National College of Ireland

H9EEAI- Agile Development and AI Engineering

Course: H9EEAI- Engineering and Evaluating Artificial Intelligence Systems

Outline

Software Development Process

- SDLC
    - SRS

Traditional Models

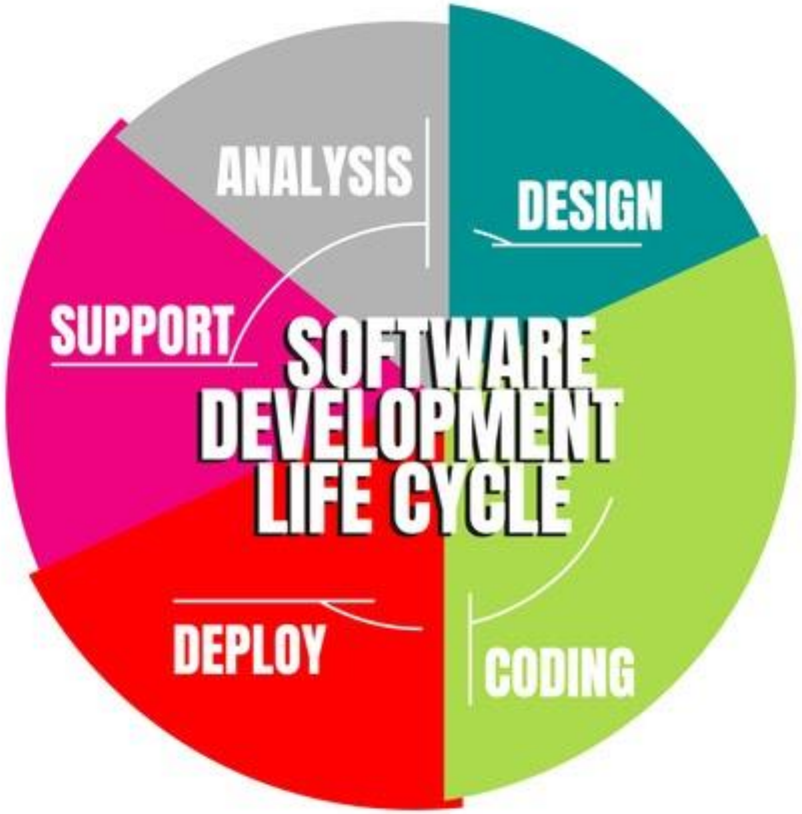
- Waterfall
- V Shape Model
- Incremental SDLC
- Prototype Model
- Spiral Model

Agile Methodologies

How Agile Methodologies are used in AI Engineering

SDLC Phases

1. Planning
2. Requirement Analysis
3. Design
4. Implementation
5. Testing
6. Deployment
7. Maintenance



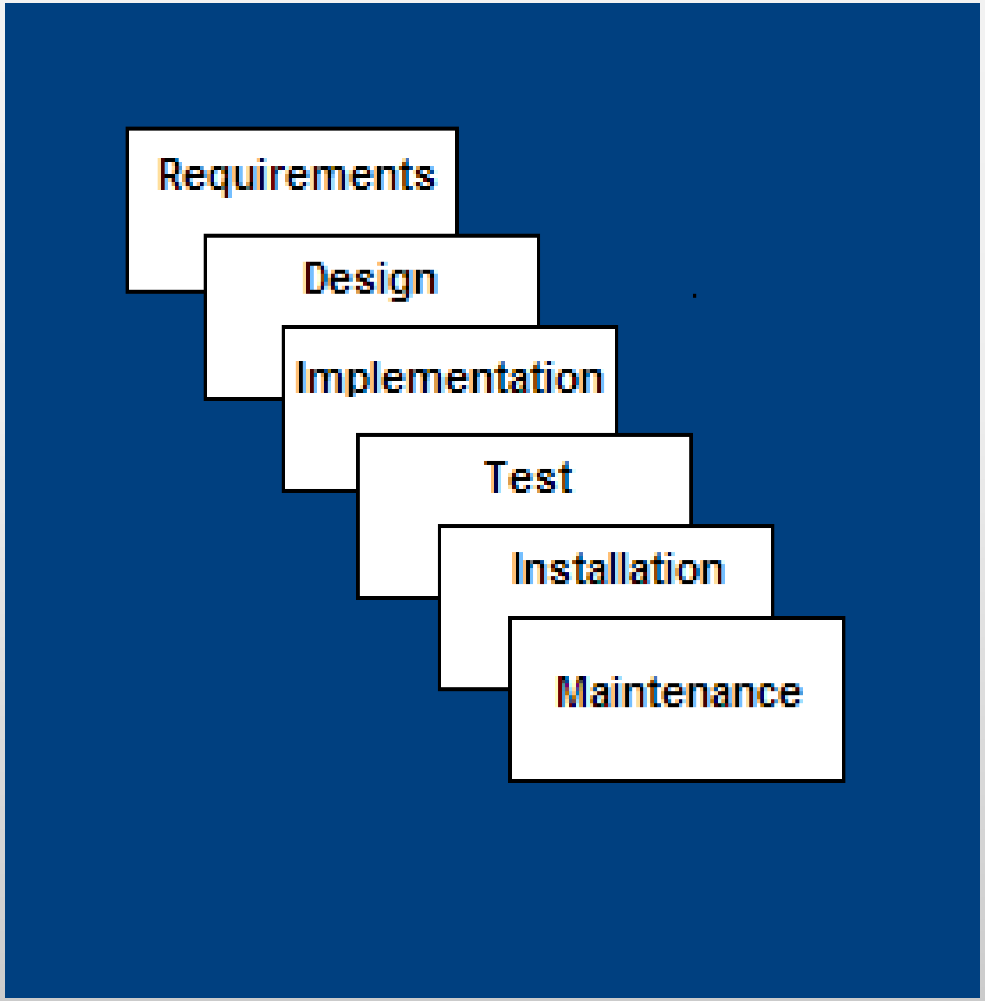
https://www.techopedia.com/definition/22193/software-development-life-cycle-sdlc

