

• Question 1

1.2 out of 3 points



Upper management has forbidden the use of scrum because they've heard nothing in scrum dictates the production of documentation of the internal APIs in a large project such as yours. As Scrum Master, your project manager has asked you to explain to upper management how scrum will address their concern. Which of the following arguments realistically support your agenda to use scrum?



Explain the use of the Definition of Done in a scrum project and provide an example of a Definition of Done requiring documentation of the internal APIs.



Tell your project manager to do it as explaining scrum is not part of your job

Answers: Explain that scrum does indeed require documentation for all APIs



Explain the use of the Definition of Done in a scrum project and provide an example of a Definition of Done requiring documentation of the internal APIs.

Tell your project manager to do it as explaining scrum is not part of your job

Response Feedback: Scrum does not specify exactly what (documentation, product code, unit-level test code, UML diagrams, whatever) must be produced. But there are at least two major approaches to getting anything produced: Put it in the Product Backlog or put it in the Definition of Done.

• Question 2

0 out of 5 points



After Sprint 5 of their 10 planned two-week sprints, a scrum team has completed 25 of the 100 Story Points that were initially in their Product Backlog. How many two-week sprints will be required to complete all of the remaining work, assuming that the team velocity does not change?

(Enter an integer number without decimals as your answer).

Selected Answer: 8

Correct Answer: 15

Answer range +/- 0 (15.0 - 15.0)

Response Feedback: $\#RemainingSprints = RemainingWork / Velocity$

Where

RemainingWork is the estimated incomplete work in the Product Backlog

Velocity is the average amount of work completed in each sprint

• Question 3


0 out of 4 points



Use the Williams_ESEM2011_Scrum_3_Microsoft_Teams.pdf paper, "Scrum + Engineering Practices: Experience of Three Microsoft Teams" to answer the following question. How many Lines of Production code did each engineer in Team B write on average per work day? (assume that a month has 20 work days)

Round your answer to 2 decimals.

Selected Answer:  [None Given]

Correct Answer:  13.37272727

Answer range +/- 2 (11.37272727 - 15.37272727)

Response Feedback: 3 engineers ... 8,826LOC.... 11 months
~13LOC engineer/day

• Question 4

0 out of 4 points



In Scrum Team A, every team member works 40 hours / week and their velocity after 5 sprints is 28.

In Scrum Team B, every team member works 80 hours / week and their velocity after 5 sprints is 60.


At this work rate, at the end of Scrum 15



Scrum Team B will likely have a velocity of around 60.



Scrum Team B will produce higher quality code (e.g., less defects) than Scrum Team A.

Answers: 

Scrum Team A will produce higher quality code (e.g., less defects) than Scrum Team B.

Scrum Team B will likely have a velocity of around 60.

Scrum Team B will produce higher quality code (e.g., less defects) than Scrum Team A.



Scrum Team A will likely have a velocity of around 30.

Response Feedback: Extreme Programming principle: sustainable pace. Programmer overwork generates fatigue and lack of productivity in the long run.

• Question 5

0 out of 4 points



Which of the following is a potential "pitfall" (liability) of a waterfall life cycle?

Selected Answers: 

Failure to groom the Product Backlog



Big bang integration

Answers: Failure to groom the Product Backlog



Changing customer requirements

Completed User Stories do not meet the team's Definition of Done



Big bang integration

• Question 6

4 out of 4 points



A UML State Chart Diagram:

Selected Answer: 

Documents the internal design of a class

- Answers:
- Quantifies the quality state of a project as a defect density
 - ☒ Documents the internal design of a class
 - Illustrates only the burndown state of a sprint (not the product)
 - Documents how objects in a system interact

• Question 7

0 out of 5 points



For Sprint 7, team Alpha agreed to implement the following four stories:

- #A (3 story points)
- #B (7 story points)
- #C (8 story points)
- #D (5 story points)

Towards the end of the sprint, team Alpha realizes that it cannot complete story #B before the Sprint 7 Review.

What was team Alpha's **Expected Velocity ([EV])** for Sprint 7 and what was team Alpha's **Actual Velocity ([AV])** at the end of Sprint 7?

