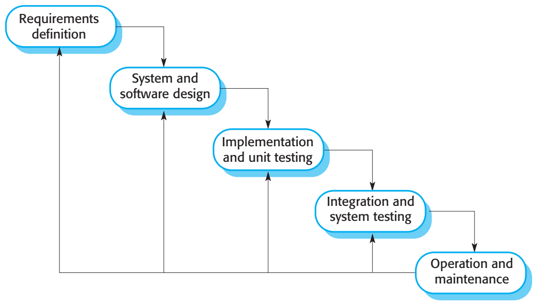
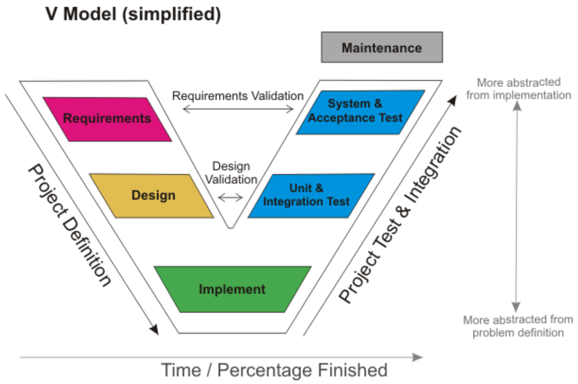
SE202
Introduction to Software Engineering

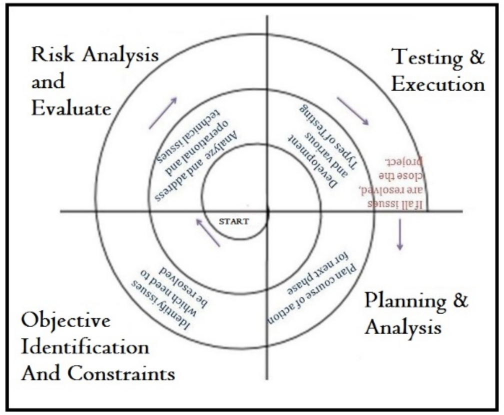
Lecture 9-2

Software Process Models
Adaptive

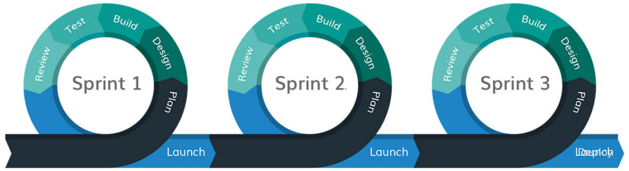
Pathathai Na Lumpoon


Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance




Risk Analysis and Evaluate
Testing & Execution
Analyze and address operational and technical issues
Development and various Types of Testing
If all issues are resolved, close the project.
START
Identify issues which need to be resolved
Plan course of action for next phase
Objective Identification And Constraints
Planning & Analysis

Agile Methodology



Software Process *Model*

- A software process model is an abstract representation of a process. It presents a description of a process from some particular perspective.
  - Ad-hoc software development, code-and-fix, waterfall, evolutionary, rapid prototyping, spiral, etc.
- Different process model is for different SW development purpose. You should know strength and weakness of each for choosing model properly.

