BITS Pilani presentation

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

BITS Pilani
Pilani Campus

innovate achieve lead

BITS Pilani

Pilani Campus

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

Module-2 Topics

innovate achieve lead

1. Project Life Cycle Models
    - Iterative, Incremental and (Adaptive or Agile) Approaches
2. Early Agile Models
3. Popular Agile Methods

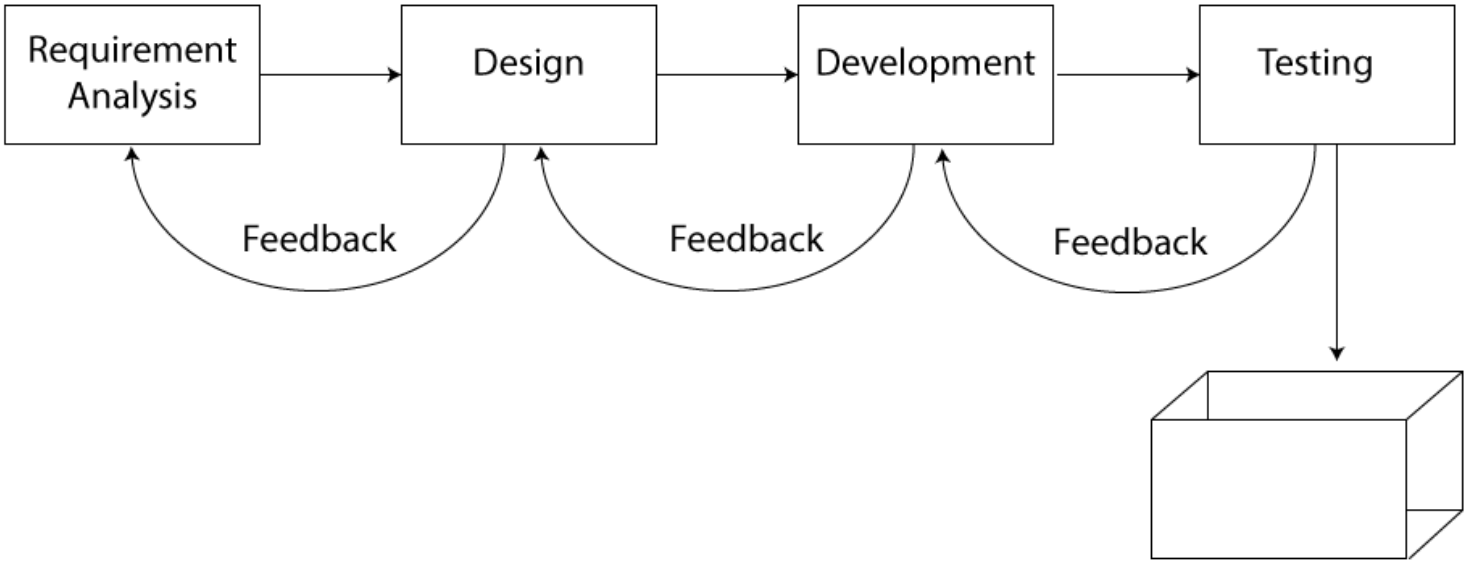
innovate achieve lead

BITS Pilani
Pilani Campus

Project Life cycle models

Predictive Project Development Life Cycle

- Fully Plan-Driven aka Waterfall - Risky Invite Failure

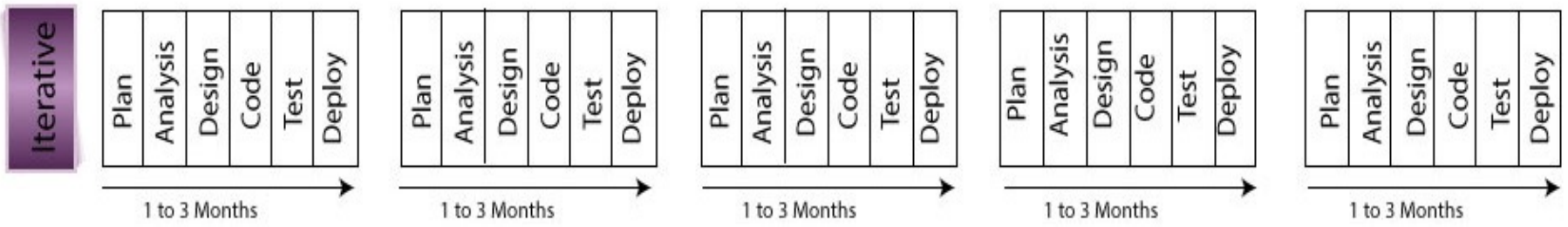


- Goal, Characteristics: Cost, Requirements-Fixed,Single Delivery

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

Iterative & Incremental Project Development Life Cycle

Plan



- Goal, Characteristics:
- Iteration Model: Accuracy/Correctness, Single Delivery
- Incremental: Speed, Multiple Deliveries
- Requirements-Dynamic

- **Iteration life cycle:** Product Increment/output may **not** be usable
- Example: Customized outfit/coat, Website

- **Incremental life cycle:** Product Increment/output is usable
- Example: Visit to a restaurant