National College of Ireland

Template for SRS document

SRS Template

Waterfall Model



- **Requirements** – defines needed information, function, behavior, performance and interfaces.
- **Design** – data structures, software architecture, interface representations, algorithmic details.
- **Implementation** – source code, database, user documentation, testing.

National College of Ireland

Waterfall: Advantages & Disadvantages

Advantages:

Kind of Base Model

Simple & easy

Works when requirements are fixed and clear

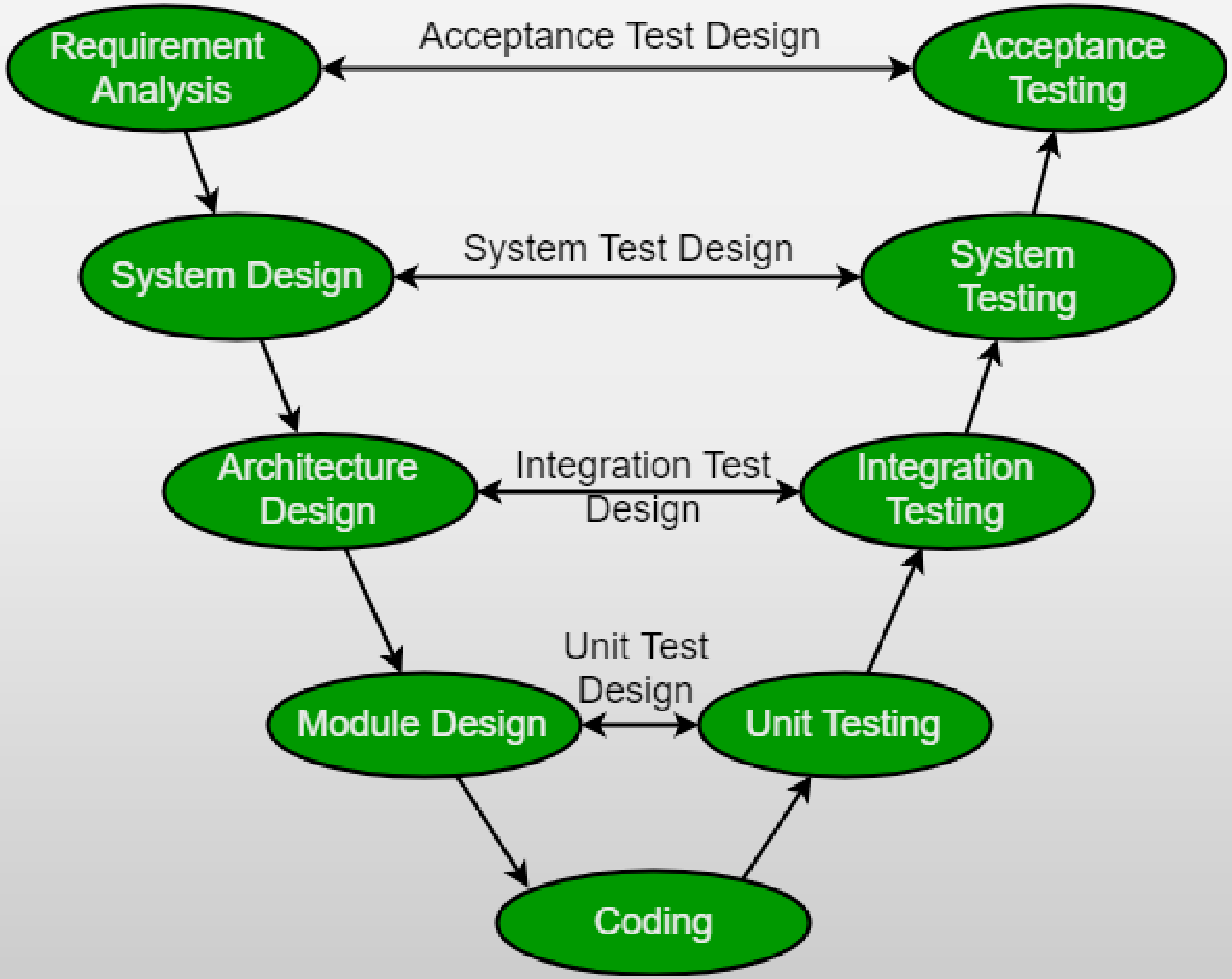
Disadvantages:

