



BITS Pilani

Pilani Campus

BITS Pilani presentation

K.Anantharaman
Faculty CS Department

kanantharaman@wilp.bits-pilani.ac.in



SE ZG544 S1-22-23 , Agile Software Processes

SE ZG544 S1-22-23

Lecture No. 1, Module-1 - Agile Methods - An Introduction

Introduction



1. Faculty introduction
2. Email Id : kanantharaman@wilp.bits-pilani.ac.in
3. e-learn portal: <https://elearn.bits-pilani.ac.in/>
4. [Course Handout](#)
5. Recorded Video Lectures in e-learn/Taxila portal
 - According to the course handout, grouped by module
 - You MUST go through each module before coming to the online session

Poll



- <https://forms.gle/wRadsyQREA3BpkE26>

Module-1 – Topics



- Traditional software development practices
- Need for Agile Methods
- Benefits of Agile Methods

Basic Project Management concepts



- What is a Project?
 - Definite Start-End date, Temporary, Scope(Produce Specific result) , Budget/Effort – Example: Building a house
- Project Management Life Cycle Phases
 - Initiation, Planning, Execution, Closeout, Monitoring & Control
- System Development Life Cycle/phases (SDLC)
 - Requirements, Design, Construction, Implementation

Questions?



- Q1,Q1_1,Q1_2
- <https://forms.gle/onYWuBBy8TAJ6QVAA>
- <https://forms.gle/oC9BhYDVc2EvsD5N9>
- <https://forms.gle/pocBLb1fA7RjdYYU7>

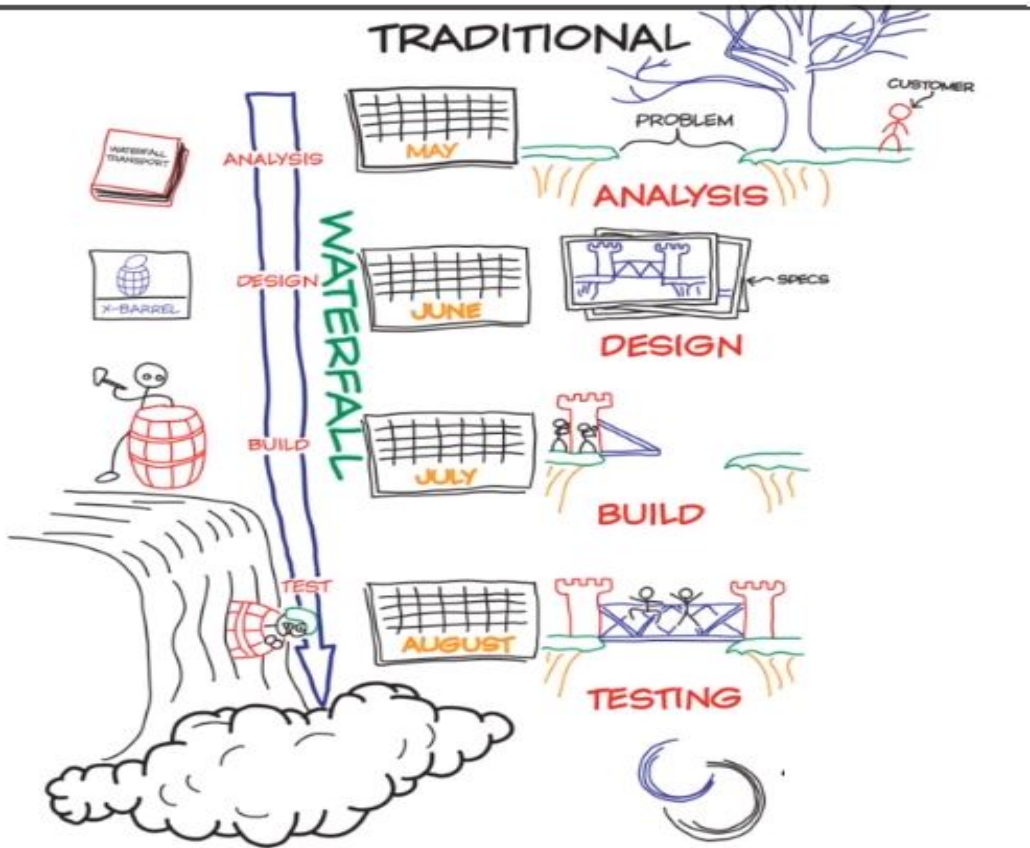


Agile SW Process SE ZG544 S1-22-23

Project Management Model

Water Fall Model and Agile

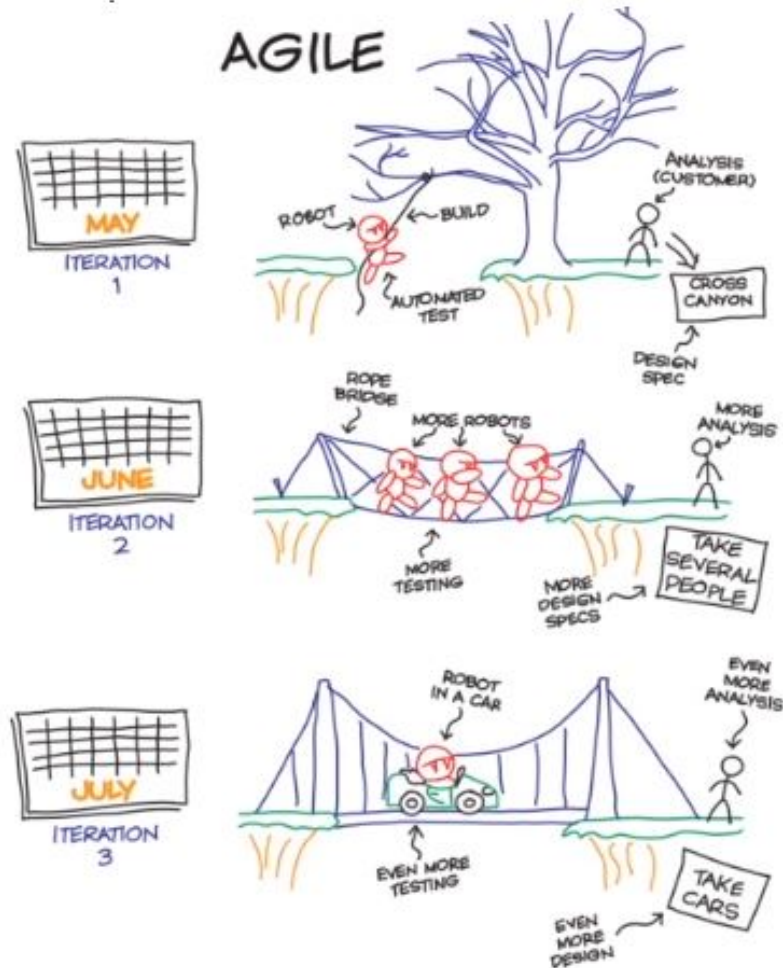
Traditional /Waterfall Development Approach(Rigid)



Reference : The Agile Sketchpad By [Dawn Griffiths](#), [David Griffiths](#), Oreilly media

Agile SW Process SE ZG544 S1-22-23

Agile Software Development Approach (Empirical)