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

Q&A

Q1,Q2

https://forms.gle/E8rdswFxp6B2pWfD9

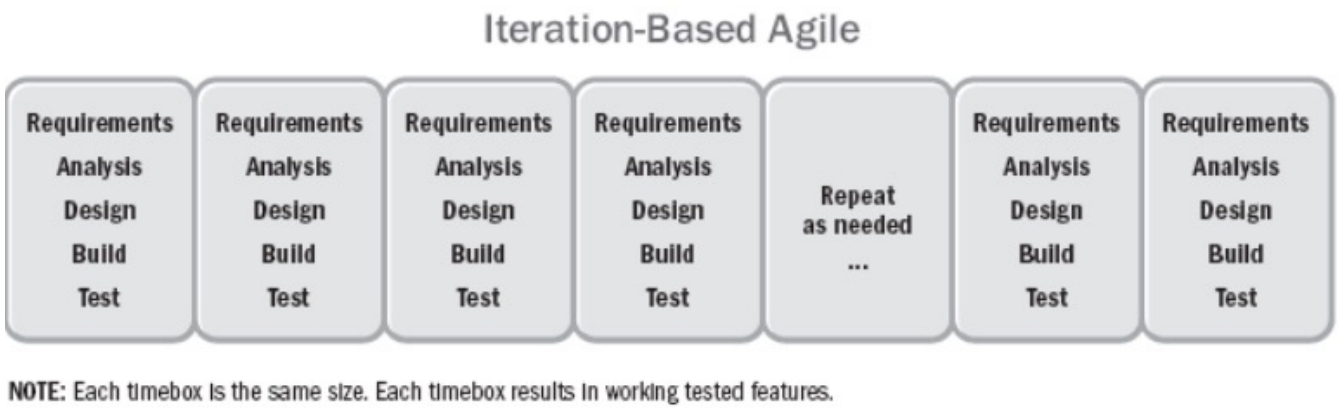
https://forms.gle/A9W5ikfCJPaT5tbg8

Agile Life Cycle Models

- Iterative
- Flow-Based

Agile/Adaptive Life Cycle-Iteration Based Agile

Plan



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

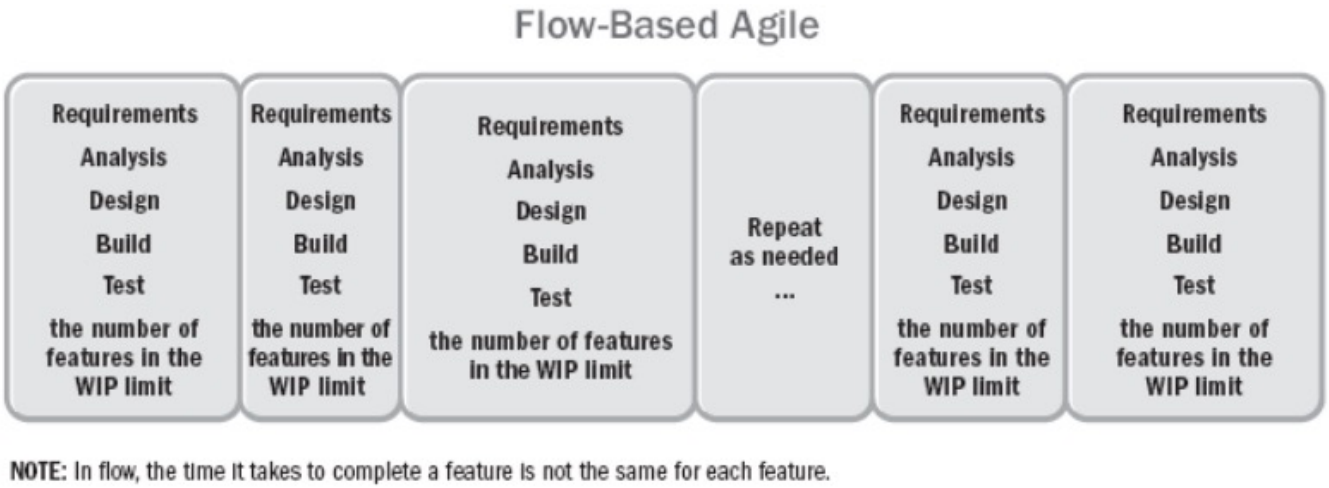
Ref: Agile Practice Guide (ENGLISH) by Project Management Institute Published by Project Management Institute, 2017 (Agile methodologies)I

