# Course 1

# Software Development Processes and Methodologies

## Week 1

## What software development looks like

1. Which of the following software development models can best respond to requirements changes?

The V-model

The Waterfall model

Agile models

2. In which of the following software development models are the software development activities performed sequentially rather than in iterations?

The Waterfall model

Agile models

3. Which of the following are limitations of the waterfall model?  Select three.

It is difficult to respond to requirements changes.

Integration issues may remain undetected until the last phase.

Misinterpretations of requirements or design can remain undetected until the later development phases.

It is not suitable for big projects.

## Why do we need requirements?

1. Which is the right description about the requirement specification process?

Requirement specification is a process that identifies and specifies the problem and possible solutions to the problem.

Requirement specification is a process to design the solution to the problem.

Requirement specification is a process to identify and specify the problem to solve.

Requirement specification is a process of solving the requirements.

2. Specifying requirements is difficult because (select three):

There is no guideline for writing a software requirement specification document.

Software is intangible, which makes it difficult to comprehend and communicate.

Terminology can be interpreted in multiple ways depending on the person or the context in which it was used.

The client or end users might not be clear about what they want.

3.Why is requirement specification important?  Select two.

Spending time upfront in requirement specification can save time in the later phases of the software development lifecycle.

Repairing a bug in the requirements can cost thousands of times less than in the later phases of the software development lifecycle.

Changing the requirement is not possible once the system is fully developed.

# Requirements vs Specification

1. Which of the following are true about requirements and specifications? (Choose all that apply)

Requirements are for users; specifications are for developers.

Specifications shall meet the requirements.

Requirements call for specification-level refinement.

Requirements shall be written in the user's language; system specifications shall be written in the system language.

2. The following are the requirements and specifications of an online banking service. Which of the following can be categorized as (user) requirements?  Select two.

At the time a query is made to check the current balance of a checking account, the owner of the checking account shall be logged in.

Activate a login session when a user logs in, and maintain the session for 60 minutes unless the user who logged in had remained inactive for more than 120 seconds.

The user shall be able to check the current balance of the checking accounts that he/she own.

A user shall be able to use the online banking system securely.

3.The following are requirements and specifications of an online banking service. Which of the following can be categorized as system specifications? Select two.

At the time a query is made to check the current balance of a checking account, the owner of the checking account shall be logged in.

Activate a login session when a user logs in, and maintain the session for 60 minutes unless the user who logged in has been inactive for more than 120 seconds.

The user shall be able to check the current balance of the checking accounts that he/she own.

A user shall be able to use the online banking system securely.

## Non-functional Requirements

1. A constraint to only use Microsoft Project during the system development is a non-functional requirement.

True.

False.

2. Which of the following are non-functional requirements:

Some product requirements, like using a specific encryption protocol, are non-functional requirements.

Organization requirements imposed by the company, like a specific coding style, are non-functional requirements.

External requirements imposed by external organization, like using a specific development style, are non-functional requirements.

All of the above.

## WRSPM Quiz

1. WRSPM stands for:

Wide, Requirement, Software, Planning and Model.

World, Requirement, Specification, Program and Machine.

None of the above.

2. Looking at the difference between user requirements and system specifications in the ATM example, we know that swiping the card and prompting for a PIN are requirements, while reading the card details and a 4-digit PIN are specifications.

True.

False.

3. The purpose of the WRSPM model is to ensure that:

Specifications meet the requirements.

Requirements meet specifications.

None of the above.

# Software Architecture: Definition

1. A subsystem in an architecture must:

be created separately and can operate individually.

have business value.

be integrated with one another or with existing subsystems.

All of the above.

2.Partitioning of a large system into smaller subsystems helps the buy-or-build decision because we can examine each subsystem and reason about possible buy-or-build options for each.

True.

False.

3. A good software architecture is important because:

It helps organize the workforce and resources.

It allows for parallelization in development.

It helps build-or-buy decisions.

It helps with funding decisions.

# Software Architecture Models

1.

A UNIX program where the output of one program is the input of another, is an example of which of the software architecture models below:

Layered Model.

Pipe-and-Filter Model.

Blackboard Model.

Event-based Model.

Client-Server Model.

2. An online banking system is best modeled by:

Pipe-and-Filter Model.

Blackboard Model.

Client-Server Model.

Event-based Model.

3. A vehicle identification and tracking system, where each moving vehicle is tracked and monitored through a shared program, is best modeled by:

Pipe-and-Filter Model.

Blackboard Model.

Layer Model

Client-Server Model

Event-based Model

# Software Architecture: Process

1. The difference between subsystems and modules are:

Subsystems can independently comprise the business logic by itself while modules can't.

Subsystems can communicate with other subsystems while modules cannot communicate with other modules.

All of the above.

2. Software quality attributes that we care about during software architecture are:

Performance.

Reliability.

Testability.

Security.

Usability.

3. Software architecture concerns itself with both estimation and quality but not partitioning.

True.

False.

# Requirements and Architecture

1. Which of the following is considered the most difficult in the software development process?

Developing a Java program to meet the designed solution

Testing the solution with users

Accurately identifying user needs

Determining the accurate market price for the finished product

2. What is a Requirements Specification?

A process, how the requirements are written (specified)

A product, a written specification of the requirements

Both

Neither

3. Based on the NASA statistics on budget and schedule overrun vs. time spent on requirements process, what is the recommended amount of time to spend on the requirements stage?

0% of the total time spent on the project

5-10% of the total time spent on the project

20% of the total time spent on the project

The same amount as you expect to spend on testing.

4. Which of the following matches the terms to the correct definition?

Requirements - user needs in user language; Specification - solution properties designed to solve problem

Requirements - solution properties designed to solve problem; Specification - user needs in user language

5. Which of these is a user requirement?

Retry posting the message up to 3 times every 5 minutes if the post is rejected by the server

Allow the user to post a message to Facebook

Allow an authorized user to post a message of no more than 136 characters to the Facebook POST API

Use the MariaDB database for internal data persistance

6. Which of the following is NOT an example of a non-functional requirement.

The developing team shall use the spiral model of software development lifecycle.

Communication between the client and server will be expressed in EBCDIC endoding.

The messages between the client and server shall be in YAML 1.2 format.

The user shall be able to select the course number.

7. What element(s) of the WRSPM model belong in the environment (including the overlap between environment and system)?

W

W,R

W,R,S

R,S

R,S,P

S

S,P

S,P,M

P,M

M

8. Which of these is proper definition of software architecture?

integrating small systems with no individual business value into larger ones

partitioning large systems into smaller ones that can be created seperately, have individual business value, and can be easily integrated

planning and pricing the resources involved with developing a large-scale software system, including presenting such plans and budgets to senior level executives for funding

all of the above

9. Which of the following models is best suited for a system with significant shared data that needs to be shared across a variety of components or sub-systems, somewhat like global variables, but with better data integrity?

event-based

blackboard

pipe-and-filter

client-server

layered

10. Which of the following models is best suited for a system that includes several subsets of functionality that are used in more than one area of the system?

blackboard

pipe-and-filter

event-based

layered

client-server

11. Which of the following models is best suited for a system that benefits from the separation of complexity and processing between work which can be done locally and that which should be completed on a shared, remote service?

client-server

layered

event-based

blackboard

pipe-and-filter

12. Which of the following would be considered a sub-system, rather than a module?

A set of classes which generates reports

A class which represents the paying customer

A class which provides a service (e.g. AccountCreationService)

A set of classes which convert various values based on environment or parameter information

# Week 2

## Software Design: Introduction

1. Software design is the process of transforming the stated problem into a ready-to-use implementation.

False

True

2. Abstract solutions do not require extensive domain knowledge and effectively reduce the costs during the software design phase.

False

True

3. It is often advised that abstract solutions do not provide optimization details regarding the implementation.

True

False

4. When it comes to software design, it is always best to follow a solution that is widely popular in the industry

False

True

5. While a solution coming from software design does not include implementation details, there are still common cases where pseudocode may be provided to correctly capture the sense of a complex algorithm.

True

False

# Software Design: Modularity

1. Check all that apply: The four aspects of modularity are...

Information Hiding

Sort Functions

Data Hiding

Components

Coupling

Cohesion

Data Encapsulation

2. Which of the four aspects of modularity is defined as: How well modules work together.

Information Hiding

Data Encapsulation

Cohesion

Coupling

3. Which of the four aspects of modularity can be described as: Abstracting away implementation details.

Data Encapsulation

Cohesion

Coupling

Information Hiding

4. Which of the four aspects of modularity can be described as: How well a module meets a single well-defined goal.

Data Encapsulation

Coupling

Information Hiding

Cohesion

5. Which of the four aspects of modularity can be described as: Containment of constructs and concepts within a module.

Coupling

Information Hiding

Data Encapsulation

Cohesion

6. Three goals of ______ can be described as (1) Decomposability, (2) Composability, and (3) Ease of Understanding.

Coupling

Cohesion

Modularity

7. You have a sort function that provides no details on which sorting algorithm is used. This is an example of which aspect of modularity?

Cohesion

Data Encapsulation

Information Hiding

Coupling

8. A benefit to using ______ is that you know if your data is corrupted, then it must have been corrupted by the module.

Information Hiding

Cohesion

Coupling

Data Encapsulation

# Software Design: Coupling

1.Choose the most accurate answer: Low coupling aids in...

abstracting away complex information.

ensuring that changes don't cross boundaries of modules.

lower corruption rate of data.

decomposability.

2.Which of the following statements about coupling are true?

Coupling measures the strength of connections between components.

Loose coupling makes it unlikely that changes will be propogated across components.

Shared variables and control information leads to tight coupling.

Loose coupling is partly achieved through message passing.

All of the above

3.The goal in low coupling is to ensure that changes don't cross the boundaries of modules.

True

False

4.The three types of tight coupling are:

Data; Message; None

Content; Common; External

Control; Data Structure; Message

There is no such thing as tight coupling

5.Module A relies directly on local data of module B. This is an example of what type of coupling?

Tight external coupling

Tight content coupling

Tight common coupling

6.Modules A and B both rely on global data or a global variable. This is an example of what type of coupling?

Tight external coupling

Tight common coupling

Tight content coupling

7.Modules rely on externally imposed format (or protocol or interface). This is an example of what type of coupling?

Tight common coupling

Tight content coupling

Tight external coupling

8.The two types of medium coupling are:

Content; Common

Message; None

Control; Data Structure

Data; Message

9.Module A controls the logical flow of module B. This is an example of what type of coupling?

Medium control coupling

Medium data structure coupling

10.Module A and B both rely on the same composite data structure. This is an example of what type of coupling?

Medium data structure coupling

Medium control coupling

11.The three types of loose coupling are:

Data; Message; None

Control; Data Structure; Loose

Data; Message; Data Structure

Content; Common; External

12.Modules only share parameters. This is an example of what type of coupling?

Loose no coupling

Loose message coupling

Loose data coupling

13.The loosest type of coupling; components only communicate through parameters or message passing. This is an example of what type of coupling?

Loose message coupling

Loose no coupling

Loose data coupling

# Software Design: Cohesion

1. Choose the most accurate answer: Cohesion describes...

how well everything within a module fits together.

loose coupling.

how changes do not cross boundaries of modules.

decomposability.

2.Which of the following statements about cohesion is true?

Cohesion measures how well everything within a module fits together.

Describes the implementation of a single logical entity or function.

Represents a desirable design attitude.

High cohesion is the aim in software development.

All of the above

3.The goal in real-life software development is to aim for object and functional cohesion and anything less than this should be justified.

True

False

4.The four types of weak cohesion are:

There is no such thing as loose cohesion

Coincidental, temporal, sequential, logical association

Communicational, sequential, object, functional

Coincidental, temporal, procedural, logical association

5.Parts of modules are together in the same file. This is an example of what type of cohesion?

Procedural cohesion

Logical association cohesion

Coincidental cohesion

Temporal cohesion

6.Different pieces of code are activated at the same time. This is an example of what type of cohesion?

Temporal cohesion

Coincidental cohesion

Logical association cohesion

Procedural cohesion

7.One piece of code runs after the other. This is an example of what type of cohesion?

Logical association cohesion

Coincidental cohesion

Temporal cohesion

Procedural cohesion

8.Components do similar but separate things. This is an example of what type of cohesion?

Procedural cohesion

Temporal cohesion

Coincidental cohesion

Logical association cohesion

9.The two types of medium cohesion are:

Sequential, logical association

There is no such thing as medium cohesion

Coincidental, temporal

Communicational, sequential

10.All elements of a component operate on the same input or produce the same output. This is an example of what type of cohesion?

Sequential Cohesion

Communicational Cohesion

11.One component's output provides the input to another component. This is an example of what type of cohesion?

Sequential Cohesion

Communicational Cohesion

12.The two types of strong cohesion are:

Object, strong

Coincidental, temporal

Communicational, sequential

Object, functional

13.Each operation in a module is provided to allow object attributes to be modified or inspected. This is an example of what type of cohesion?

Functional Cohesion

Object Cohesion

14.Every part of a component is necessary for a single well-defined behavior. This is an example of what type of cohesion?

Object Cohesion

Functional Cohesion

15.True or False: inheritance strengthens cohesion.

False

True

# Implementation

TOTAL POINTS 5

1. Why should you program when you are alert?

Sleep deprivation leads to mistakes.

Output falls drastically after working 50+ hours per week.

All of the above

2.Which of the following are examples of 'Write code for people not computers'?  Select three.

Do not code while sleep deprived.

Let your code (variable names, function names, etc.) explain HOW it does what it does.

In your comments, explain WHY the code does what it does.

Choose informative variable names.

3.Fill in the blank. Write ____, _____, and _____ before writing functional code.

System Specifications; User Guide; Exception Handling.

Comments; Benchmarks; Tips for Use.

Comments; Tests; Exception Handling.

4.If you write the same code more than once...

Put that code into a method.

It's okay.

You did something wrong.

5.True or False: Every method can be written without side effects.

False.

True.

# Deployment

TOTAL POINTS 3

1.Deployment occurs in the end stage of active development.

False

True

2.Product deployment should encapsulate three main focus areas.  What are they?

Problem areas

Planned steps

Description of the physical environment

Plan for recovery in case of failure

3.Concerns to address in your deployment plan include:

Physical environment requirements

Hardware requirements

Documentation

3rd party software requirements

Training

Software being deployed

Database-related activities

All of the above

# Deployment: Rollback

TOTAL POINTS 3

1.A rollback is the reversal of actions completed during deployment with the intent to never revert a system back to its previous working state.

False

True

2.What are some reasons why you would need a rollback plan?  Select two.

Installation doesn't go as expected.

It is the end goal in product deployment.

Fixing the problem ends up taking longer than the window allows.

Database management in complex systems.

3.Deployment without a rollback plan should occasionally occur. Especially when the installation involves deleting files.

False

True

# Deployment: Cutover Strategies

TOTAL POINTS 5

1.The three cutover strategies are...

Cold backup

Cold failover

Hot failover

Hot standby

Warm failover

Warm standby

2.If you need to have a cutover strategy where the replacement machine is ready for a full setup and install, then you would choose which of the following cutover strategies?

Hot failover

Warm standby

Warm failover

Cold backup

3.If you need to have a cutover strategy where the replacement services are ready for the redirection of data with no installation, then you would choose which of the following cutover strategies?

Warm standby

Hot failover

Cold backup

Hot backup

4.If you need to have a cutover strategy where the replacement machine is running and ready for installation, then you would choose which of the following cutover strategies?

Cold backup

Hot failover

Cold failover

Warm standby

5.There is no difference between load balancing and hot failover.

true

false

# Software Testing: Introduction

TOTAL POINTS 7

1.Test data represents possible inputs that are provided to the _____ for the purposes of testing.

system

oracle

both the system and the oracle

2.The oracle uses information provided from test cases to decide the correctness of the software under test

True

False

3.Test cases are composed of _______

test data

system (actual) outputs

expected outputs

4.Manual oracles are sufficient and effective for the majority of real-world applications

True

False

5.Choose the correct True-False pair: An error is the manifestation of a failure. A failure is the manifestation of an error.

False, False

True, True

True, False

False, True

6.It is possible for errors to remain undiscovered as long as the erroneous _________ is never used

component

test data

test case

output

7.Software testing can be used to prove the absense of failures in a module, but not the absense of faults

True

False

# Design, Implementation, and Testing

TOTAL POINTS 13

1.Where does software design fit in the traditional waterfall software development lifecycle?

Between architecture and implementation

Between implementation and deployment

Before requirements

Between specification and architecture

2.Which of the following is NOT an aspect of software design

Polymorphism

Cohesion

Coupling

Modularity

3.The ability to use a built-in function of a programming language to generate a random number is an example of which of the following?

Coupling

Cohesion

Modularity

Information hiding

4.Analyzing the extent to which other modules must change when a module is modified is an example of which of the following?

Coupling

Modularity

Cohesion

Information hiding

5.When all of the responsibilities of a module are easily classified as being strongly related, this is an example of high what?

Cohesion

Coupling

Information hiding

Modularity

6.Which of the following is most preferred?