Software development models

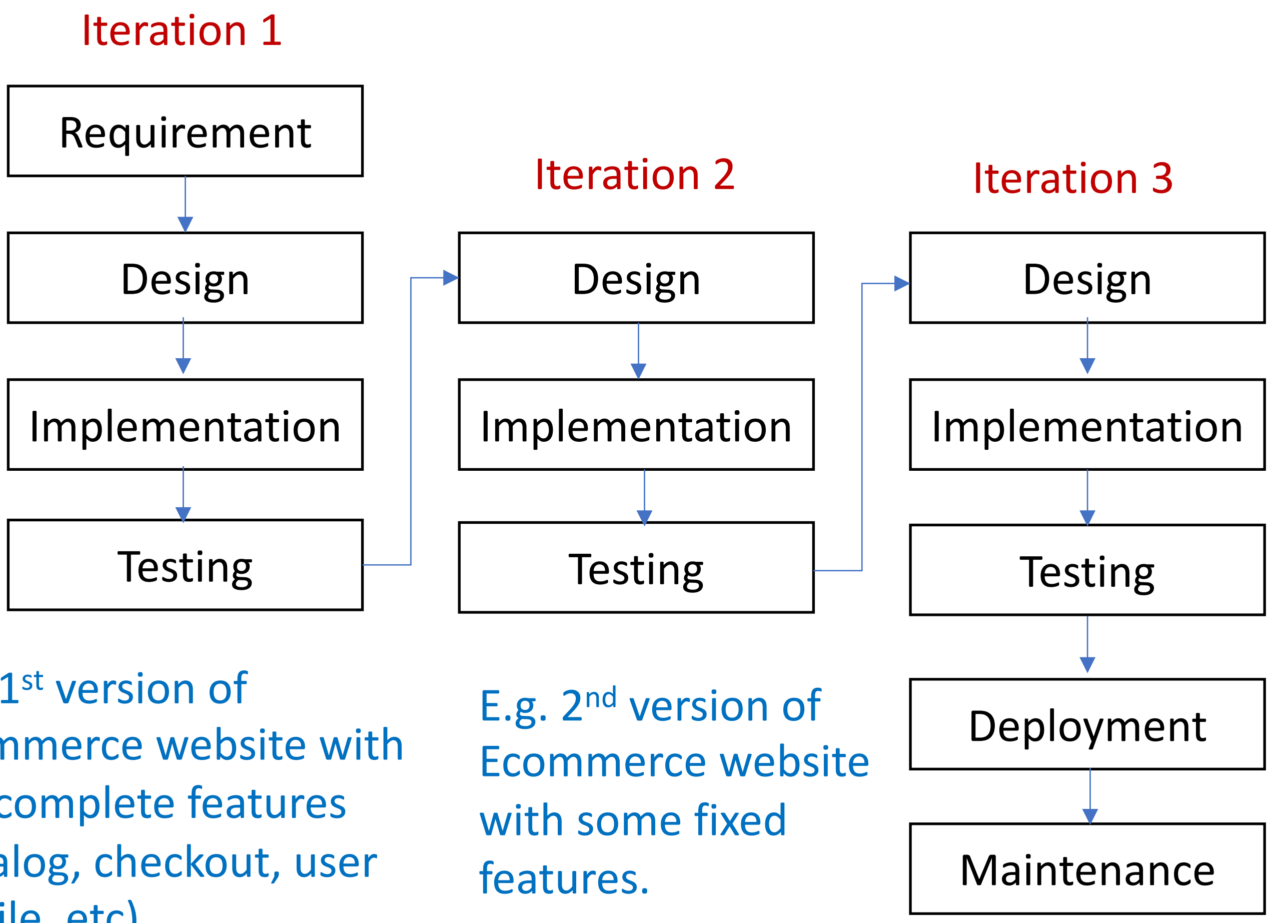
- There are two categories:
  - Predictive development model
    - E.g. Waterfall model, Incremental waterfall model and V model
    - Most requirements, processes, schedule and technologies are anticipated
    - Work well for short projects where you know everything in advance
    - Work poorly if requirements are uncertain or change during the project
  - Adaptive development model (Iterative model)
    - E.g. Prototypes, Spiral and Agile
    - Give opportunities to change direction by regularly reviewing a progress of project
    - Work well with changing or uncertain requirements

Iterative = doing something again and again, usually to improve it

Iterative model

- An approach that repeats a series of development efforts again and again in order to improve a working prototype
    - Iteration 1: Initially provides some (not all) of the software specifications and develop the first version of the software (simple prototype).
    - Iteration 2: [After the first version if there is a need to change the software] Repeat the development to get a new version of application.
    - Later iterations: Continue the changed until reach the development of accurate application
- Pro: Suitable of large project, major requirements must be defined; however, some details can evolve with time, Support user feedback
- Con: Costly system architecture or design issues