No feedback

Can’t work in parallel

High Risk (can’t see any output till end)

V-Shaped SDLC Model



https://www.geeksforgeeks.org/software-engineering-sdlc-v-model/

- Verification & Validation Model
  - **Verification**: It involves static analysis technique (review) done without executing code.
  - **Validation**: It involves dynamic analysis technique (functional, non-functional), testing done by executing code.
- Extension of Waterfall
- Testing is done at every step (parallel).
- Often used for Critical systems, such as aerospace and defence systems, because of its emphasis on thorough testing

National College of Ireland

V Model: Advantages & Disadvantages

Advantages:

Simple and easy to understand

Good understanding of the project at very beginning

Every component is testable - building an error-free and good quality product.

Progress can be tracked – due to documentation – provides a clear link between the requirements and the final product

Proactive defect tracking

Disadvantages:

No user feedback

Inflexibility - linear and sequential model, which can make it difficult to adapt to changing requirements or unexpected events.

Time-Consuming – focus on lots of documentation

Prototyping SDLC Model

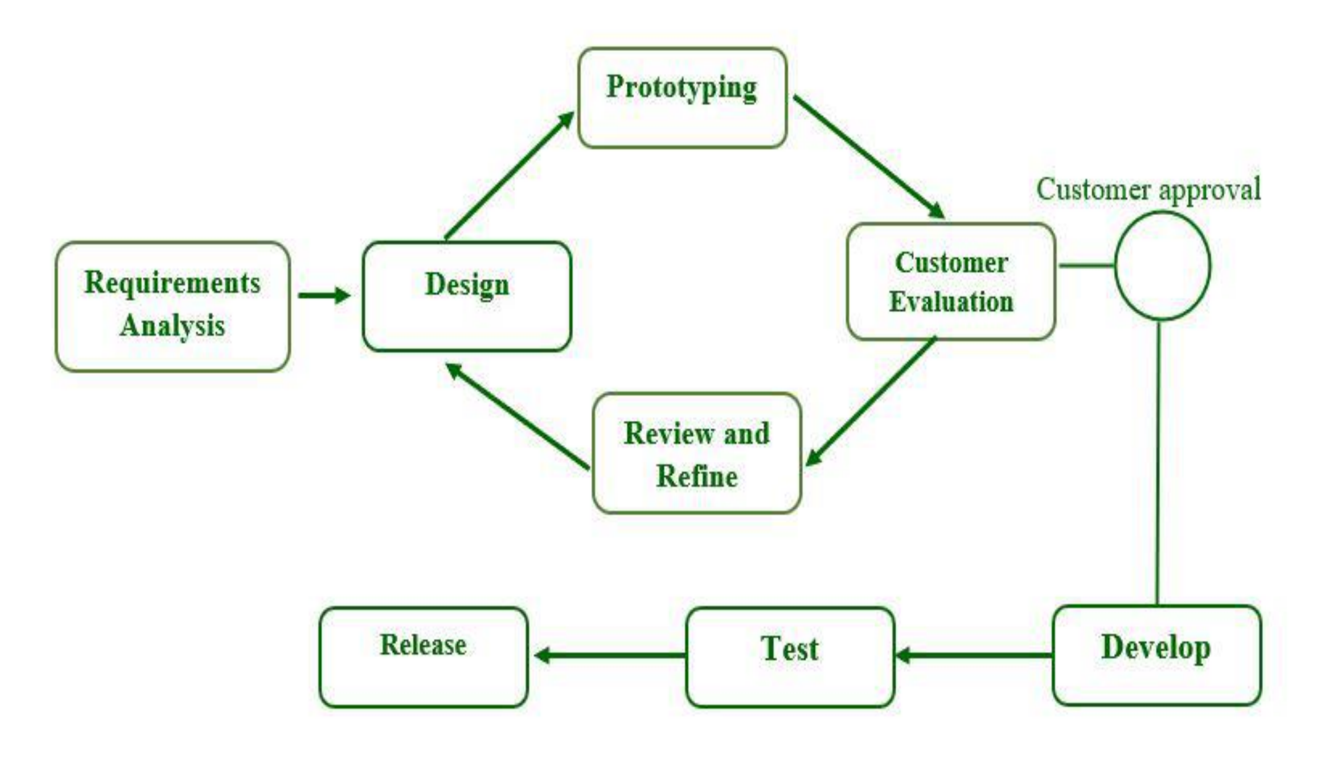
Require a high level of user interaction

Prototypes are developed for gathering and refining requirements

Once prototype is final then

SDLC phases are used to

develop complete application



https://www.geeksforgeeks.org/software-engineering-incremental-process-model/

Prototyping Model

Advantages

Users help to shape the future.