Having two modules rely on the same global information structure (common coupling)

Allowing one module to affect the program flow of another via the use of a state message (message coupling)

Having more than one module rely on the same version of the RSS standard (external coupling)

Allowing one module to affect the program flow of another via the use of a flag (control coupling)

7.Which of the following is LEAST desirable?

including functionality which occurs around the same time

including functionality which relys on the same input

including functionality which modifies an object's own attributes

including functionality which entirely encapsulates all the necessary function for a specific task

8.Which of the following are necessary before proper testing?

inputs which cause issues

inputs, expected output, an oracle, and the actual output

inputs and expected output

inputs, expected output, and an oracle

9.Which of the following is an mistake in code which is found in code review?

Latent error

Effective error

Fault

Failure

10.Which of the following is an example of validation?

ensuring the system locks out an account after three failed log-in attempts

software recognizes incorrect inputs

software is well-received by the user

use of the software provides the correct results as documented

11.In your current project, you have access to some intern development resources, which are not currently operating at full capacity. You also know that the testing timeline will be truncated, due to delays in critical-path module development. Which strategy should you employ?

Make no changes to the current project testing or development allocations, utilizing intern resources to create documentation.

Utilize the intern resources to design and develop drivers and stubs, while work continues on critical-path module development

Allow the testing team to work without (or with quick-to-develop) drivers and stubs, while using intern resources to aid critical-path development

Allow the testing team to work without (or with quick-to-develop) drivers and stubs, and utilize the intern resources to aid testing once all critical-path development is complete

12.When is testing complete?

When you run out of time

When you have completed all the tests in the test plan

When you find the last bug

When you have tested every input

13.Which of the following is an example of black-box testing?

Developing test cases to exercise error-prone constructs

Developing additional tests when every if statement does not evaluate both true and false once given the current test suite

Developing additional test cases to force a division by zero

Developing test cases based on typical user behavior

# Week 3

## Waterfall methods

TOTAL POINTS 3

1.In waterfall method, you get your product in one big bang deployment

True

False

2.Sashimi model may help decrease the time duration of the project by

Overlapping the phases

Adding more skilled resources of the project

3.Which of the following are true for the V-model?  Select two.

It is a predictive model.

Testing-related activities are started earlier in the process

Requirement changes are welcomed in all phases of this project.

# Waterfall and Incremental Software Development Models

TOTAL POINTS 7

1.In predictive models, you are not sure of user needs so requirements are expected to change during development

False

True

2.In which model might you end up building something different than what you originally planned?

Adaptive models

V-Model

Waterfall models

3.What are the BENEFITS of the Sashimi Model?  Select two.

It can help shorten development time

It supports early validation

It can support creating a learning prototype early

4.Which of the following is true for the V-model?  Select two.

In this model, you build software in increments

It is a predictive model

The V-model is useful in cases where there is ambiguity in requirements and early validation would be useful.

This models emphasizes risk analysis and resolution

5.Which of the following is true for Incremental Models?  Select three.

Incremental models may result in rework

You can overlap building of one increment with another

If deploying an increment to actual users can benefit the organization, using an incremental model is a potential candidate to consider.

You always have to use the same model for each of the increments

Incremental models are always predictive models

6.Which of the following is true for the Waterfall model, V-model and Sashimi model?  Select two.

In these models, the cost of change depends on how late we find out about the change. If a problem requiring a change is found during the implementation phase, it will be more expensive to fix than one found during the design phase

In these models, the team is expected to know the solution really well.

In these models, we assume that we know requirements really well

7.Which model is the best model to use in all situations?

None

Waterfall

Adaptive since it has most advantages

# Unified and Spiral Processes

## Software Development Lifecycle

TOTAL POINTS 5

1.In the Unified Process, all requirements work is done upfront and no requirements work is done in the construction phase.

True

False

2.What are some of the activities that happen in the elaboration phase of the Unified Process? Select two.

Decide if you are going to build the system or buy an off-the-shelf product

Address known risks

Deploy an increment of the software

Validate system architecture by building executable architecture baseline

3.Which of the following activities happen in Step 2 of the Spiral Model?  Select two.

Decide objectives and constraints

Make a decision whether to continue with the next cycle

Identify risks

Resolve risks

4.Select the four basic steps followed in each cycle of the Spiral Model.

Design

Requirements

Plan the Next Iteration

Development and Tests

Identify and Resolve Risks

Define Objective

Deployment

5.Which of the following are true for the Unified Processes and its variants?  Select two.

Agile Unified Process and Open Unified Process are lighter versions of the "Unified Process".

Rational Unified Process is a lighter version of the Unified Process.

UML Unified Process is the Ultimate Unified Process with a superset of all practices and processes.

Enterprise Unified Process adds additional practices on top of the Unified Process and Rational Unified Process.

# Traditional Software Development Models

TOTAL POINTS 11

1.Which of the following is true for adaptive models? (Select any two)

You may end-up building something different than what you originally thought

You discourage change during the development process

You are not sure about what exactly you want to build

Adaptive model is the best model in all situations

2.A team divided the application they were building into 4 parts and planning to build one part at a time. To build each part, they are planning to go through the steps of requirements, design, implementation, test, and deployment. Also, they are planning to complete each part fully with high fidelity. By fidelity we mean the completeness of the features implemented in each part. Based on this information, what type of model do you suggest they use for their software development process?

Iterative and Incremental

Iterative

Incremental

3.A team is working on a problem where the requirements are well known but they are short on time. They prefer to start the design phase when some of the requirements have been defined. Similarly, they want to start coding when the design for part of the system is ready. What model will best fit their needs?

Sashimi

V-model

Waterfall

Spiral

4.In the incremental model variation where you do requirements and design for all of the application once and then just increment on implementation, testing, and deployment, are you making the process more suitable for a predictive or adaptive situation?

Predictive

Adaptive

5.Which of following is NOT true about incremental and iterative models?

Iterative approaches release every feature with low fidelity and then improve fidelity over time. By fidelity we mean the completeness of the feature.

Incremental approaches release features as they are finished with high fidelity.

An incremental model is one where you can build each part of the system individually but you release all at once.

In a model which is both iterative and incremental, we release features when they are usable. Over time the development team improves existing features and adds new ones

6.Which of the following is NOT true for the Waterfall model, V-model and Sashimi model?

In these models, we assume that we know requirements really well.

In these models, the team is expected to know the solution really well.

These models will be a good model to be used by a startup implement a new idea that is new to market

In these models, the cost of change depends on how late we find out about the change. If problem requiring a change is found during the implementation phase, it will be more expensive to fix than one found during the design phase.

7.n the Unified Process, what does the milestone of "Lifecycle Architecture" mean?

Agreement among stakeholders about the objectives of the project and to proceed with the project.

Agreement among the stakeholders about the technical approach.

Agreement among the stakeholders about the deployment of the system to actual users.

8.Which of the following is true for the Unified Process? (select any 3)

It is a framework and not a specific model.

It is a predictive framework.

It is a very simple framework to follow.

It is an architecture-centric framework.

The construction phase supports both iterative and incremental approaches.

9.Select situations listed below where you would NOT use the Unified Process framework.

Medium-sized project with some risks involved.

Very small, simple project where the problem and solution are well known.

Situation where the organization wants to deliver part of the application early to capture the market ahead of competition.

All requirements are not known early in the project.

10.Which of the following is NOT true for Spiral Model?

You always have to perform all the steps mentioned in the Spiral Model diagram

It is suitable for very large high risk projects

Spiral Model is not a model but a process model generator.

It minimizes waste by allowing teams to tailor the effort put into different processes based on the risk involved in each cycle.

11.Which of the following is NOT the benefits of the Unified Process?

Easy to understand and implement.

Flexible for handling change.

Supports quality and reuse by putting an emphasis on architecture.

Supports / encourages risk identification and mitigation.

Supports to incorporate other models in the process for example you can use Agile in the construction phase.

# Week 4

# Agile Mindset

TOTAL POINTS 8

1.What are some of the challenges with Waterfall methods that prompted the software industry to come up with alternatives like Agile? Select three.

Wrong implementation goes undetected for a long time.

Integration issues between different components of the software go undetected for a long time. During the testing phase, when all the components are integrated, these issues are discovered but it is very late in the process.

Project teams were geographically distributed.

Predicting customer needs is difficult.

Projects cost too much.

2.What are the four Agile values according to the Agile Manifesto?

Responding to Change over Following a Plan

Planning Properly over Just Executing

People and Interaction over Processes and Tools.

Customer Collaboration over Contract Negotiation

Working Software over Comprehensive Documentation

3.Which of the following statements align with the value "Working software over comprehensive documentation” of the Agile manifesto?

Don't do documentation on an Agile project.

Create documentation when it provides value.

Must write documentation to fulfill company policy.

4.Which of the following is true about the Agile Manifesto? Select two.

The Agile Manifesto was primarily based on theoretical knowledge.

The Agile Manifesto specifies exact practices and processes that Agile teams are expected to follow

The Agile Manifesto consists of 4 values and 12 principles.

The Agile Manifesto is a work in progress and we are continuing to learn.

5.Why is it difficult to predict user needs and requirements? Select three.

Translation issues. Requirements are misinterpreted.

Sometimes the market shifts from the time when the requirements were originally defined.

Not enough time is spent on requirements.  If we spend more time early, we can define requirements very accurately.

Difficult to understand user needs.

# Agile Frameworks and models

TOTAL POINTS 7

1.Which of the following are official rituals/meetings/practices in Scrum?

Mid-Sprint Status Review Meeting

Sprint Planning Meeting

Sprint Review Meeting

Sprint Retrospective Meeting

Daily Scrum Meeting

2.Which of these are core properties of Kanban?

Retrospectives

Visualize the Work

Limit work in progress

Manage Flow

Daily stand-up

3.Which of the following are TRUE about Lean Startup? Select two.

In Lean Startup, you try to validate/invalidate your assumptions (crucial for the success of your product) as fast as you can.

Lean Startup concept is primarily applicable to startup companies.  It is not applicable for bigger organizations.

The concept of "Validated Learning" in Lean Startup means validating your learning and new ideas with your manager before you implement

Lean Startup concepts recommend to go through Build-Measure-Learn cycles as fast as you can.

4.Which of the following is true about Scrum?  Select two.

In daily stand-up, everybody talks about what they worked, what they are planning to work on and share any impediments they have.

It is encouraged to make iteration scope changes during the sprint

Burndown and Burnup chart help teams track the progress of the current sprint.

In Sprint Review, we talk about process and talk about what was working, what didn't work and how we can improve.

In Sprint Retrospective, we talk about what we build, give a demo and get customer/client feedback

5.Which of the following is true about product and sprint backlogs?  Select two.

A sprint backlog is created during the sprint planning meeting.

The product backlog is a prioritized backlog with highest priority items on the top.

Sprint and product backlogs are same thing.

A sprint backlog has all of the items contained in the product backlog.

# Agile & Lean Software Development

TOTAL POINTS 10

1.What are some of the challenges with the Waterfall methods that prompted the software industry to come up with alternatives like Agile? (select 2)

Projects were very complex and big.

Projects cost too much.

It is difficult to predict user/customer needs.

Incorrect interpretations of requirements by developers go undetected for a long time. During user acceptance tests, these issues are discovered but it is very late.

2.Which of the following statements aligns with the value "Working software over comprehensive documentation" of Agile manifesto?  (select any 2)

Agile coach says to the team “Documentation is of no use.”

A manager tells the team: “Writing certain documentation is company policy so we must create them irrespective of its use.

Value delivering software over writing documentation.

If documentation is absolutely essential then create it.

3.Which one of the following statements/situations/conversations align with an Agile mindset?

A manager e-mails her team: “Meeting face to face just for a 15 minute discussion is inefficient, let us all just dial-in from our desk.”

A software vendor says to a client: "We must define exactly what the user needs and put that in agreement."

A manager providing a status update to a customer says: “We are done with requirements and design so we are 50% done.”

*Manager: “The customer is suggesting another change in feature X which is complete as per specification. They agree that it was built as we agreed upon but it lacks some functionality and will result in user frustration. Should we implement the change or not?”*

*Developer: “Yes, if it does not fulfill user needs, we should make the change."*

4.Which of the following is TRUE about lean startup? (select all that apply)

Lean Startup concepts recommends to go through Build-Measure-Learn cycles as fast as you can.

In Lean Startup, you try to validate/invalidate your assumptions (crucial for the success of your product) as fast as you can.

The concept of "Validated Learning" in Lean Startup means validating your learning and new ideas with your manager before you implement.

Lean Startup concept is primarily applicable to startup companies. It is not applicable for bigger organizations.

5.Which of following is valid on a Kanban Board? (select any 2)

WIP limit for Step X is 3. Step X is divided into "Doing" and "Done." Number of items in "Doing" column is 2, and number of items in "Done" column is 1

WIP limit for Step X is 3. Step X is divided into "Doing" and "Done." Number of items in "Doing" column is 4, and number of items in "Done" column is 0

WIP limit for Step X is 5. Step X is divided into "Doing" and "Done." Number of items in "Doing" column is 3, and number of items in "Done" column is 1. One of the item in previous step is done and developer wants to move that item to Step X

WIP limit for Step X is 3. Step X is divided into "Doing" and "Done." Number of items in "Doing" column is 2, and number of items in "Done" column is 1. One of the items in the previous step is done and developer wants to move that item to Step X

WIP limit for Step X is 3. Step X is divided into "Doing" and "Done." Number of items in "Doing" column is 3, and number of items in "Done" column is 2

6.Which of the following Framework/process model supports Agile Mindset? (select all that apply)

DSDM

Lean Startup

Crystal

Waterfall

Kanban

V-Model

XP

Sashimi

Scrum

7.A software development team just started using Kanban. They had the following columns with WIP limit of 3 for each column on their Kanban board from left to right: Backlog(3), Analysis(3), Development(3), Testing(3), Deployment :

*There are currently 3 items in Testing (3 items being tested)*

*Similarly, there are 3 items in Development column (2 items are in development and 1 item is done being developed)*

*One of the developers does not have anything to work on so the manager wants to add one more item to the development column so the developer looking for work can work on it.*

Which one of the following would be a valid action in this situation? (Select any 3)

Add another tester in the Testing team so this bottleneck can be resolved quickly.

Manager should ask the developer waiting for work to help the testers so that they can finish one of the items from their column and move it to Deployment column.

The manager should ask the testers to move one of the items of the Test column temporarily to the Deployment column so that we can move items from the Development column. Later, we can move items back from the Deployment column to the Testing column.

It does not make sense for the developer to wait before one of the items moved to another column. The manager should add one more item to the development column and no other change needed.

Team should get together and discuss if we should increase the WIP limit of the development column as we have more developers and it will be smoother flow if WIP limit of the Development column is increased.

8.In Scrum, when is the sprint backlog created?

At the beginning of the project

During the sprint planning meeting

During the backlog refinement meeting

Whenever needed

9.In Scrum, how is Product Backlog Arranged?

Into categories, P1, P2, P3, etc.

Most important items at the top, least important items at the bottom.

Items are randomly arranged

Large items at the top, small items at the bottom

10.Which of the following is TRUE about Scrum? (Select all that apply)

Scrum is an adaptive framework/model

Scrum is the best way to implement the Agile mindset. Other Agile methods are not as effective.

Scrum has fixed time-boxed development cycles called sprints.

Daily Standup in Scrum is mainly for Scrum Master to get the status of the project.

# Course 2

# Agile Software Development

# Week 1

## Agile Values and Principles

TOTAL POINTS 4

1.What are some of the challenges with waterfall methods that prompted the software industry to come up with alternatives like agile?  Select three.

Integration issues between different components of the software go undetected for a long time. During the testing phase, when all the components are integrated, these issues are discovered but it is very late in the process.

Projects were very complex and big

Projects cost too much

It is difficult to predict user/customer needs

Incorrect interpretations of requirements by developers go undetected for a long time. During user acceptance tests, these issues are discovered but it is very late.

2.What are the four values of the Agile Manifesto?

Customer Collaboration Over Contract Negotiation

People and Interaction Over Processes and tools

Working Software over comprehensive documentation

Planning Properly Over Just Executing

Responding to Change Over Following a plan

3.Which Agile Value does the following principle align with:

*"Build projects around motivated individuals, give them the environment and support they need and trust them to get the job done."*

Individuals and Interaction Over Process and Tools

Respond to Change  Over Following a plan

Customer collaboration over contract negotiation

Working software over comprehensive documentation

4.Which of the following align with the agile principle:

*"Business people and developers work together daily throughout the project."*

Since agile methodologies do not recommend writing all detailed requirements upfront, it is really important to support greater collaboration between business people and developers so that they can build a shared understanding of what needs to be built.

Business people and developers MUST meet face-to-face daily to clarify confusions, questions etc.

Business people also need to code together with developers to make faster progress.

# Using Agile Methods

TOTAL POINTS 4

1.What are some of the benefits organizations have seen from using agile methods? Select five.

Project Visibility

Increased productivity

Increased project governance

Increased Team Morale

Improved Business / IT Alignment

Better Software Quality

2.The agile principle of "Deliver working software frequently" helps with which of the following challenges of predictive model? Select two.

It helps detect translation issues.

It helps prevent change

It helps validate user needs.

3.What new challenges does agile bring? Select two.

Very uncomfortable for leaders because of unpredictable journey

Architecture/Design/Database modeling is challenging

Agile eliminates all documentation hence makes it very difficult to use the software

# Applying an Agile Mindset

TOTAL POINTS 4

1.What are some of the tactics / concepts you can use to implement agile mindset?  Choose two.

Build in small bite-sized chunks.  These chunks could represent both iterative and incremental approach

Create lot of meetings to increase collaboration

Don't waste too much time in documenting, planning or designing. Focus your time on coding and implementation always.

