an example of iterative prototyping.

- Developing a Website
  - Develop a prototype of the Website with basic functionality
  - Demo to Customer and receive feedback
  - Add to the richness or feature to the product in subsequent iteration

Source: https://www.izenbridge.com//

Incremental Life Cycle

- In an incremental approach, one aims **to build pieces of program/product that is complete in features** and richness. Product **increment** is **usable**.
- In this case, each functionality is built to its fullest and **additional functionalities are added in an incremental fashion.**



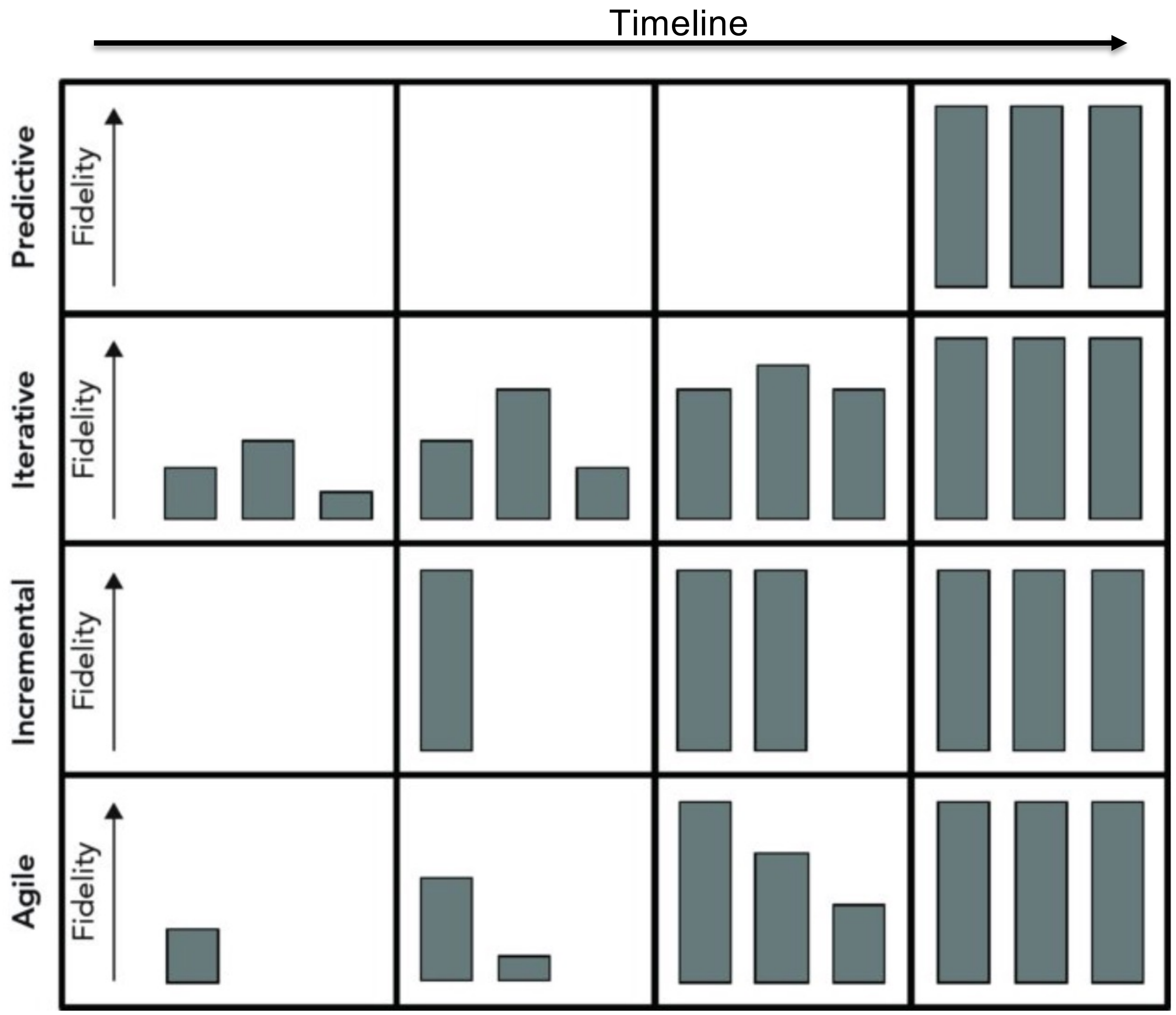
Example: You can compare this to a visit to restaurant. You get served starters first and on completion of its main course and then dessert. You get served incrementally and you consume it.

Source: https://www.izenbridge.com//

Project Life Cycle Models

- Different development approaches add features and increase fidelity in different ways.

Fidelity: the degree of exactness with which something is copied or reproduced.



Goals/Characteristics

Cost, Requirements-Fixed, Single Delivery

Accuracy/Correctness, final single delivery
Intermediate delivery – may no be usable
Example: Customized outfit/coat, Website

Accuracy/Correctness, Multiple Delivery,
Intermediate delivery –usable
Example: Visit to a restaurant

Accuracy/Correctness, Multiple Delivery,
Intermediate delivery –usable
Example: Game: Treasure Hunt with clues

Ref: Beginning Software Engineering by Rod Stephens

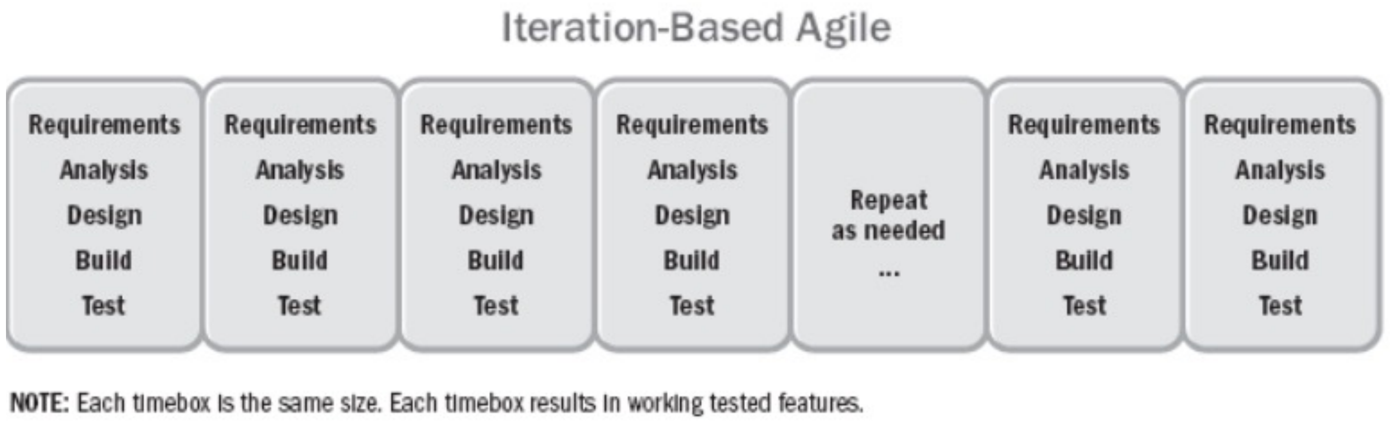
Agile Life Cycle Models

- Iterative
- Flow-Based

Agile/Adaptive Life Cycle- Iteration Based Agile

- The project life cycle that is **iterative and incremental**

Plan


Iteration-Based Agile
Requirements
Analysis
Design
Build
Test
Repeat as needed ...
NOTE: Each timebox is the same size. Each timebox results in working tested features.