Iterative model with an example



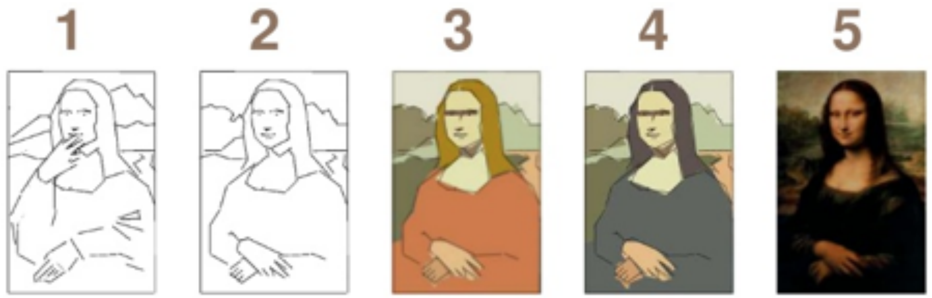
E.g. 1st version of Ecommerce website with not complete features (catalog, checkout, user profile, etc)

E.g. 2nd version of Ecommerce website with some fixed features.

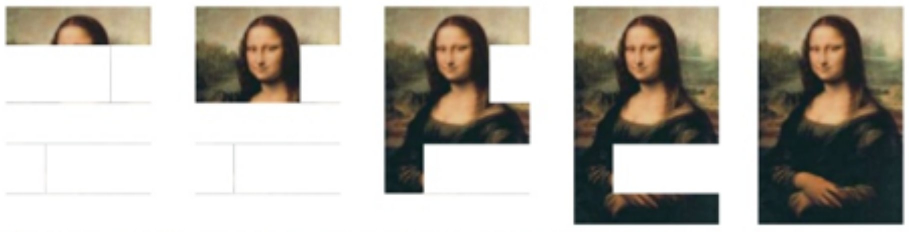
E.g. 3rd version of Ecommerce website with all fixed features.

Incremental vs iterative

- Iterative
  - Develop through repeated cycles
  - Start simple, expecting to change
  - Used to improve the candidate solution



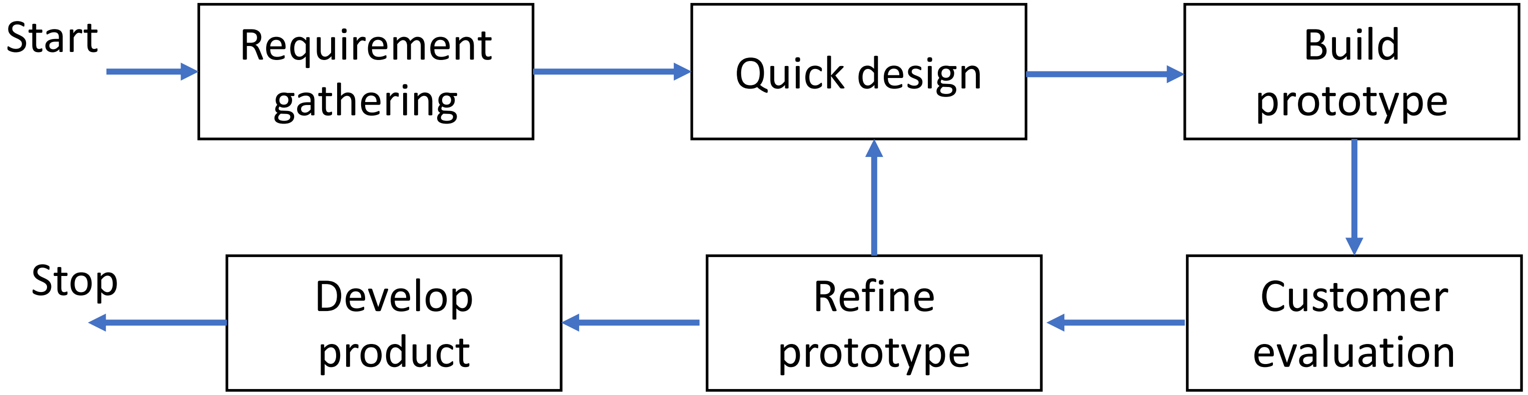
- Incremental
  - Develop smaller portions at a time
  - Gradually build up functionality
  - Allow value to be delivered early



Give me some real-world examples to compare iterative and incremental approach

Prototyping

- A prototype is a simplified model that lets you study the behavior of some part of an application.
- A software prototype is a program that mimics part of the application you want to build.