To keep the cost of change down, use engineering practices like automated tests, continuous integration, incremental design

2.If a developer says "we don't to do any discussion or upfront design" because we are agile", is he/she truely representing what agile means

Yes,  because in agile, we start coding right away

No, agile does not mean start coding right away.

3.When applying agile, we don't need to do any release planning.. We only plan for a week.

True

False

# Agile Fundamentals

TOTAL POINTS 10

1.Which of the following statements align with the value "Working software over comprehensive documentation" of Agile manifesto?

Value delivering software over writing documentation.

A manager tells the team: “Writing certain documentation is company policy so we must create them irrespective of its use.

If documentation is absolutely essential then create it.

An agile coach says to the team “Documentation is of no use.”

2.Which of the following are true about Agile Manifesto? (select any 2)

Agile Manifesto was primarily based on theoretical knowledge

The Agile Manifesto is work in progress and we are continuing to learn.

The Agile Manifesto specifies exact practices and processes that agile teams are expected to follow

The Agile Manifesto consists of 4 values and 12 principles.

3.Which one of the following statements/situations/conversations align with an agile mindset?

A manager e-mails her team: “Meeting face to face just for a 15 minute discussion is inefficient, let us all just dial-in from our desk.”

Manager: *“The customer is suggesting another change in feature X which is complete according to the specification. They agree that it was built as we agreed upon but it lacks some functionality and will result in user frustration. Should we implement the change or not?”*

Developer: *“Yes, if it does not fulfill user needs, we should make the change."*

A manager providing a status update to a customer says: “We are done with requirements and design so we are 50% done.”

A software vendor says to a client: "We must define exactly what the user needs and put that in the agreement."

4.Why is it difficult to predict user needs and requirements? (select any 3)

Translation issues. Requirements are misinterpreted.

Difficult to understand user needs.

Sometime the market shifts from time when the requirements were originally defined.

There is not enough time spent on requirements. If we spend more time early, we can define requirements very accurately.

5.Which of the following statements align with following Agile Principle:

*"Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale"*

We MUST release software to actual users every couple weeks.

We should frequently deliver software in some environment (not necessarily to production always) to gather user/client feedback

We should NOT release software to a production environment on a shorter timeframe. We should ONLY release it in lower environment (e.g., a test environment).

6.Which of the following align with this principle:

*" Working software is the primary measure of progress."*

If a team has written the requirements and done the design for 5 out of 10 stories, then the team has made some progress (they are more than 0% done)

If a team has delivered 5 features out of 10 features (or stories), team has made progress (they are more than 0% done)

On a 10 month project, the team has been working for 5 months. Thus, the project is 50% complete.

7.In which of the following situations, would you NOT recommend using an agile process?

For a project where the customer has an idea of what they want to build but is not sure of his/her exact needs.

For a project where the customer wants to build an e-commerce website but the development team has never done something like this before. Further, the company knows that they want to build an e-commerce website but are not sure about the functionality at this time.

For a project where change is not expected. requirements are well known, and the solution is both well known and repeatable.

For a project where requirements are well known but the technology and solution are unknown (i.e., changes expected in the solution).

8.Which of the following is true for agile projects?

An agile methodology may help a team to go to market early by delivering with limited functionality.

Agile methods do not require rework because the team knows exactly what they building. No refactoring of code is needed.

Scrum is the best way to implement agile.

If somebody says, they use an agile methodology, they must be doing either Scrum or XP.

9.One of the core principles of agile is to "Embrace Change". This makes architecting and designing the system challenging since you don't know all of the requirements upfront. Thus, you have to be prepared to make architecture and design changes. To effectively embrace change, agile teams must learn how to keep the cost of change low.

*From the options below, select the ones that can help mitigate this challenge. (Select 3)*

Keep the code clean and continuously improve/refactor the design as needed. Don't delay these changes for later.

Keep design simple and just build what is required. It is easier to change a simple design than to update a complicated design.

Try to define as many requirements as you can at the beginning of the project, and spend a good amount of time designing the system.

Automated Regression testing provides faster feedback which helps you detect defects quickly. This in turn, gives you confidence to make changes.

10.Which of the following is true for a typical agile project? (select 2)

No discussion sessions happen around what we want to build. We start with coding.

The team talks about the high-level software architecture / design as well as their approach for building software (the release plan) before directly jumping into coding.

The team does a detailed design before starting to code.

If required, the team may need to plan and conduct training, etc. for users before launching the system.

Release planning is never required on an agile project. We only plan for a day or for a week at the most.

# Week 2

## User Stories

TOTAL POINTS 5

1.What of the following are the qualities of good user story as mentioned in the INVEST model? Choose three.

Independent: Dependencies among stories should be minimized

Natural: Stories should feel intuitive to the user

Valuable: Stories should be valuable to user

Editable: Stories should be editable after and during development

Specific: Stories should define a very specific requirement

Testable: Once the story is implemented, we should be able to test that it is done

2.What issues do you see in  following story?

*As a grocery buyer, I want to see different food items in different colors: red for meats,  brown for grains, green for vegetables so that i can identify food items by type.*

This story is not testable.

This story breaks the quality of negotiable.

The user "grocery buyer" is not a valid user.

This story has no issues.  It has all the info we need in a story.

3.What are the 3 Cs of user stories?

Card

Confirmation

Creative

Conversation

Complete

4.What are some of the key parts of a user story?  Choose four.

Acceptance tests

The estimated development time

The design of the back-end functionality including table and field names.

The story title

The names of the developer who will implement the story and the tester who will validate it.

Who is it for?  What does the user want to do?  Why does the user want to do that?

5.Which of the following statements are true about "Spike" stories?  Choose three.

Spikes are stories that allow slack time for team members to pursue their interests.

Spike stories are timeboxed.

They can be used for knowledge acquisition.

It is recommended that these stories have well defined acceptance criteria so that team knows what is expected at the end of exploration.

# Gathering User Needs

TOTAL POINTS 6

1.The goal of a story writing workshop is to:

Write as many foundational stories as possible.

Write as many user stories as possible in priority order.

Write as many user stories as possible under the themes identified.

Write as many themes as possible.

2.What are some of the valid ways to handle non-functional requirements in agile development? Select two.

Delay handling non-functional requirements until the end and then work on them after development is done.

Add non-functional requirements to the definition of done for all stories.

Create specific stories that outline the non-functional needs with clear acceptance criteria.

3.Which of the following are true about story writing workshops? Select two.

All stories created have to be at the same level.

A story writing workshop can take anywhere between a few hours to a few days.

The whole team (including the product owner, the scrum master, and the development team) participates in the story writing workshop.

4.What are some of the ways a story map can help software development? Select all that apply.

It helps plan releases and work.

It organizes and prioritizes the story backlog.

It helps communicate user needs with the stakeholders and get feedback.

It helps discover user needs.

It builds a shared understanding among team members.

5.What are some of the activities you expect to see in a story mapping exercise? Select three.

Find variations in the paths the user can take through the system.

Find gaps in the story map by walking through the user experience (as expressed in the activities/tasks) aloud.

Plan releases/journeys.

Randomly write a story and its details.

6.What are the characteristics of good product backlog?  (Select any 4)

Emergent

Every story is assigned to developer

Prioritized

Estimated

Complete with design details

Detailed appropriately

# Agile Estimation

TOTAL POINTS 6

1.Who estimates the effort to complete a product backlog item (a user story)?

The scrum development team after clarifying the requirement.

The scrum master with input from the scrum development team.

The most senior people in the organization such as the engineering managers and the chief architect.

The product owner with input from the scrum development team.

2.What is true about Agile Estimation techniques?  Select two.

Estimates are shared.

They estimate effort and not duration.

No planning, No estimation. It is considered non productive activity.

The senior-most person should create estimates and get feedback from team members.

3.Which of the following estimates represents concept of relative sizing? Select three.

Story A is 1 apple, Story B is 5 apples, and Story C is 10 apples

Story A is bigger than Story B. Story C is smaller than Story A.

Story A will take 1.5 ideal days, Story B will take 5 ideal days and Story C will take 3.5 Ideal Days

Story A will take 1 day, Story B will take 5 Days and Story C will take 10 days

Story A will take half the time Story B will take. Story C will take double the time Story A will take.

4.What is true about the planning poker estimation technique? Select three.

It helps uncover misunderstandings.

It supports the concept of shared estimates.

It is a superfast technique to estimate lots of stories in a short amount of time.

Everyone involved in development has to participate in estimation.

5.Which of the following observations are true about Card Sorting? Select two.

It is very useful for estimating a large number of stories.

It is a very inaccurate method of estimation.

This technique benefits from using relative sizing.

Not everybody on the development team participates in this method.

6.What is true about estimation using Ideal Days? Select two.

It is very fast to estimate.

It is very natural for a new team member to understand.

It has a potential issue of "My ideal days are not your ideal days".

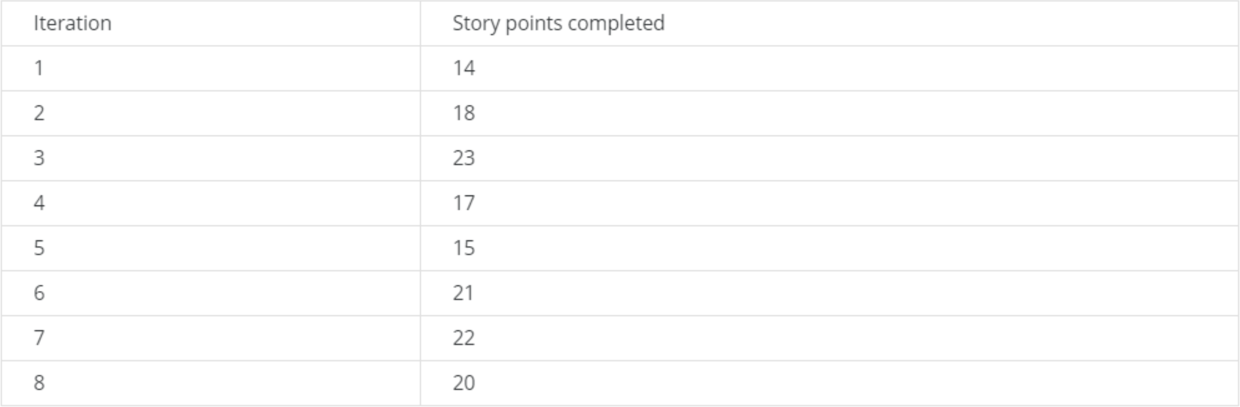
It is relative sizing technique.

# Release Planning and Tracking

TOTAL POINTS 10

1. Calculate the velocity range a team should use to select work for next iteration based on their past velocities (see chart below). The team uses the last 5 iterations to estimate their velocity. Use format "from-to" to specify the answer (e.g., 0-100).

Table 1: The velocity of previous iterations



| Iteration | Story points completed |
|---|---|
| 1 | 14 |
| 2 | 18 |
| 3 | 23 |
| 4 | 17 |
| 5 | 15 |
| 6 | 21 |
| 7 | 22 |
| 8 | 20 |

Enter answer here : 15-22

2.A team was doing release planning and they decided that the next release will include all stories from Story 1 to Story 11 (see table 2 below).

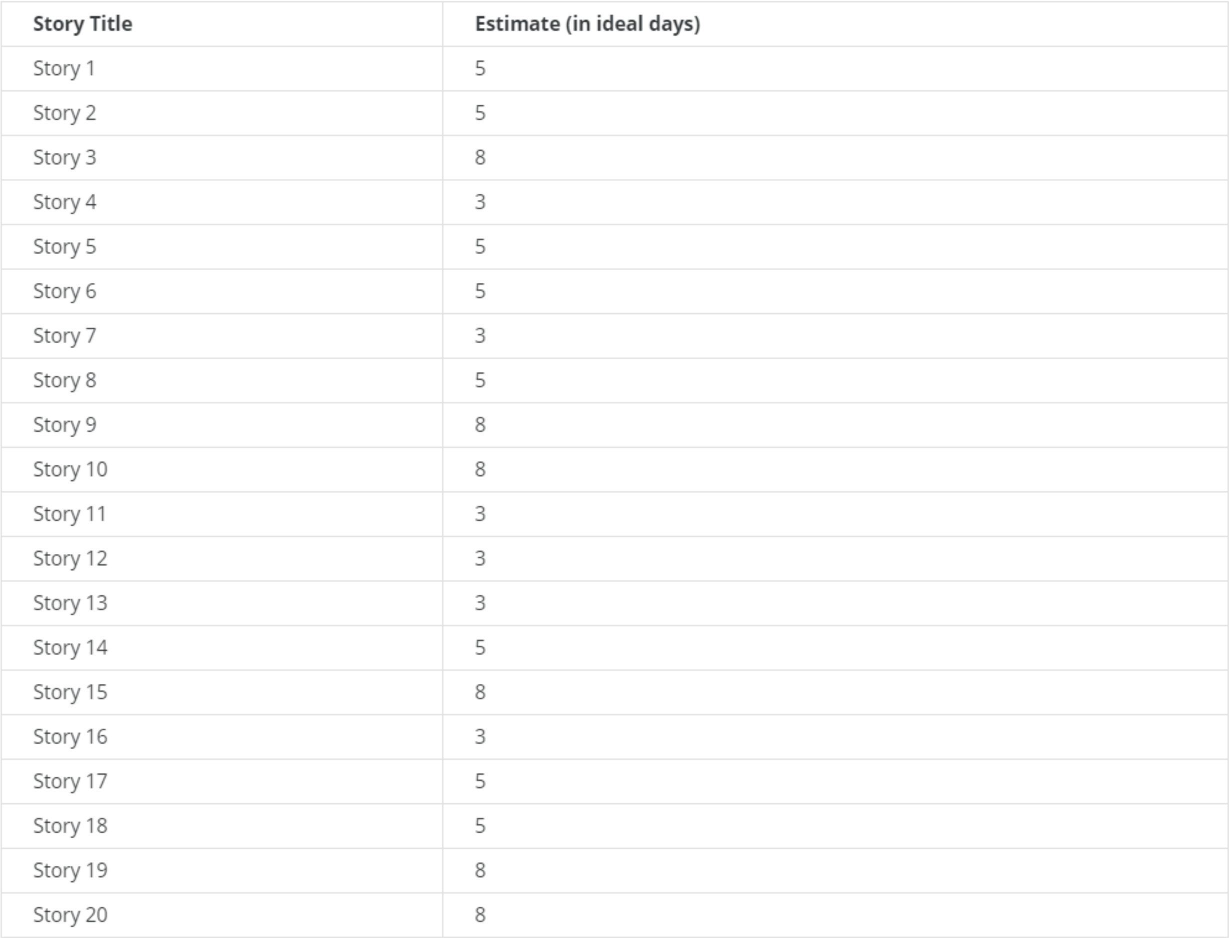
The velocity range to be used for the release planning is 15-22.

The team works in a 2 week iteration.

It costs about $50,000 per iteration to fund the entire team.

Calculate the estimated duration for next release.  Additionally, how much will this release cost?

Table 2: Prioritized Product Backlog



| Story Title | Estimate (in ideal days) |
|---|---|
| Story 1 | 5 |
| Story 2 | 5 |
| Story 3 | 8 |
| Story 4 | 3 |
| Story 5 | 5 |
| Story 6 | 5 |
| Story 7 | 3 |
| Story 8 | 5 |
| Story 9 | 8 |
| Story 10 | 8 |
| Story 11 | 3 |
| Story 12 | 3 |
| Story 13 | 3 |
| Story 14 | 5 |
| Story 15 | 8 |
| Story 16 | 3 |
| Story 17 | 5 |
| Story 18 | 5 |
| Story 19 | 8 |
| Story 20 | 8 |

Duration: 6-8 weeks,   Cost: 150K to 200K

Duration: 3-4 weeks, Cost: 150K to 200K

Duration: 10-14 weeks, Cost: 500K to 700K

Duration: 6-8 weeks,  Cost: 300K to 400K

3.Select scenarios below where the team used the correct approach to handle special cases of team velocity.

One of the team members is taking a couple days off in next iteration, but the team didn't adjust its velocity to reflect that.

The team is getting more defects / support work so they are not able to finish the stories selected for the current sprint. The team estimated that it takes 30% of team capacity to handle support work. The team decided to reduce velocity by 30% and use it for future iteration.

A new team member is going to join the team in the next iteration, so the team adjusted their velocity appropriately and selected more work to do in the iteration

To forecast velocity for first iteration, *1) the team builds deeper understanding of few stories from the backlog*

*2) From the understood stories, they select stories they feel they can get done in one sprint.*

*3) The sum the estimates of the selected stories is the team's forecasted velocity.*

4.Which of the following methods can help you track a release? (select any 2)

A cumulative flow diagram for a release

A release burn-up chart

An iteration burn-down chart

A weekly status report

# Requirements and Planning

TOTAL POINTS 10

1.What is true about user stories in Agile Software Development? (Select any 3)

User stories are tokens for conversations.

To be valid, a story MUST follow the following syntax template: "As a ____ I want to ________________ so that _____________________ ."

Stories can be written at different levels and can be refined and split into smaller stories as you move from vague idea to implemented software.

All stories have to be the same size.

User stories are used to plan, design, describe, build, and validate your product.

2.What are some of the benefits of writing acceptance tests for a story? (select 4)

They help the product owner (who is writing the story) think through the user needs.