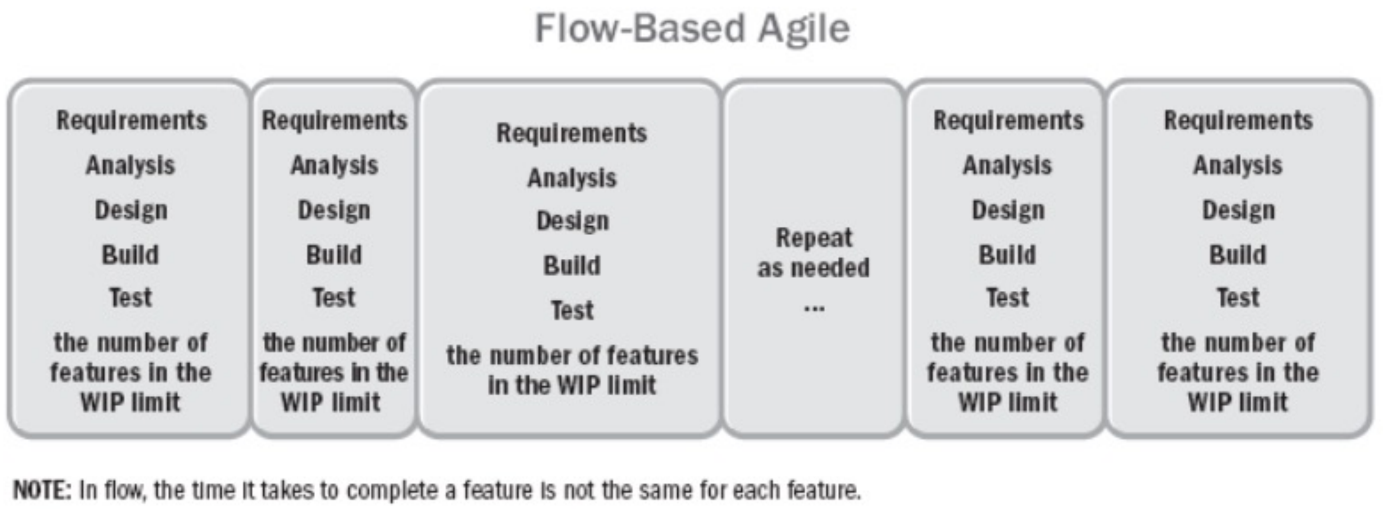
Fixed Time box : 1-4 weeks equal duration for each iteration
Goals and Characteristics: Value, Multiple deliveries

Re.f: Agile Practice Guide (ENGLISH) by Project Management Institute Published by Project Management Institute, 2017 (Agile methodologies)I

Agile/Adaptive Life Cycle- Flow-based Based Agile

- The project life cycle that is **iterative and incremental**

Plan

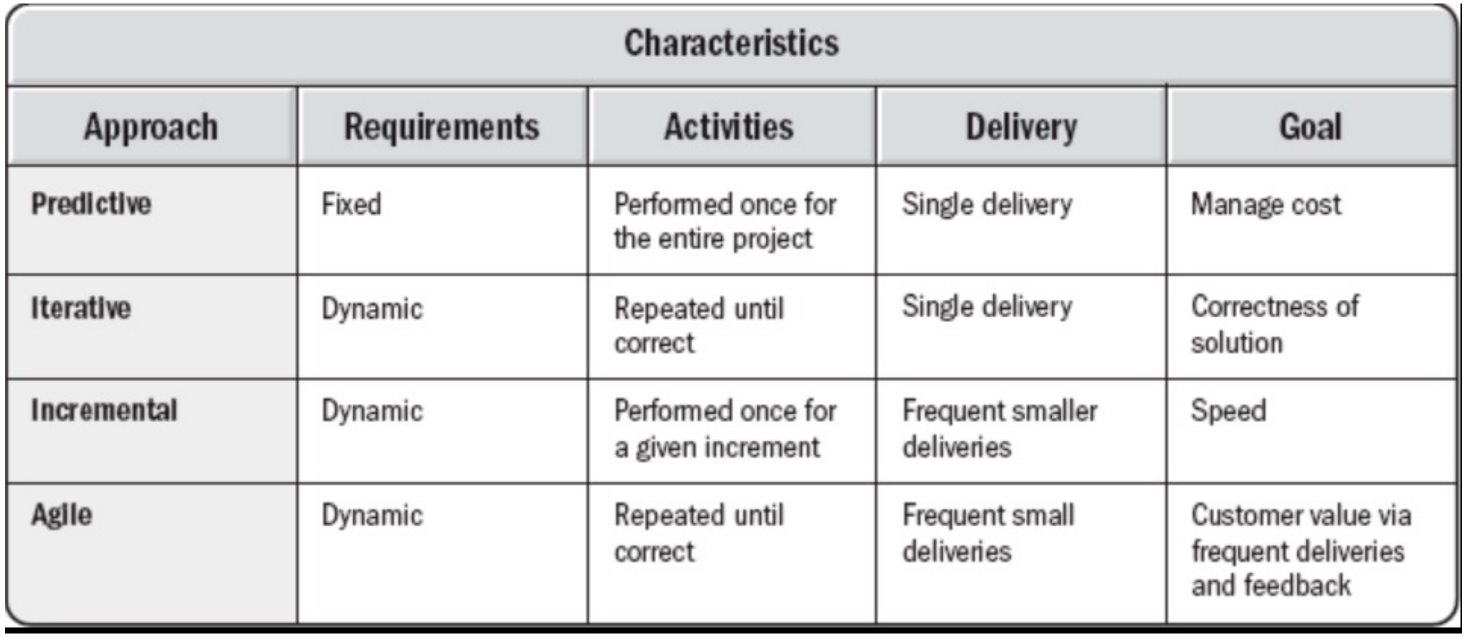

Flow-Based Agile
Requirements
Analysis
Design
Build
Test
the number of features in the WIP limit
Repeat as needed
...
NOTE: In flow, the time it takes to complete a feature is not the same for each feature.

Variable Time Box :

Goals and Characteristics: Value, Multiple deliveries

Ref: Agile Practice Guide (ENGLISH) by Project Management Institute Published by Project Management Institute, 2017 (Agile methodologies)l

Project Life Cycles Characteristics



Characteristics				
Approach	**Requirements**	**Activities**	**Delivery**	**Goal**
Predictive	Fixed	Performed once for the entire project	Single delivery	Manage cost
Iterative	Dynamic	Repeated until correct	Single delivery	Correctness of solution
Incremental	Dynamic	Performed once for a given increment	Frequent smaller deliveries	Speed
Agile	Dynamic	Repeated until correct	Frequent small deliveries	Customer value via frequent deliveries and feedback

- It should be emphasized that **development life cycles are complex and multidimensional.**
- Often, the **different phases in a given project employ different life cycles**, just as distinct projects within a given program may each be executed differently.

Ref: Agile Practice Guide (ENGLISH) by Project Management Institute Published by Project Management Institute, 2017 (Agile methodologies)l

Q&A

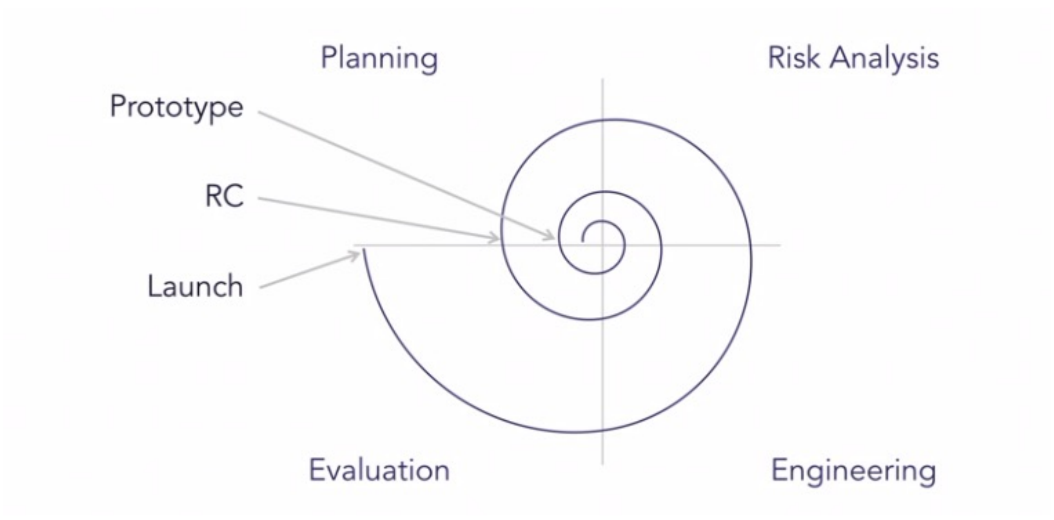
- Set1

BITS Pilani

Pilani Campus

Some Popular Iteration Models

Spiral Risk Driven Customer driven Planning



- Developed by Barry Boehm, 1986.
- Easier management of risks (Theme)
- Mix of water fall and iterations
- Y-Axis represents Cost
- X-Axis represents Review
- Prototype-1, Prototype-2 ….
- Operational Prototype
- Final Release

Four Phases
Planning: Requirements Identification and Analysis
Risk Analysis: Risk identification, Prioritization and Mitigation
Engineering: Coding, Testing and Deployment
Evaluation: Review and plan for next iteration