(Enter as answers two integer values)

Specified Answer for: EV ☒ [None Given]

Specified Answer for: AV ☒ [None Given]

Correct Answers for: EV

Evaluation Method	Correct Answer	Case Sensitivity
<input checked="" type="checkbox"/> Exact Match	23	

Correct Answers for: AV

Evaluation Method	Correct Answer	Case Sensitivity
<input checked="" type="checkbox"/> Exact Match	16	

Response Feedback: Expected velocity: 23 (estimates of proposed stories: #A + #B + #C + #D)
Actual velocity: 16 (estimates of completed stories: #A + #C + #D)

• Question 8

Needs Grading



A scrum team completed 9 five-week sprints at velocities of 5, 10, 30, 28, 31, 30, 30, 31 and 30 respectively. In preparation for a big demo with important stakeholders at the end of sprint 10 (which will also have a five-week duration), during the Sprint 10 Planning meeting the product owner urges the team to implement 4 top-priority stories worth 13, 21, 13 and 8 story points.

What should the team do? Briefly justify your answer.

Selected Answer: [None Given]

Correct Answer: [None]

Response Feedback: [None Given]

• **Question 9**

0.8335 out of 5 points



UML Diagrams are useful for:



Discussing/capturing various design decisions



Estimating the progress of a waterfall project toward its completion



Documenting a finished implementation

Answers: Charting the progress of a scrum project toward its completion



Discussing/capturing various design decisions

Estimating the progress of a waterfall project toward its completion



Documenting a finished implementation



Creating a common understanding amongst team members

• **Question 10**

0 out of 5 points



For Sprint 13, team Alpha agreed to implement the following five stories:

#A (5 story points)

#B (1 story points)

#C (3 story points)

#D (8 story points)

#E (2 story points)

These five stories were broken down into 34 tasks, with a sum of the task estimates of 104. Assuming that Sprint 13 was successful, and the team was able to complete all proposed tasks and stories, what is the **Actual velocity** at the end of the sprint?

(Enter as answer an integer value, without decimals)

Selected Answer: [None Given]

Correct Answer: 19

Answer range +/- 0 (19.0 - 19.0)

Response Actual velocity: 19 (estimates of completed stories: #A + #B + #C + #D + #E). The
Feedback: task estimates are not included in the actual velocity.

• **Question 11**

0 out of 5 points



During the implementation of project X there were 30 defects/KLOC introduced. If the defect removal model included unit testing (with 40% effectiveness) and pair programming (with 10% effectiveness), how many defects/KLOC were delivered to the client?


HINTS:

Do not round your intermediate values

Round your final answer to two decimals

You can use the excel spreadsheet (Template quality plan and defect removal model) from the last sprint to compute these values

Selected Answer:  [None Given]

Correct Answer:  16.2

Answer range +/- 0.5 (15.7 - 16.7)

Response Feedback: In 30 defects/KLOC

3 defects/KLOC removed by Pair Programming

10.80 defects/KLOC removed by Unit Tests

Pair programming is performed before unit testing.

- **Question 12**

0 out of 2 points



What is the difference between a User Story and a Use Case?

Selected
Answer:



User Stories describe what the customer needs, while Use Cases determine if a story is usable by the customer

Answers:

A User Story describes what a customer needs the System to do, while a Use Case describes all the interactions between a customer and the System

User Stories are only used with a scrum process, while Use Cases are only used with a waterfall process



A User Story describes what a customer needs the System to do, while a Use Case describes a sequence of interactions between an Actor and the System

User Stories describe what the customer needs, while Use Cases determine if a story is usable by the customer

Response
Feedback:

Perhaps oversimplified: User Stories are about what the customer needs while Use Cases are mostly about what the customer will get! Stories are usually described in the literature as descriptions of what the customer needs. Use Cases are usually described as a sequence of interactions between an Actor and the System to achieve a goal. Goals may have some commonality with User Stories. For example, "As a user, I need to login to the system to recall my account data" might be a User Story but it's similar to the Goal in a "User Login" Use Case which would *also* include a sequence of interactions describing how an end-user will interact with the system to login.

- **Question 13**

Needs Grading (Extra Credit)

Briefly interpret the message communicated the following Burndown Chart for Sprint 1



Selected Answer: [None Given]

Correct Answer: [None]

Response Feedback: [None Given]

• **Question 14**

0 out of 4 points



Why do many scrum advocates advise us to create Acceptance Criteria (AC) for the User Stories prior to beginning a sprint?