They help build a common understanding between team members.

They help you write tests cases for the story.

They can potentially help you split stories, if required.

They make sure the story is easy to develop.

3.What is wrong with the following story? (Select any 2)

*"As a product owner, I want a list of highly-rated restaurants on the brochure."*

It doesn't specify the value of the story (the "why" is missing).

The user who benefits from this story is missing (the "who" is missing).

The story is not small.

The story is not negotiable.

4.What is wrong with following user story? (Select any 2)

*"I want the software to be easy to use."*

The story is not valuable.

It is not testable.

The story is very big.

No valid user is identified.

5.How is gathering user needs / requirements different on an agile project? (select any 3)

At any given time, the level of detail may be different for different parts of the software.

The agile way supports progressive refinement---defining the right level of detail at the right time.

Agile encourages conversations as a key method for building a shared understanding.

In agile, we don't waste time on understanding user needs. Instead, we build something, show it to the user, and learn.

6.What are some of the activities that happen during a User Story Writing Workshop? (Select any 4)

The leaders from the organization lay out exactly what needs to be built.

Everybody silently writes user stories around a theme

Sometimes, you keep it free-form for people to write stories and later group them by theme.

User role analysis and defining personas

To generate stories, one of the options is to start from the top down or bottom up.

7.Which of the following are true about the Story Mapping technique? (select 3)

It is a user-centric approach where we map out the system from a user's perspective.

The horizontal axis (moving from left to right) in a story map represents time.

The vertical axis (moving from top to bottom) in a story map represents rough priority.

One story map can only have one user. It is not recommended to have multiple users on same story map.

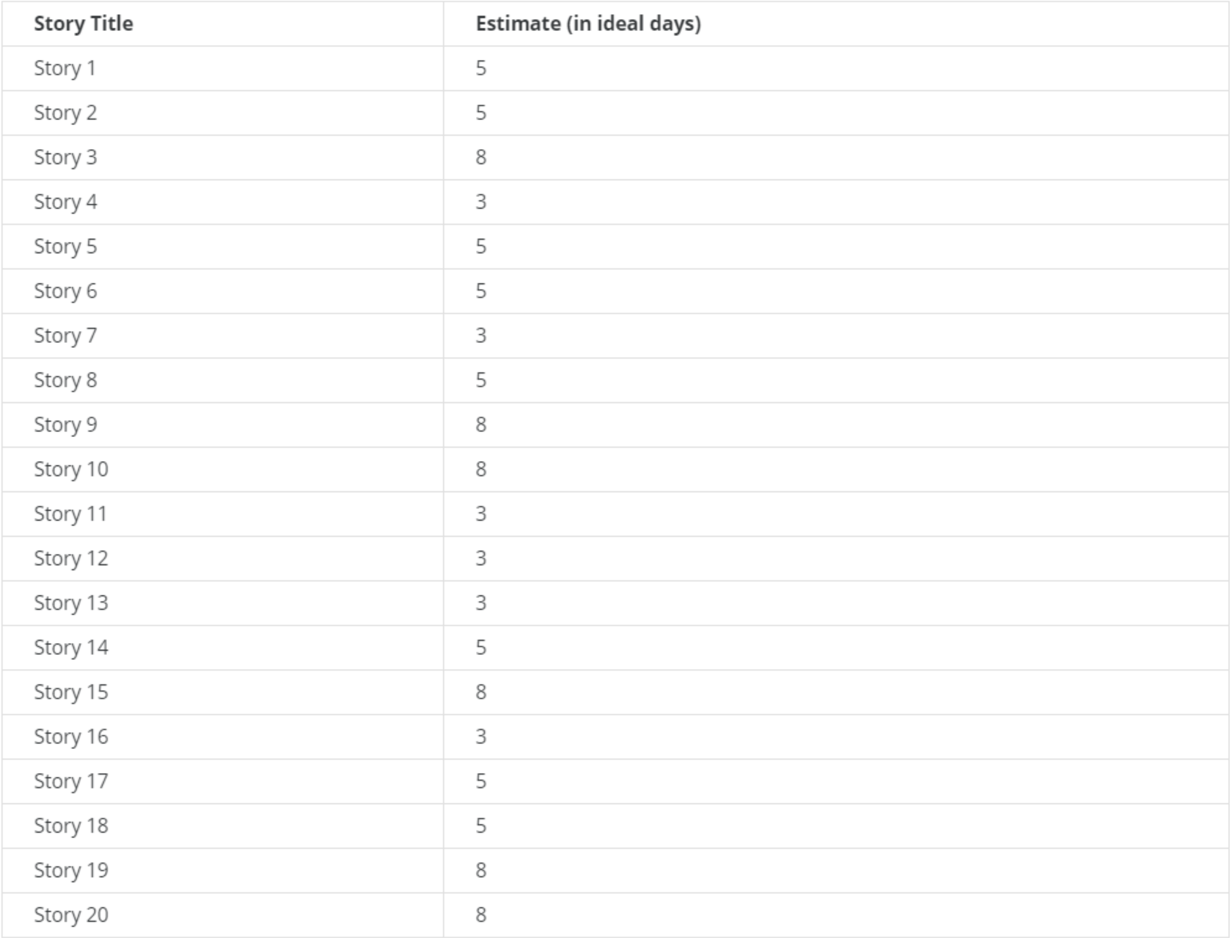
Only the product owner and users attend the story mapping exercise.

8.Due to an unpredictable market event, the Product Owner (PO) asked the team to complete the next release in 6 weeks. Assuming the backlog in Table 2 is up to date and prioritized from highest to lowest, the Product Owner wants you to estimate what can be done in next 6 weeks. Describe 1) what can be done, 2) what might be done, and 3) what will not be done.

The velocity range to be used for the release planning is 15-22.

The team works in 2 week iteration.

Table 2: Prioritized Product Backlog



| Story Title | Estimate (in ideal days) |
|---|---|
| Story 1 | 5 |
| Story 2 | 5 |
| Story 3 | 8 |
| Story 4 | 3 |
| Story 5 | 5 |
| Story 6 | 5 |
| Story 7 | 3 |
| Story 8 | 5 |
| Story 9 | 8 |
| Story 10 | 8 |
| Story 11 | 3 |
| Story 12 | 3 |
| Story 13 | 3 |
| Story 14 | 5 |
| Story 15 | 8 |
| Story 16 | 3 |
| Story 17 | 5 |
| Story 18 | 5 |
| Story 19 | 8 |
| Story 20 | 8 |

Most likely: Story 1 to Story 8; Might Be Done: Story 9 to Story 13; Not likely: Rest of the stories

Most likely: Story 1 to Story 9; Might be Done: Story 10; Not likely: Rest of the stories

Most likely: Story 1 to Story 11; Might be Done: Story 12 and 13; Not likely: Rest of the stories

9.What is true about planning with an agile mindset?  (Select any 2)

Use Adaptive Planning.

Plan as little as you can. Planning is considered a non-productive activity.

Planning is mostly done by project manager.

Plan just enough, just in time.

10.What do we mean by Velocity in agile terms?

It is calculated at the team level and represents the estimated duration of current iteration.

It is a velocity of the individual team members, not the team

It is calculated at the team level and represents how much work a team can get done in an iteration.

It represents actual work done in the first iteration of project.

# Week 3

## Scrum Overview

TOTAL POINTS 4

1.Which of the following are official rituals/meetings/practices in scrum?  Select four.

Sprint Planning Meeting

Daily Scrum Meeting

Sprint Retrospective Meeting

Sprint Review Meeting

Mid-Sprint Status Review Meeting

2.What are the three roles in Scrum?

The Team

Scrum Master

Product Owner

Analyst

Architect

3.Who prioritizes the work in Scrum?

The Team

Scrum Master

Team Manager

Product Owner

4.What artifacts are defined as part of Scrum framework?  Select two.

Product Backlog

Sprint Backlog

Iteration Burndown / Burnup

Product Specification

Test Plan

# Sprint Planning and Tracking

TOTAL POINTS 6

1.In Scrum, when is the sprint backlog created?

At the beginning of the project

During the sprint planning meeting

During the backlog refinement meeting

Whenever needed

2.In Scrum, how is the Product Backlog arranged?

Large items at the top, small items at the bottom.

Items are randomly arranged.

Into categories, P1, P2, P3, etc.

Most important items at the top, least important items at the bottom.

3.In which artifact is the customer requirements stored?

In the Product Backlog

In a database

In a Scrum Product Requirement Specification

Nowhere. The Scrum Product Owner knows them

4. What is usually plotted on the x-axis of the Sprint Burndown Chart?

Days of the sprint

Tasks

Hours

Team member names

5.What is a Sprint Burndown Chart?

A sprint plan which is burned to celebrate successful completion of sprint.

A chart showing the trend of work remaining across time in a sprint.

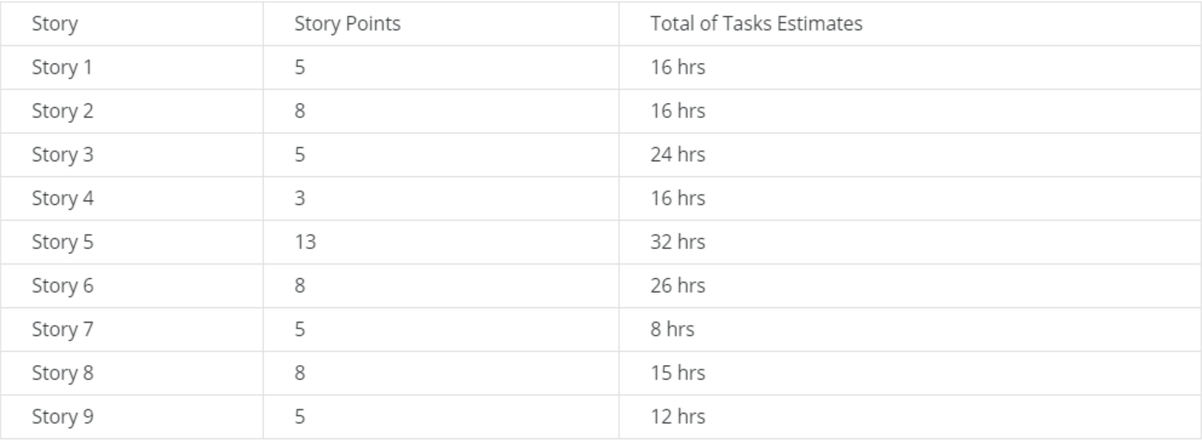
A chart showing the trend of work accomplished across time in a sprint.

A graph to measure human burnout effect due to fast pace of sprint

6.Your team is planning out the next sprint. You've chosen to fill the sprint by taking stories in priority order from the product backlog and stopping when you reach the first story that won't fit in the sprint.

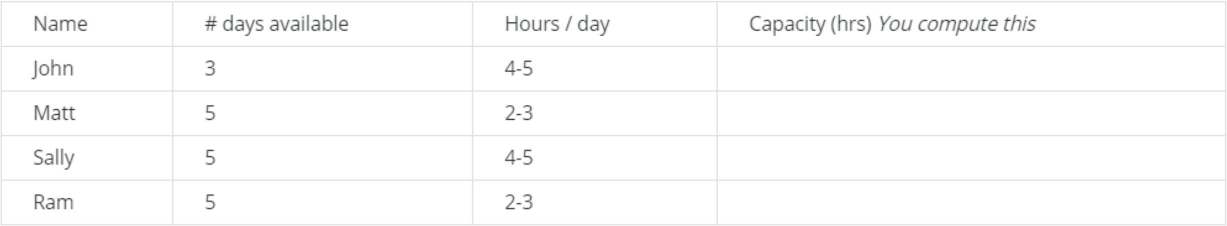
Based on following details, which stories should the team commit to for a sprint?

Table 1: Prioritized story with estimated story points and total estimate in hrs of tasks for that story.



| Story | Story Points | Total of Tasks Estimates |
|---|---|---|
| Story 1 | 5 | 16 hrs |
| Story 2 | 8 | 16 hrs |
| Story 3 | 5 | 24 hrs |
| Story 4 | 3 | 16 hrs |
| Story 5 | 13 | 32 hrs |
| Story 6 | 8 | 26 hrs |
| Story 7 | 5 | 8 hrs |
| Story 8 | 8 | 15 hrs |
| Story 9 | 5 | 12 hrs |

Table 2: Capacity of Team members for given sprint



| Name | # days available | Hours / day | Capacity (hrs) *You compute this* |
|---|---|---|---|
| John | 3 | 4-5 | |
| Matt | 5 | 2-3 | |
| Sally | 5 | 4-5 | |
| Ram | 5 | 2-3 | |

Story 1 and Story 2

Story 1, Story 2, Story 3, and Story 7

Story 1, Story 2, Story 3, and Story 4

Story 1, Story 2, and Story 3

# Sprint Review, Retrospective and Execution

TOTAL POINTS 4

1.During sprint execution, when are new tasks added to the sprint?

When the product owner identifies a new task.

When the scrum master identifies a new task.

Never. Sprint tasks are fixed and decided during sprint planning.

As soon as possible after they are identified, unless they reflect a scope change in the sprint goals.

When the scrum master approves them.

2.Who should attend the sprint retrospective?

All team members.

The scrum master and some of the team members.

Everybody in the team, except for the product owner.

Everybody who is attending the sprint review, to get as much feedback as possible.

The product owner, team members, and their managers.

3.How long should the Daily Standup be?

At least an hour.

As long as you need.

Short fixed duration (most commonly 15 min).

Change based on how many people are attending.

4.Which of the following are discussed in a sprint retrospective? Select three.

What's working?

What's not working?

New action items / What can we do better?

Finding mistakes and achievements of individual team members.

# Scrum

TOTAL POINTS 10

1.Which of the following is true about Scrum?

In the daily stand-up, everybody talks about what they worked on, what they are planning to work on, and any impediments they have.

In a Sprint Review, we talk about process and talk about what worked, what didn't work, and how we can improve.

Burn-down and Burn-up charts help the team track the progress of the current sprint.

In a Sprint Retrospective, we talk about what we built, demo it, and get customer/client feedback.

It is encouraged to make iteration scope changes during the sprint.

2.Which of the following is TRUE about Scrum? (Select two)

Scrum is an adaptive model.

Scrum has fixed, time-boxed development cycles called sprints.

The daily stand-up in scrum is mainly for the scrum master to get the status of the project.

Scrum is the best way to implement the agile mindset. Other Agile methods are not as effective.

3.You are a developer on a scrum team. Your scrum master invited you to attend a sprint planning meeting. Which of following activities would you expect in that meeting?

The developers/testers ask questions to understand the stories.

The product owner (or equivalent) shares the top priority stories for the sprint.

The team will select the stories to work on for next iteration (sprint).

Either during the meeting or afterwards, developers create tasks to further solidify what work needs to be done and make final a work commitment for the sprint.

The developers/testers provide a high-level estimate for stories.

The team talks about what went well and what didn’t go well in the sprint.

4.You need to calculating a team member's capacity for an iteration.  Which of the following activities should be EXCLUDED from the committed capacity of the team member?

Time spent on defects

Paid time off (PTO)

Organizational Meetings

Sprint Ceremonies

Basic support for the product

Helping other team members

5.What should happen in the sprint review meeting?

Feedback is gathered from stakeholders resulting in additional items added to the product backlog.

A potentially shippable product increment is demonstrated live.

Presentation slides are shown about hypothetical things.

What happened during the Sprint is reported.

6.The CEO asks a team member to do some work outside the goals of the current sprint in progress. What should the team member do?

Add it to the next sprint.

Add it to the current sprint while swapping out committed work of equal size.

Inform the product owner so he can work with the CEO and if it is still necessary to get this work done in current sprint, it should be discussed with in the team.

Add it to the current sprint.

7.When is a sprint complete?

When all tasks are complete.

When all committed product backlog items meet their definition of done.

It depends.

When the sprint duration ends.  For example, if sprints are 2 weeks long, then the sprint is complete at the end of those 2 weeks.

8.Which of the following are goals of the daily stand-up meeting?

Report progress to the manager.

Assign work to people.

Share status information.

Set team direction and synchronize the day's activities.

Identify impediments.

Remove every single impediment in this meeting.

9.Which of following is true about Sprint Reviews?

Only team members attend the Sprint Review.

The Sprint Review is to show stakeholders the documentation and design. The demo should be given to customer before the sprint review.

Individual team members should be encouraged to demo the work they did.

The Sprint Review should be done every week even if your sprint duration is 3 weeks or longer.

10.Which of the following statements are true about the Sprint Retrospective? (select all that apply)

Managers should be required to attend retrospectives so they know what problems team members have.

There is one format you should use to conduct retrospectives: What's working, what's not working, and what can we do better?

It is important to follow up on action items the team decided to work on.

Team members should avoid blaming (finger pointing) other team members in the retrospective.

# Week 4

## XP

TOTAL POINTS 10

1.Which of the following is the right sequence when developing software using the XP practice of Test First Programming?

Write code

Refactor as necessary

Write a test

Make the test pass

Write code

Write a test

Make the test pass

Refactor as necessary

Design

Write code

Write a test

Write a test

Make sure the test fails

Write enough code so the test passes

Refactor as necessary

2.Which XP practice prescribes that "the code [always be] written by two programmers at one machine"?

Buddy Programming

Twin Programming

Peer Programming

Pair Programming

3.Which of the following are primary practices prescribed by XP?