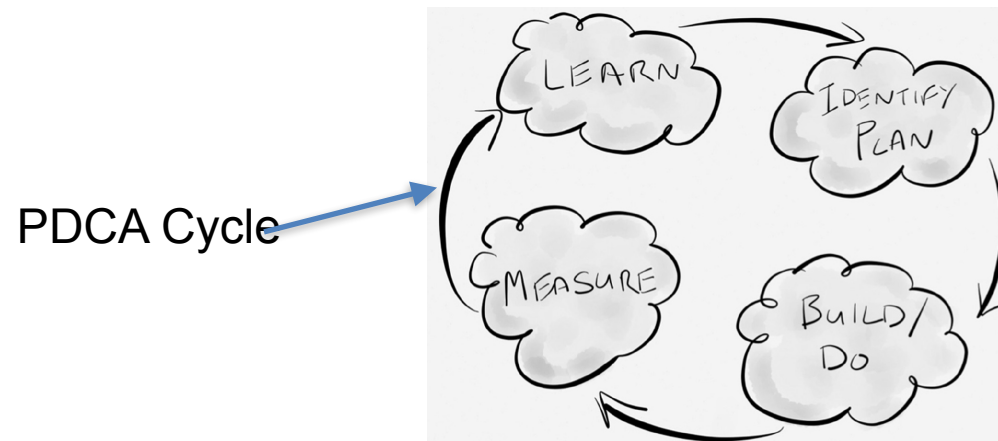


reference : The Agile Sketchpad By [Dawn Griffiths](#), [David Griffiths](#), Oreilly media

Agile SW Process SE ZG544 S1-22-23

Empirical Process Control

- Inspection
 - inspect the product being created and how it is being created
- Adaption
 - adapt the product being created or the creation process if required
- Transparency
 - ensure everyone can easily see what is happening



Questions?

- Q2, Q3
- <https://forms.gle/gaqQUVnLeB1uoCpT9>
- <https://forms.gle/pKRRh3cFn6xCJrdj6>

Advantages and Disadvantages of Waterfall



Advantages:

- Sequential, Upfront planning
- Good Documentation
- Scope of work is generally fixed

Disadvantages:

- Error propagation
- Missing requirements
- Error correction is costly
- Late customer feedback

Advantages and Disadvantages of Agile Model



Advantages:

- Early delivery of business value
- Continuous improvement
- Scope flexibility
- Team input
- Delivering well-tested products

Disadvantages:

- Poor Resource planning
- Less Documentation
- Fragmented output

Application of Waterfall Model



- Most common Project Management approach
- Surpassed by Agile approach after 2008.
- Simple and small systems.
- Enchantments to software systems
- Mission critical systems.

Application of Waterfall and Agile Model



- Fast Changing deliverables - New Technology Emerging projects
- Projects without clear requirements in the beginning
- New Product Development Projects
- Early Visibility, Quality, Risk identification



Need for Agile Methods

Software Project Success and Failure



- In 2015, Standish Group did a study of 10,000 projects in USA. The results showed that:
- 29% of traditional projects failed outright
- 60 percent of traditional projects exceeded the budget
- 11 percent of projects succeeded.

Questions?

- Q4, Q5
- <https://forms.gle/hNGMkyCTuXdXHTP86>
- <https://forms.gle/N2diLqfF994mi7ZR7>



Benefits of Agile Methods

Corporate World - Challenges and Inefficiencies



- Missed (or rushed) deadlines.
 - Budget blow-outs
 - Overworked and stressed employees.
 - Knowledge silos.
-
- Technology innovations and Agile approaches that have enabled to overcome these challenges (IT and Manufacturing industries)

Benefits of Agile Methods/Approaches/ Practices/Techniques



- Responsive planning
- Business-value-driven work
- Hands-on business outputs
- Direct stakeholder engagement
- Immovable deadlines
- Management by self-motivation
- Just-in-time' communication
- Immediate status tracking
- Waste management
- Constantly measurable quality
- Continuous improvement

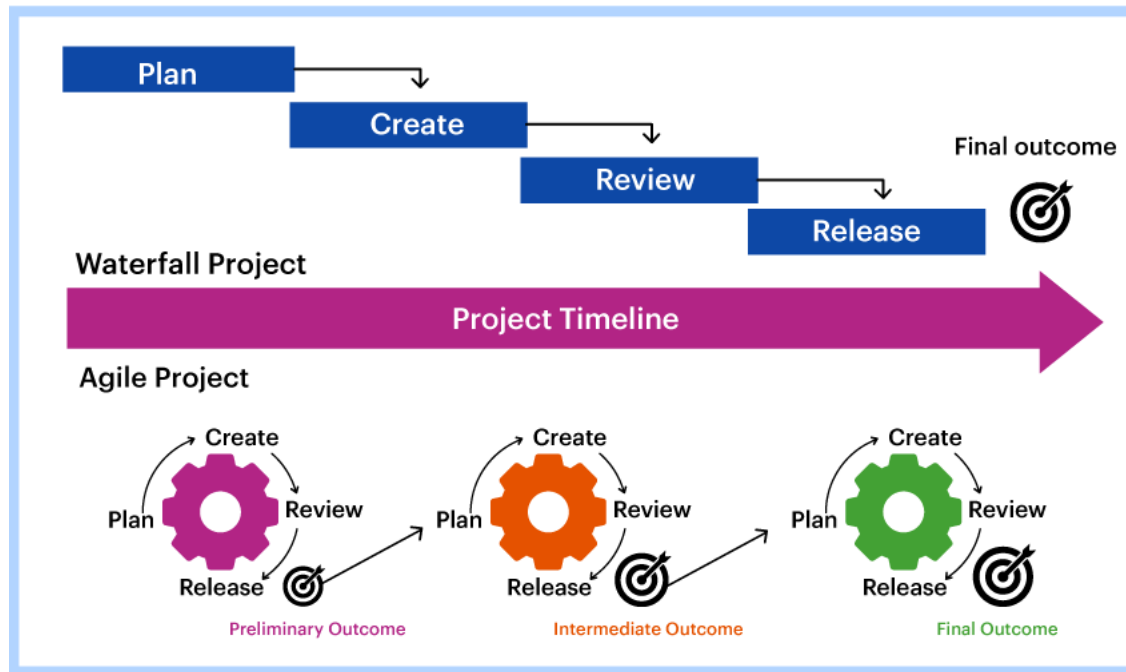


Module-1-Additional Notes

Topics Module-1



- Traditional software development practices
- Need for Agile Methods
- Benefits of Agile Methods

