## Team details

|  |  |
| --- | --- |
| Batch & Team name | MS2 & Group 8 |
| Team members | Mahender(40018074), Praveen(40017986) |
| Date | 24/09/2019 |
| Status | In-progress |

Notes:

* In this form, ‘your code’ is the code you own now – i.e., Phase-2 code
* If you link the evidence (e.g., to a file in GitHub), you don’t need to update the doc every time

## Compiler warnings and Linting

|  |  |  |
| --- | --- | --- |
|  | Assertion | Exceptions |
|  | Metric: The warning + linting count is zero |  |
|  | Sustain: The ‘treat warnings as errors’ setting is enabled |  |

#### What linting tool is used?

Sonar Lint tool is used

#### How have the warnings fared, since you took over in Phase-2?

There are few warnings when we received the code but later we resolved all the warnings.

### Evidence

<https://github.com/MahenderKadaganchi/AlertingSystemCaseStudy2/tree/master/ScreenShots>

## Static Analysis / Coding standards

#### What is the max cyclomatic complexity of a function in your code?

3

#### Where did you reduce the complexity of the code in Phase2?

We did not reduce the complexity of the code

#### What are the instances where you improved the naming for enhanced clarity / consistency?

In the PatientMonitorServerApi We have improved the naming convention of Get and Post methods.

### Evidence

<https://github.com/MahenderKadaganchi/AlertingSystemCaseStudy2/tree/master/ScreenShots>

## Duplication

|  |  |  |
| --- | --- | --- |
|  | Assertion | Exceptions |
|  | Metric: The duplication count is zero |  |
|  | Sustain: We have a mechanism to eliminate duplication |  |

#### How much duplication did you reduce in Phase2? List the instances.

No duplication is reduced as it was zero initially

### Evidence

<https://github.com/MahenderKadaganchi/AlertingSystemCaseStudy2/tree/master/ScreenShots>

## Unit testing

|  |  |  |
| --- | --- | --- |
|  | Assertion | Exceptions |
|  | Metric: We met our code-coverage target  (if you didn’t fix a target, take 75% as reference) | Unable to cover the code for the view model class |
|  | Sustain: We have a mechanism to enhance coverage and not let it slip. |  |

#### For each not-covered-code: What is the consequence of not-covering?

We did not cover the code for View model classes so the ui test is not done properly.

### Evidence

<https://github.com/MahenderKadaganchi/AlertingSystemCaseStudy2/tree/master/ScreenShots>

## Dynamic Analysis

|  |  |  |
| --- | --- | --- |
|  | Assertion | Exceptions |
|  | Metric: We have timing- and usability-related tests |  |
|  | Sustain: We have a mechanism to avoid slippage |  |

### Evidence

Attach/link: timing- and usability- tests and checks

## Functional / end-to-end tests

#### How many end-to-end tests do you execute, to verify all the functionality?

#### How many are automated?

### Evidence

Attach/link: test-scripts and/or test-spec

## Retrospection

If you had to do Phase-1 again (with the same time & scope), what would you do different?