Assists team members in effectively communicating.

Customer satisfaction exists, and he can feel the product from the start.

Minimum risk of software loss.

Disadvantages

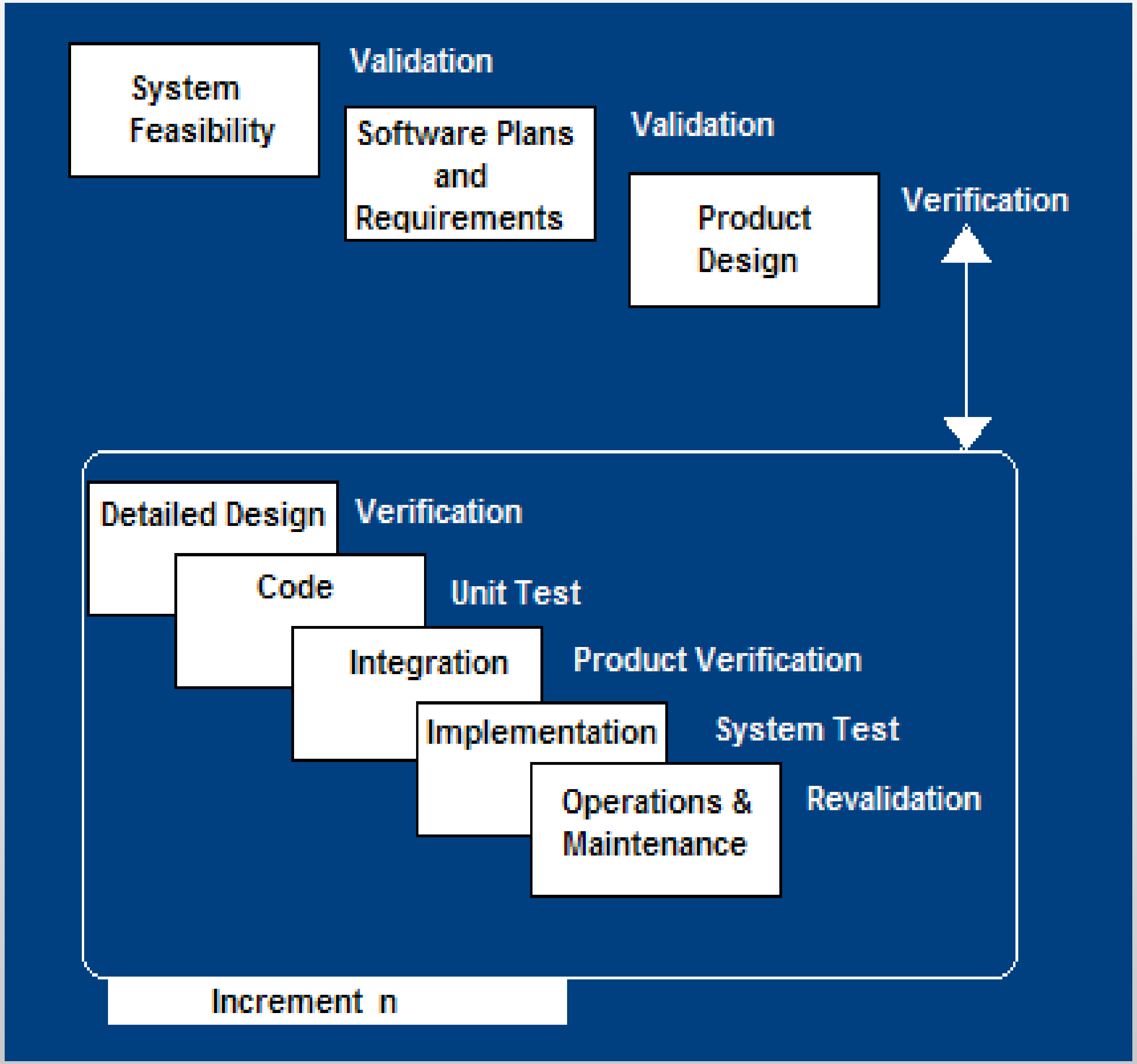
Prototyping is a time-consuming and labor-intensive process.