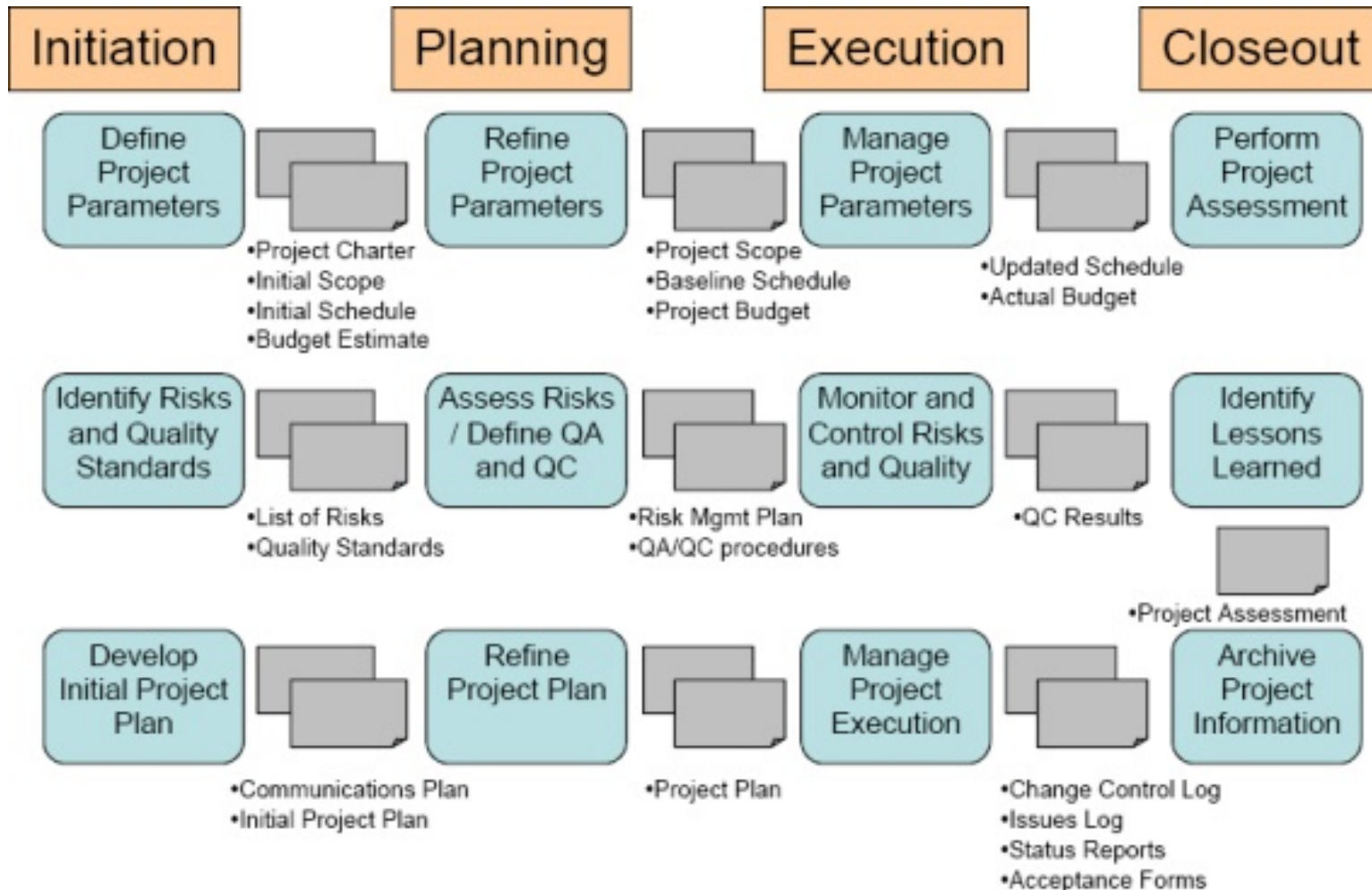


<https://kissflow.com/project/agile/traditional-vs-agile-project-management/>

What is a Project?

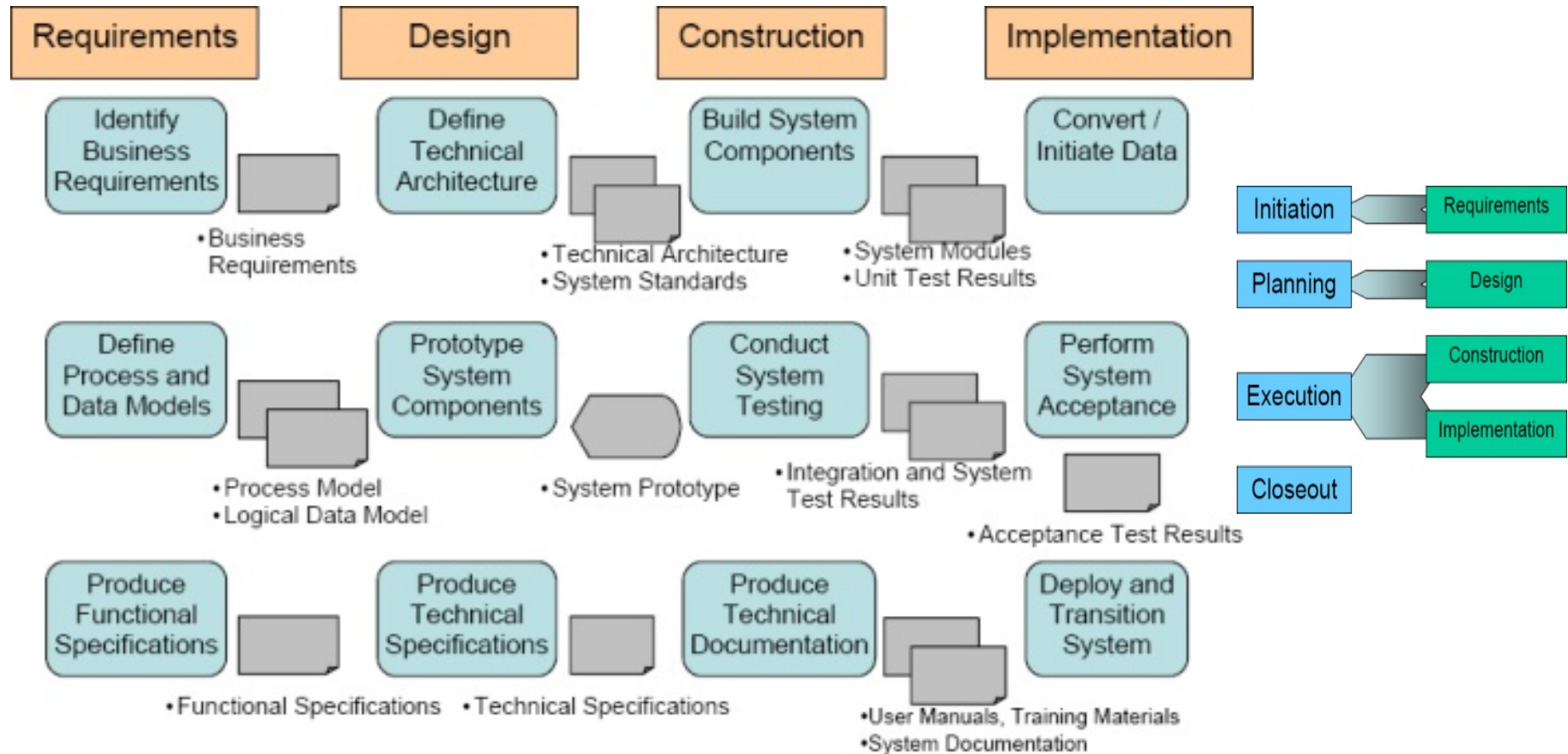
- A project is a planned program of work that requires a **definitive amount of time, effort, and planning** to complete.
- Projects have **goals and objectives** and often must be completed in **some fixed period of time and within a certain budget**.
- Development Project
- Maintenance or Support Project (Operational work)

Project Management Phases



<https://www.pmi.org/learning/library/project-managing-sdlc-8232>

System Development Phases (Engineering activities)