Ref: Agile Software Development with Shashi Shekhar,LinkedIn Learning

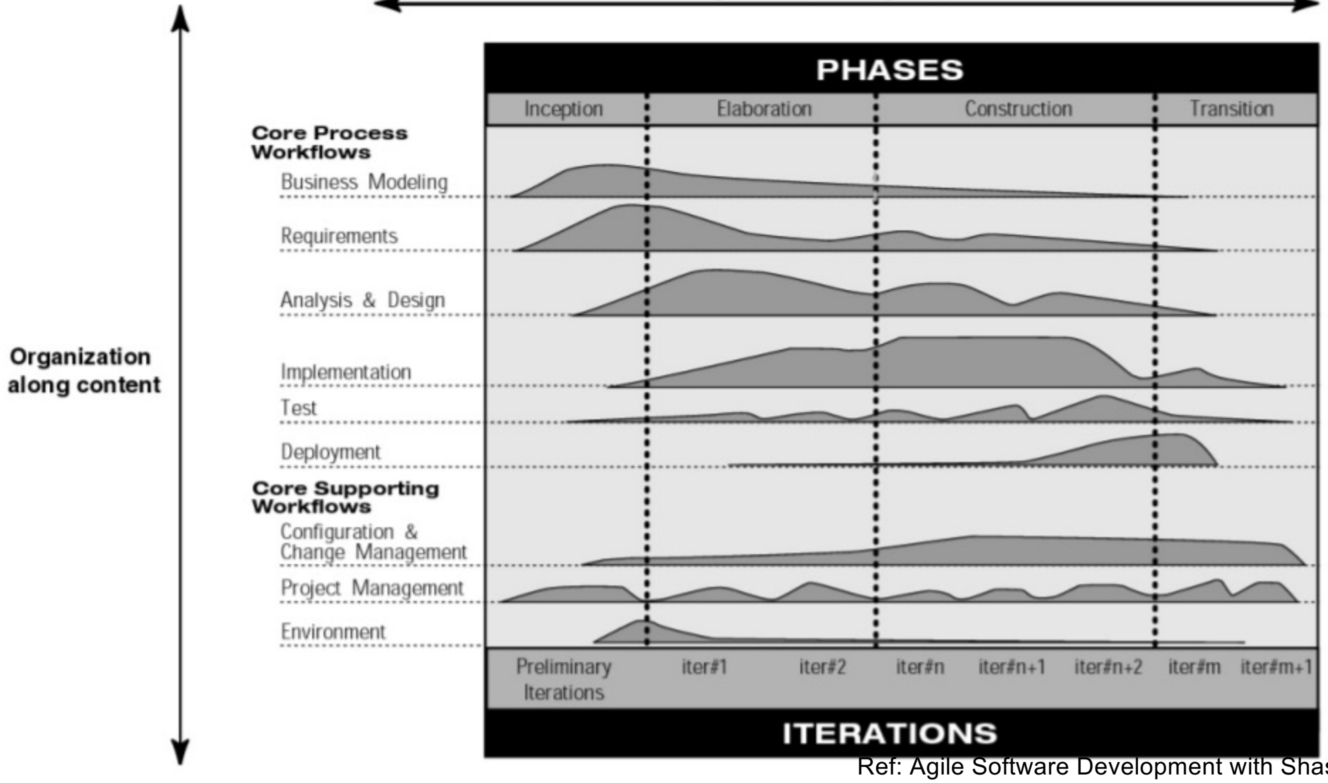
Rational Unified Process (RUP)

- 1990s, Rational Software developed the Rational Unified Process as a software process product.
- IBM acquired Rational software in 2006 (**era of OOAD,UML**)
- Rational Unified Process, or RUP, was an attempt to come up with a comprehensive **iterative software development** process.
- RUP is essentially a **large pool of knowledge**. RUP consists of **artifacts, processes, templates, phases**, and disciplines.
- RUP is defined to be a **customizable** process that would work for building small, medium, and large software systems.

Ref: Agile Software Development with Shashi Shekhar,LinkedIn Learning

RUP Iterative Model

Inception	Elaboration	Construction	Transition
Lifecycle Objectives Milestone	Lifecycle Architecture Milestone	Initial Operational Capability Milestone	Production Release Milestone



X-Axis: RUP Phases, Dynamic

Y-Axis: Organization Process, Static

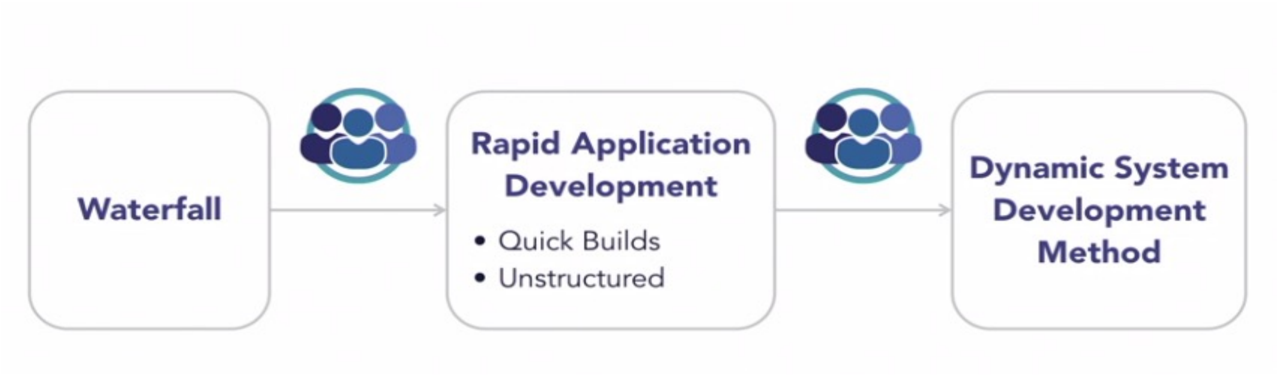
Ref: Agile Software Development with Shashi Shekhar,LinkedIn Learning

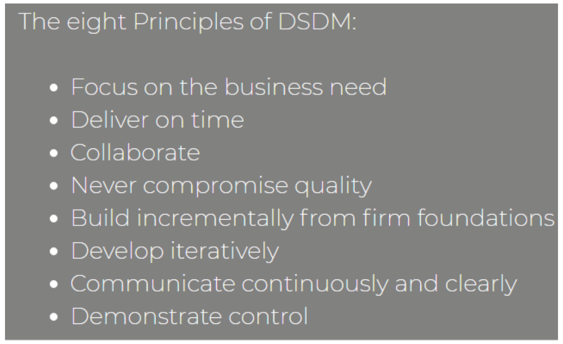
innovate achieve lead

BITS Pilani
Pilani Campus

Early Agile Methods

Dynamic System Development Method (DSDM)


Waterfall
Rapid Application Development
• Quick Builds
• Unstructured
Dynamic System Development Method


The eight Principles of DSDM:

- Focus on the business need
- Deliver on time
- Collaborate
- Never compromise quality
- Build incrementally from firm foundations
- Develop iteratively
- Communicate continuously and clearly
- Demonstrate control

- Developed in 1994
- Era where organization slowly moving away from waterfall model
- During this time RAD model came into existence
- RAD approach is very agile but has no formal process
- DSDM was formed by group of organizations
- **Project development standard in Europe** for several years
- In 2016 DSDM is changed its name to **Agile business consortium**

https://www.agilebusiness.org/

Source:Agile Lynda.com, Agilebusiness.com

Feature Driven Development (FDD)

- Lightweight Agile process
- Software is a collection of features
- Software feature ="working functionality with business value"

Feature Example:
Calculate monthly interest on the account balance
(action) (result) (object)

- Deliver working software (working feature)
- Short iterative process with five activities
  - **Develop over all, Build Feature list, Plan by feature, Design by Feature Build by feature**
- FDD is used to build large banking systems successfully

Ref: Agile Software Development with Shashi Shekhar,LinkedIn Learning

Crystal Method- Selecting a Model

Criticality

Life		🟨	🟧	🟥	🟫	🟪
Essential Money		🟨	🟧	🟥	🟫	🟪
Discretionary Money		🟨	🟧	🟥	🟫	🟪
Comfort		🟨	🟧	🟥	🟫	🟪
	1-6	7-20	21-40	41-80	81-200	Large

Team Size

- **Different crystal methodologies based on team size.**
- **If Criticality increases tweak the process to address the extra risk**

Comfort: System malfunction
Discretionary Money: Extra savings
Essential Money: Revenue loss
Life: Loss of life , Critical software

- Crystal methods are people-centric, light-weight, and highly flexible. Focus on People, Interactions,Colloborations.
- Developed by Alistair Cockburn , 1991

