LA TROBE UNIVERSITY

Project Management : Lecture 2

Part 1: Project Management Approaches

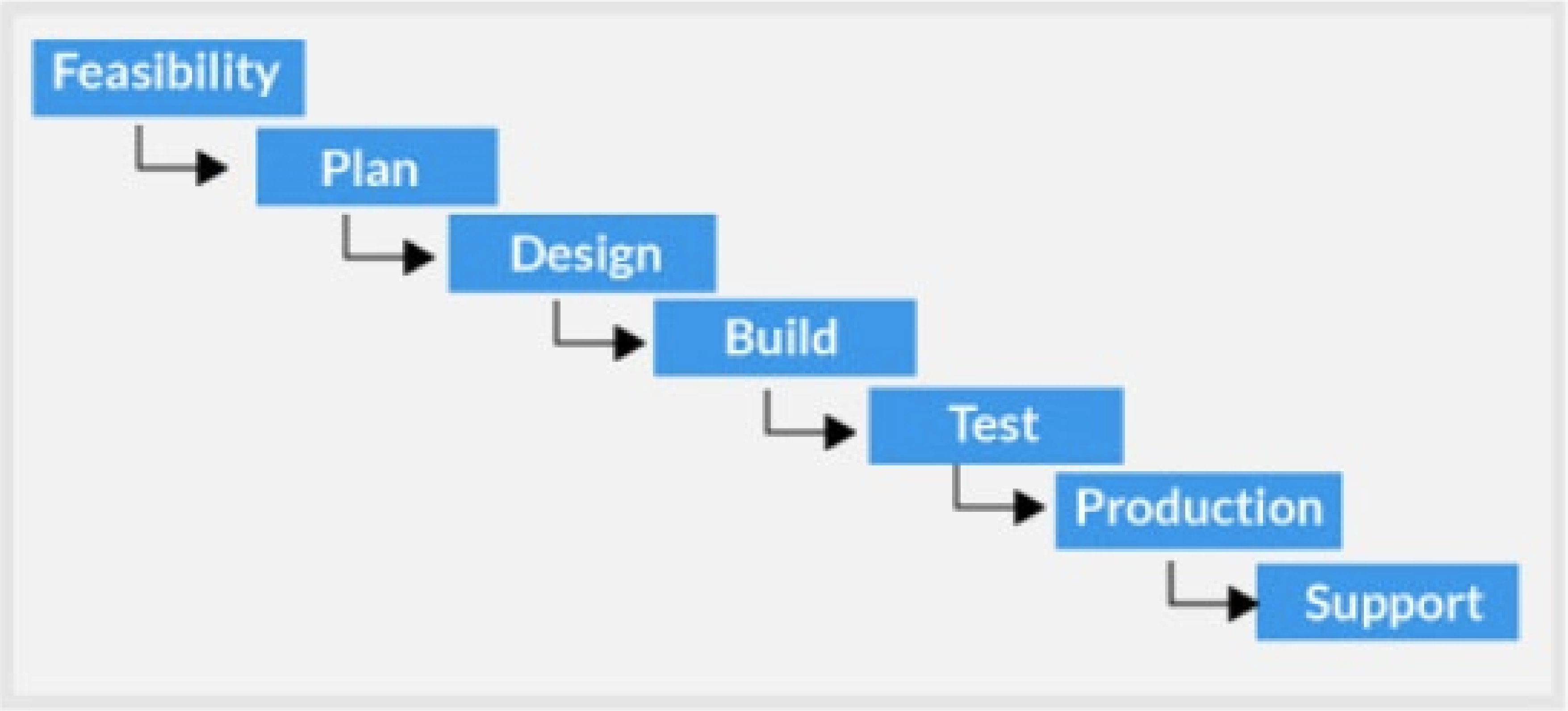
Topic Overview

- Phased approach
- Lean project management
- Iterative and incremental project management
- Agile Project Management

Traditional/phased approach (waterfall)

- It can be argued project management is as old as civilisation. All the ancient monuments are evidence of this.
- As a discipline project management started to take form in the 1950s.
- In the business world, organisations started to study the elements that made for more cost effective, timely and successful projects, and thus frameworks were developed.
- Traditional Project Management (also known as the phased approach or the waterfall approach) is linear in nature, and the phases of the project occur sequentially
- Each project follows the same stages which includes the stages from feasibility to support. Planning is done upfront without with no room for changing requirements and/or circumstances.
- This approach assumes that requirements are fixed, and time and cost are dynamic. It is because of this, traditional project management faces budget and timeline issues.

Traditional/phased approach



https://www.knowledgehut.com/blog/agile/agile-project-management-vs-traditional-project-management

Traditional/phased approach (waterfall)

Advantages of Traditional Project Management:

- Thorough planning and insight into the critical factor for success are the hallmarks of this approach.
- As a result accurate timelines and budgets are usually features of traditional project management
- The extensive documentation associated with this approach, also means project knowledge is more readily transfers and key person risk is reduced.

Disadvantages of Traditional Project Management

- The need for precision means this model is not suited to complex projects where forecasting is difficult.
- The longer the project the harder it is to look out into the future and set accurate timelines, so it should be avoided here too.
- Being a rigid approach, and it does not have the flexibility to adapt to fast changing and dynamic environments.

Lean Project Management

What is lean?

- The term “Lean” gained traction in the 1990s, it’s a philosophy that developed and refined by Toyota between 1948 and 1975, and commonly known as the Toyota Production System (TPS). Their production system is based on their philosophy and is focused on maximising value and minimising waste.
- Lean methodology can also be applied to projects. Projects that are structured to meet objectives while minimising waste and maximising value are known as “lean” projects.