Traditional Software Development Approaches

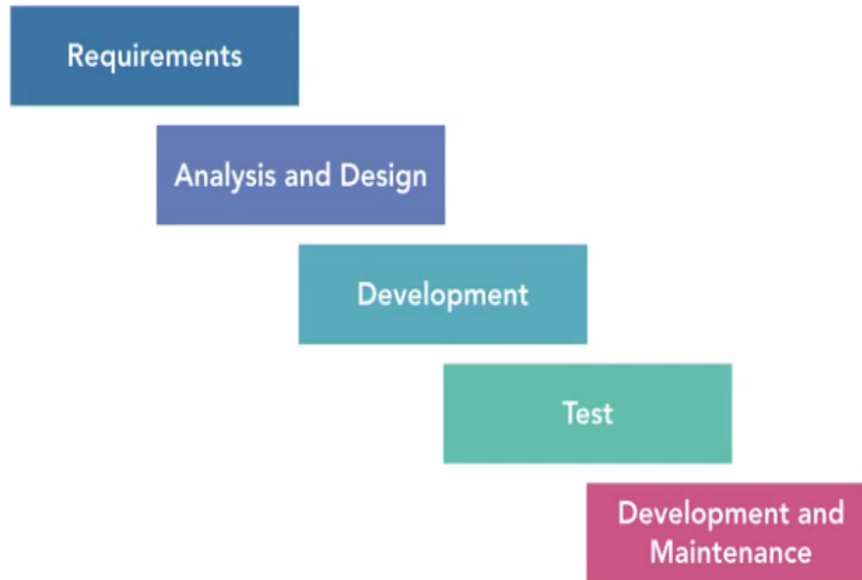
BITS Pilani

Pilani Campus



Traditional Software Development Model – Waterfall Model

Waterfall Approach



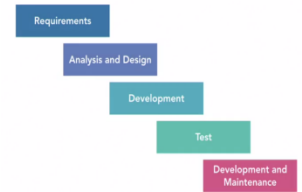
- Move to the next phase only when the prior one is complete — hence, the name waterfall.
- Origin from manufacturing like production plant
- **Upfront Planning**
- **Detailed documentation**
- **Scope of work is generally fixed.**
- Output of a phase becomes input to next phase
- Include well defined checklists, process and tools

<https://www.lynda.com/Developer-tutorials/Software-Development-Life-Cycle-SDLC/5030981-2.html>

Issues with Waterfall approach



- **Error** in one phase will **propagate** to next phase
- **Missing requirements** will result in **missing software feature**
- **Error correction** is **costly** if it is detected at later phase
- **Customer does not get to see the product** before the **early** testing phase which is usually two-thirds the way through the product time line.

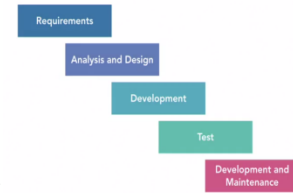


<https://www.lynda.com/Developer-tutorials/Software-Development-Life-Cycle-SDLC/5030981-2.html>

Issues with Waterfall approach ...



- You could be in the Deployment and Maintenance phase when you could realize that the product you are building was **no longer viable** due to **change in market conditions, or organizational direction**, or changed computer landscape
- (OR) You could realize that the product had a major **architectural flaw** that prevented it from being deployed.
- In other words, your product development initiative **could completely fail** after a lot of money and time had been spent on it.



<https://www.lynda.com/Developer-tutorials/Software-Development-Life-Cycle-SDLC/5030981-2.html>

Impact of Waterfall



- **Project failures**
 - Many organizations treated this failure as if there was a failure in a production factory. So they tried to fix their waterfall approach, by adding more comprehensive documentation.
 - **Comprehensive documentation**
 - Having a well documented software system is good. But the documentation by itself adds no value to the stake holders.
 - **Checklists and Coding standards**
 - Many software teams resorted to maintaining comprehensive checklist, to make sure they were producing systems of high quality. Checklist such as coding standards and architectural reviews are helpful. But you cannot produce a single recipe book for building software
- **More time should be spent on delivering working software features early and often. And enlisting customer feedback**

<https://www.lynda.com/Developer-tutorials/Software-Development-Life-Cycle-SDLC/5030981-2.html>

Application of Waterfall Model



- **Simple and small** systems.
- **Enchantments to software** systems — specifically applicable if the development team has good domain knowledge.
- **Mission critical systems.** Where you need gated checks to avoid catastrophic failures. An example is a software system where a defect can cause human causality. Comprehensive documentation is also very applicable here.
- *Waterfall model is the **most common project management approach** in software development until it was surpassed by improved approaches based on agile techniques around 2008.*

Application of Waterfall Model



- **Simple and small** systems.
- **Enchantments to software** systems — specifically applicable if the development team has good domain knowledge.
- **Mission critical systems.** Where you need gated checks to avoid catastrophic failures. An example is a software system where a defect can cause human causality. Comprehensive documentation is also very applicable here.
- *Waterfall model is the **most common project management approach** in software development until it was surpassed by improved approaches based on agile techniques around 2008.*

Software Project Success and Failure

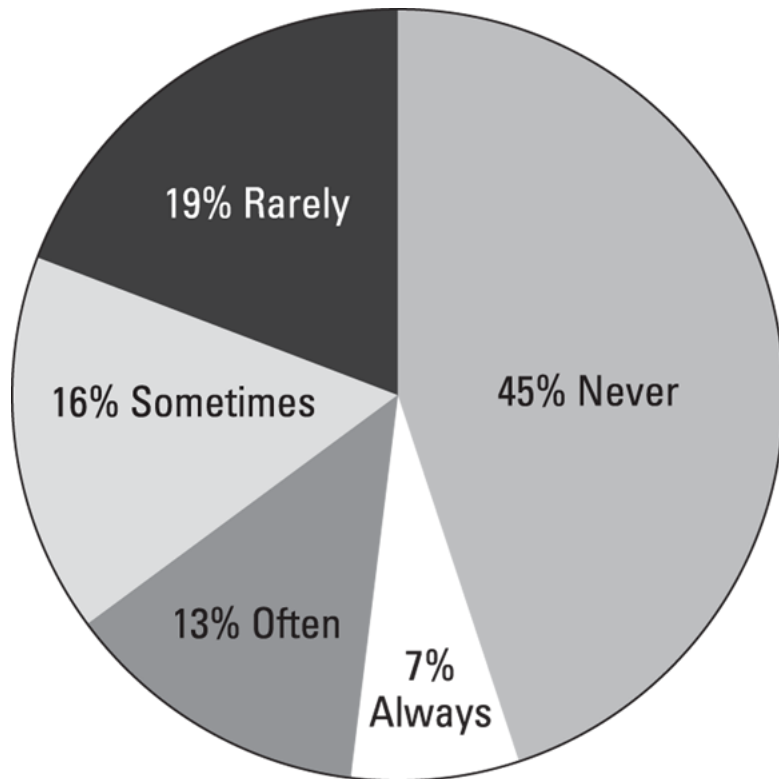


- In 2015, Standish Group did a study of 10,000 projects in USA. The results showed that:
- 29% of traditional projects failed outright
 - The projects were cancelled before they finished and did not result in any product releases. These projects delivered no value whatsoever
- 60 percent of traditional projects exceeded the budget
 - The projects were completed, but they had gaps between expected and actual cost, time, quality, or a combination of these elements. The average difference between the expected and actual project results — looking at time, cost, and features not delivered — was well over 100 percent.
- 11 percent of projects succeeded.
 - The projects were completed and delivered the expected product in the originally expected time and budget.