Selected
Answers:



AC provide an opportunity for management to evaluate the performance of each Developer



To avoid problems arising from Big Bang integration



AC promote a common understanding of what the Team plans to build



AC promote improved estimates which leads to better Sprint Planning

Answers:

AC provide an opportunity for management to evaluate the performance of each Developer

To avoid problems arising from Big Bang integration



AC promote a common understanding of what the Team plans to build



AC promote improved estimates which leads to better Sprint Planning

• **Question 15**

4 out of 4 points



A *use case* describes:

Selected Answer:

An interaction by one or more actors with the product

Answers:



An interaction by one or more actors with the product

What the customer needs

The designs proposed by the development team to build the product

How management will use the product in the business plan

• **Question 16**

4 out of 4 points



An alternate sequence can be used to:

Selected
Answer:



Describe how a system will respond to its user's illegal input

Answers:

Describe an alternate sequence of user stories for implementing the product

Describe the *happy path*



Describe how a system will respond to its user's illegal input

Describe an alternate UML sequence diagram illustrating the system's design in more detail

• **Question 17**

2.5 out of 5 points



Agile software development processes are sometimes accused of omitting planning activities. Which of the following scrum artifacts and activities have a role in planning?

Selected Answers:



User Stories



Acceptance Criteria



Daily Scrum



Definition of Done



Sprint Planning Meeting

Answers:



User Stories



Acceptance Criteria



Daily Scrum



User Story Estimates (In Story Points)



Task Estimates



Tasks



Definition of Done



Sprint Planning Meeting



User Story Priorities



Task Owners

Response
Feedback:

They all either contribute to the plan or comprise the plan. Scrum is packed with planning, but what-is-planned-when differs from some other software life cycles.

• **Question 18**

3.75 out of 5 points



The Williams_ESEM2011_Scrum_3_Microsoft_Teams.pdf paper, "Scrum + Engineering Practices: Experience of Three Microsoft Teams," reports that the quality from one of the three examined scrum teams was significantly less than the other two. Which of the following metrics likely indicate that failure?



The **Source LoC** was 8826, well below the 24952 achieved by another team



The **Test LoC / Source LoC** ratio was 0.46, well below the 0.84 achieved by another team



The **% of Code Coverage** was 53%, well below the 82% achieved by another team

Answers: The project team failed to use the scrum practices (Sprints, Daily Scrum, Planning Poker, etc) employed by the other teams

The **Source LoC** was 8826, well below the 24952 achieved by another team

The project's team members were distributed between Redmond and China, unlike the other teams which were located locally in Redmond, Washington.



The **Test LoC / Source LoC** ratio was 0.46, well below the 0.84 achieved by another team



The **% of Code Coverage** was 53%, well below the 82% achieved by another team

Response Team B test/source LoC ratio was only 0.46 while the other teams achieved 0.84. Team
Feedback: B's test coverage was only 53% while another team achieved 82%. The best performing teams were surprisingly those distributed between Redmond and China; the failing team's members were co-located in Redmond. All teams employed scrum; the differences lie in their application of software engineering practices!

• Question 19

0 out of 5 points



Which of the following statements are true:

Selected
Answers:



It is significantly better to remove defects at the beginning of the sprint versus the end of the sprint.



It is a good practice to remove a defect as soon as the defect is discovered assuming that it is feasible to remove it.

Answers:

It is significantly better to remove defects at the beginning of the sprint versus the end of the sprint.



It is cheaper to remove a defect before integration rather than after integration.



It is a good practice to remove a defect as soon as the defect is discovered assuming that it is feasible to remove it.

It is better to remove a defect after integration rather than before integration.

• Question 20

0 out of 3 points



Choose the best workflow that any scrum team should use:

Selected
Answer:



Sprint-branch workflow

Answers:

Sprint-branch workflow

"Master" workflow (i.e., committing directly to master and created release branches when needed)

GitHub flow (using pull-requests)



A customized workflow based on the project/team needs

Story-based workflow

Task-based workflow

- **Question 21**

0 out of 5 points



After finishing 4 two-week sprints, Team Mach1 completed 80 of the 560 Story Points that were initially in their Product Backlog. Based on historical data related to velocity, how many Story Points team Mach1 is expected to complete in the fifth two-week sprint, assuming that the team does not change? (Enter an integer number without decimals as your answer).

Selected Answer: [None Given]

Correct Answer: 20

Answer range +/- 0 (20.0 - 20.0)

Response Feedback: Velocity = (InitialWork - RemainingWork) / #Sprints

which is the same as:

Velocity = CompletedWork / #Sprints

The velocity can be used to estimate how many story points will be completed in the next sprint.