Agile/Adaptive Life Cycle- Flow-based Based Agile

innovate achieve lead

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

Plan



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)I

Popular Early* Iterative and Agile Models

* Year ~2000 before

- Iterative
  - Spiral
  - RUP
- Agile
  - DSDM
  - FDD
  - Crystal

Q&A

Q3,Q4,Q5

https://forms.gle/bz3784DRQYHD5BnN9

https://forms.gle/TZpHt55eRhKijnZT9

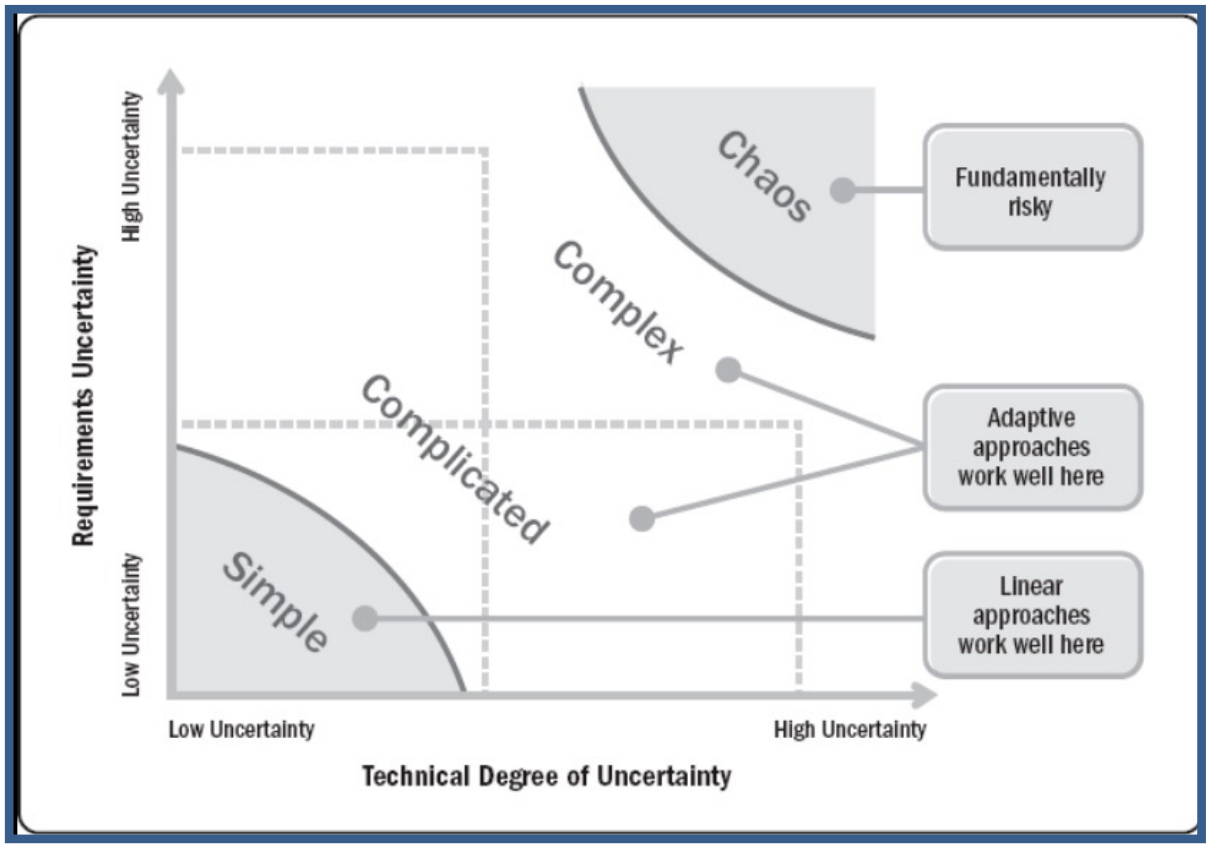
https://forms.gle/7XCdp9E7yr9ABG2r8

innovate achieve lead

BITS Pilani
Pilani Campus

Project Classifications/Decision making models

Agile Suitability - Project Environment
Stacey's Complexity Model



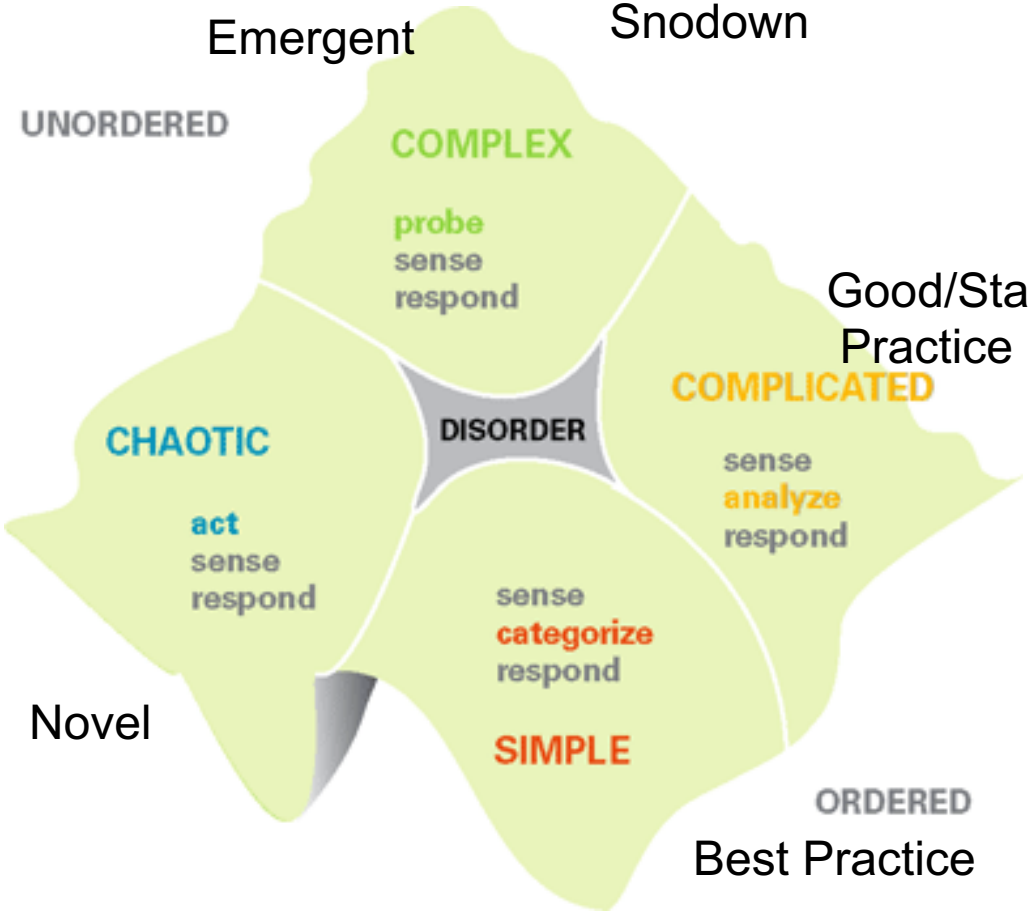
Ref: Agile Practice Guide (ENGLISH) by Project Management Institute Published by Project Management Institute, 2017 (Agile methodologies)