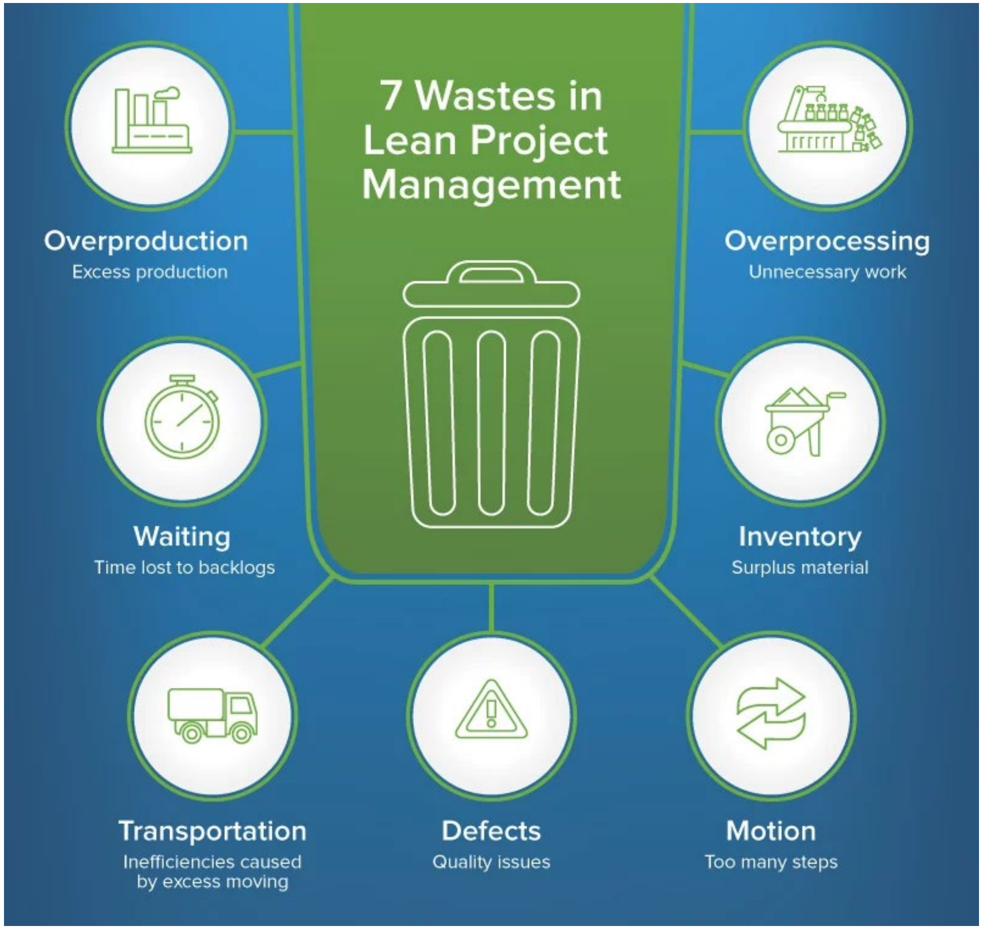
What is value?

Value was defined from the customers perspective. For something to have value, a customer had to be willing to pay for it**.**

What is waste?

Waste is anything that is done, and does not add value. It only contributes to cost and time.

Lean Project Management

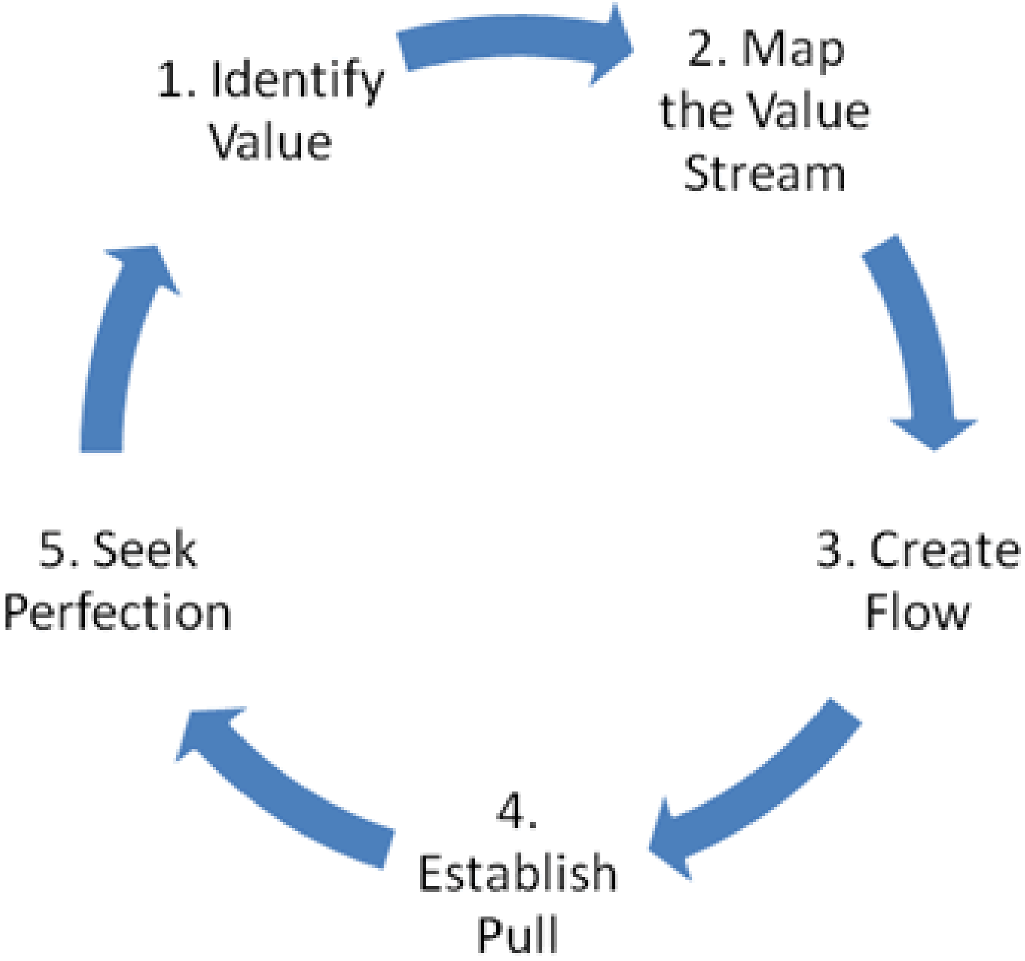


https://www.smartsheet.com/guide-to-lean-project-management

An 8th Waste is sometimes identified, which is the underutilised skills or knowledge of employees.

This is a result of organisations failing to utilize a workers full capabilities. Ex: I.T developer may be also really good at sales or relationship management but confined within the bounds of their job description. Or a company failing to take action on their employees ideas for improvement and innovation.

Lean Project Management



1. Identify Value

2. Map the Value Stream

3. Create Flow

4. Establish Pull

5. Seek Perfection

https://www.lean.org/WhatsLean/Principles.cfm

Lean Project Management

Advantages of Lean Project Management:

- The focus on value from the client's perspective makes decision making more objective and simpler, projects can always defer back to what is valuable from a client's perspective to move past divisive decision.
- Lean's focus on efficiency, means it can still be of benefit when resources are scarce, and budgets are low. Quality work can still be delivered while lowering costs.
- The philosophy of reducing the waste of peoples knowledge and skills means the project plan is created by incorporating the collective knowledge and experience of staff creating a sense of ownership and accountability.
- Delays and stoppages in work can be managed or mitigated through early identification of potential bottlenecks.
- Hand offs are well marked out and through pull planning project team members identify what they need and when they need it, allowing busy times to be well resourced and dead time to be re-provisioned.