The cost of creating prototyping

overabundance of change requests.

Poor documentation as a result of changing customer needs.

Incremental SDLC Model



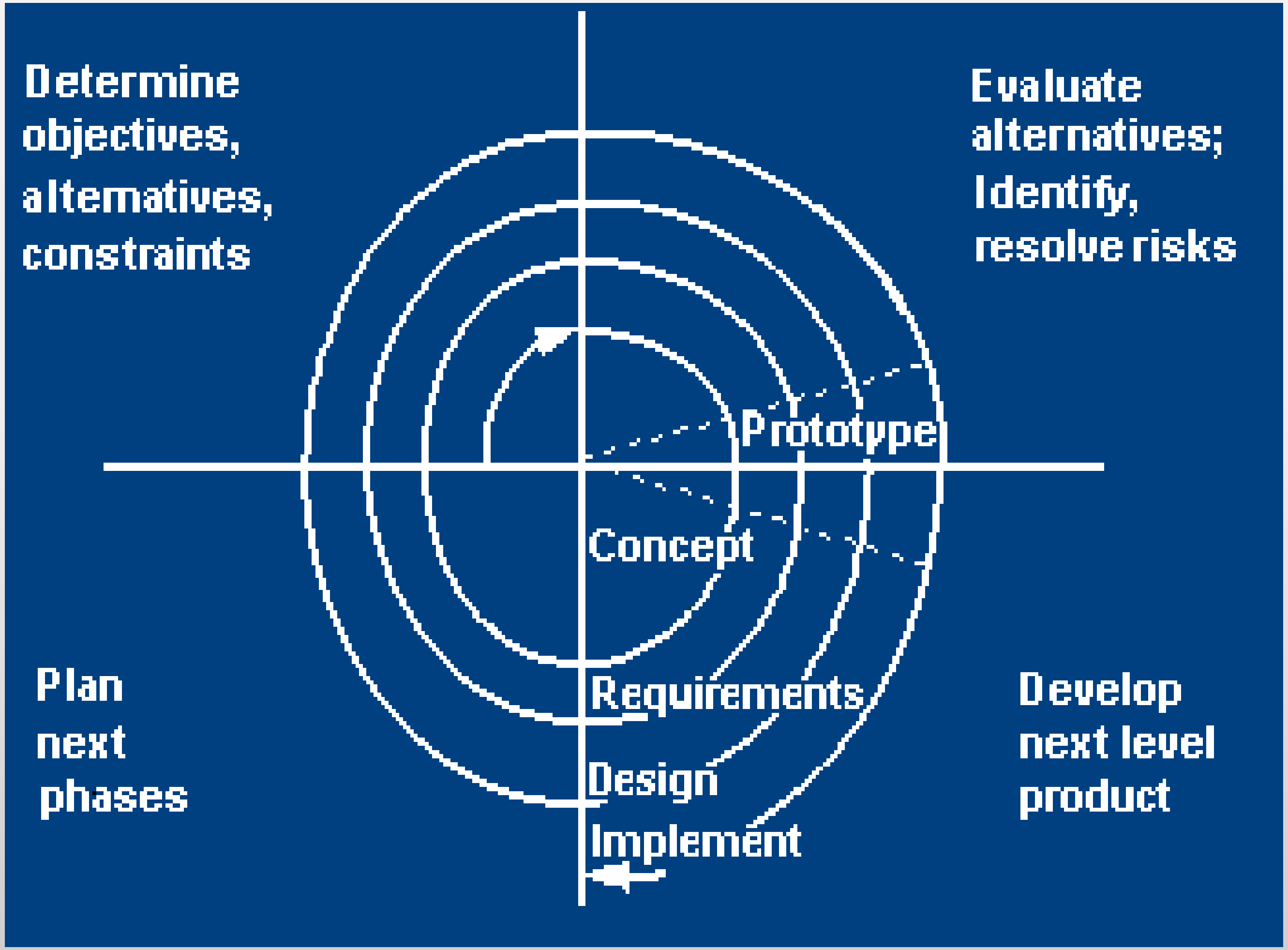
- Construct a partial implementation of a total system
- Then slowly add increased functionality
- The incremental model prioritizes requirements of the system and then implements them in groups.
- Each subsequent release of the system adds function to the previous release, until all designed functionality has been implemented.