Ref: Agile Software Development with Shashi Shekhar,LinkedIn Learning

Q&A

- Set2

BITS Pilani

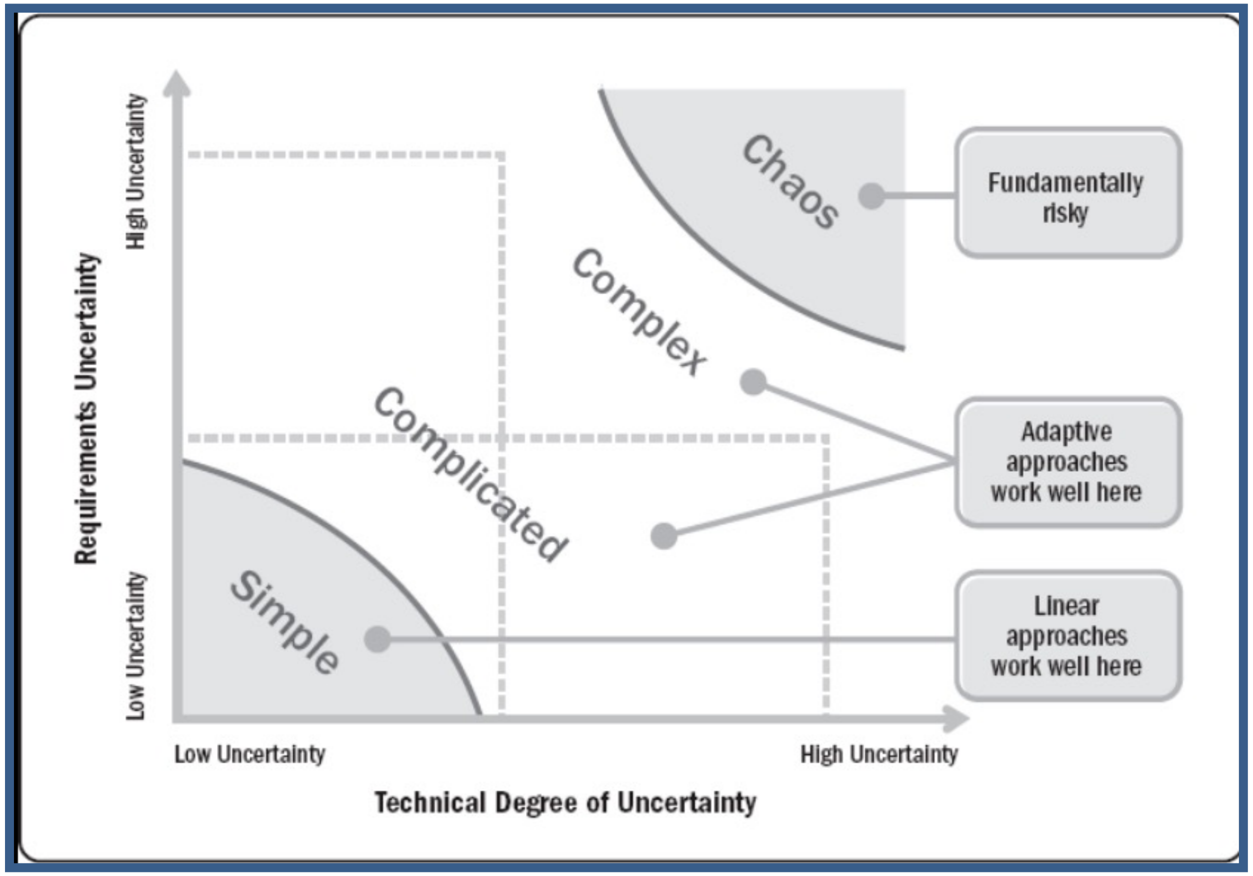
Pilani Campus

Project Classifications/Decision making models

Decision Making Models

- **Stacey's Complexity Model:** Organizes problems into four categories (Simple, Complicated, Complex, Chaotic) based on their level of clarity and predictability.
- **Cynefin framework:** Helps leaders make decisions by categorizing problems into domains (Simple, Complicated, Complex, Chaotic, Disorder) based on their nature and relationships between cause and effect.
- **Use cases of these Models:**
  - Leaders can use these models to understand the nature of a problem they face and make appropriate decisions.
  - Project managers can use the models to select the most suitable approach based on the complexity of their projects.
  - Teams can use the frameworks to recognize when a problem is shifting from one domain to another, requiring a change in their approach.

Agile Suitability - Project Environment Stacey's Complexity Model



Project Examples:

•**Simple:** Setting up a new employee onboarding process with clear steps to follow.

•**Complicated:** Building a large-scale IT infrastructure with the help of experienced IT consultants.

•**Complex:** Developing a new product in a rapidly changing market, requiring continuous iteration and learning.

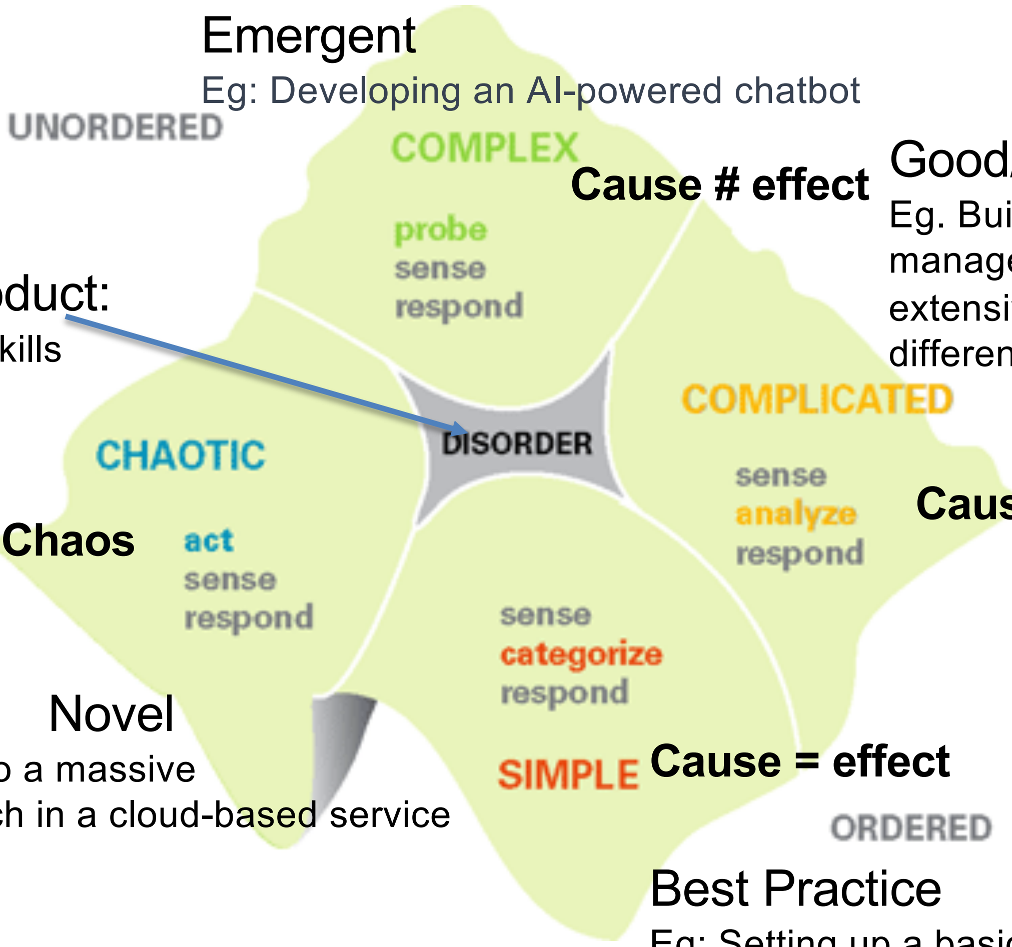
•**Chaotic:** Crisis management during a natural disaster or cyber-attack.

Ref: Agile Practice Guide (ENGLISH) by Project Management Institute Published by Project Management Institute, 2017 (Agile methodologies)

Cynefin framework - A Leader's Framework for Decision Making

Developed in the early 2000s by David Snodown

Pronounced as: *kun-ev-in*



Emergent

Eg: Developing an AI-powered chatbot

Cause # effect

Good/Standard Practice

Eg. Building a customer relationship management (CRM) system with extensive customization options for different clients.