Lean Project Management

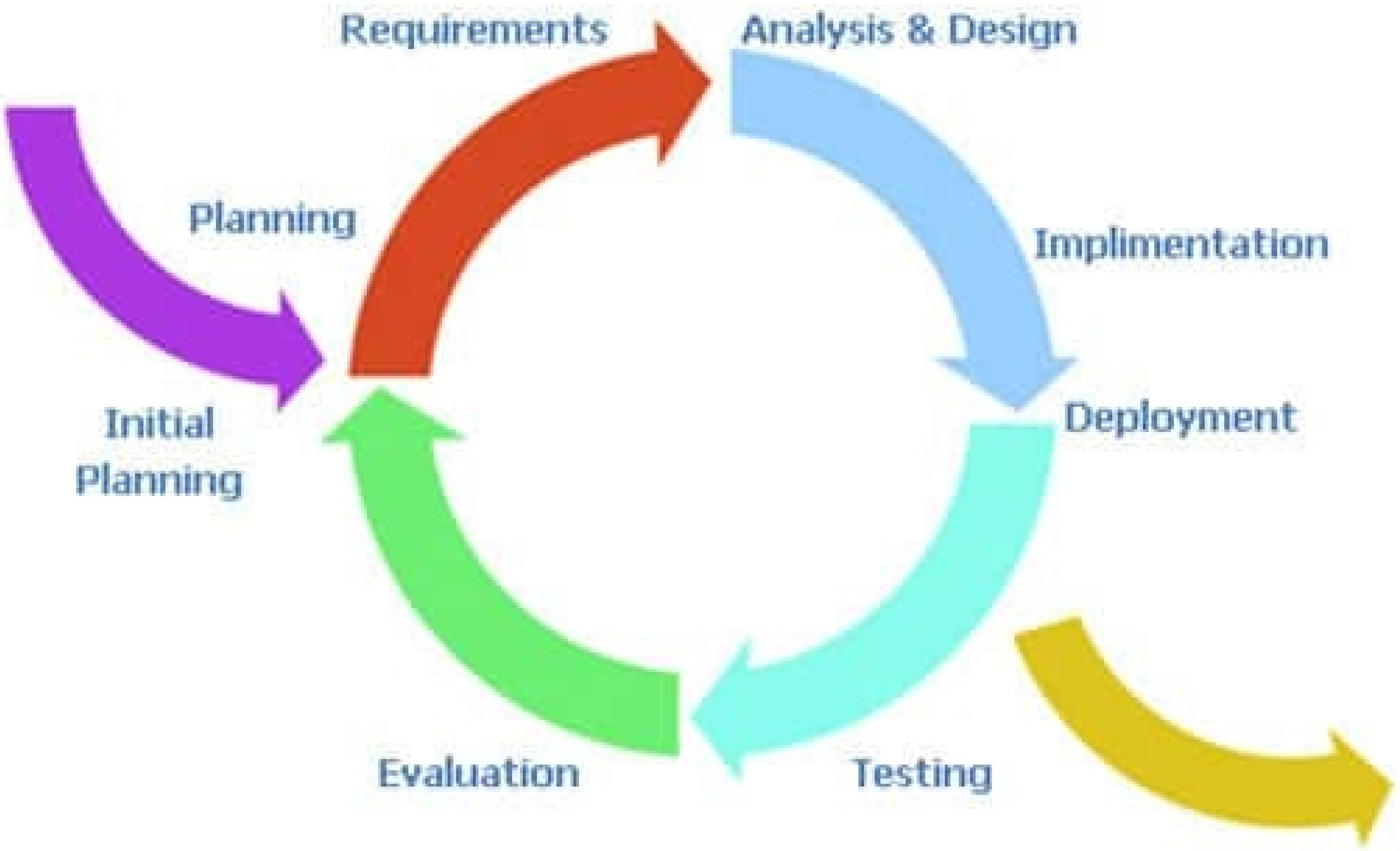
Disadvantages of Lean Project Management

- Project managers need be comfortable with playing the role of more of a facilitator and share more of their power and control. This can take time, for all stakeholders new to lean projects.
- The framework typically involves major modifications to existing processes, stake holders need to be prepared for the time and effort required to get these in line with new specifications.
- Decision making needs to be swift and precise to avoid slowing down the process

Iterative and incremental project management

- This project management approach is based on designing, building and testing through iterations (repeated cycles), and incrementally (in small time windows where a little more is added each time).
- By breaking the project work into small chunks, the project team and its stakeholders are able to learn from their each cycle and apply their learnings and improvements in the next iteration
- It involves extensive planning, design and risk elimination up-front and the project is run to the plan, any updates and changes require the scope and plans to be updated so the project is managed to the updated plan.
- Iteration's are sometimes called sprints or time-boxes. Time or scope are used to limit the size iterations. When time is used, the iterations conclude after the set time has lapsed regardless of how much progress has been made. If scope, the iterations take as long as is necessary until the scope has been developed and tested, and deemed complete.
- Usually iterations conclude with a demo to stakeholders. Feedback from this, good and bad is incorporated into the next cycle, and the scope is updated if necessary. This makes it easier to catch and correct errors early in the piece before things have gone too far and become too costly.

Iterative and incremental project management



https://www.testingexcellence.com/incremental-model/

Iterative and incremental project management

Advantages of Iterative and incremental project Project Management:

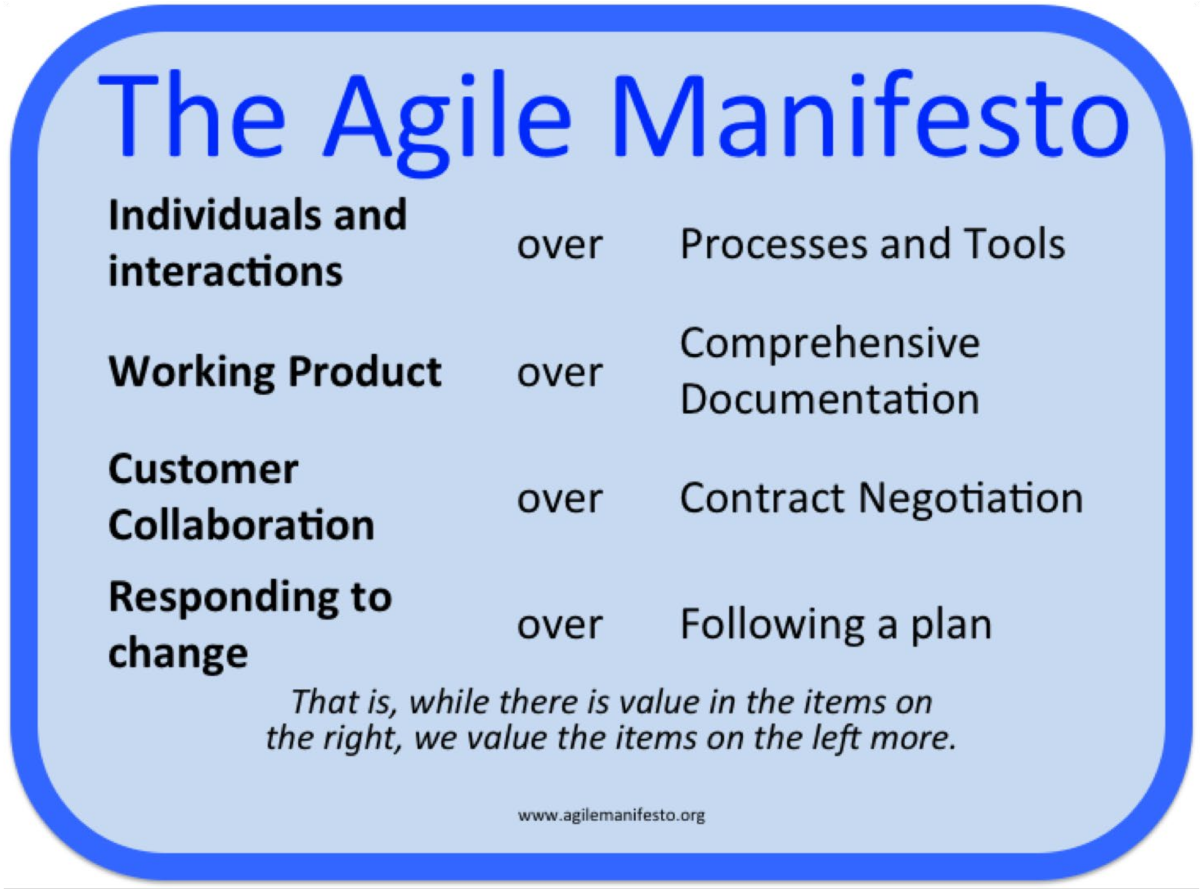
- Risk is more easily managed.
- The project team to be nimble and more easily incorporate changes. The short cycles allow the team to learn as they go and self correct.
- It's easier to measure progress.
- The ability to demo products and functionalities in each iteration, usually results in end products that are more in sync with client needs.
- Iterations allow the team to get better at working together, increase the pool of shared knowledge and improve at their given roles. Successful iterations mean repeatable processes can be developed, and utilisation and automation incorporated.

Disadvantages of Iterative and incremental project Project Management:

- It is usually an intensive approach and will most likely require intensive project management and more resources.
- It is harder to implement effectively. The project team need to be technically competent. Usually it requires multiskilled project teams, with members that are can perform more than one type of task e.g skilled in both front end and back end work.

Agile Project Management

- Agile is an iterative project management method built on the four principles from the Agile Manifesto:



The Agile Manifesto

Individuals and interactions	over	Processes and Tools
Working Product	over	Comprehensive Documentation
Customer Collaboration	over	Contract Negotiation
Responding to change	over	Following a plan

That is, while there is value in the items on the right, we value the items on the left more.

www.agilemanifesto.org

https://blog.itil.org/2014/08/allgemein/what-it-service-management-can-learn-from-the-agile-manifesto-and-vice-versa/