National College of Ireland

Incremental Model

- **Advantages:**
  - It is easy for breakdown of tasks because of divide and conquer approached used.
  - It has lowers initial delivery cost.
  - It is good to use when requirements are known up-front.
  - It is good to use when projects having lengthy developments schedules.
- **Disadvantages of using Incremental process model :**
  - It requires a good planning designing.
  - Definition of system should be complete and clear.
  - Higher costs: Since each increment requires its own planning, design, coding, testing, and deployment

Spiral SDLC Model


Determine objectives, altematives, constraints
Evaluate alternatives; Identify, resolve risks
Prototype
Concept
Requirements
Design
Implement
Plan next phases
Develop next level product

- It is a risk-driven model
- Each cycle involves the same sequence of steps as the waterfall process model
- Meta Model

National College of Ireland

Spiral Model

Advantages

Risk Handling - risk analysis and risk handling at every phase

Good for large projects

Flexibility in Requirements- Change requests in the Requirements at later phase

Customer Satisfaction- Customer can see the development of the product at the early phase

Improved Communication - regular evaluations and reviews, which can improve communication

Improved Quality – multiple iteration

Disadvantages

Complex

Expensive-

Too much dependability on Risk Analysis

Difficulty in time management- iteration are unknown

Resource Intensive –significant investment in planning, risk analysis, and evaluations

The need of new methodologies

Traditional SDLC Models Applications Issues:

- Single point of failure Applications & monolithic applications
- Time required to push the changes - Maintenance time. Ideally systems should never be down like amazon, Netflix, Google etc.

So, in very simple term, the objective of agile to address these issues.

What are the 4 pillars of Agile?

Individuals & Interactions over processes and tools

Agile value team collaboration and teamwork over working independently and doing things "by the book.”

Working software over comprehensive documentation

The software that Agile teams develop should work. Additional work, like documentation, is not as important as developing good software.

Customer collaboration over contract negotiation

Customers are extremely important within the Agile methodology. Agile teams allow customers to guide where the software should go.

Responding to change over following a plan

One of the major benefits of Agile project management is that it allows teams to be flexible.

What are the 12 Agile principles?

1. **Satisfy customers through early, continuous improvement and delivery.**
2. **Welcome changing requirements, even late in the project.**
3. **Deliver value frequently.**
4. **Break the silos of your projects.**
5. **Build projects around motivated individuals.**
6. **The most effective way to communicate is face-to-face.**
7. **Working software is the primary measure of progress.**
8. **Maintain a sustainable working pace.**
9. **Continuous excellence enhances agility.**
10. **Simplicity is essential.**
11. **Self-organizing teams generate the most value.**
12. **Regularly reflect and adjust your way of work to boost effectiveness.**

https://agilemanifesto.org/principles.html
https://asana.com/resources/agile-methodology

Advantages of Agile

Following are some of the benefits:

- Adaptability
- Customer Satisfaction
- Faster Time to Market
- Improved Quality
- Enhanced Visibility and Transparency
- Risk Mitigation

Some Agile Methods

Scrum

Extreme Programming (XP)

Feature Driven Development (FDD)

Rapid Application Development (RAD)

Adaptive Software Development (ASD)

Crystal Clear

Dynamic Software Development Method (DSDM)

Scrum

It is an iteration of:

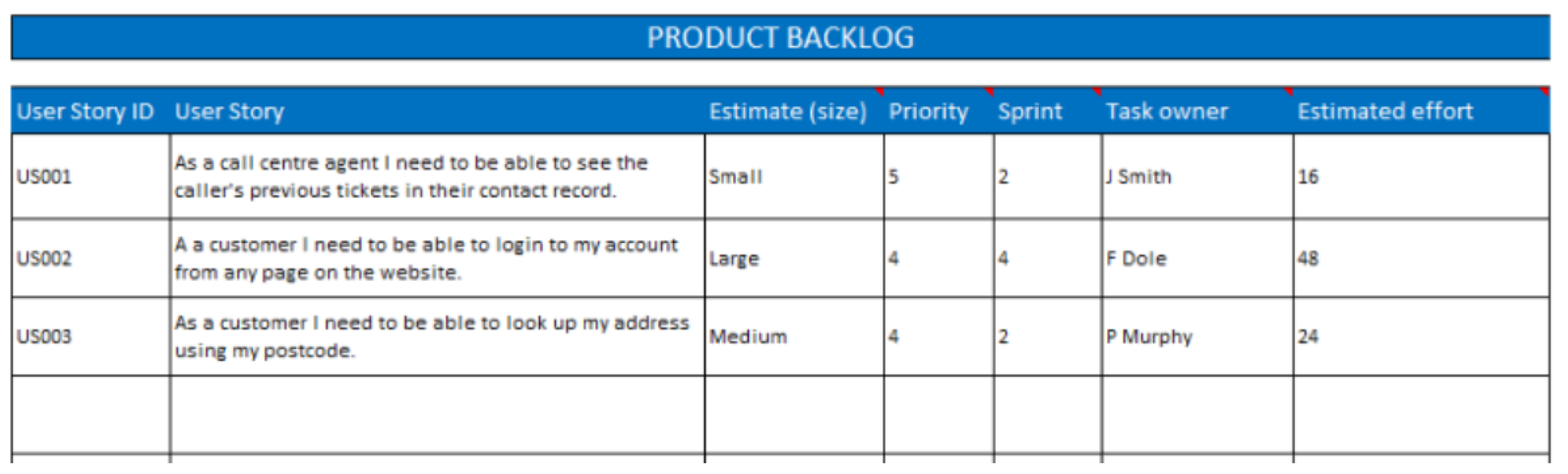
- Plan
- Build
- Test
- Review

- Scrum Team and Roles
  - **Product owner** – it is a manager, and it could be non-technical person
  - **Scrum master** – Team lead responsible for manage team and daily operations
  - **Actual Team member** – developer/testers etc.