- **Question 22**

Needs Grading



Enumerate at least 3 different strategies that a novice developer could utilize for identifying the location of a bug on a large (i.e., 10 millions lines of code) and unfamiliar system.

Selected Answer: [None Given]

Correct Answer: [None]

Response Feedback: [None Given]

- **Question 23**

4 out of 4 points



Match project descriptions with the most appropriate life cycle

Question	Correct Match	Selected Match
Contractual software development for a government agency in which the developer will specify exactly what the finished product will do, when it will be available and what it will cost	A. waterfall	A. waterfall

web application for a startup business that hasn't yet created revenue

✓ B.
scrum

✓ B.
scrum

Next generation version of an existing client-side application facing rapid, emerging competition

✓ B.
scrum

✓ B.
scrum

A next generation client-side product from an experienced team that knows their customer well in a stable business

✓ A.
waterfall

✓ A.
waterfall

All Answer Choices

A.
waterfall

B.
scrum

• Question 24

3 out of 3 points



Which of the following are scrum roles?

Selected Answers: ✓

Scrum Master



Developer



Product Owner

Answers:



Scrum Master

Test Engineer

Database Administrator

System Architect

Project Manager



Developer

Sponsor



Product Owner

• Question 25

0 out of 4 points



Check all the answers that are true:



Continuous Integration is the process of updating my local branch with the master branch as often as possible (i.e., local branch integrates changes from the master).



The absolute best workflow is the one where each user story is developed in its own branch.



Integration tests are useful for testing code on a shared branch.

Answers: Continuous Integration only works for projects with up to 100 developers.

Continuous Integration is the process of updating my local branch with the master branch as often as possible (i.e., local branch integrates changes from the master).

The absolute best workflow is the one where each user story is developed in its own branch.



Integration tests are useful for testing code on a shared branch.

The absolute best workflow for all projects is the GitHub workflow.



A builder is a tool that can compile code and run unit tests if needed.

• **Question 26**

3 out of 5 points



Which of the following activities occur during the Sprint Planning meeting?



Rewrite an overly complex User Story (an *epic*) as two or more simpler User Stories



Developers volunteer to implement the Tasks



Create new Tasks in the Sprint Backlog that will implement the User Stories selected for the upcoming sprint

Answers: The Scrum Master defines what process improvements are necessary in the upcoming sprint

The Scrum Master assigns Tasks to the Developers



Rewrite an overly complex User Story (an *epic*) as two or more simpler User Stories



Developers volunteer to implement the Tasks



Select items (e.g., User Stories) from the Product Backlog that can likely be implemented in the upcoming sprint

The Project Manager assigns Tasks to the Developers



Create new Tasks in the Sprint Backlog that will implement the User Stories selected for the upcoming sprint

Each team member reports what they did yesterday, what they plan to do today, and what impediments they were unable to resolve



The team creates a new Sprint Backlog for the upcoming sprint

Team uses Planning Poker to estimate tasks.

Developers write the tests required to exercise the Acceptance Criteria


Response Feedback: The Scrum Master is not a Project Manager (The Elements of Scrum: Ch6)


• **Question 27**



2 out of 4 points



A task should always be linked to a [x] to ensure [y] between artifacts
(for each variable input a single lower case word)

Specified Answer for: x  user stories

Specified Answer for: y  [None Given]

Correct Answers for: x		
Evaluation Method	Correct Answer	Case Sensitivity
 <i>Exact Match</i>	story	
Correct Answers for: y		
Evaluation Method	Correct Answer	Case Sensitivity
 <i>Exact Match</i>	traceability	

Response Feedback: A task should always be linked to a story to ensure traceability between artifacts

- **Question 28**

Needs Grading (Extra Credit)



Briefly describe some insights you gained or an “Aha” moment you had during the lectures describing how to use Information Retrieval (IR) techniques to support software maintenance tasks in order to make developers more productive.