Whole Team

Continuous Integration

Weekly Retrospectives

Daily Standups

Pair Programming

4.One of the practices of XP is "Whole Team".  Which of the following statements align with its meaning?

The whole team should be energized and passionate about the product they are building.

The whole team should be working together to meet the team's commitment

All the skills necessary to deliver the software product should be present on the team.

The whole team should always sit together in a room.

5.If an XP team cannot provide an estimate for a story, what should they do to gain a better understanding of the story?

Skip that story and exclude that story from the release.

No action required; keep the story with no estimates.

Provide their best guess based on what they know.

Create a spike story---a new story under which the team will do some research or other work to gain a better understanding of the original story.

6.According to XP's principles, what should you do when a story's acceptance test fails?

Release the software without fixing; a few bugs are ok.

As a team, update the acceptance test so the test passes.

Create a bug report so it can be fixed.

Extend the iteration's duration so the bug(s) can be fixed in the same iteration.

7.What activities occur as part of XP's "Weekly Planning" practice?

Selecting the next week's work.

Breaking stories into smaller tasks.

Reviewing the previous week's progress.

Planning who is going to facilitate meetings for the rest of the week

8.According to the concept of "Whole Team", which of the following statements are true?

It is not recommended for a team member to be part of multiple teams.

All of the skills the team needs to be successful should be in the team.

The team members never change. Once a team member is part of the team, that team member always remain part of the team.

If the team finds out that they need a particular skill in the team, they can add a person with that skill to the team.

9.An XP team is getting feedback more frequently than they can handle. What should they do?

Work extra hours each day to address all of the feedback.

Further increase the frequency and quantity of feedback.

Slow down the frequency of feedback.

Ignore the extra feedback.

10.Which of the following statements are true about the XP value of "Simplicity"?

No matter what you are building, just choose a simple design even if it only partially solves the problem.

Select the simplest design that could possibly work.

Generalize your solution so it can work for unanticipated future requests.

The software design for each component should fit on a single page.

# Course 3

# Lean Software Development

## Week 1

# Lean Fundamentals

TOTAL POINTS 3

1.Which of the following are 5 Lean principles? (Not the 7 principles of Lean software development)

Weekly Cadence

Seek Perfection

Kaizen

Map the Value Stream

Identify Value

Create flow

Establish Pull

2.In terms of "Lean", what is the best description of Waste?

Waste is a part of the Process and is always inevitable.

Waste is something you put in the rubbish bin.

Waste is the usage of any resource during the process, which are not recognized by the customer as adding value to the product or service they receive.

3.Lean techniques define "Value" from the perspective of who?

Team Members

The Customer

All stakeholders

The Manager

# Lean Principles

TOTAL POINTS 12

1.In Lean software development, how can amplifying learning occur? (Select all that apply)

The developers make different solutions, each with the exact same features.

The developers use short iterations.

The developers write detailed specifications before writing code.

The developers choose technologies that force them to learn new skills.

2.How can waste arise in software development? (Select all that apply)

Developers work on rarely used product features.

There are product defects found late in the development.

Knowledge about the developers' skills is shared among the team.

Required features expected to be developed in a given time period are fully "done".

3.In Lean software development, what does the principle of "defer commitment" mean?

Decisions are made to choose the modern alternative.

Decisions are made just before a deadline.

Decisions are made after having enough information from considering the alternatives.

Decisions are made to delay the product delivery to a later date.

4.In Lean software development, what does the principle of "deliver fast" mean?  (select 2 that apply)

A simpler version of software product (an MVP - Minimum Viable Product) is developed first to reach the market rapidly.

Iterations are short, so feedback is frequent, and product evolution is rapid.

The software product is delivered rapidly via courier.

Working alternatives are rapidly created.

5.Which of the following align with the lean principle of "build quality in"? (select 2 that apply)

External inspectors determine whether the software product is high quality.

The developers refactor the source code to be simpler and easier to modify.

Instead of focusing on refactoring every sprint, certain sprints are dedicated to focus on quality.

The developers apply practices to avoid or quickly catch errors while making the software product.

6.In Lean software development, what does "optimize the whole" mean? (Select all the apply)

Rather than trying to optimize one part, look at the whole system for optimization opportunities.

The ability of a system to achieve its purpose depends on how well the parts work together, not just how well they perform individually.

Leaders look at the whole picture while the other team members only look at the parts of the system.

The whole software product is merely the sum of its individual features.

7.Which of the following tools can help you find waste in a process?

5 Whys

Kaizen

Exploring multiple options

Value Stream Mapping

8.What are some of the tools to amplify learning? (Select all that apply)

Daily Builds and Smoke Tests

Iteration

Sustainable development

Set Based Development

9.How do you deliver fast (i.e reduce cycle time) without compromising the quality?

Finish the work and leave quality for later.

Have people work on multiple things at the same time.

Ask people to work overtime.

Minimize the number of items in-process.

10."The Last Responsible Moment" is a tool for which Lean Software Principle?

See the Whole

Defer Commitment

Deliver Fast

Empower People

11.Lean helps increase productivity by:

Adding additional resources to help decrease the burden of the workload.

Increasing inputs to directly increase outputs.

Giving staff more work to help create more output.

Reducing the amount of useless activity that our resources (the team members) are currently burdened with.

12.Which practices waste time during software development? (Choose two)

Delays in receiving work products.

Everyone is aware of the progress through an updated Kanban board.

Weekly meetings that are held whether or not they are needed.

The working software is shown to the client, who offers constructive feedback.

# Week 2

## Kanban

TOTAL POINTS 3

1.Which of these are core properties of Kanban? (Select all that apply)

Limit work in progress

Visualize the Work

Manage Flow

Retrospectives

Daily stand-up

2.Which of following is valid on a Kanban Board?  Select two.

The WIP limit for Step X is 3. Step X is divided into "Doing" and "Done". The number of items in the "Doing" column is 4, and the number of items in the "Done" column is 0.

The WIP limit for Step X is 3. Step X is divided into "Doing" and "Done". The number of items in the "Doing" column is 3, and the number of items in the "Done" column is 2.

The WIP limit for Step X is 3. Step X is divided into "Doing" and "Done". The number of items in the "Doing" column is 2, and the number of items in the "Done" column is 1.

The WIP limit for Step X is 3. Step X is divided into "Doing" and "Done". The number of items in the "Doing" column is 2, and the number of items in the "Done" column is 1. One of the items in the previous step is done and the developers want to move that item to Step X.

The WIP limit for Step X is 5. Step X is divided into "Doing" and "Done". The number of items in the "Doing" column is 3, and the number of items in the "Done" column is 1. One of the items in the previous step is done and the developers want to move that item to Step X.

3.A software development team just started using Kanban. They had the following columns on their Kanban board, in order from left to right: Backlog, Analysis, Development, Testing, Deployment. Each column has a WIP limit of 3.

*There are currently 3 items in testing (3 items being tested). Similarly, there are 3 items in the development column (2 items that are in development and 1 item is done with its development).*

*One of the developers does not have anything to work on, so the manager wants to add one more item to the development column so the developer looking for work can work on it.*

Which of the following are valid next actions in this situation? Select all that apply.

It does not make sense for the developer to wait for items to move out of testing. The manager should add one more item to the development column with no other change needed.

The manager should ask the developer waiting for work to help the testers so that they can finish one of the items in their column and move it to the deployment column.

The team should get together and discuss if they should add another tester to the test team so this bottleneck can be resolved quickly.

The manager should ask the testers to temporarily move an item from the test column to the deployment column so that we can move a different item from development to test. Later, we can move the item back from the deployment column to the testing column

The manager should ask the deployment team if a done item in the testing column can be deployed so there is space to move an item from development to testing.

# Kanban, Value Stream Mapping and Kaizen

TOTAL POINTS 7

1.The 5 "Whys" to be considered during a Root Cause Analysis are...?

Who, What, When, Where and Why

Who, Why, Why not, When, What

Who, Who, Who, Who, and Who

Why, Why, Why, Why and Why

2.What is the purpose of the Lean tool 'Kaizen'?

To create a map of the physical path taken by the product as it passes through the process.

To ensure that the workstation does not make anything until the next work center requests supply.

An improvement activity to create more value and remove waste.

To analyze the forces supporting or resisting a particular change.

3.Which of the following can be used to do 5 Whys analysis?

A node graph

A grid diagram

A fishbone diagram

A sin diagram

4.In Kanban, a developer pulls the next item to work on from the:

previous column on the board

managers who determine which tasks will be performed

product backlog

clients determining the tasks

5.You are a developer on a team using a Kanban board to manage flow. You work on Step X within the process and it just so happens that you are looking for more work.

*The Kanban board is set up as follows:*

*Step X has a WIP limit of 3.*

*Step X is divided into "Doing" and "Done."*

*The number of items in the "Doing" column is 2.*

*The number of items in the "Done" column is 1.*

*Without violating the rules of the Kanban board, what should you do? Select all that apply.*

Ignore the WIP limit of 3. Pick the next item to work on from the previous column.

Help the group (that picks work from the following column) finish their work so they can pull an item from the "Done" column of Step X. That will allow you to pick an item from the previous step.

Talk to the team about adjusting the WIP limit because increasing the WIP limit will create a better flow of work

Push the done item to the next column/step without considering the WIP limit of the next column/step.

6.Which of the following statements are true about Value Stream Mapping? Select all that apply.

Cycle Time is the total time taken for Value Stream.

Value-added activities are activities that add value from the customer perspective.

It is recommended that you go to the place where work is happening because that will distract the people working and will skew your value stream map.

There is no need to identify the non-value-added map as it does not add value for the customer.

7.What is the formula to calculate the Process Cycle Efficiency?

Value Added Time / Cycle Time

Value Added Time / Non Value Added Time

(Value Added Time - Non Value Added Time) / Total Cycle Time

Value Added Time - Non Value Added Time

# Week 3

# Lean Startup

TOTAL POINTS 15

1.What is the main goal of the Build-Measure-Learn Cycle?

To make sure we do quality work through every step of this cycle.

To make sure we get lot of work done effectively in this cycle.

To go through the build-measure-learn cycle as cheaply as possible.

To go through the build-measure-learn cycle as fast as possible.

2.What do we mean by Validated Learning?

Validating the assumptions by conducting an experiment and collecting data.

Listening to what the customer is saying.

Launch a product and then decide what we want to measure to validate.

Validate what you learned from your product with company leaders.

3.Which of the following are principles of Lean startup?  Select all that apply.

Invest as little as possible and build inexpensive products

Validated Learning

Innovation Accounting

Entrepreneurs are everywhere

4.Which of the following are true about Lean Startup?  Select two.

It relies on actual data rather than gut feelings to validate assumptions or ideas.

It is a concept applicable to startup companies only.

It is based on going through the Build-Measure-Learn cycle as fast as you can.

It is the best method to use for all kinds of development.

5. Why are traditional management principles ill-suited to the world of startups?

Because startups can only be built with a “just do it” attitude.

Because startups face chaos and uncertainty.

Because building a startup is not the same as institution building.

6.When a startup 'pivots', what is it doing?

Optimizing its product.

Employing a revised strategy in an attempt to achieve its vision.

Ignoring its vision.

7.According to Ries, which of the following adjectives describes good startup metrics? Select all that apply.

Audience-Focused

Actionable

Accessible

Auditable

8.Which of the following are levels of Pirate Metrics? Select all that apply.

Referral

Registration

Retention

Revenue

Activation

Acquisition

9.Which of the following are true about Pivot or Persevere? Select two.

Platform pivot: A change from an application to a platform or vice versa.

Zoom-out pivot: What was previously considered a single feature in a product becomes the whole product.

Customer segment pivot: The product hypothesis is partially confirmed; it solves the right problem but for a different customer than originally anticipated.

Pivot means to close the startup company and start a new one.

10.How do Vanity metrics prevent/inhibit pivoting?

Vanity metrics allow entrepreneurs to draw incorrect conclusions. Based on these metrics, they assume that the product is going in the right direction when it really isn't.

Vanity metrics are the metrics that matter.

Vanity metrics are designed by leadership so no pivoting is necessary.

Vanity metrics are user-based collection which is always the right metrics to measure.

11.What are three ways that a past customer can drive sustainable growth?

By reporting defects in your product

Repeat purchase

As side advertising due to usage of the product

Word-of-mouth referral

12.What are the characteristics of  "Early Adopters"?

They only use your product if it is free.

Early adopters will give you their time and honest feedback.

They use the product only if it is reliable and of good quality.

Early adopters are actively seeking a solution to the problem your product solves.

13.What are leap-of-faith assumptions?

Assumptions that, if true, would result in a lot of revenue.

Assumptions that the organization made after getting the blessing of the stakeholders.

Assumptions that are the riskiest elements of a startup plan.

14.Which of these attributes are favored by Lean startup? Select two.

Waterfall development

Customer feedback

Experimentation and Iteration

Elaborate upfront planing

15.Who is an "Intrapreneur" ?

Entrepreneurs that work with other entrepreneurs.

A manager who promotes innovations within the company.

The opposite of an entrepreneur: someone who prevents innovation.

# Week 4

## Design Thinking

TOTAL POINTS 5

1.Which of the following are true about Design Thinking? Select 2.

It is a technique for designers to design software.

It is a process for creative problem solving.

It recommends that a cross-functional team work together in the design thinking process.

It consists of five steps: requirements, design, development, testing and deployment.

2.In design thinking, what are two of the main reasons to empathize with users?

To discover the real problem to be solved.

To make sure users know that we are building the solution for them.

To build a good relationship with users.

To prevent team member bias from impacting their understanding of the problem and influencing the solution.

3.Design thinking results in innovation by integrating which three things?

Business Viability

Human Desirability

Practical Ability

Technical Feasibility

4.What are the two dimensions for categorizing assumptions?

Probability that it will be wrong

Probability that it will be right

Impact on our solution if it is right

Impact on our solution if it is wrong

5.What are the two dimensions for categorizing tests?

Must have Tests

Cost/Effort of conducting a test

Nice to have Tests

Quality of data collected from the test

# Course 4

# Engineering Practices for Building Quality Software

## Week 1

## Quality in Design

TOTAL POINTS 18

1.The logical (intended) dependency is that observer depends on subject.

True

False

2.Up until now, the great state of Foo has held a lottery to help fund education in the state. The corporation tasked with the drawing of these numbers (the non-televised ones) is XYZ Numerical Tasking.  You are the technical lead for the system which handles the drawings for the state's lotteries. One of your developers, with a mathematical tilt, comes to you with a proposal: change how the random numbers are generated.

He suggests that the generation of random numbers could be better. NASA has released a random number generator which has been proven to be better than the one used by the company. He suggests that you make the change.

One of your more senior developers notes that using the new generator will require a change. He suggests that the team connect the existing lotto system, as seen in Figure 1, to the new generator (Figure 2) using the Facade pattern.

Figure 1

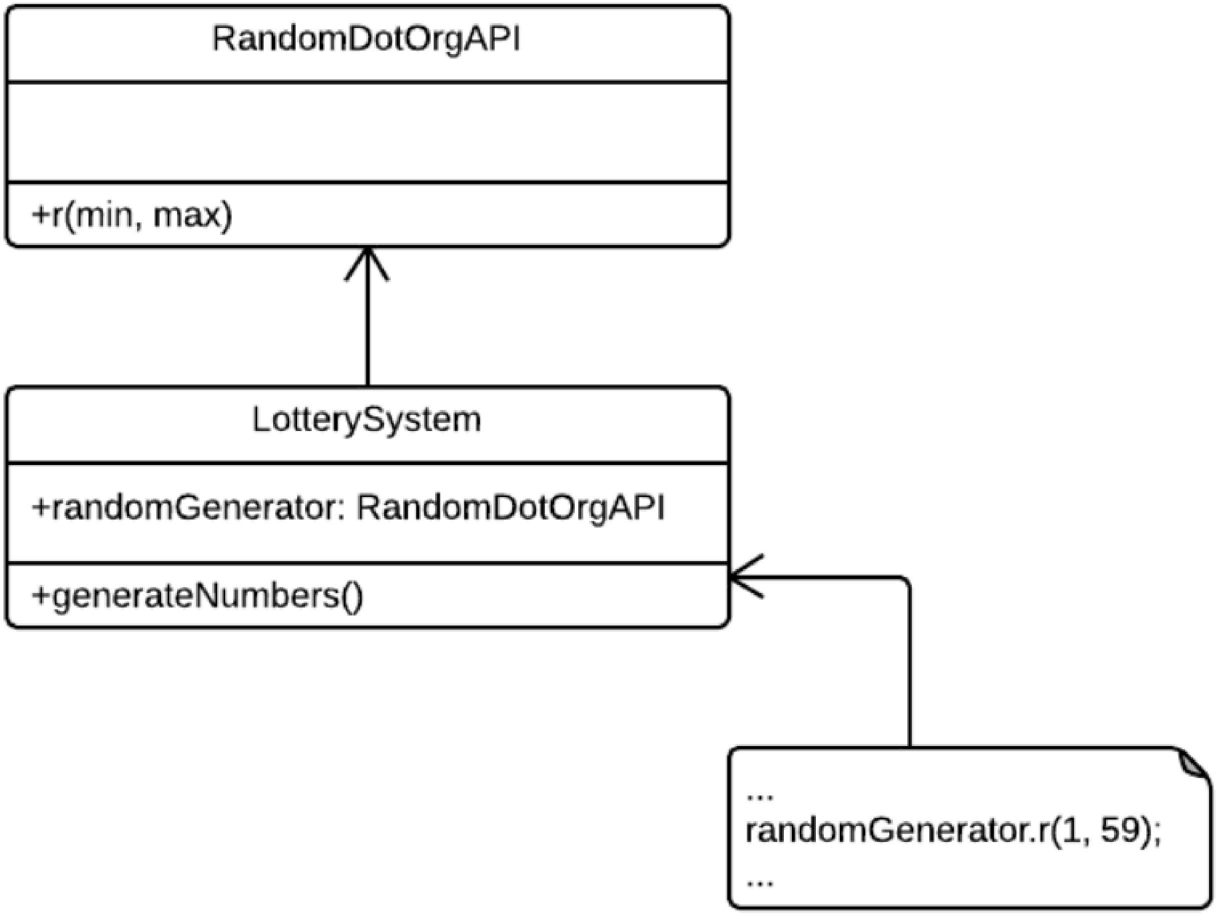
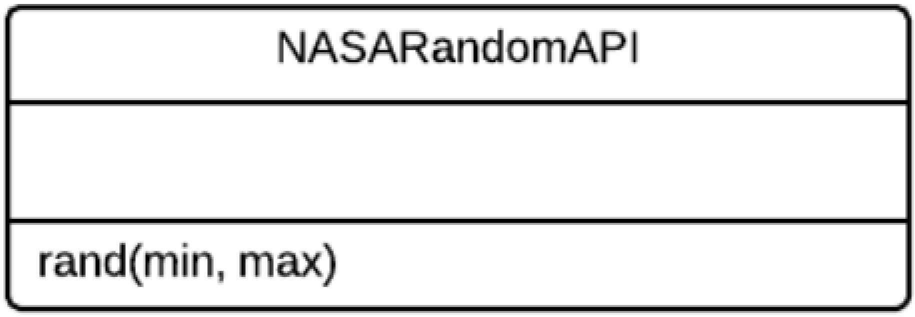


Figure 2



Facade, which seeks to simplify complex processes for a client by providing a API which hides the complicated calls/work needed, doesn't seem to fit.