Cynefin framework - A Leader's Framework for Decision Making

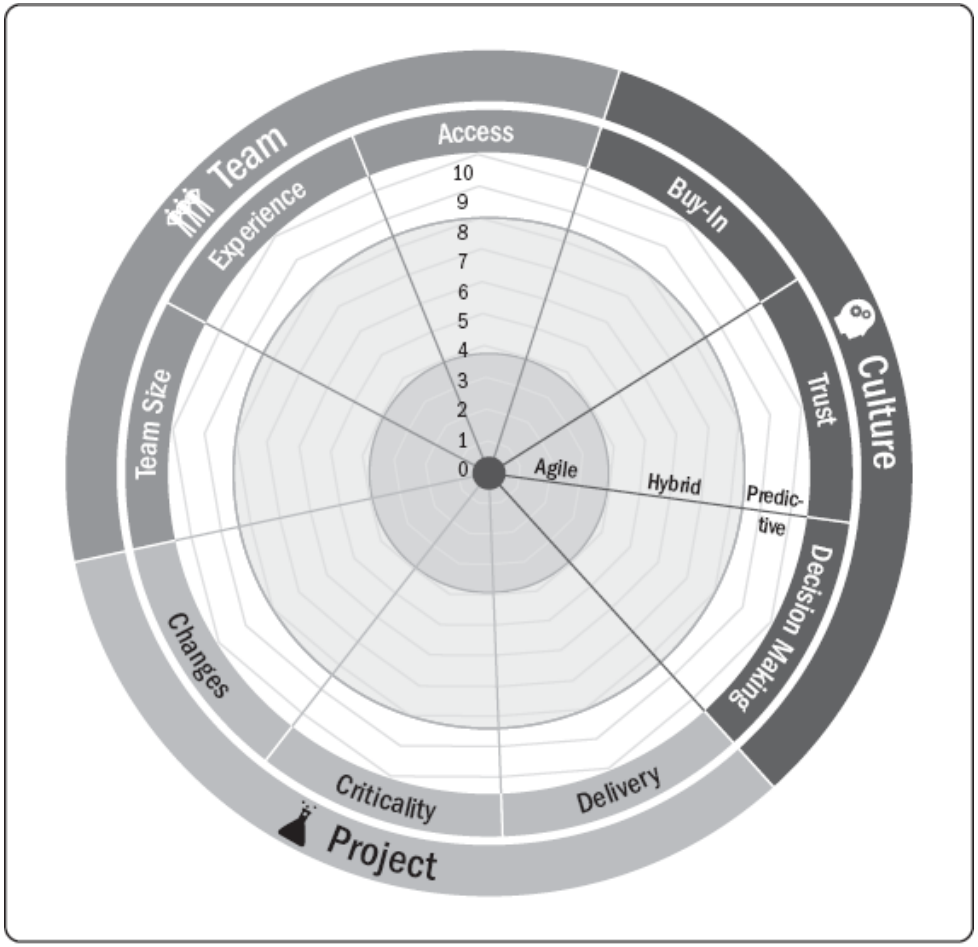
innovate achieve lead

Pronounced as: kun-ev-in

Developed in the early 2000s by David Snodown

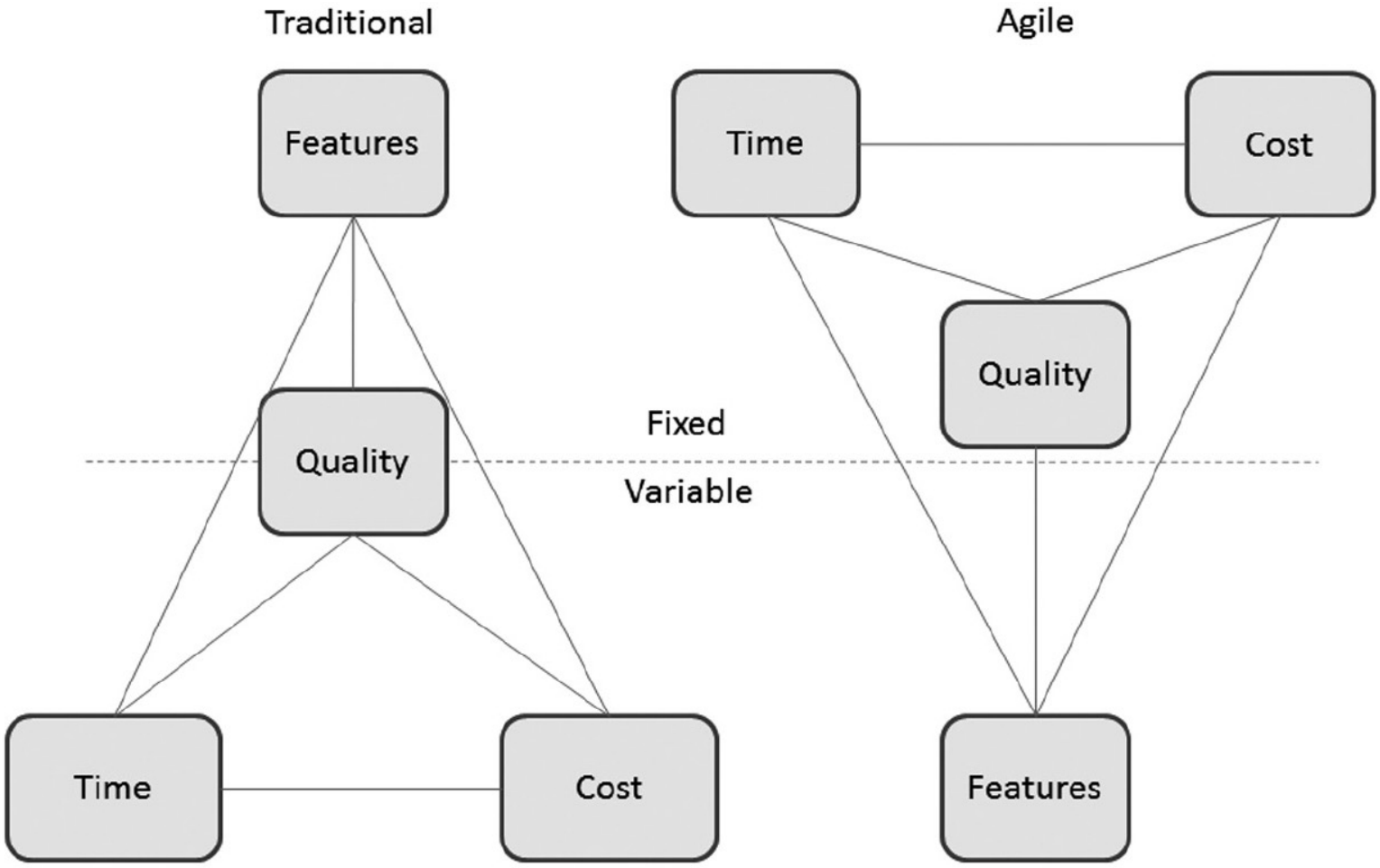


Agile Suitability Filter



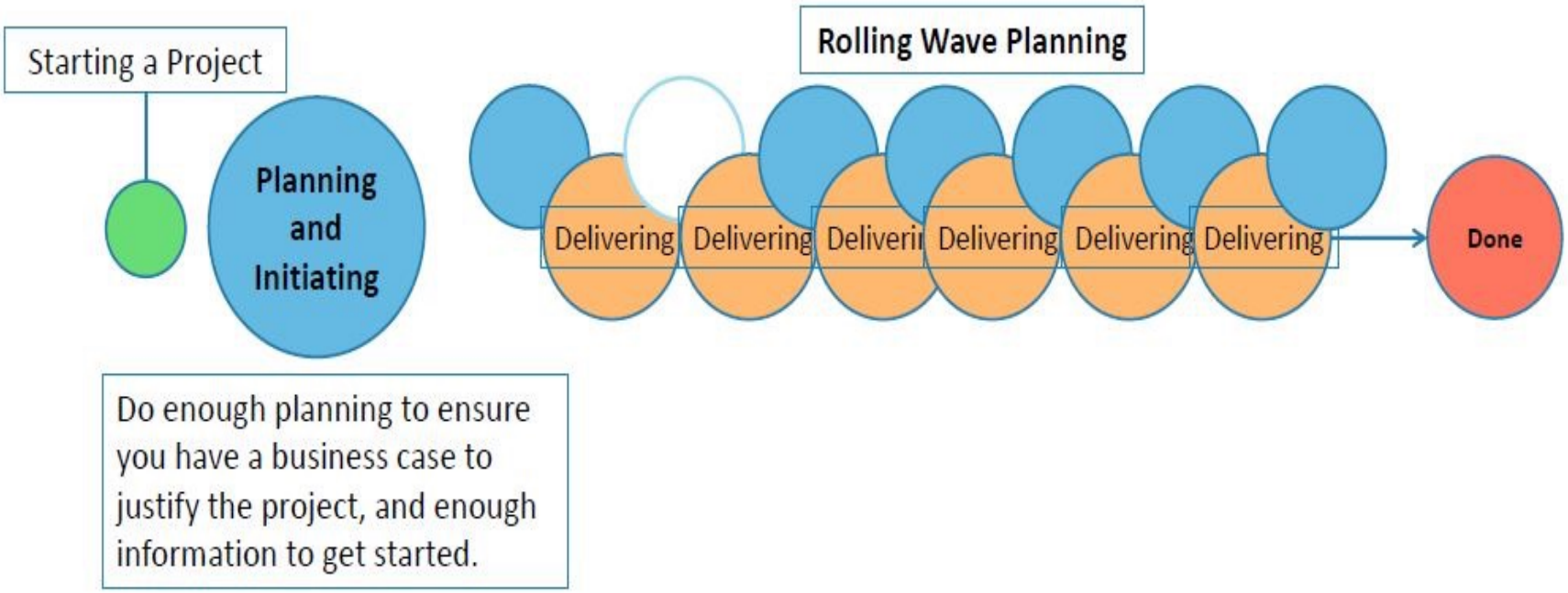
Attributes	Assessment
Buy-In	0-Yes, 5-Partial, 10-No
Trust	0-Yes, 5-Probably, 10-No
Decision Making	0-Yes, 5-Probably, 10-Unlikely
Incremental Delivery	0-Yes, 5-Maybe/Sometimes, 10-Unlikely
Criticality	0-Low, 5-Medium, 10-High
Changes	0-High, 5-Medium, 10-Low
Team Size	1-Small (<10), 5-Medium (>80),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