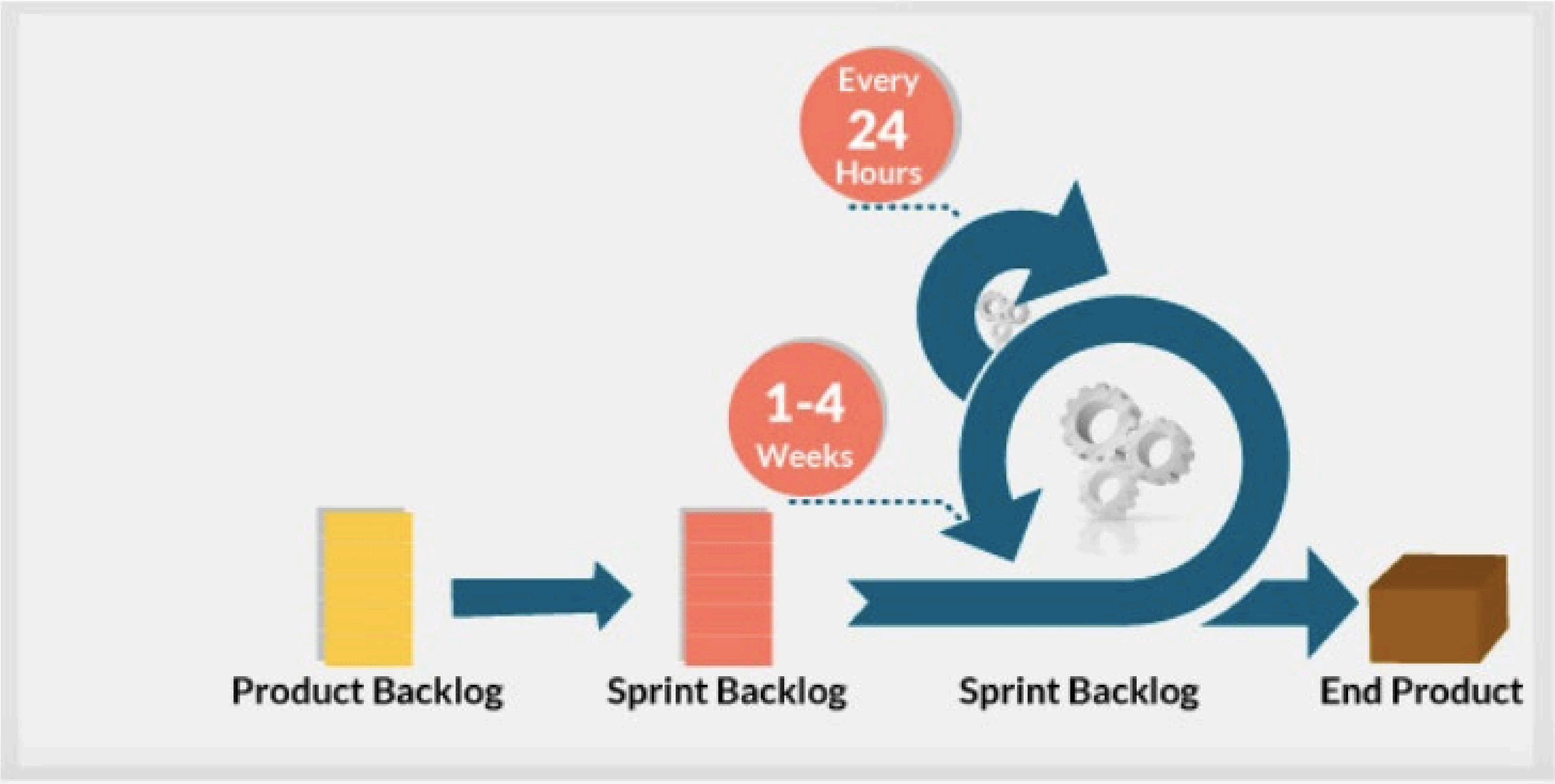
Agile Project Management

- In practice this means
  1. While both are important, having capable team member collaborating effectively is more important than tools and frameworks
  2. While good documentation is beneficial in guiding development, the purpose of development is to engineer software, not create documentation.
  3. A contract are legally important but it is more important to work closely with customers to identify and address their needs.
  4. Project plans are important, but they need to have room for changes in, stakeholders' needs and priority, and team members understanding of the problem and how to solve it.

Agile Project Management

- Project work is chunked up in to small sprints that usually last 2-3 weeks .
- There is less focus on detailed plans and documentation, instead the focus is on visual modelling, and collaboration. The project is controlled through feedback
- The team iterates for as long as it takes for the desired goal to be reached.
- Work is done on a "best endeavours" basis. All work is bill-able and as long as the customer has the budget for iterations, refinements can continue to be done with each cycle until the customer is content.

Agile approach



https://www.knowledgehut.com/blog/agile/agile-project-management-vs-traditional-project-management

Agile Project Management

Advantages of Agile Project Management:

- Risk is more easily managed.
- The project team to be nimble and more easily incorporate changes. The short cycles allow the team to learn as they go and self correct.
- It’s easier to measure progress.
- The ability to demo products and functionalities in each iteration, usually results in end products that are more in sync with client needs, sopner.
- Iterations allow the team to get better at working together, increase the pool of shared knowledge and improve at their given roles. Successful iterations mean repeatable processes can be developed, and utilisation and automation incorporated.
- Stakeholders are able to progressively prioritise work and output to match evolving market conditions.
- The project team is more likely to feel valued, since they work on things that have been validated to actually matter and receive detailed feedback from customers using the product frequently.

Agile Project Management

Disadvantages of Agile Project Management:

- Without a fixed plan and timeline to work to, there is an increased of risk of getting stuck in a limbo of redevelopment and re-work.
- The ambiguity surrounding the number of iterations, makes it hard to pin down a schedule and estimate costs for the scope of the project.
- Requires a highly disciplined project team. Flexibility can leave to a lack of accountability and thus create room for misbehaviour.
- Stakeholders and customers have to be accessible for review and feedback sessions, putting pressure on resourcing.
- Other departments such as sales and marketing need to be able to operate and plan, and support the product without a fixed road map and timeline. This complicates budgeting, communication plans and marketing complaints amongst other things.
- Management and Executives need know and understand their opinions do not drive the project but that Agile is a “pull” model and priorities are from customer and stakeholder feedback.
- Due to the above it is quite difficult to scale, or apply to larger projects and organisations.

Further reading…..

- https://activecollab.com/blog/project-management/traditional-project-management
- https://www.knowledgehut.com/blog/agile/agile-project-management-vs-traditional-project-management
- http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.74.2745&rep=rep1&type=pdf
- https://journals.sagepub.com/doi/abs/10.1002/pmj.21302
- http://www.strategosinc.com/toyota_production.htm
- https://ieeexplore.ieee.org/abstract/document/1204375
- https://technologyconversations.com/2014/01/21/software-development-models-iterative-and-incremental-development/

Next Lecture…..

Project Management in Software Development

LA TROBE UNIVERSITY

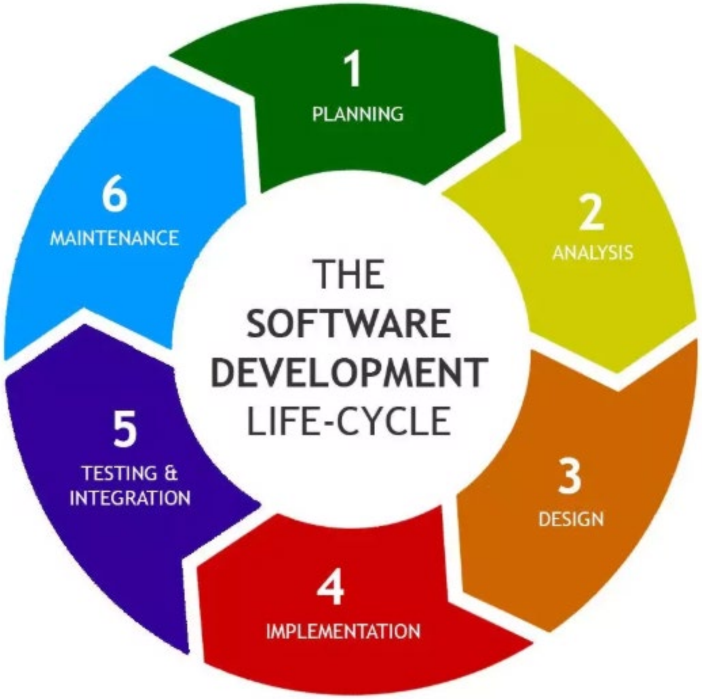
Part II: Project Management in Software Development

Topic Overview

- Software Development Lifecycle
- Project Management in Software Development
- Waterfall methodology
- Rapid Application development
- Agile methodology
- Secure Development Lifecycle (SDL)
- CI/CD
- DevOps and SecOps

Software Development Lifecycle

- Software Development Life Cycle (SDLC) is a process used to design, develop and test software. SDLC is also referred to as the Software Development Process.
- The SDLC framework consists of 6 phases including the following:
  - Planning/Requirement gathering
  - Analysis
  - Design
  - Implementation or coding
  - Testing & Deployment
  - Maintenance



SDLC Phases

1. Planning/Requirement gathering: Define customer requirements. During this phase, the BRS (Business Requirement Specification) document is created to capture the business/customer requirements.