What pattern does?

Enter answer here : Adapter

3.Within the Custodial Management System, there are are a variety of items that are being watched at any given time. Certain items can receive notifications that a response is needed immediately (rather than just tracking whether the regular upkeep has been done that week/month/etc.). These items may then need one of several possible actions: rapid cleaning, bodily fluid containment and/or removal, addressing traction/slip-danger, urgent repair, etc. Each of these actions fit the same profile (action) but not necessarily the same steps.

Which pattern would fit well as a solution to this systems need?

Enter answer here : Observer

4.Early in our proof-of-concept work on the new game we are building, we effectively hard-coded the game to ask the user what action to take next, for both players. In this case, both users are tied into a single account, so that we can play against ourselves while we build out the game.

Now, however, we want to add a new player type to the system, EasyDifficultyBot. While the "player" actions fit the same high-level format for both humans and our AI, the details of making it happen vary pretty widely.

In order to allow for this change, we want to make use of a pattern that will allow this kind of expansion while minimizing changes to the existing code.

What pattern would be best here?

Enter answer here : Strategy

5.Consider the statement "In the Strategy pattern, the context should hold references to the ConcreteStrategies."

Under what circumstances would this be true? Select all that apply.

When the Strategy base class uses a static method to return the correct derived type

Under no circumstances should the Context hold references to the ConcreteStrategies

When using a Factory object to create Strategies

When the client is tasked with deciding which strategy to use

When the context is tasked with deciding which strategy to use

6.In the Observer pattern, one difficulty is that the Subject cannot pass an instance of itself to the Observer being updated, due to circular dependencies.

True

False

7.Once a class has been had an Adapter pattern solution applied, it cannot be adapted again.

True

False

8. Factories are no longer useful when you apply the Strategy Pattern to a family of algorithms.

True

False

9.Observer is an implementation of the Dependency Inversion principle.

True

False

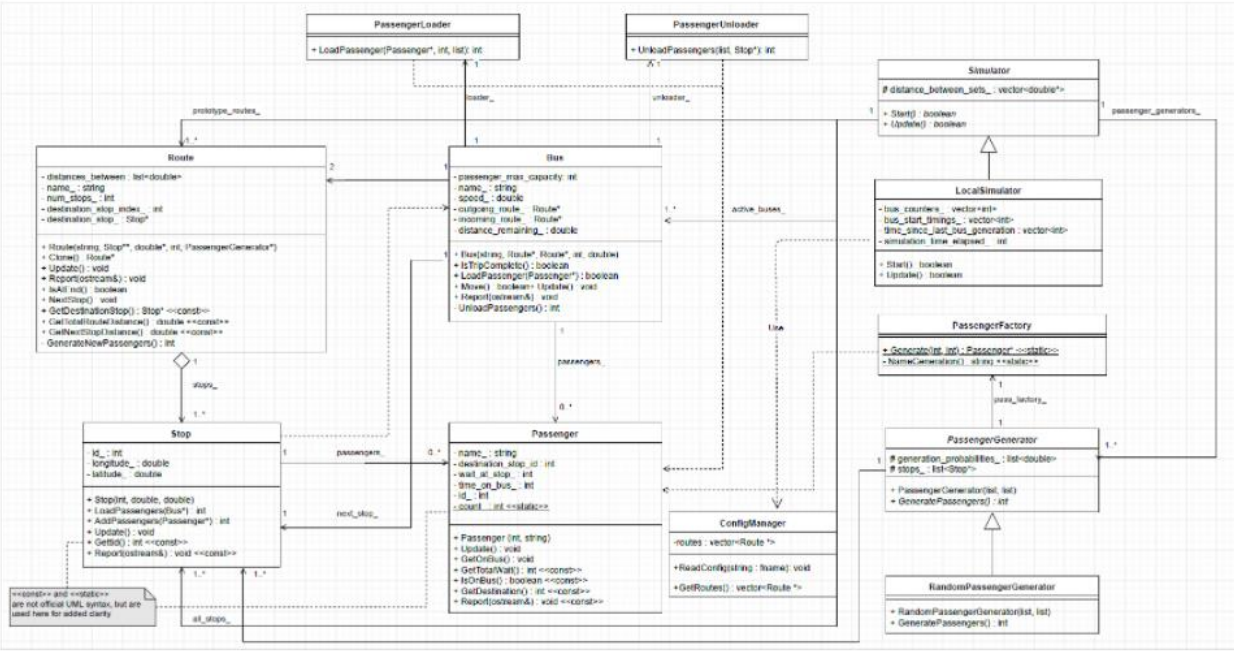
10.Strategy Pattern helps maintain the Open/Closed Principle.

True

False

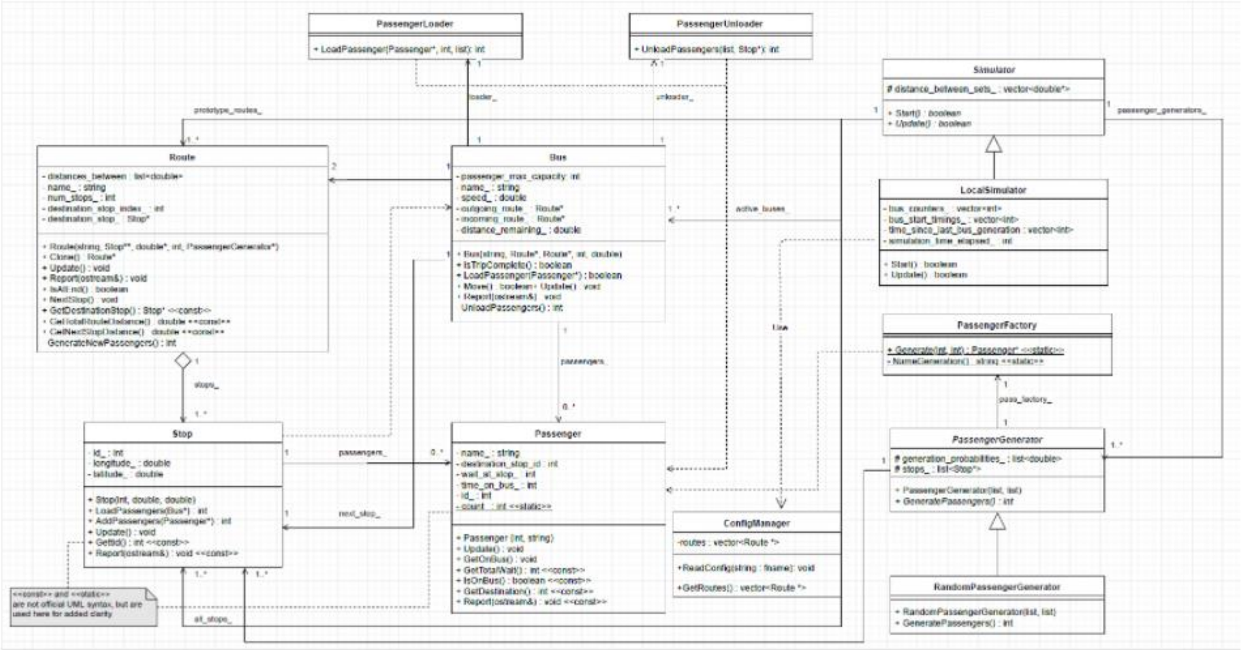
11.Notes: For questions 11-14, the diagram is a standard UML Class Diagram. You can view a larger version by right clicking and selecting "View image" (or similar) or by holding on the image and selecting "Open image in new tab" (or similar) if you're on mobile. Use up to 3 significant digits (e.g. .5 for 1/2 and .333 for 1/3 are acceptable).

For the following class diagram, calculate the Instability value for the Bus class.


PassengerLoader
+ LoadPassenger(Passenger*, int, list): int
PassengerUnloader
+ UnloadPassengers(list, Stop*): int
Simulator
# distance_between_sets_ : vector<double*>
+ Start() : boolean
+ Update() : boolean
prototype_routes_
loader_
unloader_
passenger_generators_
1
1..*
2
Route
- distances_between_ : list<double>
- name_ : string
- num_stops_ : int
- destination_stop_index_ : int
- destination_stop_ : Stop*
+ Route(string, Stop**, double*, int, PassengerGenerator*)
+ Clone() : Route*
+ Update() : void
+ Report(ostream&) : void
+ IsAtEnd() : boolean
+ NextStop() : void
+ GetDestinationStop() : Stop* <<const>>
+ GetTotalRouteDistance() : double <<const>>
+ GetNextStopDistance() : double <<const>>
- GenerateNewPassengers() : int
Bus
- passenger_max_capacity_ : int
- name_ : string
- speed_ : double
- outgoing_route_ : Route*
- incoming_route_ : Route*
- distance_remaining_ : double
+ Bus(string, Route*, Route*, int, double)
+ IsTripComplete() : boolean
+ LoadPassenger(Passenger*) : boolean
+ Move() : boolean
+ Update() : void
+ Report(ostream&) : void
- UnloadPassengers() : int
active_buses_
LocalSimulator
- bus_counters_ : vector<int>
- bus_start_timings_ : vector<int>
- time_since_last_bus_generation : vector<int>
- simulation_time_elapsed_ : int
+ Start() : boolean
+ Update() : boolean
Use
PassengerFactory
+ Generate(int, int) : Passenger* <<static>>
- NameGeneration() : string <<static>>
pass_factory_
stops_
passengers_
0..*
Stop
- id_ : int
- longitude_ : double
- latitude_ : double
+ Stop(int, double, double)
+ LoadPassengers(Bus*) : int
+ AddPassengers(Passenger*) : int
+ Update() : void
+ GetId() : int <<const>>
+ Report(ostream&) : void <<const>>
next_stop_
Passenger
- name_ : string
- destination_stop_id : int
- wait_at_stop_ : int
- time_on_bus_ : int
- id_ : int
- count_ : int <<static>>
+ Passenger (int, string)
+ Update() : void
+ GetOnBus() : void
+ GetTotalWait() : int <<const>>
+ IsOnBus() : boolean <<const>>
+ GetDestination() : int <<const>>
+ Report(ostream&) : void <<const>>
PassengerGenerator
# generation_probabilities_ : list<double>
# stops_ : list<Stop*>
+ PassengerGenerator(list, list)
+ GenerateePassengers() : int
ConfigManager
-routes : vector<Route *>
+ReadConfig(string : fname): void
+GetRoutes() : vector<Route *>
RandomPassengerGenerator
+ RandomPassengerGenerator(list, list)
+ GeneratePassengers() : int
<<const>> and <<static>>
are not official UML syntax, but are
used here for added clarity
all_stops_

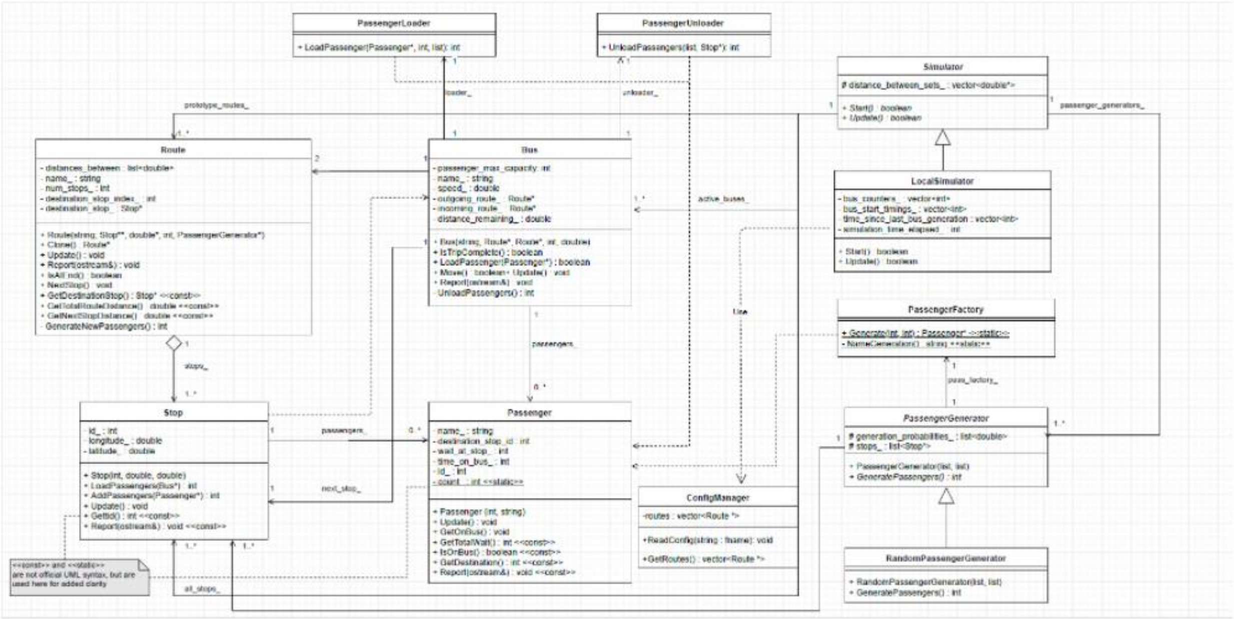
Enter answer here : .714

12. For the following class diagram, calculate the Instability value for the Stop class. Use up to 3 significant digits (e.g. .5 for 1/2 and .333 for 1/3 are acceptable).