The problem with Status Quo



- Traditional projects that do succeed often suffer from scope bloat.



- The numbers in Figure illustrate an enormous waste of time and money.
- Direct result of traditional project management processes that are unable to accommodate change.
- Project managers and stakeholders at the start of a project ask for :
 - Everything they need
 - Everything they think they may need,
 - Everything they want,
 - Everything they think they may want

Actual use of requested software features.

Project management Needed Makeover



- In software development, **everything changes**. Requirements, skills, people, environment, business rules, et cetera.
- As **time progresses, you learn better** techniques of doing things.
- Your stakeholders need to change requirements to match changing **organizational strategy or Technology trends or changing market conditions**.
- In other words, the only **guaranteed thing is change** and the shown process to refine our work.
- Software development is **inherently an iterative process** and does not work like a Waterfall cycle.
- **Over emphasis on checklists and controls does not help** because software development is human centric and is heavily dependent on judgment and creativity.
- Software is not a product designed to be built by assembly lines.



Need for Agile Methods

Software Project Success and Failure using Traditional Approach



- In 2015, Standish Group did a study of 10,000 projects in USA. The results showed that:
- 29% of traditional projects failed outright
- 60 percent of traditional projects exceeded the budget
- 11 percent of projects succeeded.
- Also, projects that do succeed often suffer from scope bloat. – Only 20% of features is often used, 80% - Sometime/Rarely/Never used.

Project management Needed Makeover



- **In software development:**
- Everything changes.
- As time progresses, you learn better techniques of doing things.
- Organizational strategy changes or Technology trends or changing market conditions. (e.g. Covid19 Situation)
- Software development is inherently an iterative process
- Over emphasis on checklists and controls does not help.
- Software is not a product designed to be built by assembly lines.

Definable Work



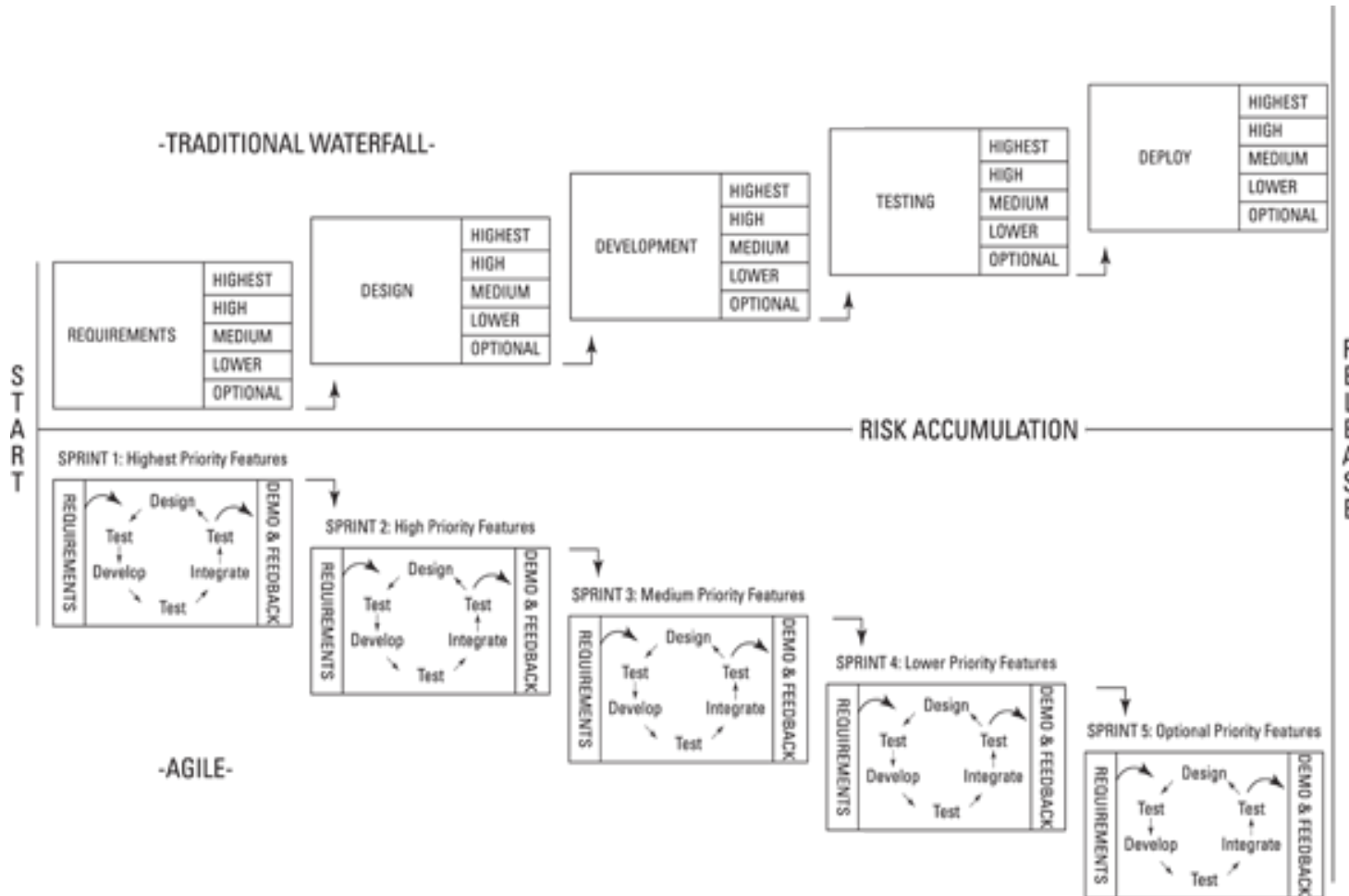
- Definable work projects are characterized by **clear procedures** that have proved successful on similar projects in the past.
- The production of a **car, electrical appliance, or home** after the design is complete are examples of definable work.
- The production domain and processes involved are usually **well understood** and there are typically low levels of execution uncertainty and risk.
- Definable work is **automated**.