Selected Answer: [None Given]

Correct Answer: [None]

Response Feedback: [None Given]

- **Question 29**

6 out of 6 points



Given a git repository with the following branches visualization (i.e., this is the original state of the repository):

Sourcetree representation

or Abstract (markdown) representation

What git commands should be used to transform the repository as follows (i.e., below is the expected state of the repository):

Sourcetree representation

Abstract (markdown)
or
representation

Selected Answer: 

```
$ git checkout master  
$ git merge someBranch
```

Answers:

```
$ git checkout someBranch  
$ git merge master
```



```
$ git checkout master  
$ git merge someBranch  
  
$ git checkout someBranch  
$ git rebase -i HEAD~1  
  
$ git checkout someBranch  
$ git rebase master  
  
$ git checkout master  
$ git rebase someBranch  
  
$ git checkout master  
$ git rebase -i HEAD~2 someBranch
```

`git rebase -i HEAD~3`

- **Question 30**

3 out of 3 points



Describe in a few words one of the major problems of expressing/documenting requirements using natural language (e.g., English), instead of using more formal methods.

Selected Answer: It is ambiguous to leads to different interpretations.

Correct Answer: [None]

Response Feedback: [None Given]

- **Question 31**


5 out of 5 points



Explain the difference between User Stories and Tasks

Selected Answer: 

Answer: User Stories describe what the customer needs in the customer's business language, while Tasks describe in engineering language how those needs will be implemented

Answers: 

User Stories describe what the customer needs in the customer's business language, while Tasks describe in engineering language how those needs will be implemented

A User Story describes who is responsible for implementing it while a Task describes what code changes will be required

A User Story defines a User Persona, a class of similar users in a role, while a Task each test that must pass to implement that story per the Definition of Done

A User Story defines what the customer needs while Tasks define what the Project Manager has snowplowed into a future sprint

• Question 32

2 out of 5 points



Which of the following are motivations for or characteristics of the scrum process?



Incremental development through a series of small, ever-more functional versions



Each incremental version, (also known as the *Product Increment*) is a full-tested, potentially releasable, subset of the final product

Answers:

The customers' needs change during the course of a long project

Because plans are likely to change, scrum teams don't waste time creating them



Incremental development through a series of small, ever-more functional versions

The Project Manager, as the most experienced member of the scrum team, estimates the User Stories



Customers don't always know what they need

The Project Manager assigns tasks in the Sprint Backlog to developers

The Definition of Done requires all product code to be completely written before any of it can be tested (since incomplete code is likely to change)

The Daily Standup can be skipped if team needs to catch up with their work.



The development team's initial approach to building the product does not always work



Each incremental version, (also known as the *Product Increment*) is a full-tested, potentially releasable, subset of the final product

Response
Feedback:

Most of this appeared in the lecture and/or Elements of Scrum. Skipping Standup Daily meeting is a bad practice, and is indicative of a much larger problem within the team.

• Question 33

4 out of 4 points



A UML Sequence Diagram illustrates:

Selected
Answer:



Interactions between objects

Answers:

The sequence of scrum life-cycle activities (e.g. Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective) in UML

The sequence of an object's state transitions



Interactions between objects

The sequence of a class's state transitions

• **Question 34**

3 out of 3 points



Which of the following are responsibilities of the Scrum Master?



Ensuring the scrum process is followed by the team



Removing impediments to the team's progress

Answers: Prioritizing User Stories for the upcoming sprint

Hiring new team members

Identifying what Tasks are necessary in the upcoming sprint



Ensuring the scrum process is followed by the team



Removing impediments to the team's progress

• **Question 35**

1.68333 out of 5 points



Careful scrutiny of *The Elements of Scrum* reveals that the word "design" rarely, if ever, appears in this description of the widely used scrum process. Yet design is an important aspect of software development. Which of the following are true about software design in scrum?



Scrum expects the team will design the entire product prior to the first sprint



Complex (time-consuming) design choices could be planned as Tasks in the Sprint Backlogs



Design decisions can be refactored as a product evolves in subsequent sprints

Answers:

Scrum leaves trivial (obvious) design decisions that don't require a plan to the Developer volunteering for a Task

Scrum expects the team to use the final sprint to produce the design

Scrum leaves design choices are the responsibility of the experienced Scrum Master

Scrum eliminated the need for software design because it was painful in waterfall

Scrum expects the team will design the entire product prior to the first sprint



Complex (time-consuming) design choices could be planned as Tasks in the Sprint Backlogs



Design decisions can be refactored as a product evolves in subsequent sprints

• **Question 36**

4 out of 4 points

A UML Class Diagram illustrates:

Selected



Answer:

The static relationships between classes

Answers:

The internal operation of classes



How a class transitions between its states

The static relationships between classes as well as the dynamic interactions between classes.



The static relationships between classes