The "IRON" Triangle – Triple Constraints



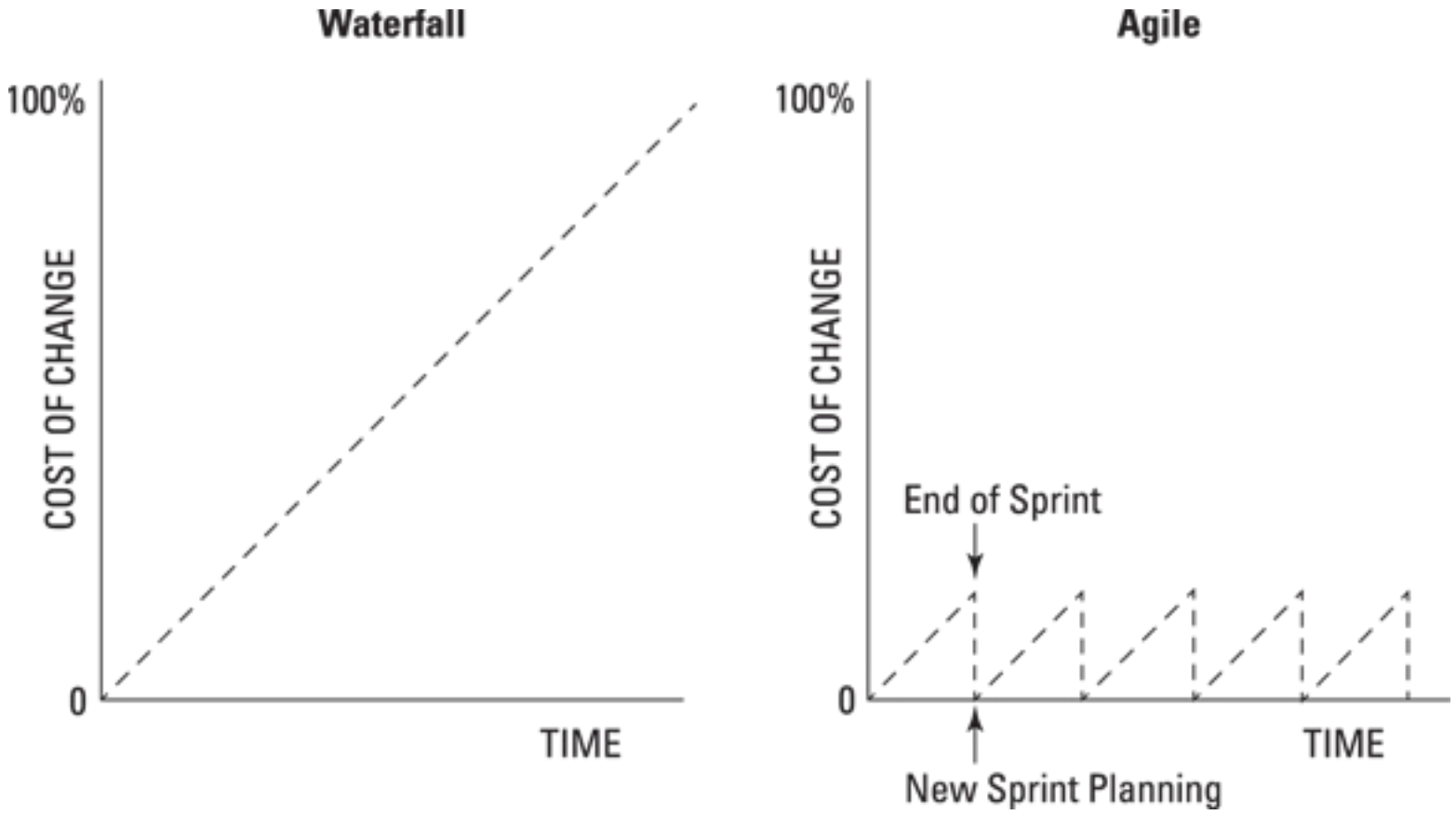
Ref: Agile Foundations - Principles, practices and frameworks by Peter Measey

Rolling Wave Planning or Progressive Elaboration

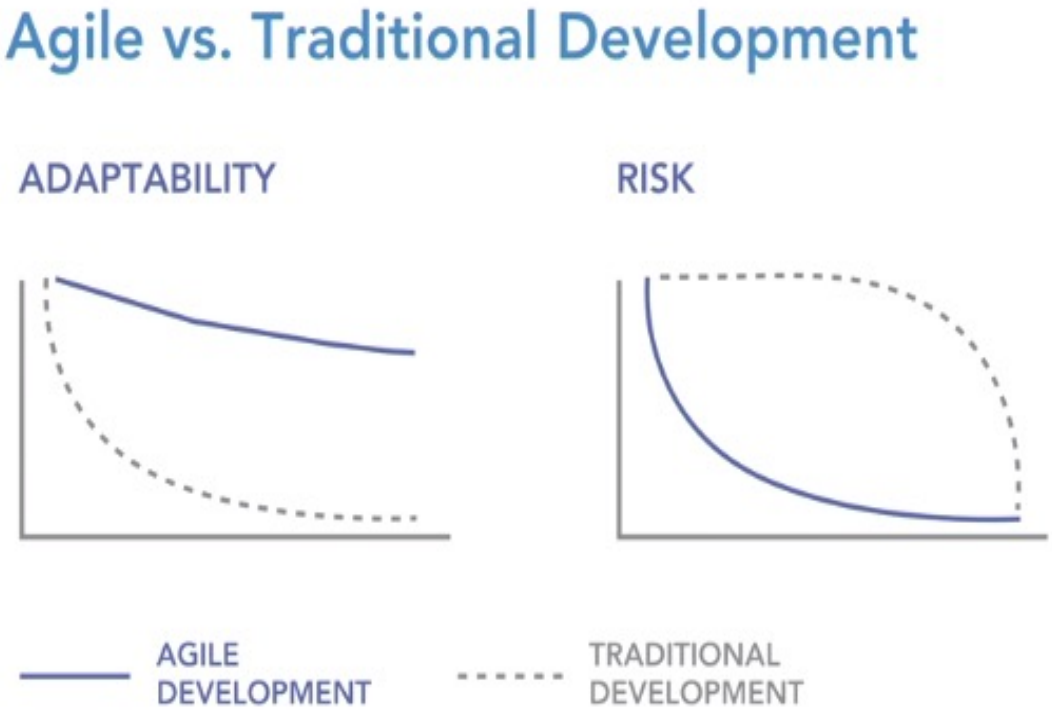


https://www.simplilearn.com/adaptive-planning-part-1-tutorial

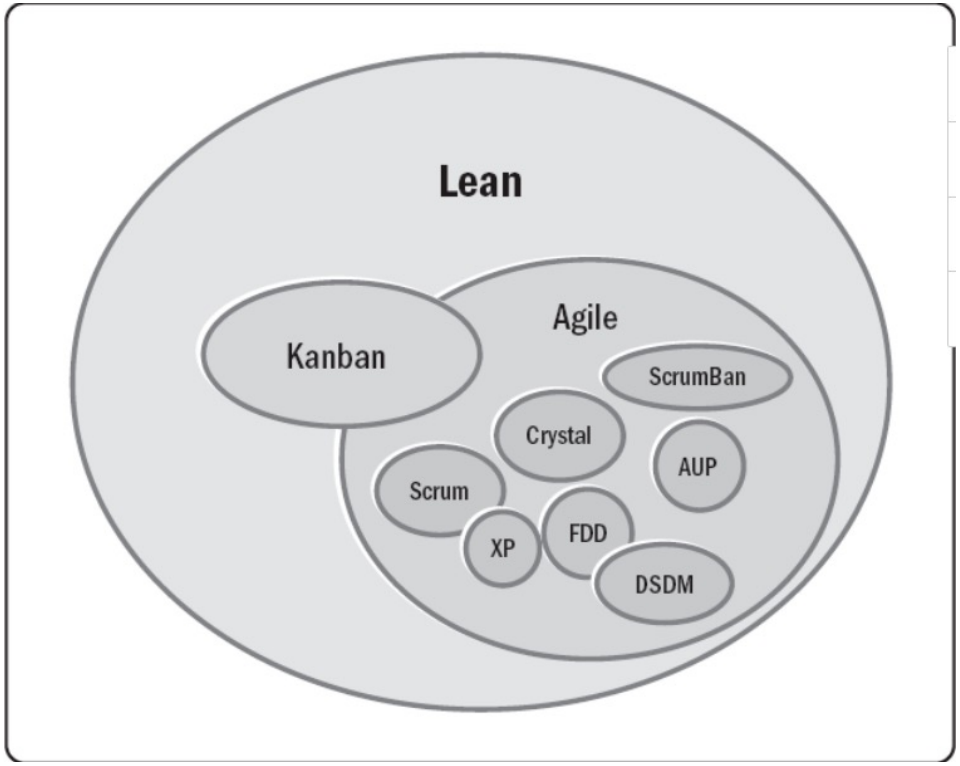
Cost of Changes: Agile vs Traditional Development