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

High Uncertainty Work

- **New design, problem solving, and not-done-before work is *exploratory*.** It requires subject matter experts to collaborate and solve problems to create a solution.
 - Examples of people encountering high-uncertainty work include software systems engineers, product designers, doctors, teachers, lawyers, and many problem-solving engineers.
- High-uncertainty projects have **high rates of change, complexity, and risk.**
 - These characteristics present problems for traditional predictive approaches that aim to determine the bulk of the requirements upfront and control changes through a change request process.
- **Instead, agile approaches were created to explore feasibility in short cycles and quickly adapt based on evaluation and feedback.**

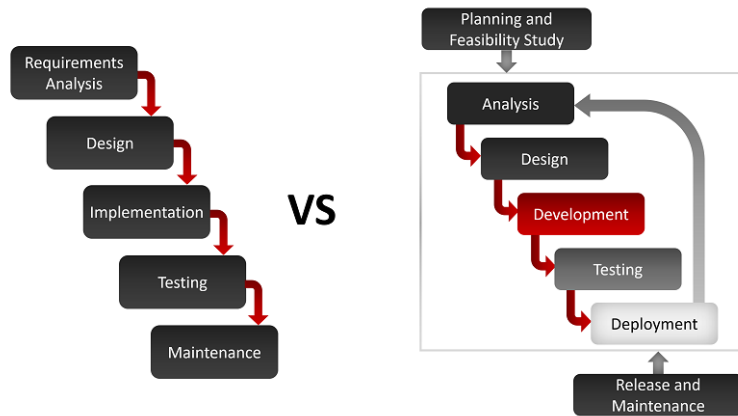
Waterfall vs agile project



Mixing traditional project management methods with agile approaches:

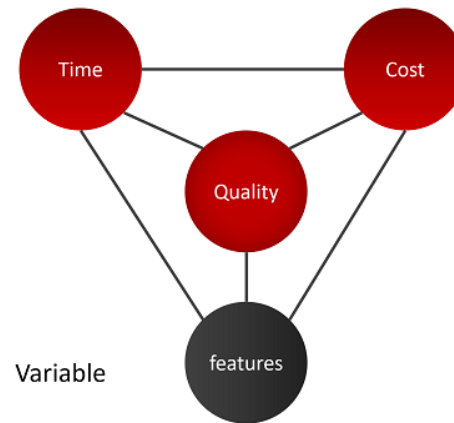
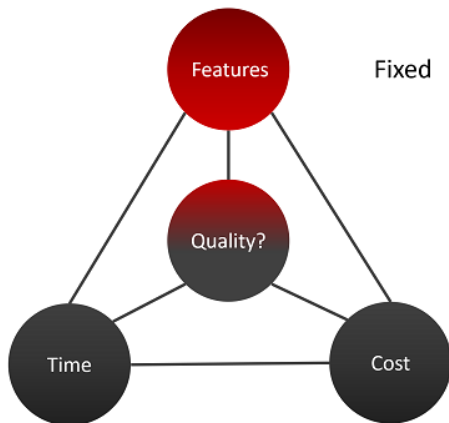
The answer, of course, is **you can't**. If you fully commit to an agile approach, you will have a better chance of deriving benefits of Agile project

Summary: Difference between Traditional and Agile Project Management



Traditional Approach

Agile Approach



1. Flexibility (Rigid Vs Adaptive)
2. Ownership & Transparency (Project Manager vs Team ownership)
3. Problem Solving (Unexpected obstacles- Escalation vs Team take decision)
4. Checkpoints and Monitoring progress: (No Frequent check-ins vs Quicker Iteration delivering value)

Ref: <https://www.kpipartners.com/blog/traditional-vs-agile-software-development-methodologies#:~:text=The%20main%20difference%20between%20traditional.in%20Agile%2C%20it%20is%20iterative.>



Evolution of Agile Project Management

Agile Project Management

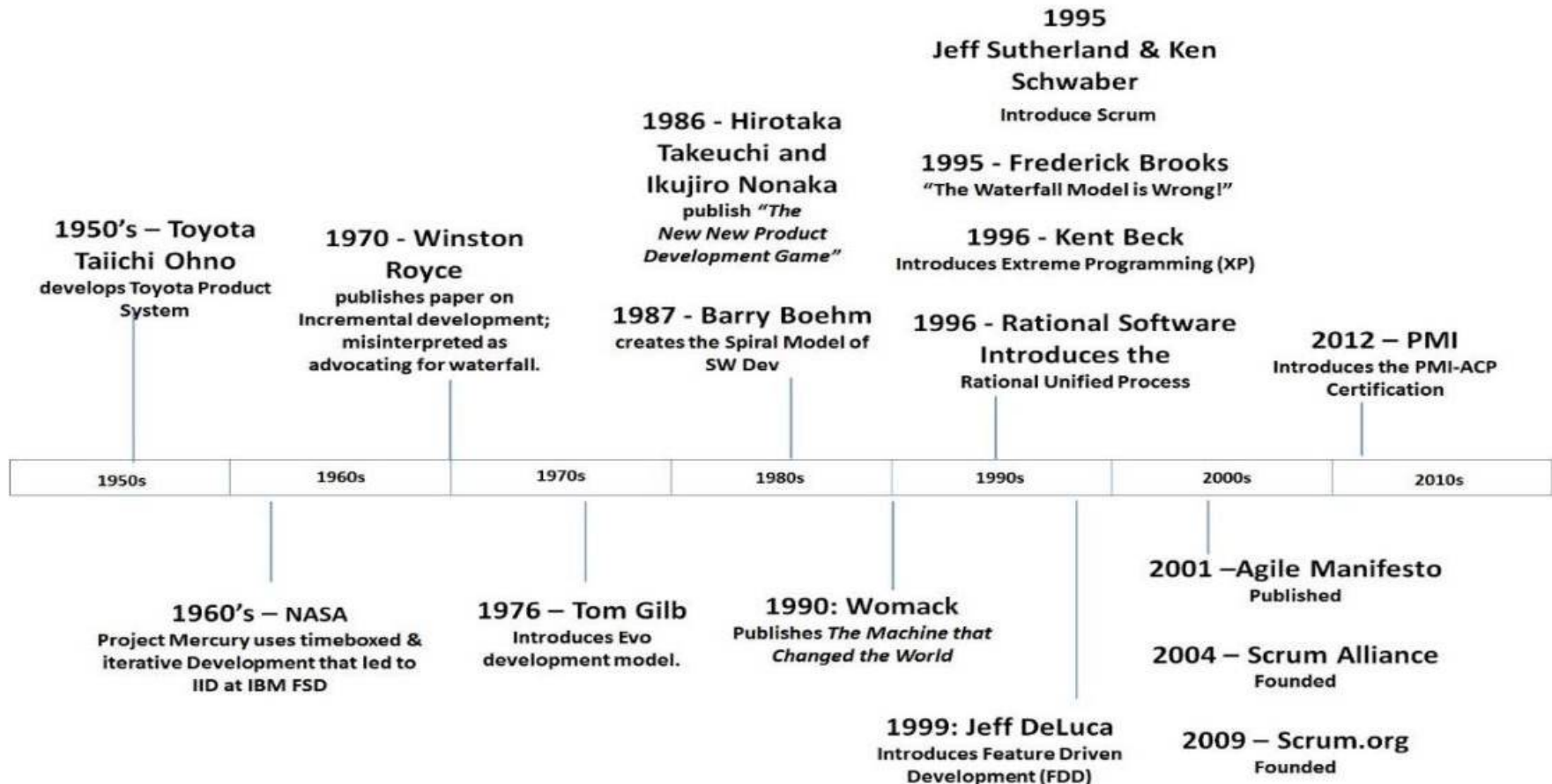


- *Agile project management* is a style of project management that focuses on :
- **Early delivery of business value**
- **Continuous improvement** of the project's product and processes
- **Scope flexibility**
- **Team input**
- **Delivering** well-tested products **frequently** that reflect customer needs.

Evolution of Agile Frameworks



A Brief History of Agile



Source Material e-learn(Recorded)

Evolution of Agile Frameworks ...

- In 1986, Hirotaka Takeuchi and Ikujiro Nonaka published an article called “**New New Product Development Game**” in the **Harvard Business Review**.
- Takeuchi and Nonaka’s article **described a rapid, flexible development strategy** to meet **fast-paced product** demands.
- This article first paired the term **scrum** with product development. (Scrum originally referred to a player formation in **rugby**.)
- **Scrum** eventually became one of the **most popular** agile project management frameworks.