2. Analysis: Define the technical requirements. During this phase, the SRS (Software Requirement Specification) document is created to capture the product requirements.

3. Design: Define the high level design (architecture) and low level design (product features/functions).

4. Implementation or coding: Translate the design into code. Implement the software components.

5. Testing & Deployment: Evaluate the software and fix defects. Once product is tested, deploy for production use.

6. Maintenance: Manage enhancements, updates and operation of the product.

Software Development methodologies

A Software development methodology is a framework used to structure the processes involved in developing a software system (or product). There are multiples of these methodologies, each with a slightly unique way of organising and varying the SDLC processes.

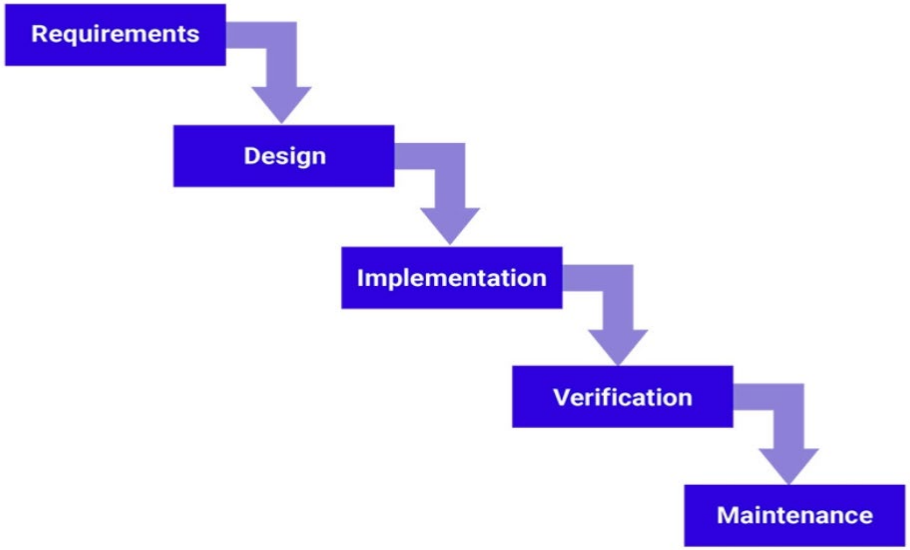
Some of the software development methodologies include:

- Waterfall development
- Rapid development
- Agile development
- Spiral development
- Lean development
- Xtreme programming

There are large overlaps within these methodologies and it often requires specific project contexts to separate two methodologies.

It helps to understand how each methodology is organised but it's even more important to understand what approach to use when managing a software development project. This approach may combine concepts of different methodologies.

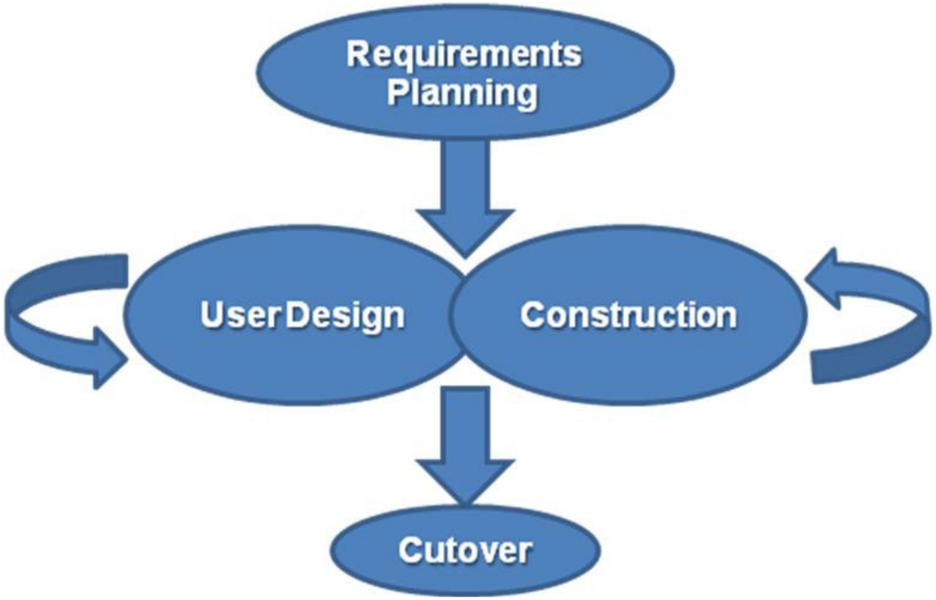
Waterfall



- The waterfall methodology includes a number of linear, sequential phases in which one phase is started after the previous phase has been completed.
- The waterfall method is considered simpler and easier to follow and includes the following phases: requirements, design, implementation, verification, maintenance.
- Although it suits simpler projects and PMs with little experience, the waterfall approach is also inflexible in that there is usually no modification/recalibration of completed phases.
- The waterfall model has also been blamed for large failed projects which run over budget and time due to the inflexibility of the controls.

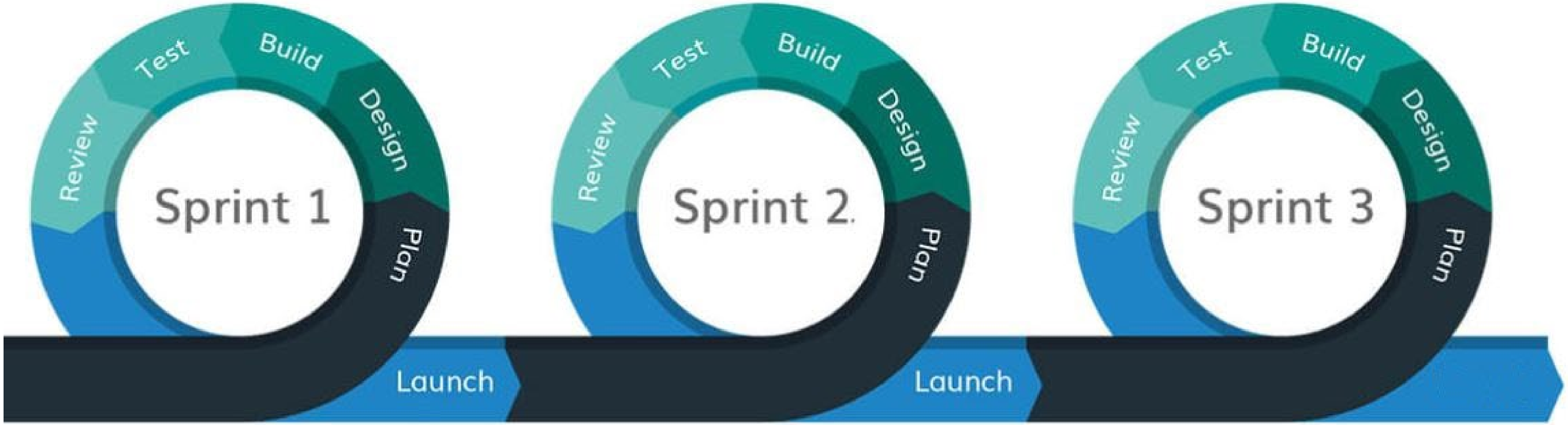
Rapid Application development

- The rapid application development methodology aims to produce prototypes/early versions of a software product quicker by focusing iteratively on user design and construction.
- This framework includes four phases: requirements planning, user design, construction, and cutover. The user design and construction phases repeat until customer confirms that the product meets requirements.
- Given the quick turnaround requirements and constant customer feedback, this approach suits projects with experienced and knowledgeable delivery team.