AR Product:

Eg: No Skills

Cause == effect

Chaos

Novel

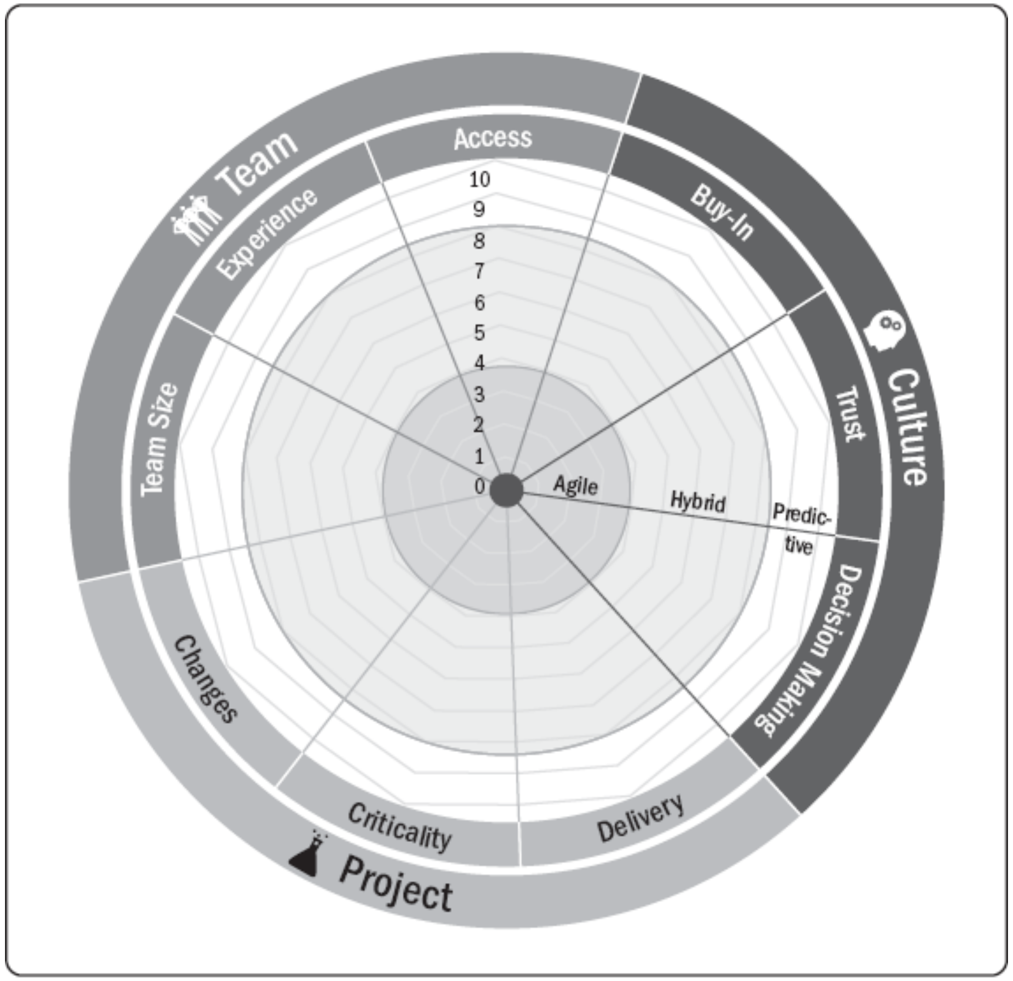
Responding to a massive security breach in a cloud-based service

Cause = effect

Best Practice

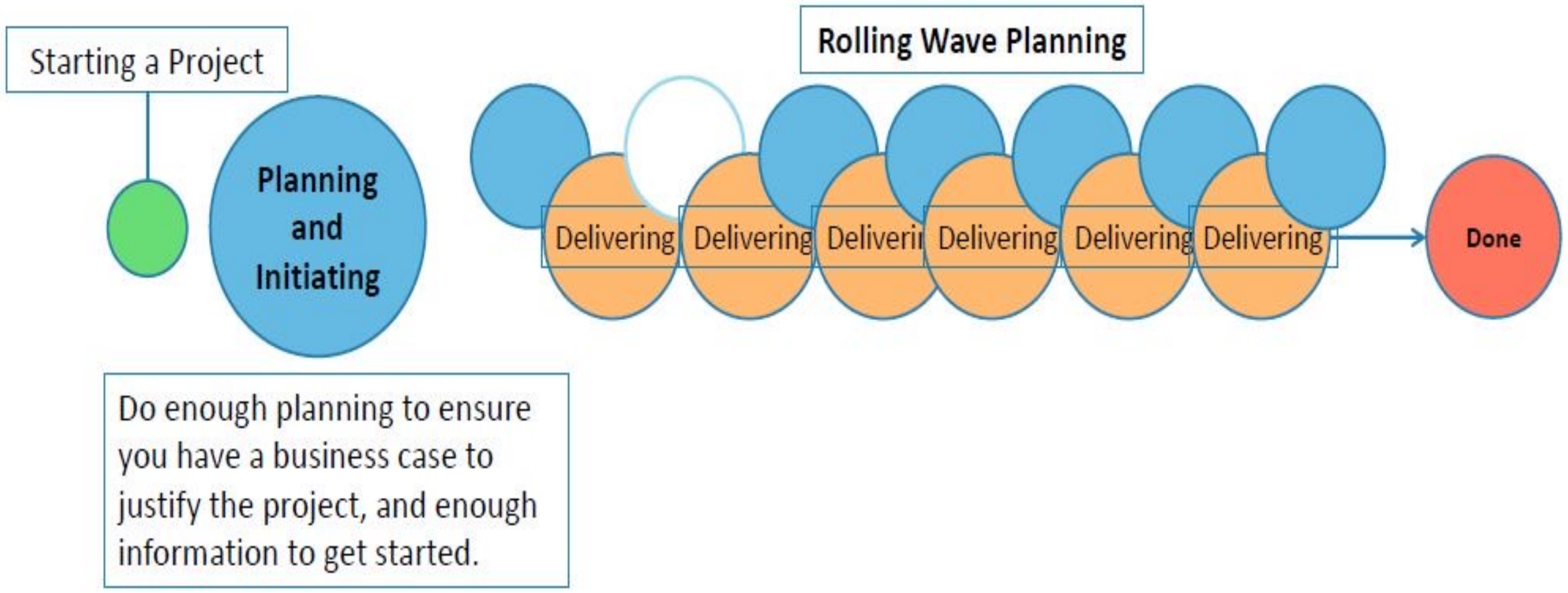
Eg: Setting up a basic website for a small business,

Agile Suitability Filter



Attributes	Assessment (wrt Agile)
Buy-In	0-Yes, 5-Partial, 10-No
Trust	0-Yes, 5-Probabily, 10-No
Decision Making	0-Yes, 5-Probably, 10-Unlikely
Incremental Delivery	0-Yes, 5-Maybe/Sometimes, 10-Unlikely
Criticality	0-Low, 5-Medium, 10-High
Changes	0-High, 5-Medium, 10-Low
Team Size	1-Small (<10), 5-Medium (>80), 10- Large (>200)
Experience	0-Yes, 5-Partial, 10-No
Access to business Info/Project Info	0-Yes, 5-Partial, 10-No

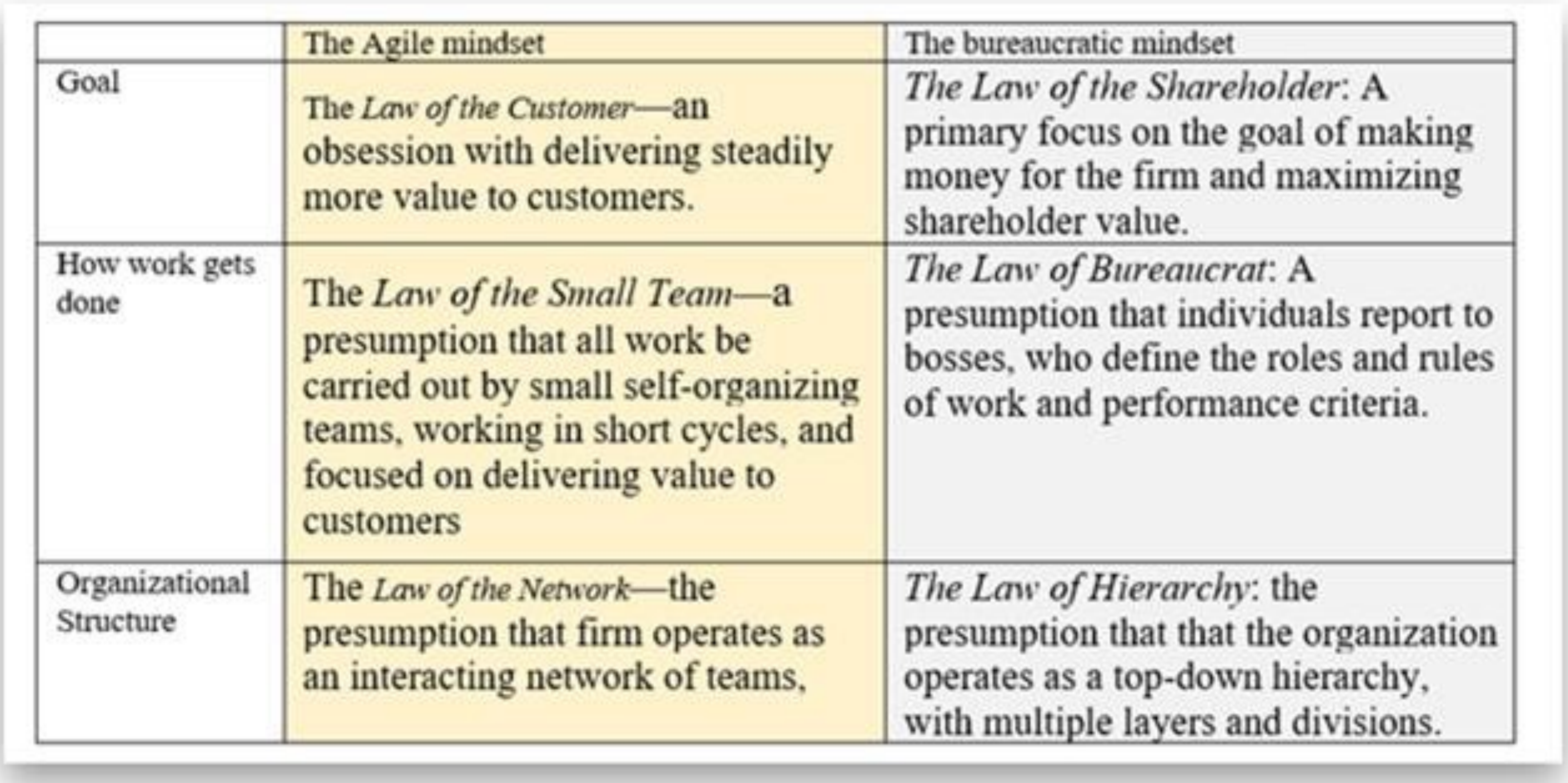
Rolling Wave Planning or Progressive Elaboration