PassengerLoader
+ LoadPassenger(Passenger*, int, list): int
PassengerUnloader
+ UnloadPassengers(list, Stop*): int
Simulator
# distance_between_sets_ : vector<double*>
+ Start() : boolean
+ Update() : boolean
prototype_routes_
loader_
unloader_
passenger_generators_
1
1..*
2
Route
- distances_between_ : list<double>
- name_ : string
- num_stops_ : int
- destination_stop_index_ : int
- destination_stop_ : Stop*
+ Route(string, Stop**, double*, int, PassengerGenerator*)
+ Clone() : Route*
+ Update() : void
+ Report(ostream&) : void
+ IsAtEnd() : boolean
+ NextStop() : void
+ GetDestinationStop() : Stop* <<const>>
+ GetTotalRouteDistance() : double <<const>>
+ GetNextStopDistance() : double <<const>>
- GenerateNewPassengers() : int
Bus
- passenger_max_capacity_ : int
- name_ : string
- speed_ : double
- outgoing_route_ : Route*
- incoming_route_ : Route*
- distance_remaining_ : double
+ Bus(string, Route*, Route*, int, double)
+ IsTripComplete() : boolean
+ LoadPassenger(Passenger*) : boolean
+ Move() : boolean
+ Update() : void
+ Report(ostream&) : void
- UnloadPassengers() : int
active_buses_
LocalSimulator
- bus_counters_ : vector<int>
- bus_start_timings_ : vector<int>
- time_since_last_bus_generation : vector<int>
- simulation_time_elapsed_ : int
+ Start() : boolean
+ Update() : boolean
Use
PassengerFactory
+ Generate(int, int) : Passenger* <<static>>
- NameGeneration() : string <<static>>
pass_factory_
stops_
passengers_
0..*
Stop
- id_ : int
- longitude_ : double
- latitude_ : double
+ Stop(int, double, double)
+ LoadPassengers(Bus*) : int
+ AddPassengers(Passenger*) : int
+ Update() : void
+ GetId() : int <<const>>
+ Report(ostream&) : void <<const>>
next_stop_
Passenger
- name_ : string
- destination_stop_id : int
- wait_at_stop_ : int
- time_on_bus_ : int
- id_ : int
- count_ : int <<static>>
+ Passenger (int, string)
+ Update() : void
+ GetOnBus() : void
+ GetTotalWait() : int <<const>>
+ IsOnBus() : boolean <<const>>
+ GetDestination() : int <<const>>
+ Report(ostream&) : void <<const>>
PassengerGenerator
# generation_probabilities_ : list<double>
# stops_ : list<Stop*>
+ PassengerGenerator(list, list)
+ GeneratePassengers() : int
ConfigManager
-routes : vector<Route *>
+ReadConfig(string : fname): void
+GetRoutes() : vector<Route *>
RandomPassengerGenerator
+ RandomPassengerGenerator(list, list)
+ GeneratePassengers() : int
<<const>> and <<static>>
are not official UML syntax, but are
used here for added clarity
all_stops_

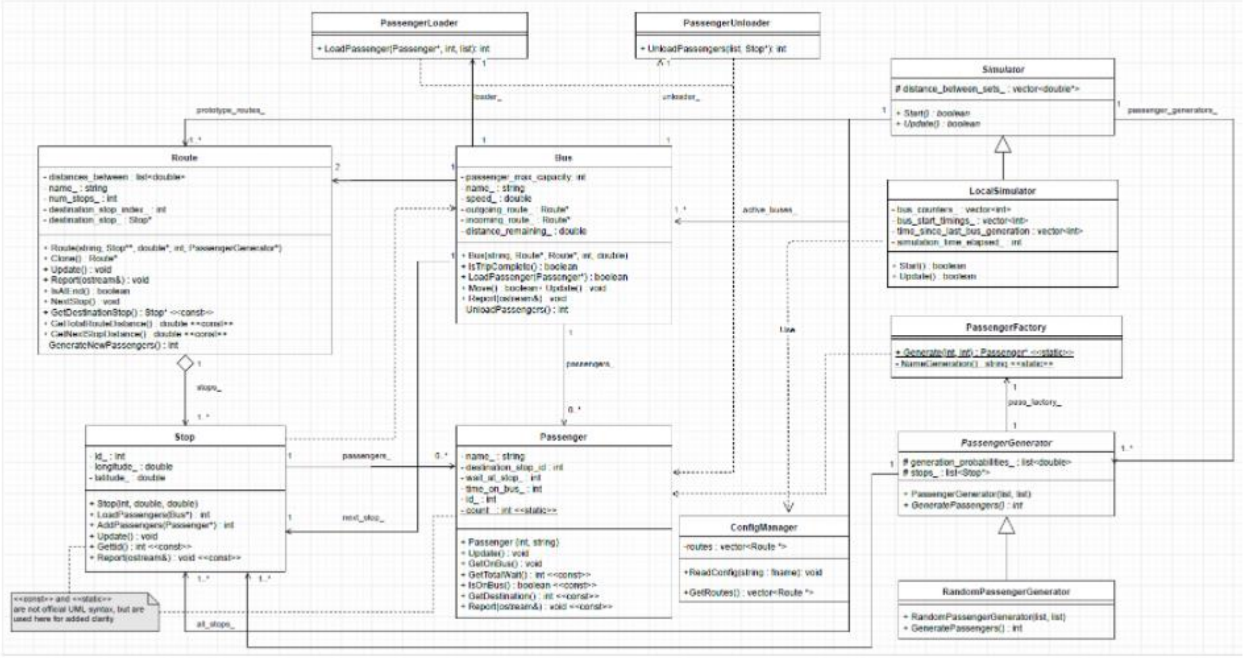
Enter answer here : .333

13.For the following class diagram, calculate the Instability value for the PassengerUnloader class. Use up to 3 significant digits (e.g. .5 for 1/2 and .333 for 1/3 are acceptable).


PassengerLoader
+ LoadPassenger(Passenger*, int, list): int
PassengerUnloader
+ UnloadPassengers(list, Stop*): int
Simulator
# distance_between_sets_ : vector<double*>
+ Start() : boolean
+ Update() : boolean
LocalSimulator
- bus_counters_ : vector<int>
- bus_start_timings_ : vector<int>
- time_since_last_bus_generation : vector<int>
- simulation_time_elapsed_ : int
+ Start() : boolean
+ Update() : boolean
PassengerFactory
+ Generate(int, int) : Passenger* <<static>>
- NameGeneration() : string <<static>>
PassengerGenerator
# generation_probabilities_ : list<double>
# stops_ : list<Stop*>
+ PassengerGenerator(list, list)
+ GeneratePassengers() : int
RandomPassengerGenerator
+ RandomPassengerGenerator(list, list)
+ GeneratePassengers() : int
ConfigManager
-routes : vector<Route *>
+ReadConfig(string : fname): void
+GetRoutes() : vector<Route *>
Route
- distances_between_ : list<double>
- name_ : string
- num_stops_ : int
- destination_stop_index_ : int
- destination_stop_ : Stop*
+ Route(string, Stop**, double*, int, PassengerGenerator*)
+ Clone() : Route*
+ Update() : void
+ Report(ostream&) : void
+ IsAtEnd() : boolean
+ NextStop() : void
+ GetDestinationStop() : Stop* <<const>>
+ GetTotalRouteDistance() : double <<const>>
+ GetNextStopDistance() : double <<const>>
- GenerateNewPassengers() : int
Bus
- passenger_max_capacity_ : int
- name_ : string
- speed_ : double
- outgoing_route_ : Route*
- incoming_route_ : Route*
- distance_remaining_ : double
+ Bus(string, Route*, Route*, int, double)
+ IsTripComplete() : boolean
+ LoadPassenger(Passenger*) : boolean
+ Move() : boolean
+ Update() : void
+ Report(ostream&) : void
- UnloadPassengers() : int
Stop
- id_ : int
- longitude_ : double
- latitude_ : double
+ Stop(int, double, double)
+ LoadPassengers(Bus*) : int
+ AddPassengers(Passenger*) : int
+ Update() : void
+ GetId() : int <<const>>
+ Report(ostream&) : void <<const>>
Passenger
- name_ : string
- destination_stop_id_ : int
- wait_at_stop_ : int
- time_on_bus_ : int
- id_ : int
- count_ : int <<static>>
+ Passenger (int, string)
+ Update() : void
+ GetOnBus() : void
+ GetTotalWait() : int <<const>>
+ IsOnBus() : boolean <<const>>
+ GetDestination() : int <<const>>
+ Report(ostream&) : void <<const>>
<<const>> and <<static>> are not official UML syntax, but are used here for added clarity
prototype_routes_
loader_
unloader_
passenger_generators_
active_buses_
Use
pass_factory_
passengers_
stops_
next_stop_
all_stops_
1
1..*
2
0..*

Enter answer here : .5

14. For the following class diagram, calculate the Instability value for the Simulator class. Use up to 3 significant digits (e.g. .5 for 1/2 and .333 for 1/3 are acceptable).


PassengerLoader
+ LoadPassenger(Passenger*, int, list): int
PassengerUnloader
+ UnloadPassengers(list, Stop*): int
Simulator
# distance_between_sets_ : vector<double*>
+ Start() : boolean
+ Update() : boolean
LocalSimulator
- bus_counters_ : vector<int>
- bus_start_timings_ : vector<int>
- time_since_last_bus_generation : vector<int>
- simulation_time_elapsed_ : int
+ Start() : boolean
+ Update() : boolean
PassengerFactory
+ Generate(int, int) : Passenger* <<static>>
- NameGeneration() : string <<static>>
PassengerGenerator
# generation_probabilities_ : list<double>
# stops_ : list<Stop*>
+ PassengerGenerator(list, list)
+ GeneratePassengers() : int
RandomPassengerGenerator
+ RandomPassengerGenerator(list, list)
+ GeneratePassengers() : int
ConfigManager
-routes : vector<Route *>
+ReadConfig(string : fname): void
+GetRoutes() : vector<Route *>
Route
- distances_between_ : list<double>
- name_ : string
- num_stops_ : int
- destination_stop_index_ : int
- destination_stop_ : Stop*
+ Route(string, Stop**, double*, int, PassengerGenerator*)
+ Clone() : Route*
+ Update() : void
+ Report(ostream&) : void
+ IsAtEnd() : boolean
+ NextStop() : void
+ GetDestinationStop() : Stop* <<const>>
+ GetTotalRouteDistance() : double <<const>>
+ GetNextStopDistance() : double <<const>>
- GenerateNewPassengers() : int
Bus
- passenger_max_capacity_ : int
- name_ : string
- speed_ : double
- outgoing_route_ : Route*
- incoming_route_ : Route*
- distance_remaining_ : double
+ Bus(string, Route*, Route*, int, double)
+ IsTripComplete() : boolean
+ LoadPassenger(Passenger*) : boolean
+ Move() : boolean
+ Update() : void
+ Report(ostream&) : void
- UnloadPassengers() : int
Stop
- id_ : int
- longitude_ : double
- latitude_ : double
+ Stop(int, double, double)
+ LoadPassengers(Bus*) : int
+ AddPassengers(Passenger*) : int
+ Update() : void
+ GetId() : int <<const>>
+ Report(ostream&) : void <<const>>
Passenger
- name_ : string
- destination_stop_id_ : int
- wait_at_stop_ : int
- time_on_bus_ : int
- id_ : int
- count_ : int <<static>>
+ Passenger (int, string)
+ Update() : void
+ GetOnBus() : void
+ GetTotalWait() : int <<const>>
+ IsOnBus() : boolean <<const>>
+ GetDestination() : int <<const>>
+ Report(ostream&) : void <<const>>
<<const>> and <<static>> are not official UML syntax, but are used here for added clarity
prototype_routes_
loader_
unloader_
passenger_generators_
active_buses_
Use
pass_factory_
passengers_
stops_
next_stop_
all_stops_
1
1..*
2
0..*

Enter answer here : 0.8

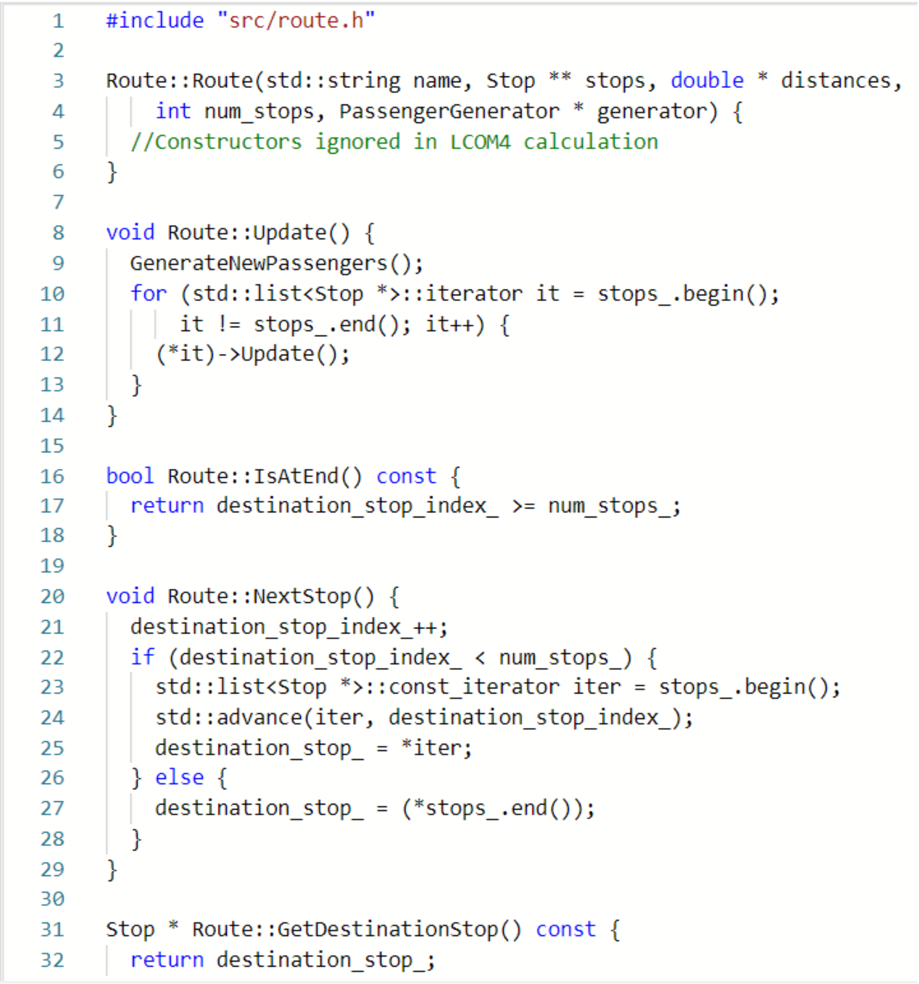
15.While calculating LCOM4, we ignore constructors and destructors. Constructors and destructors frequently set and clear all variables in the class, making all methods connected through these variables, which increases cohesion artificially.

LCOM4 = 1 indicates a cohesive class, which is the "good" class.

LCOM4 >= 2 indicates a problem. The class should be split into so many smaller classes.

LCOM4 = 0 happens when there are no methods in a class. This is also a "bad" class.

Calculate the value of the LCOM4 measurement for the following code:



```
#include "src/route.h"

Route::Route(std::string name, Stop ** stops, double * distances,
    int num_stops, PassengerGenerator * generator) {
  //Constructors ignored in LCOM4 calculation
}

void Route::Update() {
  GenerateNewPassengers();
  for (std::list<Stop *>::iterator it = stops_.begin();
      it != stops_.end(); it++) {
    (*it)->Update();
  }
}

bool Route::IsAtEnd() const {
  return destination_stop_index_ >= num_stops_;
}

void Route::NextStop() {
  destination_stop_index_++;
  if (destination_stop_index_ < num_stops_) {
    std::list<Stop *>::const_iterator iter = stops_.begin();
    std::advance(iter, destination_stop_index_);
    destination_stop_ = *iter;
  } else {
    destination_stop_ = (*stops_.end());
  }
}

Stop * Route::GetDestinationStop() const {
  return destination_stop_;
```

Enter answer here : 2

16.Cyclomatic complexity is calculated by the formula:

M = E − N + 2P,

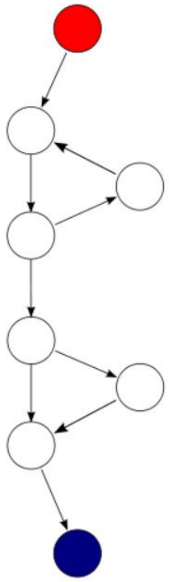
where

E = the number of edges of the graph.

N = the number of nodes of the graph.

P = the number of connected components.

Calculate the cyclomatic complexity for the following control flow graph:



Enter answer here : 3

17. Cyclomatic complexity is calculated by the formula:

M = E − N + 2P,

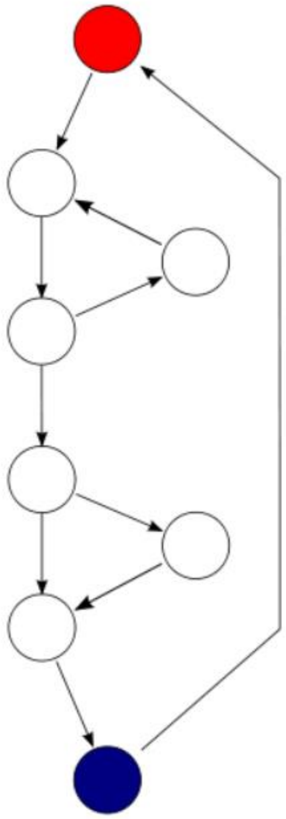
where

E = the number of edges of the graph.

N = the number of nodes of the graph.

P = the number of connected components.

Calculate the cyclomatic complexity for the following control flow graph:



Enter answer here : 4

18. Cyclomatic complexity is calculated by the formula:

M = E − N + 2P,

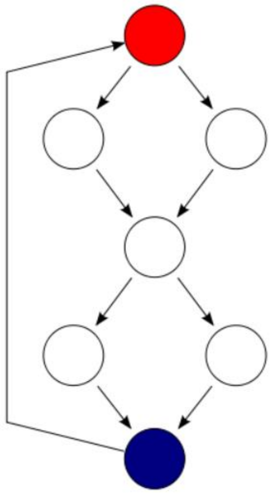
where

E = the number of edges of the graph.

N = the number of nodes of the graph.

P = the number of connected components.