Agile

- Agile development is a methodology for developing software in short timeboxes/iterations in which features/functionality are deployed incrementally.
- Agile development essentially combines iteration and the continuous feedback to incrementally refine and deliver a software system.
- There are several different forms of agile development including Kanban, Scrum, Dynamic systems development method (DSDM) and eXtreme Programming (XP).
- Agile development's multiple iterations improve efficiency by finding and fixing defects, hence improving the product incrementally.



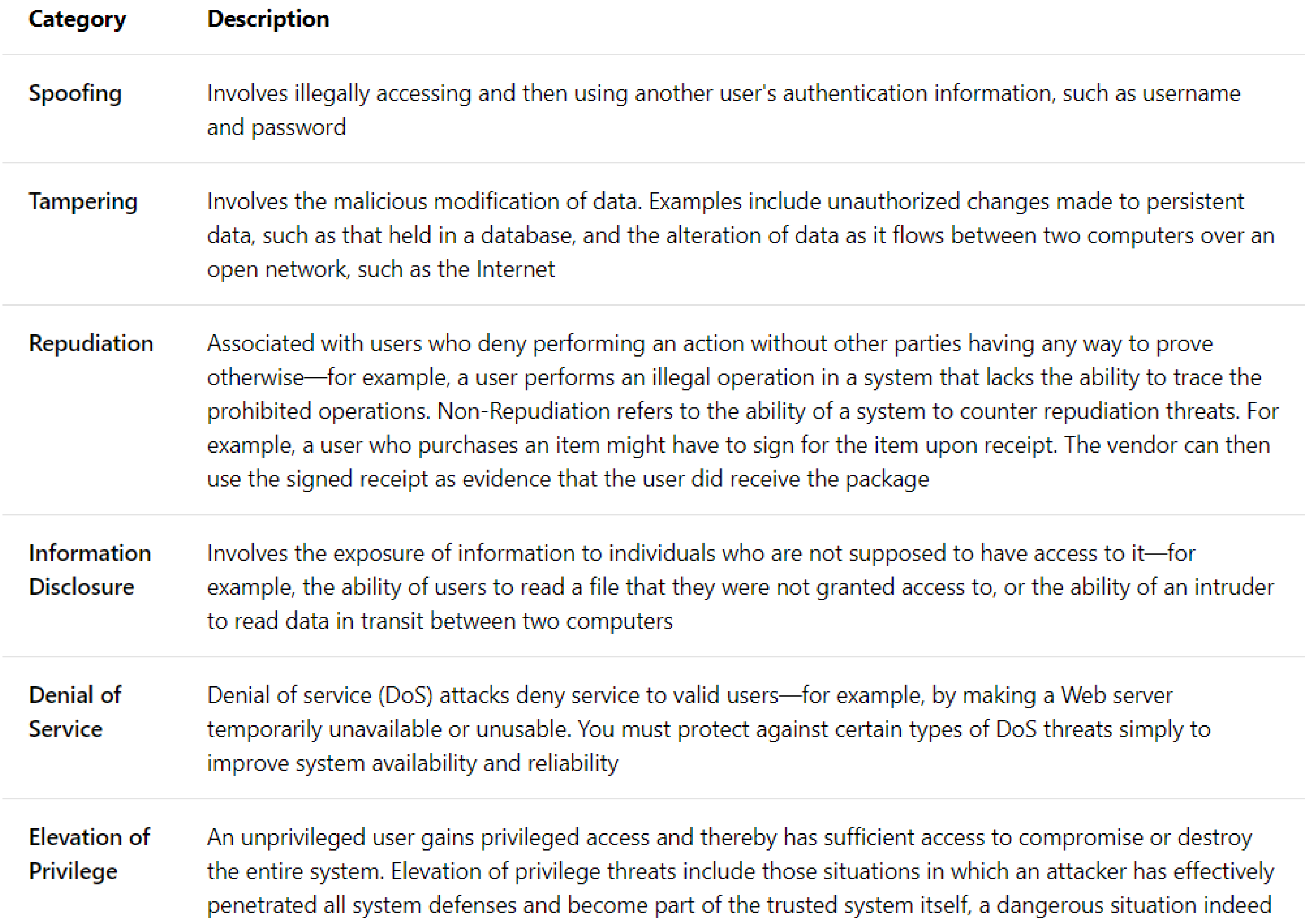
Secure Development Lifecycle (SDL)

- SDL is a Microsoft initiative and aims to introduce security and privacy considerations throughout all development phases.
- The objective of SDL is to help developers build secure software, to proactively address security compliance requirements, and reduce development costs.
- Some of the SDL practices include the following:
    - Define Security requirements
    - Define metrics and compliance reporting
    - Perform threat modelling to identify risk and mitigation actions
    - Manage 3rd party risk
    - Perform Static and Dynamic security testing

Threat Modelling in SDL

- Threat Modelling is a key part of the SDL and is a method of identifyng threats, attacks and vulnerabilities that could affect the application being developed; and formulating countermeasures to help mitigate or remove the threat.
- Threat Modelling includes 5 steps:
  - Defining security requirements.
  - Creating an application diagram.
  - Identifying threats.
  - Mitigating threats.
  - Validating that threats have been mitigated.
- The SDL methodology uses the STRIDE (Spoofing, Tampering, repudiation, Information disclosure, Denial of Service, Elevation of Privilege) method to categorise different types of threats and to simplify the development/implementation of countermeasures to the risks resultant from these threats.