Do enough planning to ensure you have a business case to justify the project, and enough information to get started.

https://www.simplilearn.com/adaptive-planning-part-1-tutorial

Mindset



	The Agile mindset	The bureaucratic mindset
Goal	The *Law of the Customer*—an obsession with delivering steadily more value to customers.	*The Law of the Shareholder*: A primary focus on the goal of making money for the firm and maximizing shareholder value.
How work gets done	The *Law of the Small Team*—a presumption that all work be carried out by small self-organizing teams, working in short cycles, and focused on delivering value to customers	*The Law of Bureaucrat*: A presumption that individuals report to bosses, who define the roles and rules of work and performance criteria.
Organizational Structure	The *Law of the Network*—the presumption that firm operates as an interacting network of teams,	*The Law of Hierarchy*: the presumption that that the organization operates as a top-down hierarchy, with multiple layers and divisions.

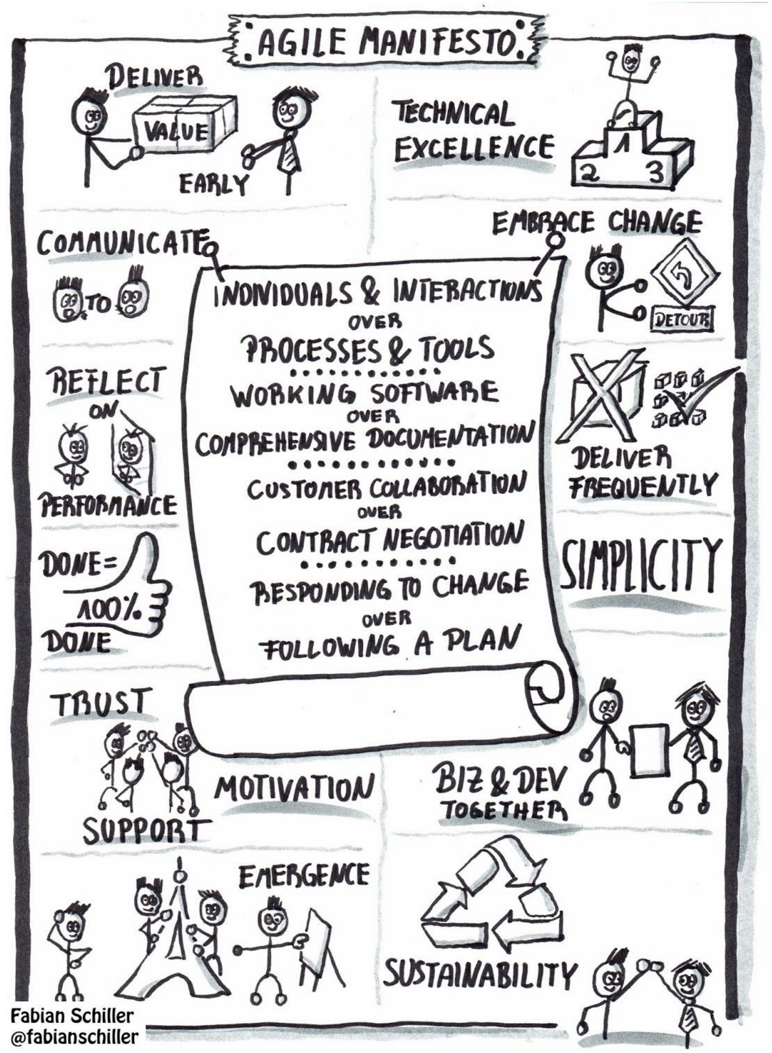
Source: https://www.forbes.com/sites/stevedenning/2019/08/13/understanding-the-agile-mindset/?sh=5a66a5545c17

BITS Pilani
Pilani Campus

Agile Mindset

s

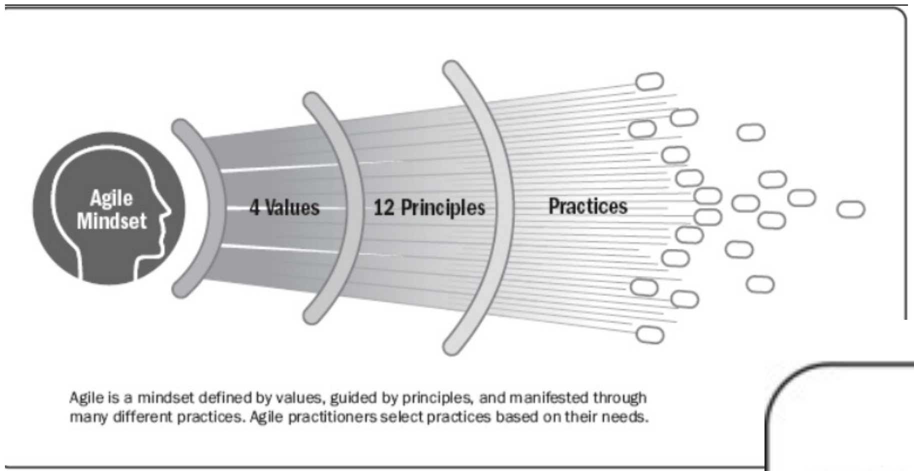
Agile Manifesto & Principles (Basis of Agile Mindset)



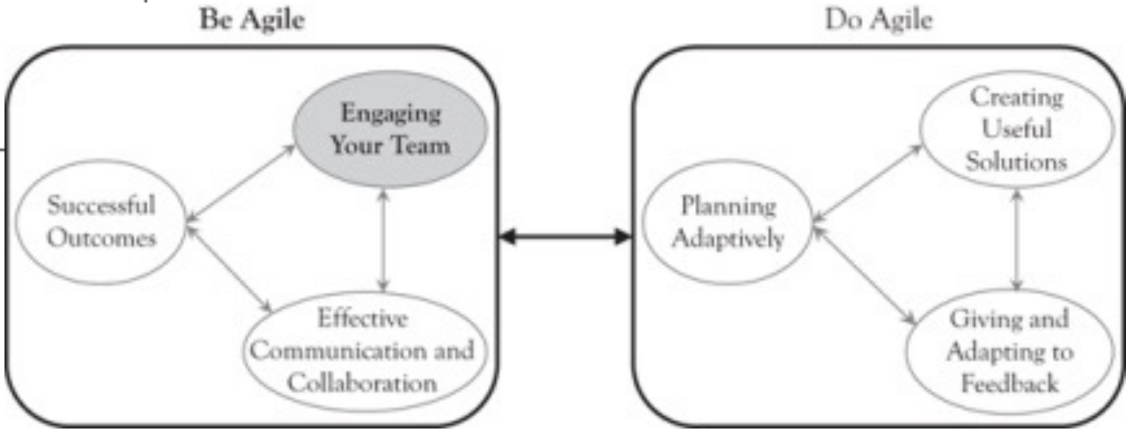
- In 2001, a group of software and project experts got together to talk about what their successful projects had in common.
- This group created the Agile Manifesto, a statement of values for successful software development
- Then they wrote Agile Principles

We will discuss this topic in more detail in the next module

Agile Mindset (Be Agile and Do Agile)



Agile is a mindset defined by values, guided by principles, and manifested through many different practices. Agile practitioners select practices based on their needs.



Be Agile
Engaging Your Team
Successful Outcomes
Effective Communication and Collaboration
Do Agile
Creating Useful Solutions
Planning Adaptively
Giving and Adapting to Feedback

An Agile mindset implies that an organization or person has absorbed Agile to the extent that it becomes part of their ‘identity’, i.e. their ‘business-as-usual’ state and the default way they interact with the world.

Agile Mindset …

- It is important to understand that, for **Agile to be successful,** the **right mindset is equally important than merely implementing Agile tools, practices or principles**.