Adaptability and Risk
Agile vs Traditional Development



Popular Agile Methods



- They all have one important thing in common:
- they focus on changing your team's **mindset.**

Mindset

innovate achieve lead

	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

Q&A

Q6,Q7,Q8

https://forms.gle/9SUn7mtfuxcZ3kZUA

https://forms.gle/2f6AqYXnGMynNbKdA

https://forms.gle/3USQSAg2vKScGPZf7

End Contact Session-2

innovate achieve lead

BITS Pilani

Pilani Campus

Module-2 Additional Notes

Life Cycle

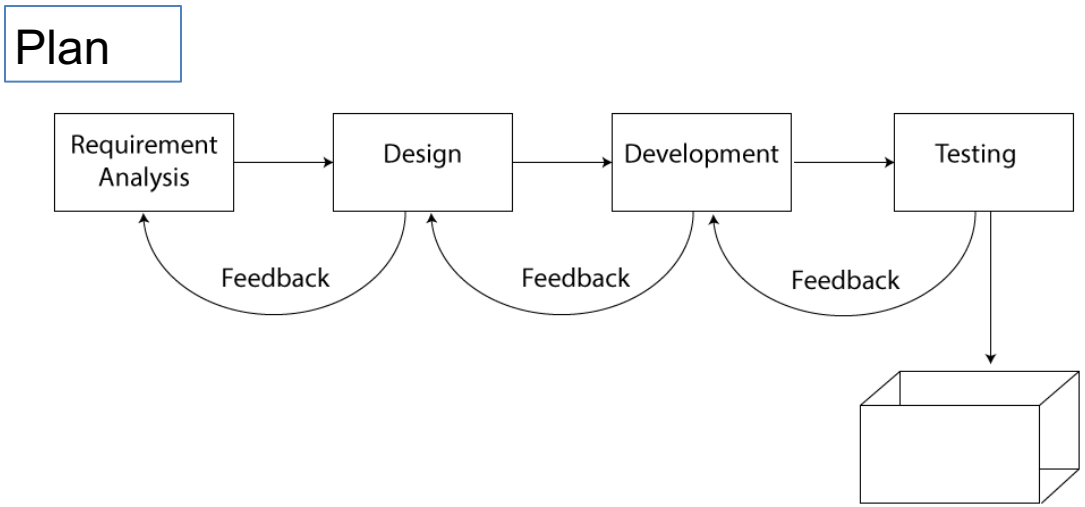
- The **sequence of actions** that must be **performed** in order to **build a software** system

- **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

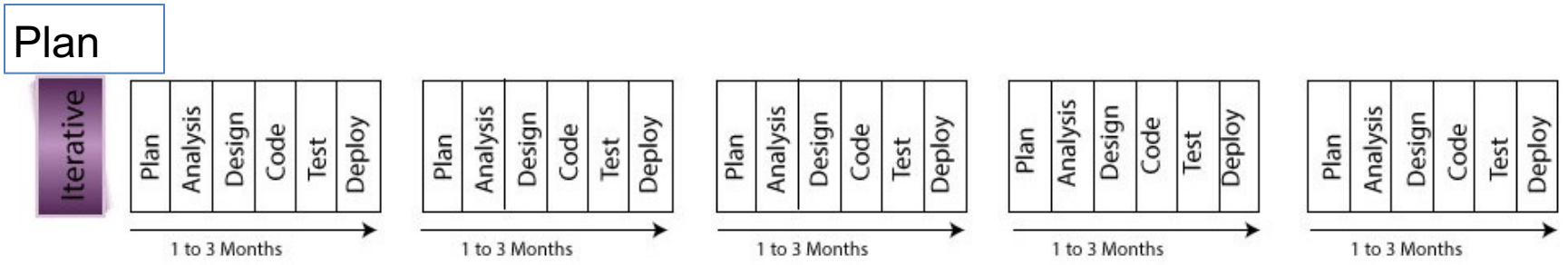


- Requirements/Scope is fixed
- Single delivery
- Goal: Manage Cost
- Minimal feedback changes
- Team is matured in estimation, technology etc..
- Project governance model exists
- **Don't expect long feedback cycle**, If this happens, this lifecycle not suitable for the project

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

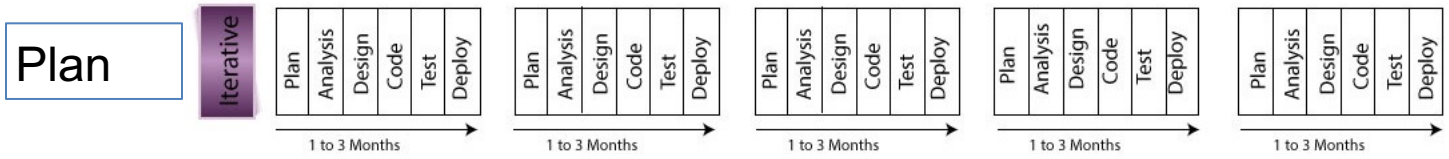
- When you are getting a customized coat made
  - You may be required to go for a trail to check for the fitting.
  - Even though the you may find the coat fitting well, you may not be able to use it as it has not been finished.
  - The fitting test was to give you an idea of the final product, which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