STRIDE



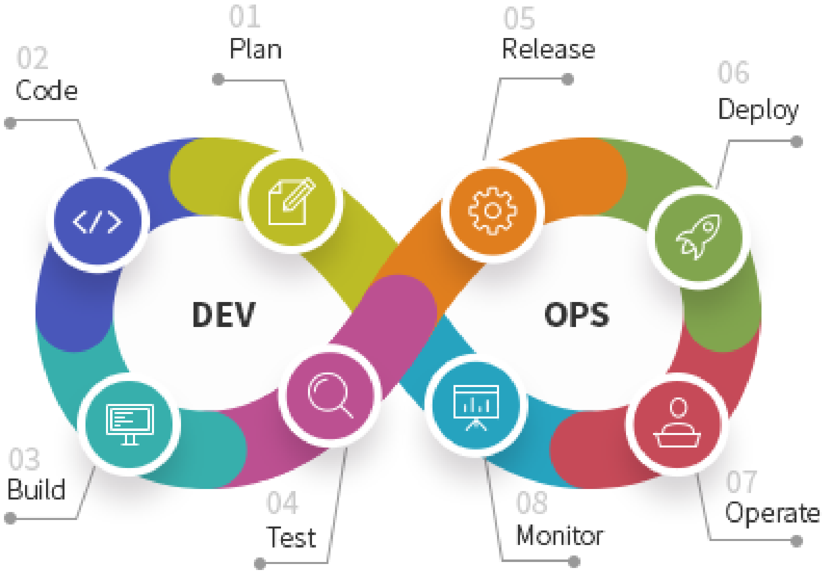
Category	Description
Spoofing	Involves illegally accessing and then using another user's authentication information, such as username and password
Tampering	Involves the malicious modification of data. Examples include unauthorized changes made to persistent data, such as that held in a database, and the alteration of data as it flows between two computers over an open network, such as the Internet
Repudiation	Associated with users who deny performing an action without other parties having any way to prove otherwise—for example, a user performs an illegal operation in a system that lacks the ability to trace the prohibited operations. Non-Repudiation refers to the ability of a system to counter repudiation threats. For example, a user who purchases an item might have to sign for the item upon receipt. The vendor can then use the signed receipt as evidence that the user did receive the package
Information Disclosure	Involves the exposure of information to individuals who are not supposed to have access to it—for example, the ability of users to read a file that they were not granted access to, or the ability of an intruder to read data in transit between two computers
Denial of Service	Denial of service (DoS) attacks deny service to valid users—for example, by making a Web server temporarily unavailable or unusable. You must protect against certain types of DoS threats simply to improve system availability and reliability
Elevation of Privilege	An unprivileged user gains privileged access and thereby has sufficient access to compromise or destroy the entire system. Elevation of privilege threats include those situations in which an attacker has effectively penetrated all system defenses and become part of the trusted system itself, a dangerous situation indeed

CI-CD

- Continuous Integration and Continuous Delivery (CICD) is a set of practices to help development teams to deliver code changes more frequently and reliably. CICD is also known as Continuous Deployment.
- Most modern applications are developed in different platforms and using different tools. There needs to be a mechanism to integrate and validate these changes.
- The goal of CICD is to establish a consistent and automated way to build, package, and test applications and automatically deliver applications to various platforms.
- For developers, CICD help developers deliver value faster and more transparently

Dev Ops

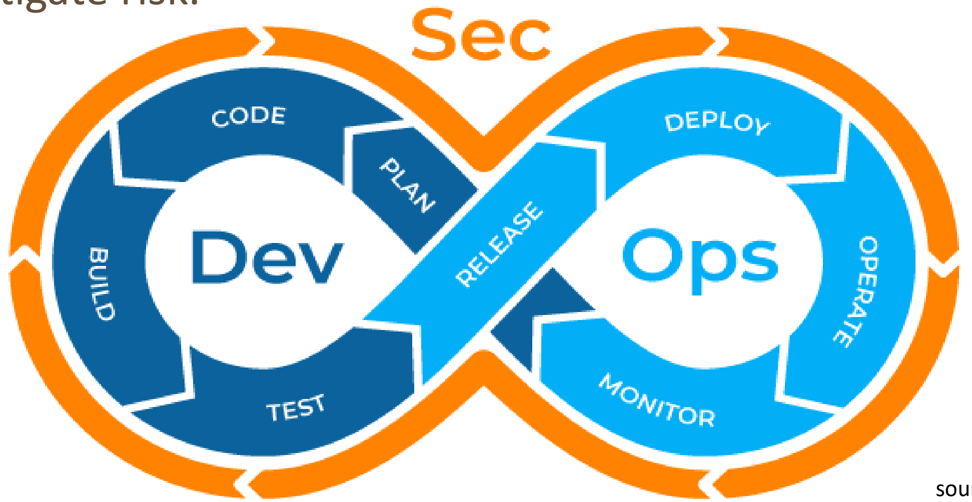
- In the context of Agile development and CI-CD, Dev-Ops is a set of software development practices that combine software development (Dev) and IT operations (Ops) to shorten the development life cycle and incrementally release features, fixes, and updates frequently.
- Dev-Ops helps organisations to deliver software applications faster while also evolving and improving products at a faster pace. This faster deployment also helps organizations to keep up with frequently changing customers preferences and trends and to compete more effectively with alternative products.



source: https://www.diaspark.com/devops-service/

Sec Ops

- A common enterprise security approach in software development has been to;
  - Scan code while the software is being built, then
  - Conduct a dynamic scan and a full penetration test before production release or in production before go-live.
- With most companies adopting agile software development and CI-CD, the security of the software must be tested and bolstered with each incremental release of the software.
- SecOps (Security Operations) is a joint effort between IT security and operations teams using processes and technology to manage an organisation's security function.
- SecOps involves ongoing collaboration between IT Security and IT Operations to effectively mitigate risk.



source: https://secops-isa.org

Why Sec-Ops

- Historically, the IT security and IT operations teams have had complementary roles and occasionally an antagonistic relationship.
- In some instances, Security can be viewed as compliance driven and most of the compliance requirements need to be implemented by IT operations.
- The goal of SecOps is to foster collaboration between the Security and IT operations teams.
- SecOps also aims to improve security by baking security into the software development lifecycle.

Further reading…..

- https://www.brightwork.com/blog/what-are-the-characteristics-of-a-project
- https://moodle2016-17.ua.es/moodle/pluginfile.php/80324/mod_resource/content/2/agile-manifesto.pdf
- https://www.pmi.org/learning/library/project-complexity-determine-rigor-9874
- https://www.irbnet.de/daten/iconda/CIB_DC24048.pdf
- http://www.asapm.org/asapmag/articles/PrezSez10-07.pdf

Next Lecture…..

Project Management phases

Thank you