Scrum based development

- **Product backlog:**
  - Work that needs to be done. In order to have the best Scrum sprint possible, make sure you have your product backlog clearly documented in one place.
  - Each task would iterate over its plan, build, test, and review. For example, front end, catalogue, payment etc.
  - The product owners should frequently reorder and refresh the product backlog based on new information from customers, from the market, or from the project team.

Product backlog



PRODUCT BACKLOG

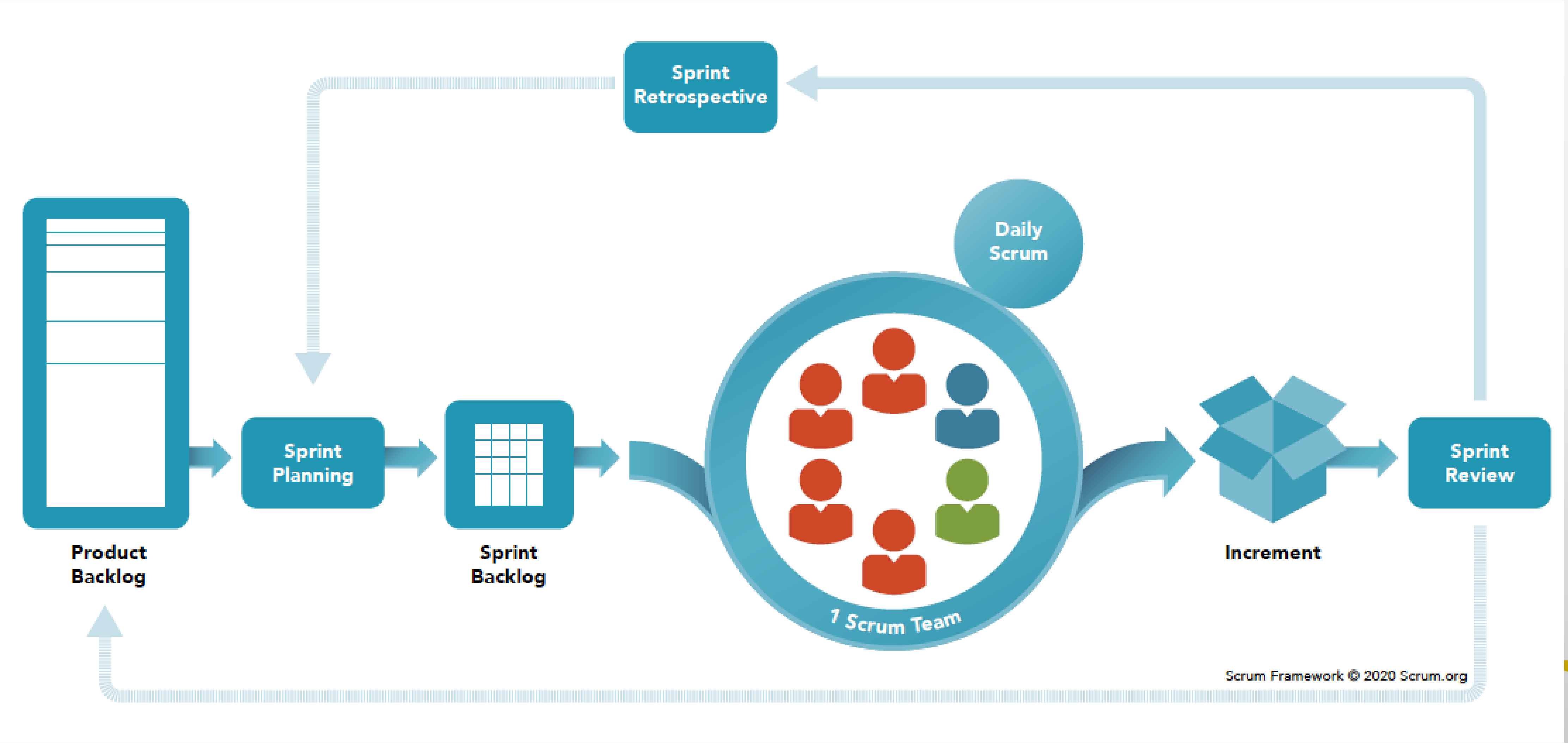
User Story ID	User Story	Estimate (size)	Priority	Sprint	Task owner	Estimated effort
US001	As a call centre agent I need to be able to see the caller's previous tickets in their contact record.	Small	5	2	J Smith	16
US002	A a customer I need to be able to login to my account from any page on the website.	Large	4	4	F Dole	48
US003	As a customer I need to be able to look up my address using my postcode.	Medium	4	2	P Murphy	24