Start

Requirement gathering

Quick design

Build prototype

Customer evaluation

Refine prototype

Develop product

Stop

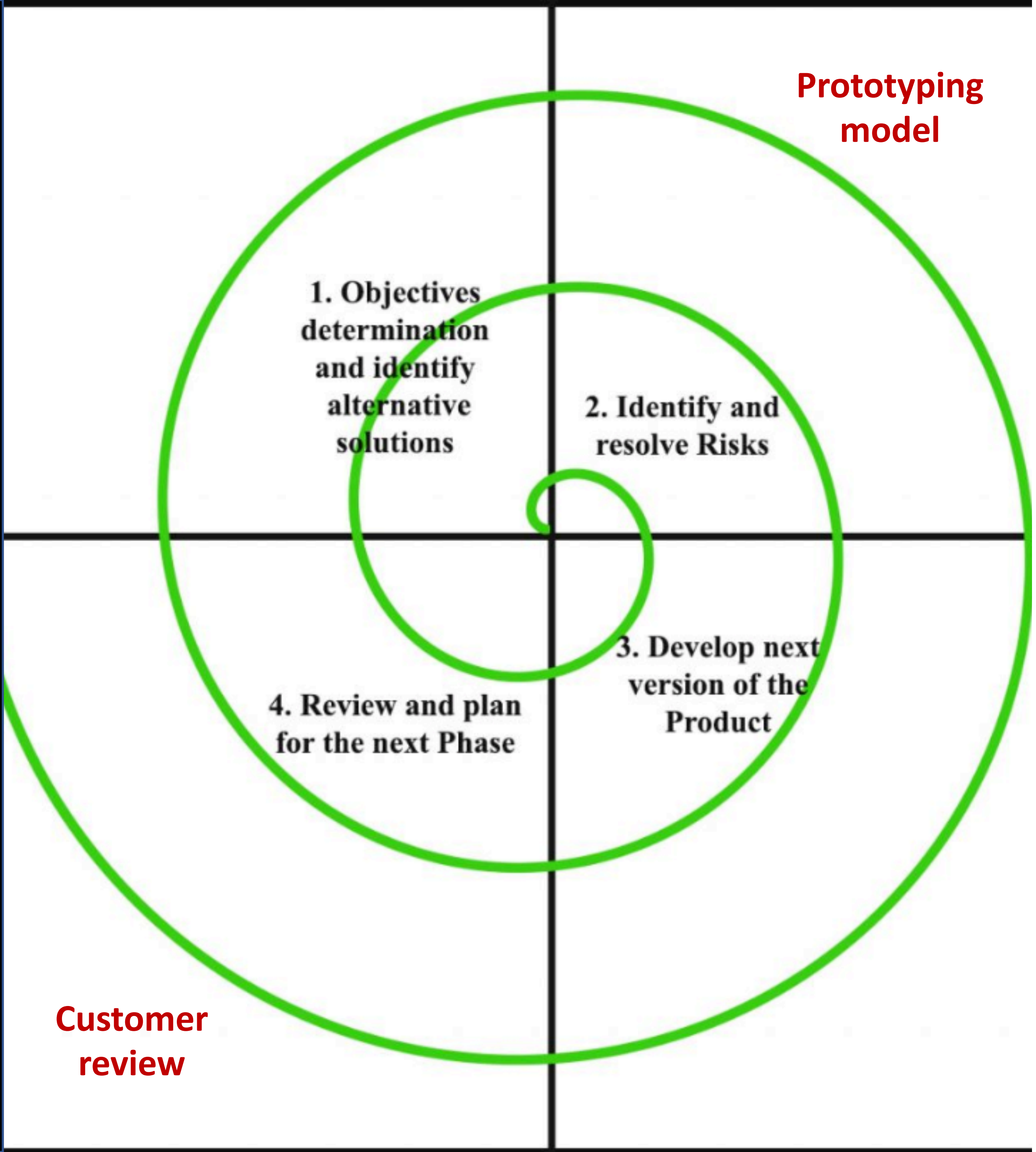
Type of prototypes

- Throwaway prototypes
  - The prototype to study some aspect of the system and then you throw it away and write code from scratch
- Evolutionary prototypes
  - The prototype demonstrates some of the application’s features. As the project progresses, you refine those features and add new ones until the prototype transform into the finished application.

Spiral

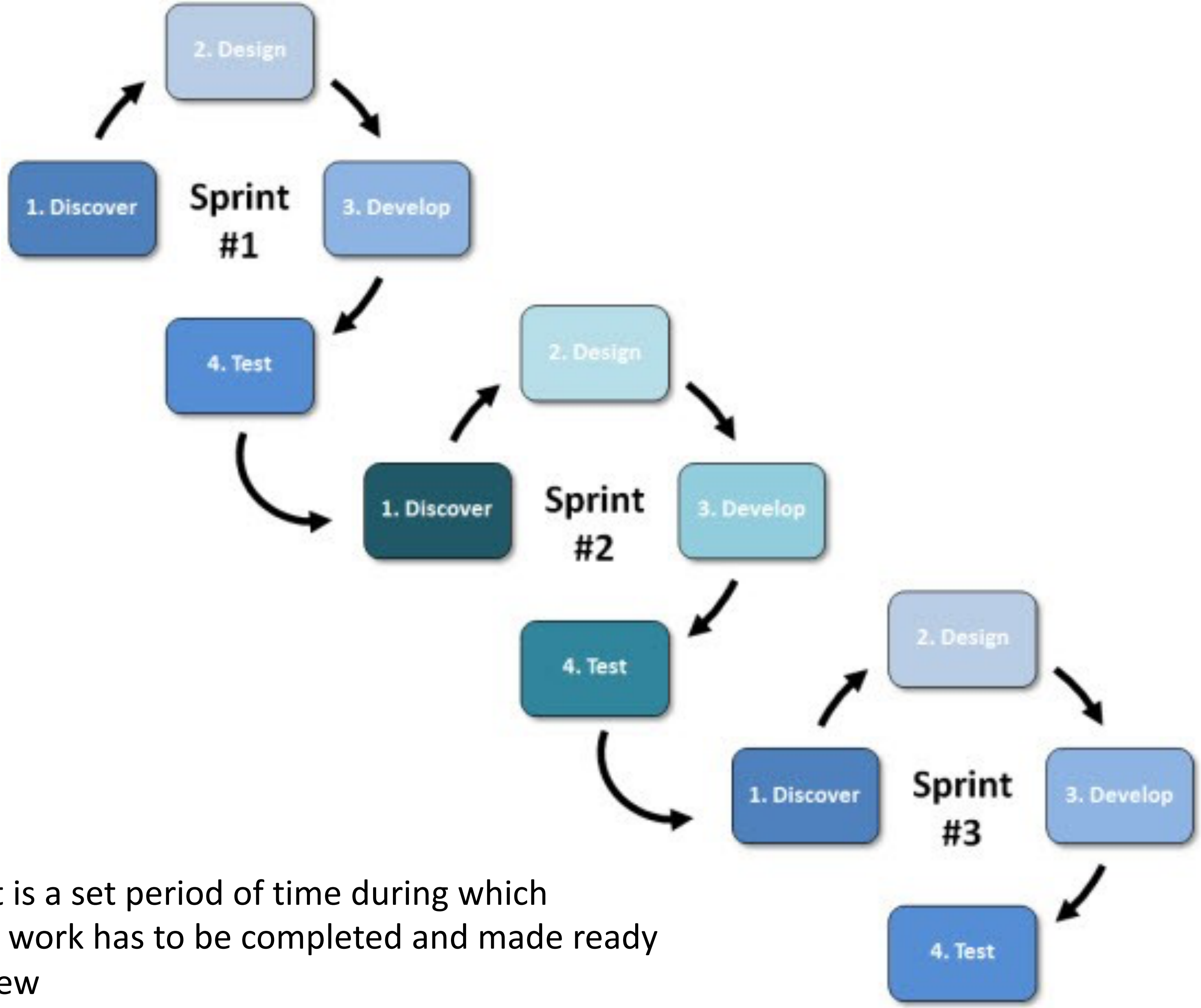
- The spiral model is a risk-driven approach that uses a sequence of repeating phases to identify project risks to reduce the chances of the project failing
- Each loop of the spiral is called a phase of the software development process (planning, risk analysis, engineering, and evaluation).
- The exact number of phases can be varied by the PM estimating the project risks.
- Pro: many points of reviewing and making decisions, focus on risk analysis, support changing very well
- Con: It’s complicated, risk analysis can be difficult, not suitable for small project