Evolution of Agile

- 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:
- ***We will see more details about Agile Manifesto in the next Module***



How Agile Project Work

How agile projects work

- Agile approaches are based on an ***empirical control method*** — a process of making decisions **based on the realities observed in the project**.
- In the context of software development methodologies, an empirical approach can be **effective in both new product development and enhancement and upgrade projects**.
- By using **frequent and firsthand inspection** of the work to date, you can make immediate adjustments, if necessary.

Why Agile Projects Work Better



- The Standish Group study, mentioned earlier slide “Software project success and failure,” found that while **29 percent of traditional projects failed outright, that number dropped to only 9 percent** on agile projects.
- The decrease in failure for agile projects is a result of agile project teams making **immediate adaptations** based on **frequent inspections** of progress and **customer satisfaction**.

Why Agile Projects Work Better ...



- Some key areas where agile approaches are superior to traditional project management methods:
 - **Project success rates: The risk of catastrophic project failure falls to almost nothing on agile projects.** Agile approaches of prioritizing by business value and risk ensure early success or failure. Agile approaches to testing throughout the project help ensure that you find problems early, not after spending a large amount of time and money.
 - **Scope creep: Agile approaches accommodate changes throughout a project, minimizing scope creep.** On agile projects, you can add new requirements at the beginning of each sprint without disrupting development flow. By fully developing prioritized features first, you prevent scope creep from threatening critical functionality.
 - **Inspecting and adaptation:** Agile project teams — armed with **frequent feedback from complete development cycles and working, shippable functionality** — can improve their processes and their products with each sprint.



Benefits & Challenges of Agile Methods

Corporate World - Challenges and Inefficiencies



- Most organizations (Small/Large/Public/Private/Startup) share the same core challenges and inefficiencies, including:
 - Missed (or rushed) deadlines.
 - Budget blow-outs
 - Overworked and stressed employees.
 - Knowledge silos.
- Technology innovations and Agile approaches that have enabled them: (IT & Manufacturing industries)
 - Genuinely create more efficient work environments, to consistently manage their work within allocated budgets, and to regularly deliver high business-value (and high-quality) outputs on time.

Benefits of Agile Methods/Approaches/ Practices/Techniques



- ***Responsive planning***: involves breaking down long-term objectives into shorter delivery cycles; and then *adapting* ongoing work (and funding) based on the outcomes of each delivery cycle.
- ***Business-value-driven work***: involves prioritizing work in accordance with the amount of primary and secondary business value that each activity is likely to bring to the organization.

Benefits of Agile Methods/Approaches/ Practices/Techniques ...



Hands-on business outputs: involves regularly inspecting outputs firsthand in order to determine whether business requirements are being met – and whether business value is being delivered for the organization.

Direct stakeholder engagement: involves actively engaging internal and external customers throughout a process to ensure that the resulting deliverables meet their expectations.

Benefits of Agile Methods/Approaches/ Practices/Techniques ...



Immovable deadlines: are fixed time commitments that encourage staff members to deliver regular ongoing value to the organization.

Management by self-motivation: involves using the power of self-organized teams to deliver outcomes under the guidance and oversight of the customer.

‘Just-in-time’ communication: replaces traditional corporate meetings with techniques for more effective communication and knowledge transfer (**Differ Commitment**)

Benefits of Agile Methods/Approaches/ Practices/Techniques ...



Immediate status tracking: provides tools that enable staff to keep others in the organization continuously aware of the status of the work that they are doing.

Waste management: involves maximizing the value of the organization's resources by reducing and, where possible, eliminating low business-value activities.

Constantly measurable quality: involves creating *active checkpoints* where organizations can assess outputs against both qualitative and quantitative measurements.

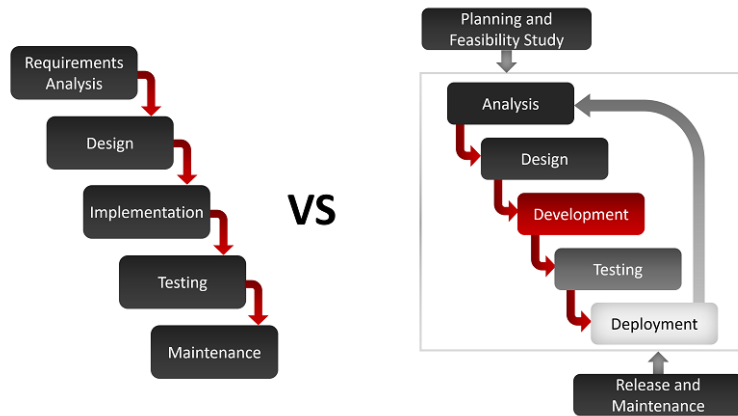
Benefits of Agile Methods/Approaches/ Practices/Techniques ...



Rearview mirror checking: provides staff with tools for regularly monitoring and self-correcting their work.

Continuous improvement: involves regularly reviewing and adjusting business activities to ensure that the organization is continuing to meet market and stakeholder demand.

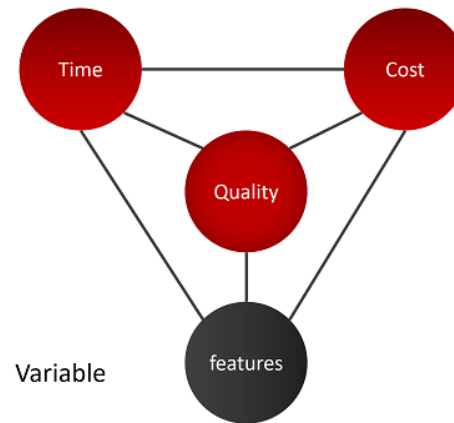
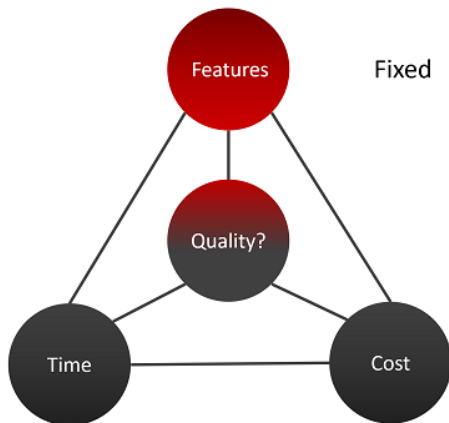
Summary – Agile Methods (Module-1)



VS

Traditional Approach

Agile Approach



Difference between Traditional and Agile Project Management:

1. Flexibility (Rigid Vs Adaptive)
2. Ownership & Transparency (Project Manger vs Team ownership)
3. Problem Solving (Unexpected obstacles-Escalation vs Team take decision)
4. Checkpoints and Monitoring progress: (No Frequent check-ins vs Quicker Iteration delivering value)

<https://www.kpipartners.com/blog/traditional-vs-agile-software-development-methodologies#:~:text=The%20main%20difference%20between%20traditional,in%20Agile%2C%20it%20is%20iterative.>