- For example, **having a visual board** does **not necessarily mean that a team is Agile.**

- Experiments have shown that **people can strongly influence** each other to adopt **or reject** an Agile mindset .

Agile Mindset …

- **Agile is a journey, not a destination**
- Teams become more Agile by **embedding the Agile mindset deeper inside themselves and the organization.**
- This process is **facilitated** by **applying Agile values**, practices, **principles** and so on.

Agile is not just a set of rituals

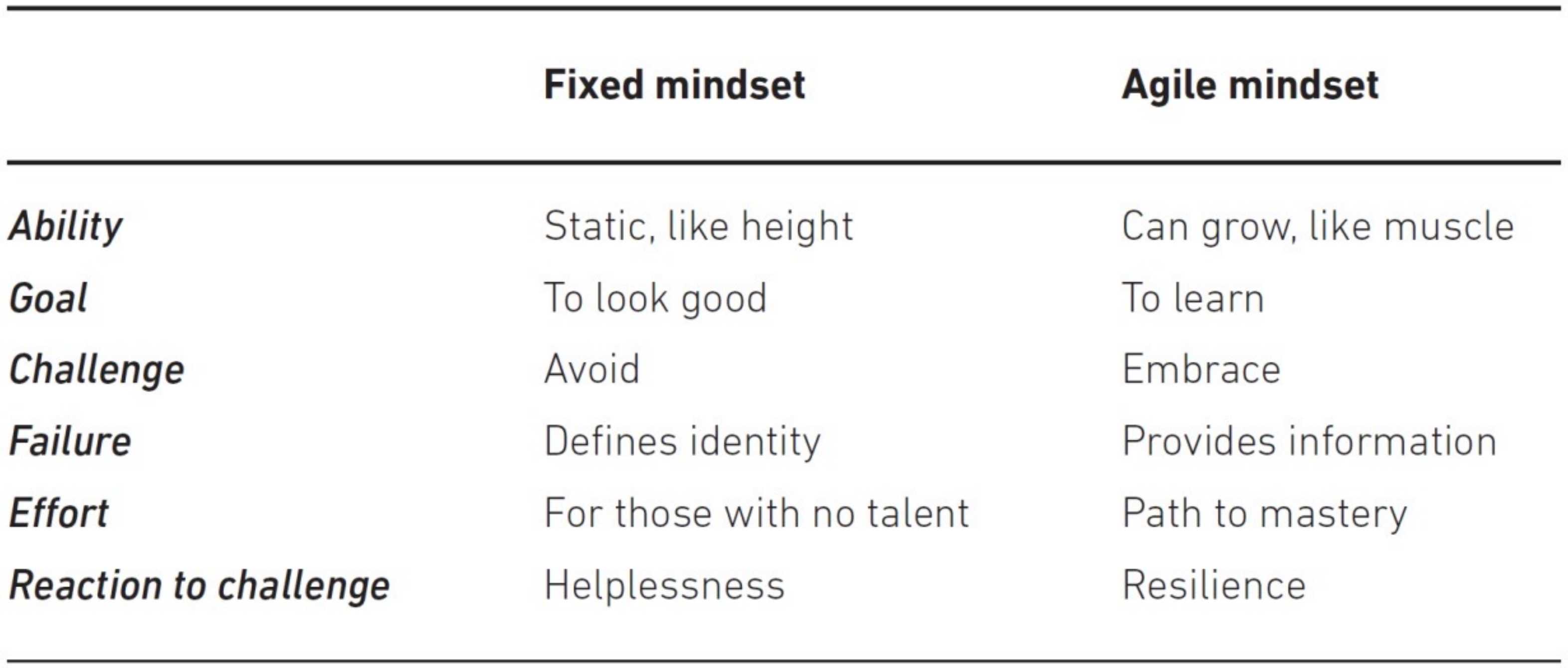
- Where teams may understand what to do, but don’t understand why they are doing it. This will result in **partial success**.
- If this happens teams tend to use Agile practices for a short period of time, but over the **longer term they fall back into their previous ways of working.**
- Because they don’t understand why they are doing what they are doing.

Agile is not just techniques

- People often prefer to **pick out the elements of a framework they can easily understand** rather than understanding an entire framework and why it should be implemented.

- Some Agile frameworks will only **work effectively if all the key activities, roles and artefacts of the framework are in place.**
  - So, for example, people might think that just prioritizing stories within a backlog is the same as 'being Agile', instead of it being just one part of the whole. Additionally, while prioritization of stories is indeed a core practice in Agile, it is also used in more traditional non-Agile delivery approaches.

What is being Agile? Mindset?

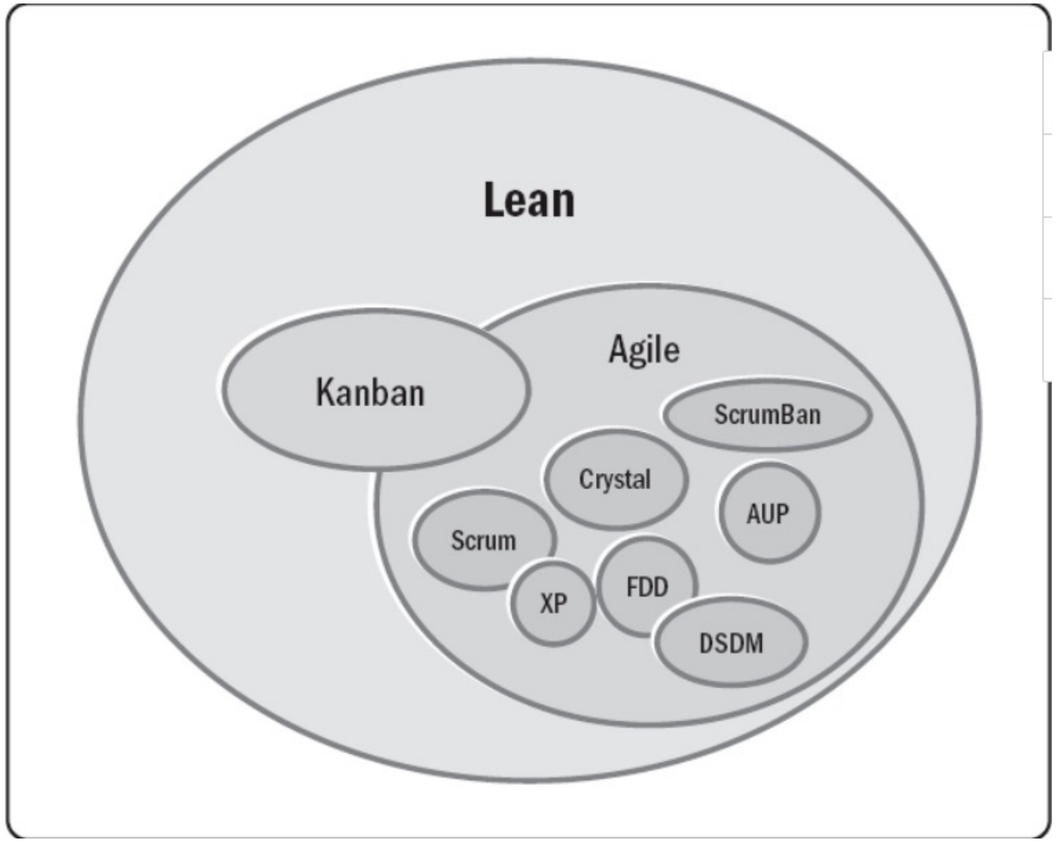


	Fixed mindset	Agile mindset
Ability	Static, like height	Can grow, like muscle
Goal	To look good	To learn
Challenge	Avoid	Embrace
Failure	Defines identity	Provides information
Effort	For those with no talent	Path to mastery
Reaction to challenge	Helplessness	Resilience

- The most important aspect of an Agile mindset is understanding that Agile is neither just a set of rituals that are repeated, nor is it merely based on techniques.

Source: Agile Practice guide, PMI

Agile is a blanket of many approaches



- Is agile an **approach, a method, a practice**, a **technique**, or a **framework**?
- Any or all of these terms could apply depending on the situation.

Source: Agile Practice guide, PMI

Two Approaches to fulfill Agile Values and Principles

- **Adopt a formal agile approach** (Designed and Proven)
  - Take time to understand the agile approaches before changing or tailoring them. Premature and haphazard tailoring can limit benefits.

- **Implement changes to project practices** in a manner that fits the project context **to achieve progress on a core value or principle.**
  - Use time boxes to create features
  - Specific techniques to iteratively refine features.
  - Consider dividing up one large project into several releases

➢ The **end goal** is not to be agile for its own sake, but rather to deliver a **continuous flow of value to customers** and achieve better business outcomes.