Spiral model


Prototyping model
1. Objectives determination and identify alternative solutions
2. Identify and resolve Risks
3. Develop next version of the Product
4. Review and plan for the next Phase
Customer review

Agile

- Agile model is a combination of both incremental and iterative model and promotes more involvement of customer during the software development.
- Pro: Rapid development, Allows changes easily, Strong communication of software team with the customer, Focus on user and customer
- Con: Mainly depends on the customer, Not suitable for large projects, Lack of documentation

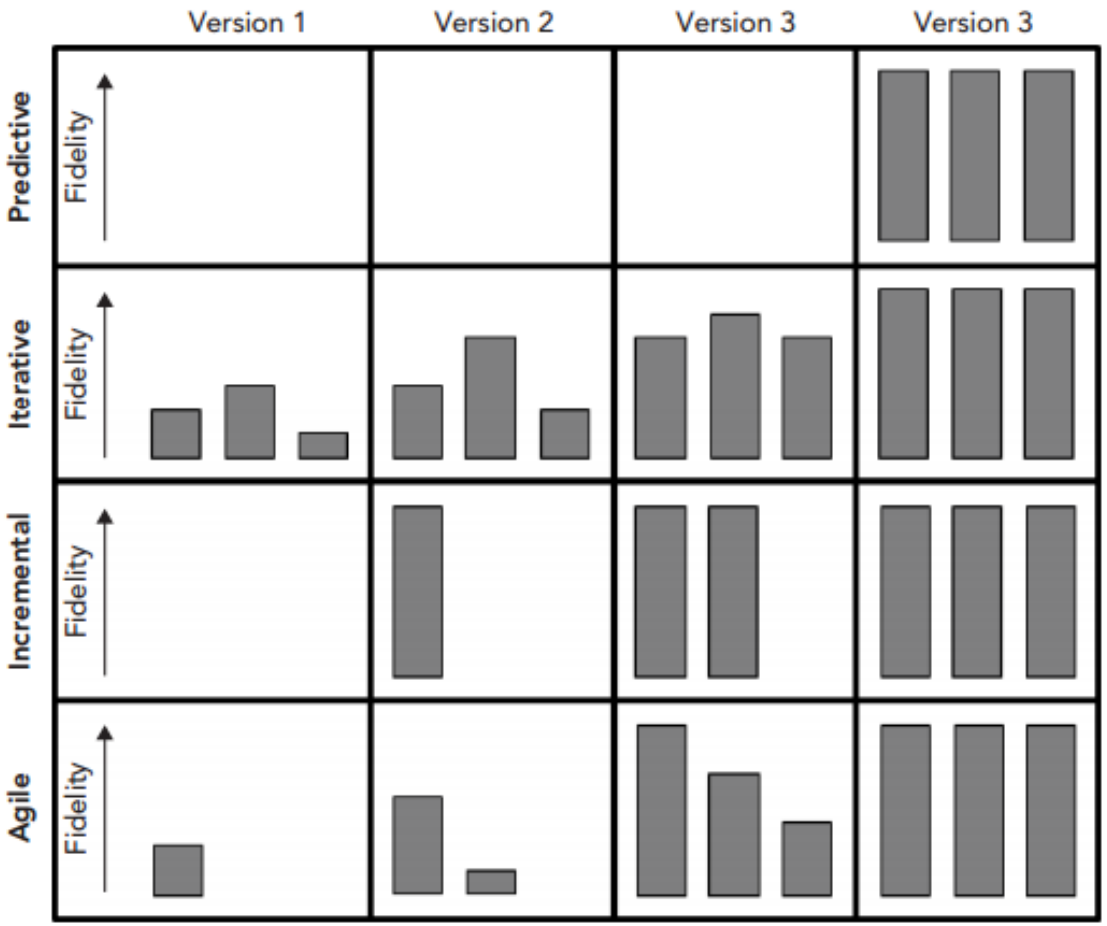


A sprint is a set period of time during which specific work has to be completed and made ready for review

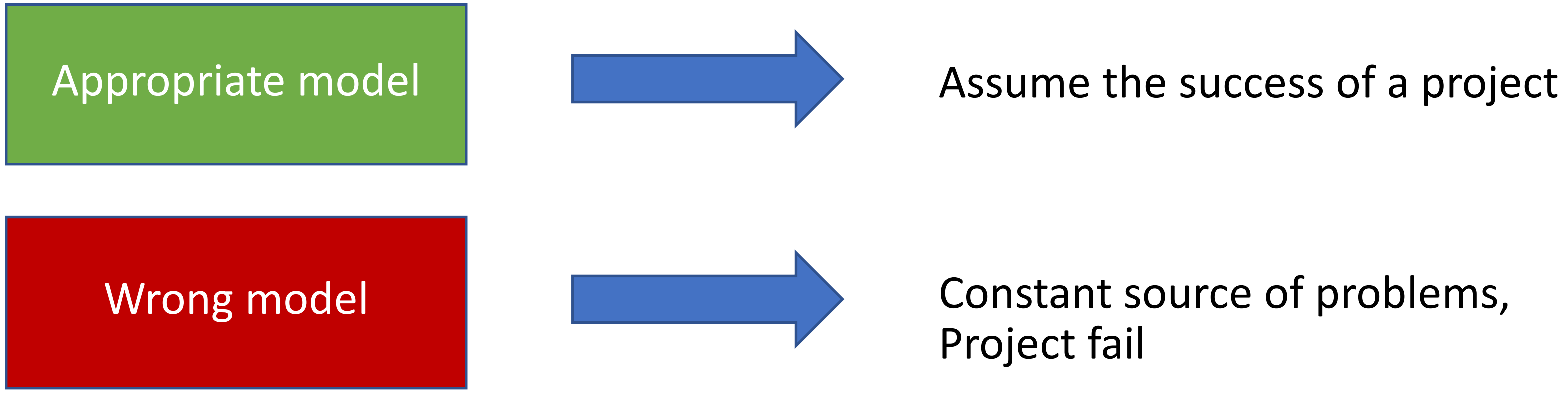
Different development approaches add features and increase fidelity in different ways

Fidelity = The completeness
of feature

- **Predictive approaches** make one big release when everything is done.
- **Iterative approaches** release every feature with low fidelity and then improve fidelity over time.
- **Incremental approaches** release features as they are finished with high fidelity.
- **Agile approaches** combine iterative and incremental approaches. Features are released as soon as they are useful. Over time, existing features are improved, and new features are added.



Choosing a model



Appropriate model → Assume the success of a project

Wrong model → Constant source of problems,
Project fail

- Guideline to choose right model
  - Aware of what level of understanding we have of requirement
  - Expect of lifetime of the project
  - Risk
  - Schedule constraints
  - Interaction with customers
  - Expertise

Model usage prerequisites

- Make sure everyone on the team is on board
  - Everyone needs to agree on what the rules are and what procedures you will use to make sure the rules are followed.