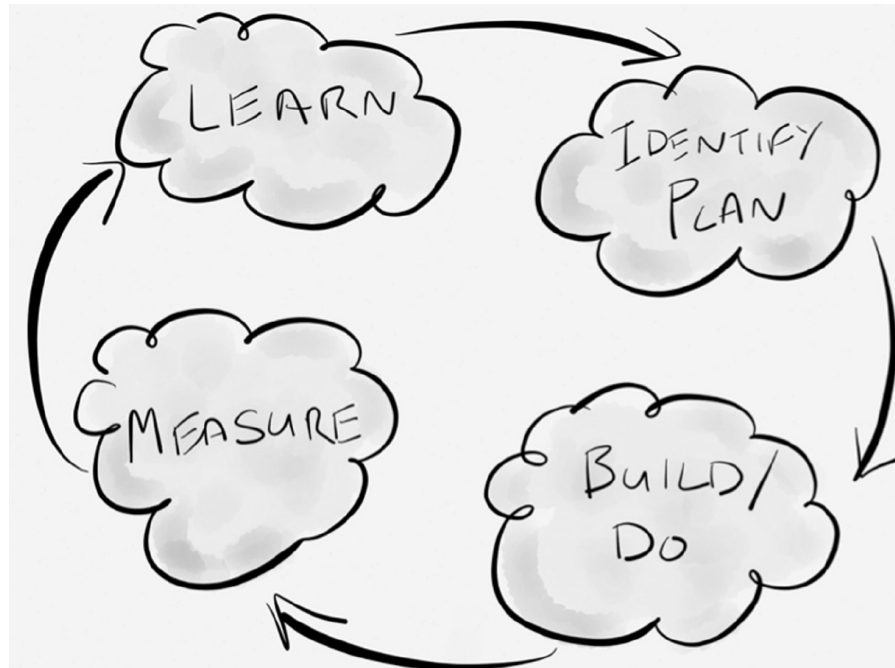
How agile projects work

- Agile approaches are based on an ***empirical control method*** — a process of making decisions **based on the realities observed in the project**.
- In the context of software development methodologies, an empirical approach can be **effective in both new product development and enhancement and upgrade projects**.
- By using **frequent and firsthand inspection** of the work to date, you can make immediate adjustments, if necessary.

Empirical Process



- Empirical processes (see Figure) incorporate repeated inspection and adaptation of a product to ensure the right product is delivered in the right way. This is especially important in **environments** that **experience high variability** and are therefore **most suited to Agile** working.



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

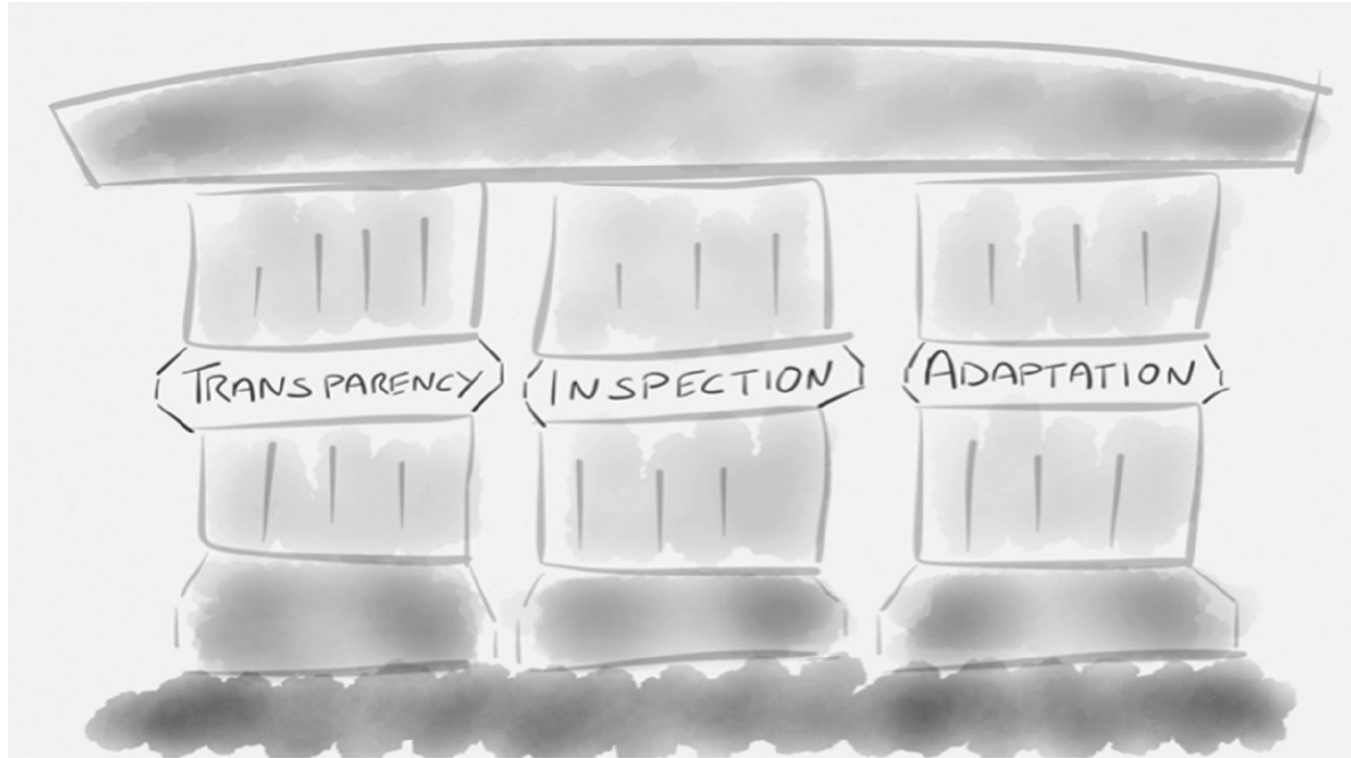
23-Aug-22

Agile Software Process SE SG544 S1-22-23

63

BITS Pilani, Pilani Campus

Pillars of Empirical control Method



Inspection – inspect the product being created and how it is being created **Adaptation** – adapt the product being created or the creation process if required
Transparency – ensure everyone can easily see what is happening

Frequent Iterations

- To accommodate frequent inspection and immediate adaptation, agile projects work in ***iterations*** (smaller segments of the overall project).
- An agile project involves the **same type of work** as in a **traditional waterfall** project:
 - You create **requirements and designs**, **develop the product**, **document it**, and if necessary, integrate the product with other products. You test the product, fix any problems, and deploy it for use.
 - However, instead of completing these steps for all product features at once, as in a waterfall project, you **break the project into iterations**, also called ***sprints***.

Examples of Empirical models



- **PDCA** Plan, Do, Check, Act – Edward Deming (Deming, n.d.).
- **POOGI** Process of On-Going Improvement – Theory of Constraints (Goldratt and Cox, 1984).
- **OODA** Observe, Orient, Decide, Act – John Boyd (Boyd, n.d.).
- **BML Build**, Measure, Learn – Lean Start-up (Ries, 2011).
- **DMAIC** Define, Measure, Analyse, Improve, Control (Six Sigma, 2006).
- **TAC** Thought, Action, Conversation – DSDM Agile Project Framework (DSDM Consortium, 2014b).
- **Kaizen** A Japanese word which means ‘good change’, used to describe a philosophy of continuous improvement (Liker, 2004).

Thank you

Thank you