Source: Agile Practice guide, PMI

Q&A

- Set 3

BITS Pilani
Pilani Campus

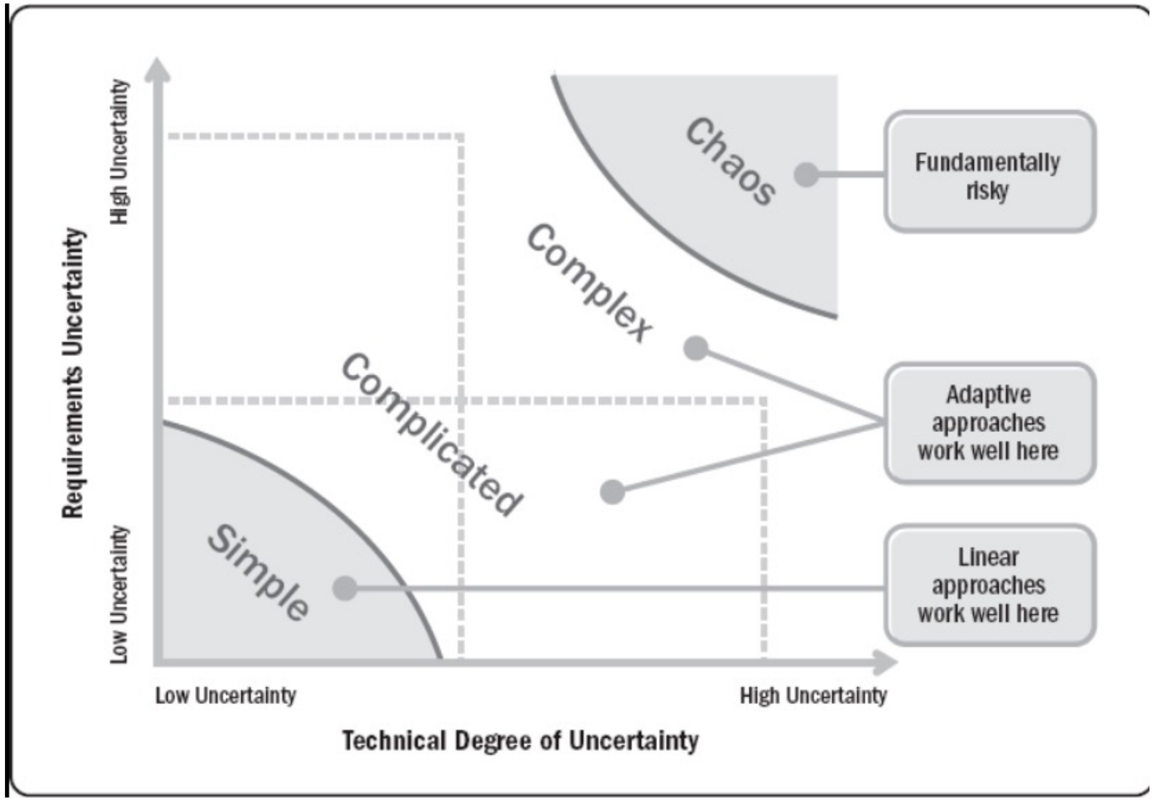
Delivery Environments and Agile Suitability

Delivery Environments

- The environment within which **Project/Product delivery** will occur should largely drive the delivery and **governance framework(s)** that will be implemented.
- For example, in a delivery environment where **high variability** is likely to be encountered (like IT product development), an **Agile framework** would be suited
- In an environment where **variability** is likely to be **low**, a **more defined process** may be more suited (like **'Waterfall')**.

Ref: Agile Foundations - Principles, practices and frameworks by Peter Measey

Understanding the Delivery Environments: Stacey's Complexity Model



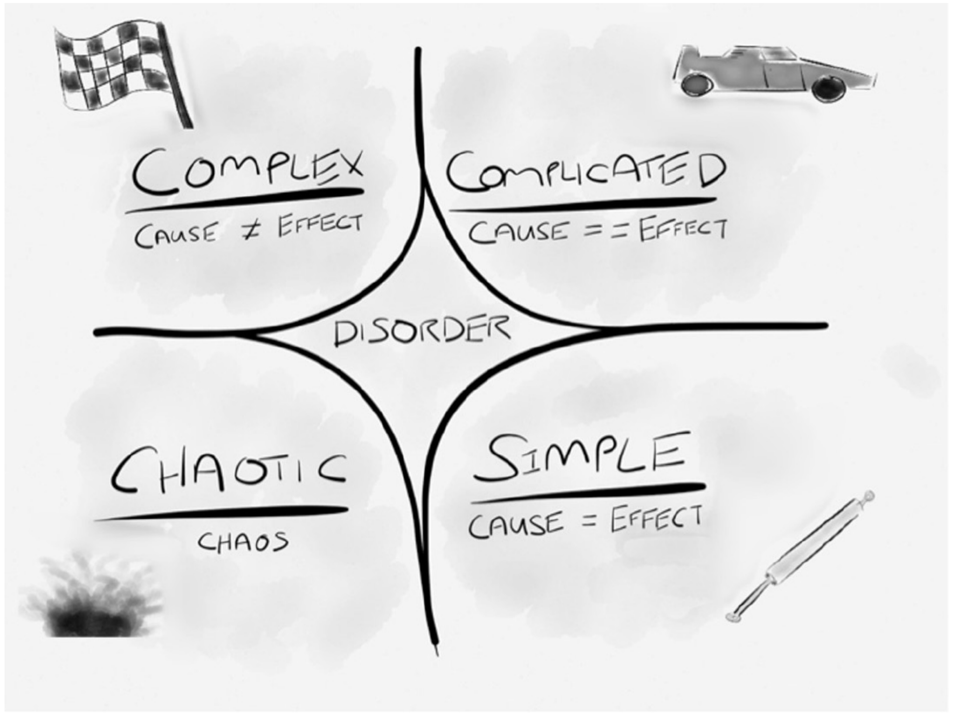
- Simple Environment: Use defined Process like Waterfall
- Complicated/Complex/Anarchy Environment: Use Empirical process like Agile. Example: New IT product development

When trying to understand types of environments, it is important to take into account the amount of innovation that is being sought or considered for a new product or service. As the level of innovation increases, so does the move towards complexity, and a high variability is likely to be present.

Ref: Agile Foundations - Principles, practices and frameworks by Peter Measey

Cynefin Framework for Decision Making

- The Cynefin framework (Snowdon and Boone, 2007) gives an alternative framework for determining and understanding simple, complicated and complex environments



- The central idea of the framework is to offer decision-makers a “sense of place” to view their perceptions in dealing with a situation or problem. Not all situations are equal, and this framework helps to define which response is required for a given situation or problem.

https://txm.com/making-sense-problems-cynefin-framework/

Cynefin identifies five domains:

- **Simple (obvious) domain:**
  - In this domain the relationship between cause and effect is obvious and therefore it is relatively easy to predict an outcome. In this domain predictive planning works well as everything is pretty well understood. Teams can define up front how best to deliver a product, and they can then create a defined approach and plan. The Waterfall model works well in these types of environments with little variability.
- **Complicated domain**
  - In this domain, the relationship between cause and effect becomes less obvious; however, after a period of analysis it should generally be possible to come up with a defined approach and plan. Such a plan will normally include contingency to take into account the fact that the analysis may be flawed by a certain amount. Again, the Waterfall model is suitable for this environment as there is an element of definition up front; however, a more empirical process, like Agile, may be more suited.

https://txm.com/making-sense-problems-cynefin-framework/

Cynefin identifies five domains:

Complex domain

- In this domain the relationship between cause and effect starts to break down as there tend to be many different factors that drive the effect. While it may be possible to identify retrospectively a relationship between cause and effect, the cause of an effect today may be different to the cause of the same effect tomorrow. Creating a defined up-front approach and plan is not effective within this domain and therefore an Agile way of working is recommended.

Chaotic domain

- In this domain, there is no recognizable relationship between cause and effect at all, making it impossible to define an approach up front or to plan at all. Instead, teams must perform experiments (e.g. prototyping, modelling) with the aim to move into one of the other less chaotic domains. An Agile approach can work in this domain, for example Kanban which does not require up-front plans.

Disorder

- Being in this environment means that it is impossible to determine which domain definition applies. This is the most risky domain as teams tend to fall into their default way of working, which may prove unsuitable for what they are trying to achieve.

https://txm.com/making-sense-problems-cynefin-framework/

A Note on
Cynefin identifies five domains:

- During a product's development and evolution there may be **elements of delivery spread across** all the Cynefin domains at the same time.

- There **may be aspects of a large system that are simple**, while **others may be in the complicated domain**; and there could also be areas where innovation is necessary and which require a move towards the complex or even towards the chaotic domain.

https://txm.com/making-sense-problems-cynefin-framework/

Thank you