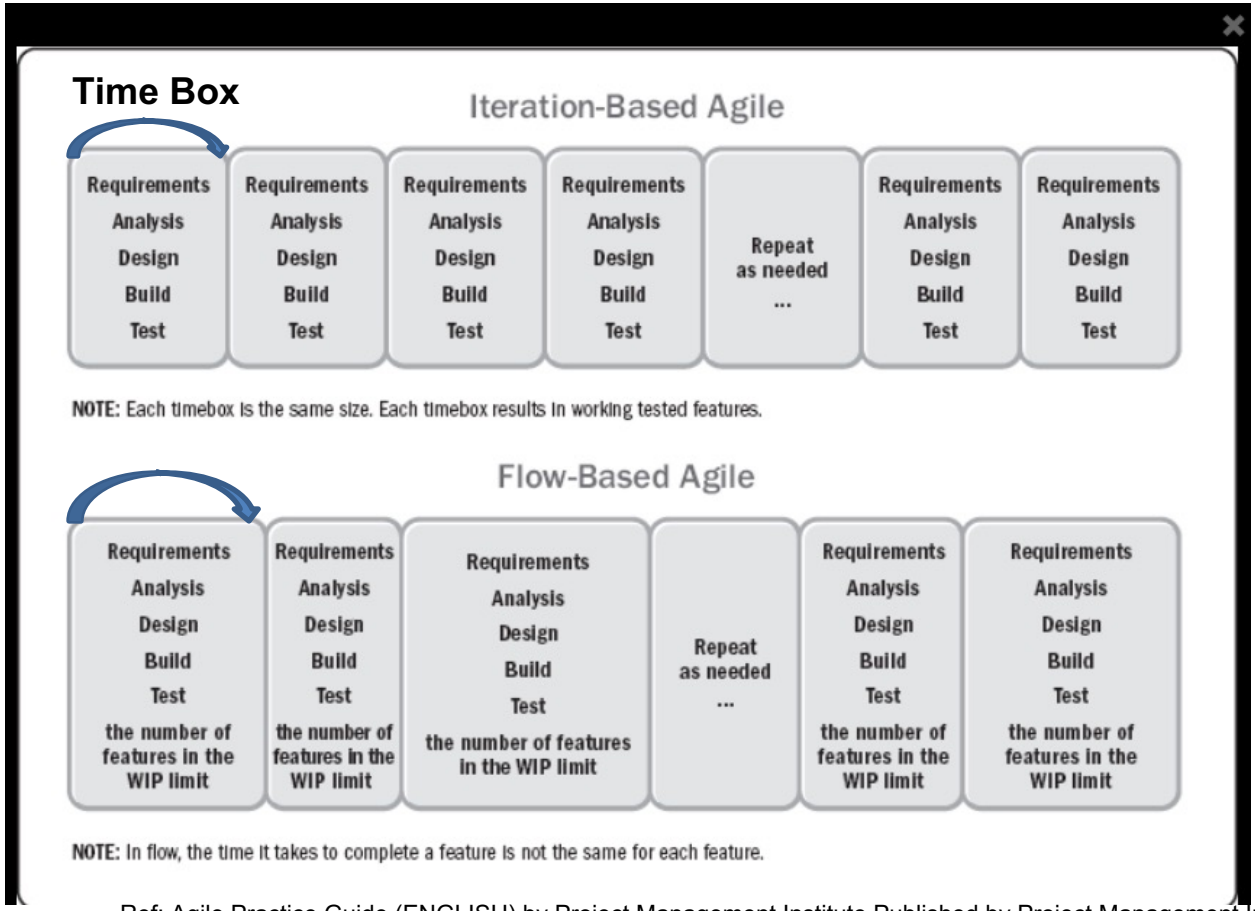
Example: You can compare this to a visit to restaurant. You get served starters first and on completion of its main course and then dessert. You get served incrementally and you consume it.

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

Agile/Adaptive Life Cycle

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

Plan



Fixed Time box : 1-4 weeks equal duration for each iteration

Limit WIP (work in Progress)
Optimize the flow

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

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)I

Delivery Environments and Agile Suitability

Delivery Environments

innovate achieve lead

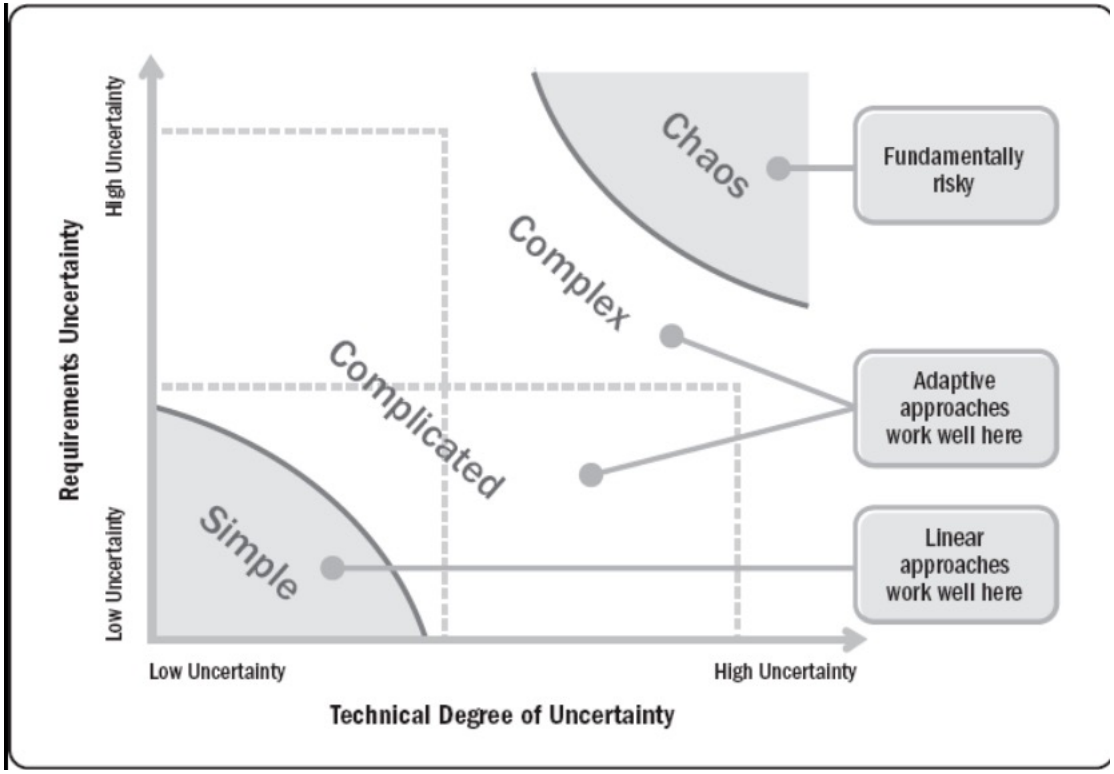
- The environment within which **Project/Product delivery** will occur should largely drive the delivery and **governance framework(s)** that will be implemented.

- 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



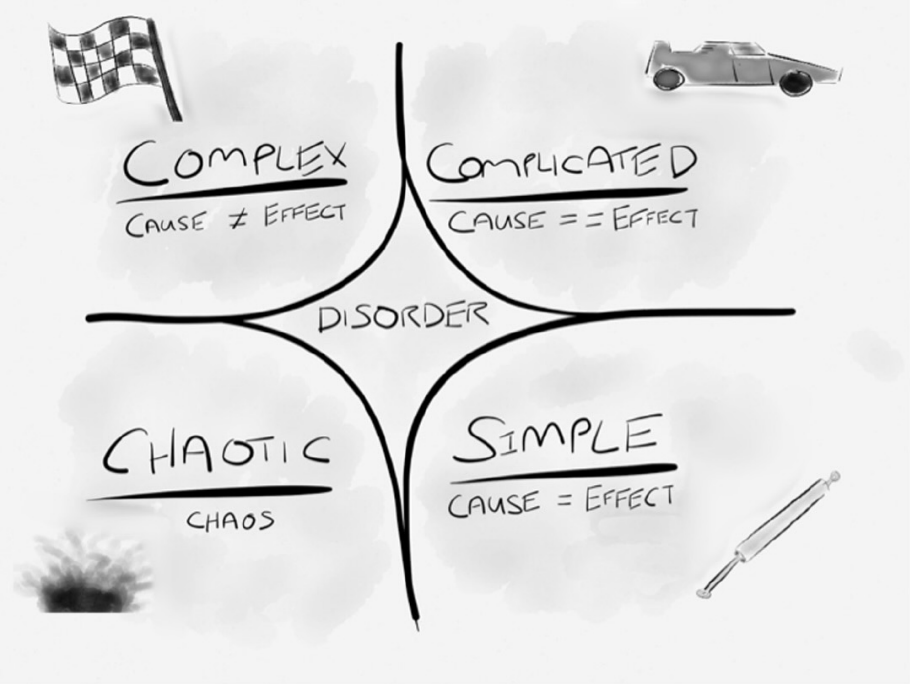
- Simple Environment: Use defined Process like Waterfall
- Complicated/Complex/Anarchy Environment: Use Empirical process like Agile. Example: New IT product development