https://www.stakeholdermap.com/agile-templates/product-backlog-template.html

Sprint backlog

- **Sprint Planning** – the product owner/scrum master will explain the objective of the task
- **Daily Scrum** – 15 minutes meeting everyday to discuss progress, status, issues etc.
- **Sprint Review** – these meetings are held in around 2-week times. During this time, scrum team will present the work that's "Done" for stakeholder approval or inspection.

Scrum workflow



Scrum Framework © 2020 Scrum.org

Sprint Retrospective

- During the Sprint Retrospective, the team discusses:
  - What did you like about the last sprint?
  - What did you learn from the last sprint?
  - What was lacking in the last sprint?

Extreme Programming - XP

Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team.

Extreme Programming is successful because it stresses customer satisfaction.

When Applicable/or should use?

- Dynamically changing software requirements
- Risks caused by fixed time projects using new technology
- Small, co-located extended development team
- The technology you are using allows for automated unit and functional tests

XP Values

Communication

XP stresses the importance of the appropriate kind of communication – face to face discussion with the aid of a white board or other drawing mechanism.

Simplicity

Simplicity means “what is the simplest thing that will work?”

Feedback

Through constant feedback about their previous efforts, teams can identify areas for improvement and revise their practices.

Courage

Kent Beck defined courage as “effective action in the face of fear” (Extreme Programming Explained P. 20). - you need courage to raise organizational issues that reduce your team’s effectiveness.

Respect

The members of your team need to respect each other in order to communicate with each other, provide and accept feedback

XP Practices (1-6)

1. **Planning game** – determine scope of the next release by combining business priorities and technical estimates
2. **Small releases** – put a simple system into production, then release new versions in very short cycle
3. **Metaphor** – all development is guided by a simple shared story of how the whole system works
4. **Simple design** – system is designed as simply as possible (extra complexity removed as soon as found)
5. **Testing** – programmers continuously write unit tests; customers write tests for features
6. **Refactoring** – programmers continuously restructure the system without changing its behavior to remove duplication and simplify

XP Practices (7 – 12)

7. **Pair-programming** -- all production code is written with two programmers at one machine
8. **Collective ownership** – anyone can change any code anywhere in the system at any time.
9. **Continuous integration** – integrate and build the system many times a day – every time a task is completed.
10. **40-hour week** – work no more than 40 hours a week as a rule
11. **On-site customer** – a user is on the team and available full-time to answer questions
12. **Coding standards** – programmers write all code in accordance with rules emphasizing communication through the code

XP is “extreme” because

Commonsense practices taken to extreme levels

If code reviews are good, ***review code all the time*** (pair programming)

If testing is good, everybody will ***test all the time***

If simplicity is good, keep the system in the simplest design that supports its current functionality. (***simplest things that work***)

If design is good, everybody will design daily (***refactoring***)

If architecture is important, everybody will work at defining and refining the architecture (***metaphor***)

If integration testing is important, build and ***integrate test several times a day*** (continuous integration)

If short iterations are good, ***make iterations really, really short*** (hours rather than weeks)

Agile AI Engineering

How to reach the maximum potential of a business using AI?

A set of principles need to be employed that are aimed at deriving maximum benefits from all organizational procedures.

- Depending on the nature of a business' processes, Agile methodologies can be adopted if conditions permit.

To scale the rate of AI development and implementation in enterprises, the iterative processes as prescribed under the Agile manifesto can be implemented.

- Although there are some teething issues when it comes to the implementation of Agile to deliver value through AI, there are benefits that are worth taking the time to adapt these methods.

Bringing Agile and AI Together: Benefits

Improved quality - It is not enough in AI development to put out the initial version of any software without constantly developing and improving its functionality for later versions.

Agile talks about how development is an iterative process, and how it is necessary to refine your products constantly.

Flexibility - Agile principles have a wide range of uses and applications. Depending on the function of AI in your enterprise, Agile AI practices can be formulated as per those requirements.

Fixed Timeframe - Agile Artificial Intelligence allows you to work within smaller time frames, known as sprints. The primary focus of Agile is customer satisfaction. Efficient work directly correlates to improved customer satisfaction, as delivery and feedback can be met with greater consistency.

Better monitoring - Agile AI allows organizations to collect, organize and compute enormous amounts of data. Upon analysis, this can be used to pinpoint problems and design solutions, recommend courses of action, and in some scenarios, also initiate responses.