Calculate the cyclomatic complexity for the following control flow graph:



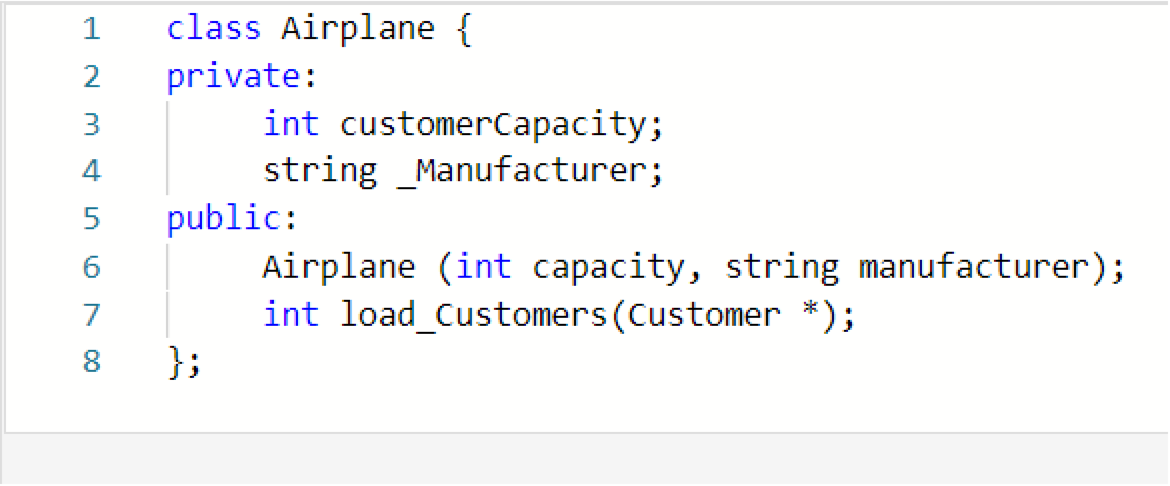
Enter answer here : 4

## Week 3

## Quality in Implementation

TOTAL POINTS 15

1.



```
1   class Airplane {
2   private:
3       int customerCapacity;
4       string _Manufacturer;
5   public:
6       Airplane (int capacity, string manufacturer);
7       int load_Customers(Customer *);
8   };
```

Which of the following are style errors for the code on Line 2?

Spacing between elements

variable should have trailing underscore

indentation

variable should have no underscore

method names should be CamelCase

variable name should be separated by underscore

order of access restrictions

class name should be lowercase

method Airplane should be private

method names should be lower case

method names should be separated by hyphens

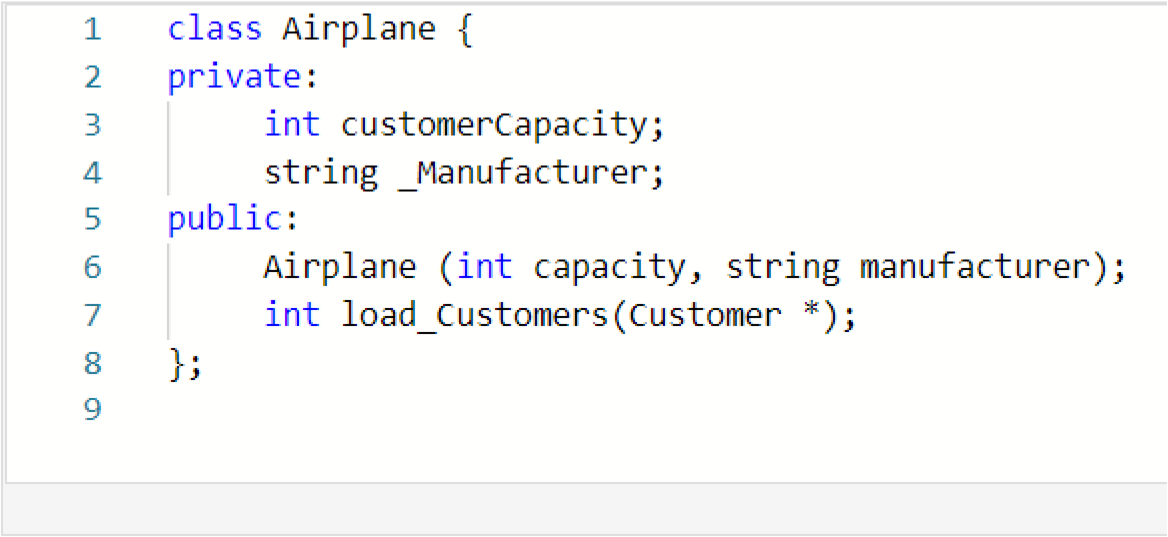
variable name should be lowercase

variables should be public

variable name should not be separated

variable name should be uppercase

2.



```
1   class Airplane {
2   private:
3       int customerCapacity;
4       string _Manufacturer;
5   public:
6       Airplane (int capacity, string manufacturer);
7       int load_Customers(Customer *);
8   };
9
```

Which of the following are style errors for the code on Line 3?

variable name should be separated by underscore

method names should be CamelCase

indentation

class name should be lowercase

variable should have no underscore

method Airplane should be private

method names should be separated by hyphens

variables should be public

order of access restrictions

method names should be lower case

variable should have trailing underscore

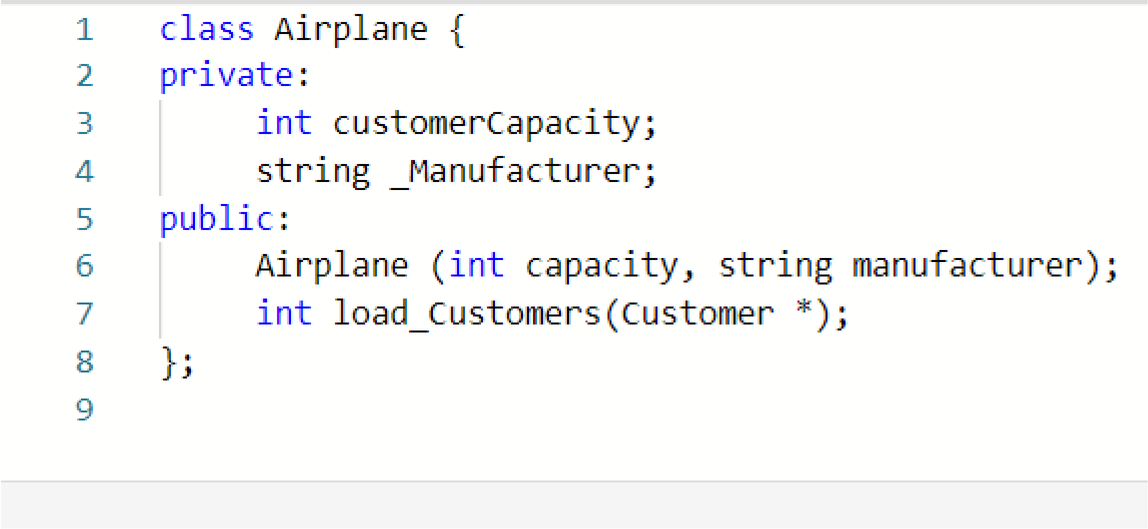
variable name should be lowercase

variable name should not be separated

variable name should be uppercase

Spacing between elements

3.



```
1   class Airplane {
2   private:
3   |     int customerCapacity;
4   |     string _Manufacturer;
5   public:
6   |     Airplane (int capacity, string manufacturer);
7   |     int load_Customers(Customer *);
8   };
9
```

Which of the following are style errors for the code on Line 4?

variable name should be uppercase

variable should have trailing underscore

method names should be CamelCase

variable should have no underscore

class name should be lowercase

method names should be separated by hyphens

method Airplane should be private

variable name should be lowercase

Spacing between elements

indentation

variable name should not be separated

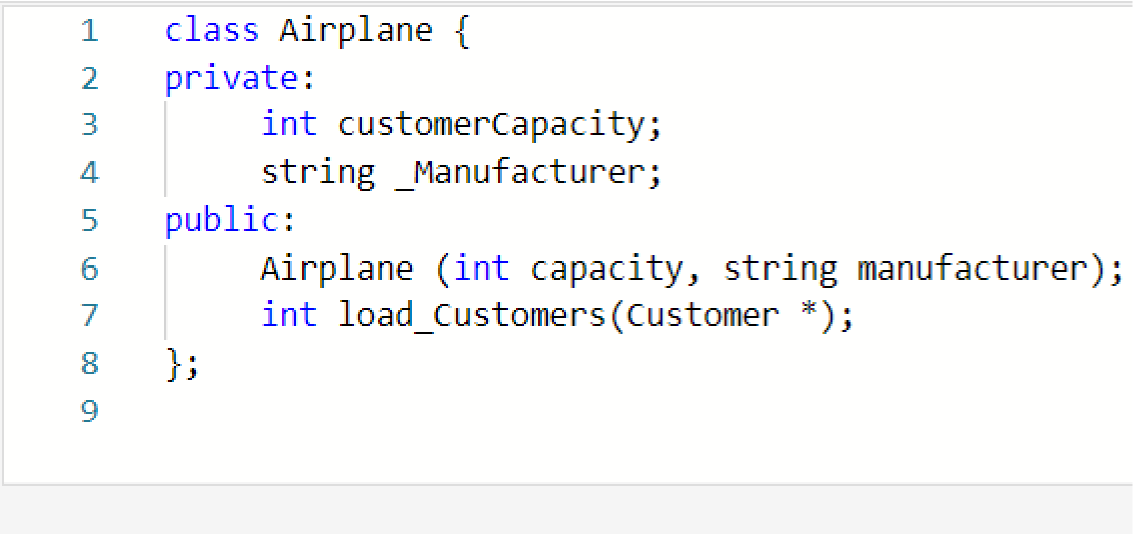
order of access restrictions

variables should be public

method names should be lower case

variable name should be separated by underscore

4.



```
1   class Airplane {
2   private:
3       int customerCapacity;
4       string _Manufacturer;
5   public:
6       Airplane (int capacity, string manufacturer);
7       int load_Customers(Customer *);
8   };
9
```

Which of the following are style errors for the code on Line 5?

variable name should be uppercase

class name should be lowercase

method names should be CamelCase

variable name should be separated by underscore method Airplane should be private

variable should have trailing underscore

indentation

method names should be lower case

Spacing between elements

variables should be public

variable name should not be separated

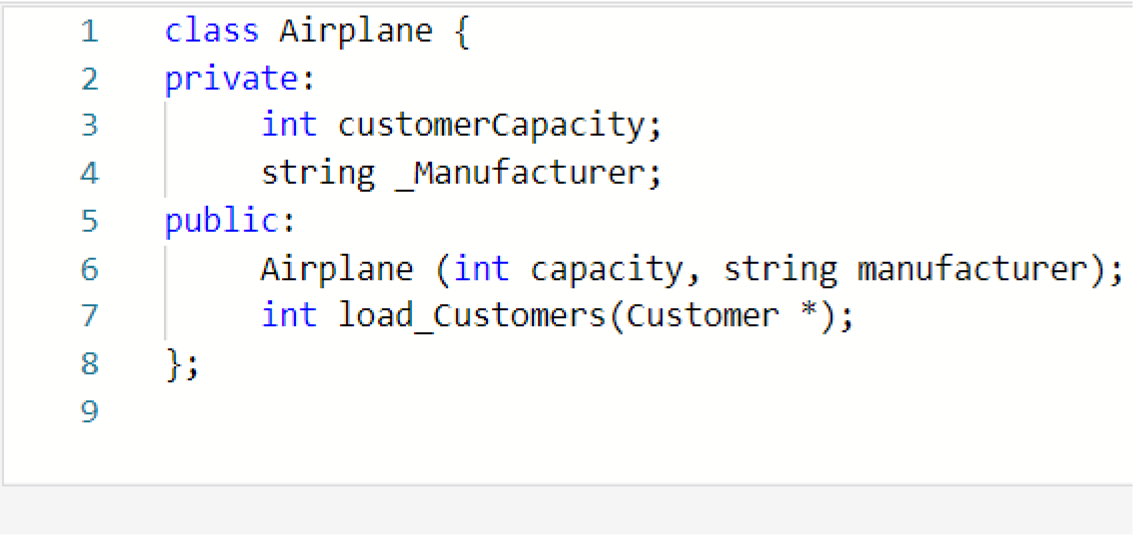
method names should be separated by hyphens

variable name should be lowercase

order of access restrictions

variable should have no underscore

5.



```
1   class Airplane {
2   private:
3         int customerCapacity;
4         string _Manufacturer;
5   public:
6         Airplane (int capacity, string manufacturer);
7         int load_Customers(Customer *);
8   };
9
```

Which of the following are style errors for the code on Line 6?

order of access restrictions

variable name should not be separated

method names should be separated by hyphens

variable name should be uppercase

method Airplane should be private

Spacing between elements

variable name should be lowercase

variable should have no underscore

method names should be CamelCase

indentation

variable should have trailing underscore

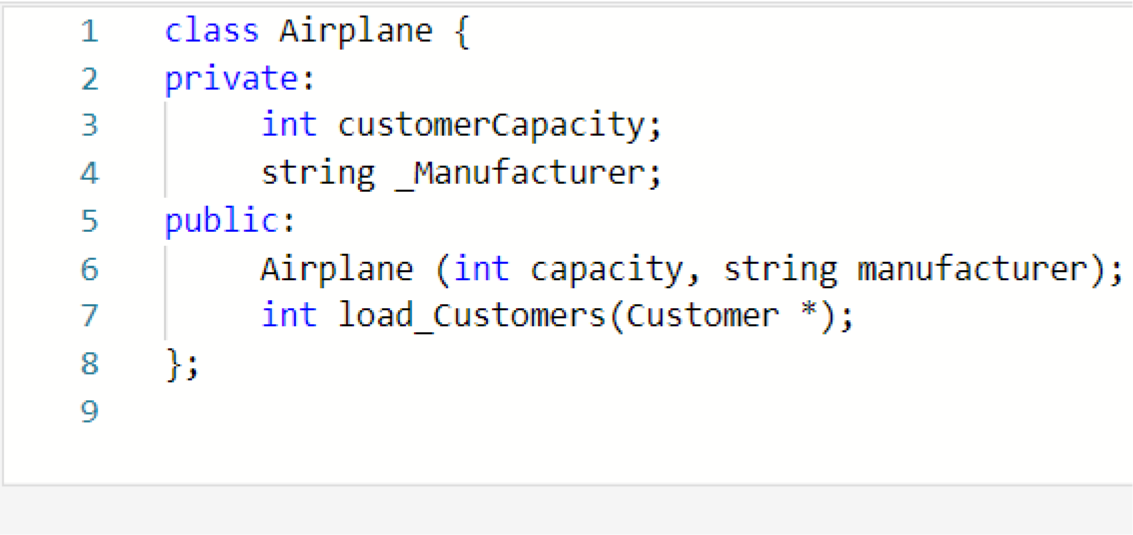
class name should be lowercase

method names should be lower case

variable name should be separated by underscore

variables should be public

6.



```
1   class Airplane {
2   private:
3       int customerCapacity;
4       string _Manufacturer;
5   public:
6       Airplane (int capacity, string manufacturer);
7       int load_Customers(Customer *);
8   };
9
```

Which of the following are style errors for the code on Line 7?

order of access restrictions

indentation

variable name should be lowercase

class name should be lowercase

method names should be CamelCase

method names should be lower case

variable should have trailing underscore

method names should be separated by hyphens

variable name should be uppercase

variable name should not be separated

variables should be public

variable name should be separated by underscore

variable should have no underscore

Spacing between elements

method Airplane should be private

7.Using a debugger can find all defects in code.

True

False

8.What is created to allow a compiler to temporarily stop processing code that is being executed to allow for the developer to see current state?

Debugging

Breakpoint

Pause

Debugging break

Stoppoint

9.Commits should happen only at the end of a daily coding session.

True

False

10.Only one line of code should change per commit.

True

False

11.Commit messages are only helpful to you at the time you make them.

True

False

12.Branching aids developers seeking to work on the same code simultaneously.

True

False

13.Project materials are stored in a remote repository through the WebHook functionality.

True

False

14.Compilers perform static analysis.

True

False

15.Static analysis can only be performed while code is being executed.

True

False

# Week 4

# Quality in Testing and Deployment

TOTAL POINTS 20

1. The goal of test selection is to find the maximum number of tests that can be successfully run.

True

False

2.Randomly testing is the poorest form of testing.

True

False

3.The developer of the code is the best person to test that code.

True

False

4. Manually created tests can use randomly selected inputs to maximize defect finding.

True

False

5.Code coverage includes statement coverage, ensuring all code statements are executed at least once by at least one test case and all tests pass.

True

False

6.Requirements testing is difficult because the tests can't be created until the code is written, which is quite far into the development process.

True

False

7.Program testing helps find defects, but testing cannot prove there are no bugs.

True

False

8.A test case is a set of inputs written to try and "break the code", i.e. find a defect.

True

False

9.Test obligations come from Structural Analysis, that is, from the code itself.

True

False

10.Tests which meet the code coverage criteria can still be poor tests.

True

False

11.New processes/reports are able to be added to the Jenkins Pipeline through plug-ins.

True

False

12.Continuous Integration alerts to submitting developer of build or test failure, but cannot remove the code committed.

True

False

13.Continuous Integration is a subset of the capabilities in a Continuous Delivery pipeline.

True

False

14.Pushing code to production without the need for developer action is one of the primary benefits of Continuous Delivery.

True

False

15.The difference between Continuous Delivery and Continuous Deployment is whether the deployment of code to production is manual or automated, respectively.

True

False

16.Canary is a system of alerts to developers based on build, test, release and/or deployment. The alerts are similar to "tweets" as on Twitter, hence the similarity of the names.

True

False

17.Since the goal is for every test to pass, tests should only include inputs which will result in successful operation/behavior when executed on correct code.

True

False

18.The developer should first run the tests before adding anything new, as in to ensure that all tests passed before adding any new tests or code.

True

False

19.Continuous Delivery is an update to the Blue-Green Deployment paradigm.

True

False

20.Statement coverage is the strongest form of code coverage, which is why it is required for many FAA and FDA regulated software projects.

True

False