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:

innovate achieve lead

- **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:

innovate achieve lead

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/

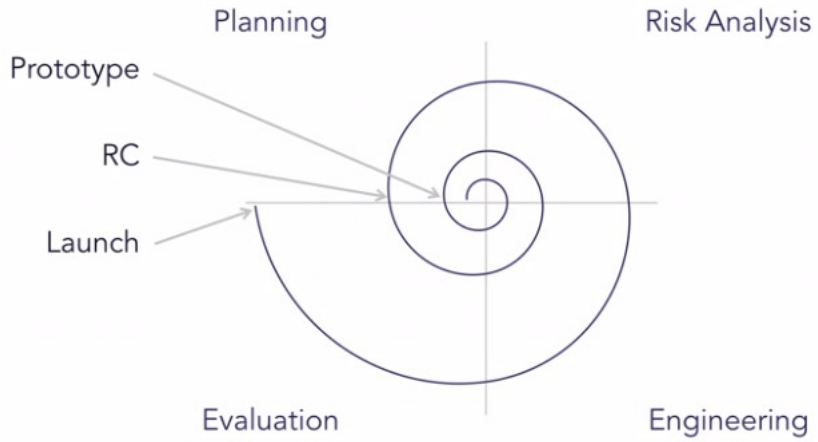
Some Popular Iteration Models

innovate achieve lead

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

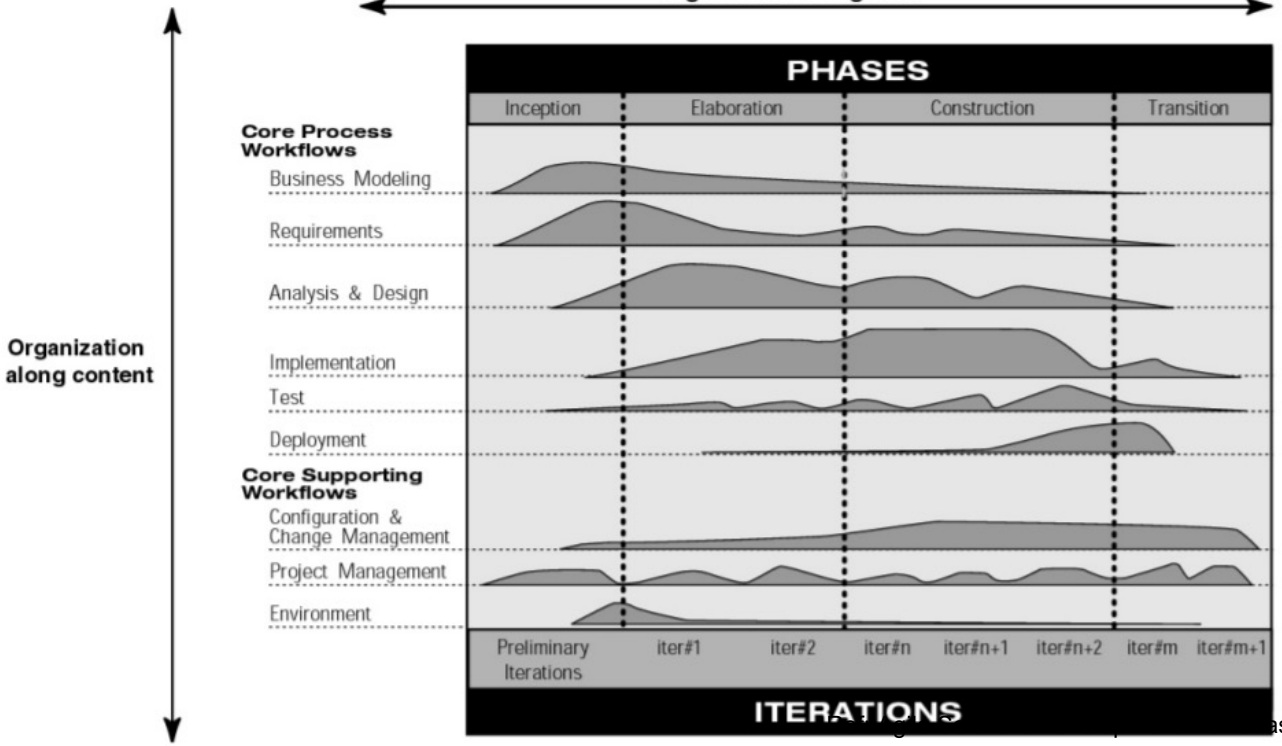
Rational Unified Process (RUP)

- 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

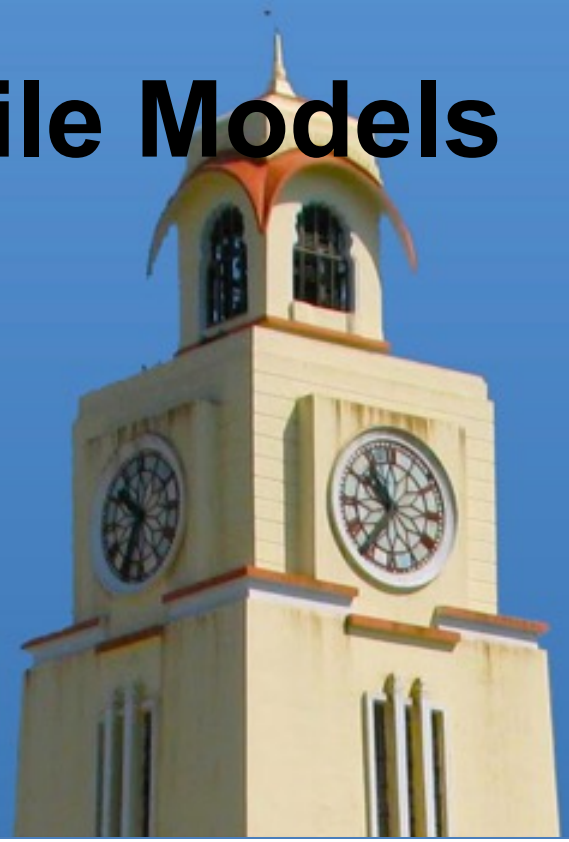


X-Axis: RUP Phases, Dynamic

Y-Axis: Organization Process, Static

ashi Shekhar,LinkedIn Learning

Early Agile Models



innovate achieve lead

BITS Pilani
Pilani Campus

Early Agile Methods

Dynamic System Development Method (DSDM)

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)

innovate achieve lead

- 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

Thank you