Reliability:

http://www.radiatedigital.com/every-website-crashes-and-why-they-crash/

Maintainability

https://t4tutorials.com/software-maintainability-in-software-engineering/

Reusability

https://en.wikipedia.org/wiki/Reusability

**Non-functional Requirements**

1. **Reliability**
   1. Definition of a failure

The website crashes, which means it stops serving data.

* 1. Reasons for the website crash
     1. Code errors
        1. Reason

The developer does something wrong while doing maintenance or updating the website.

* + - 1. Precautions

the developer should make sure the website works in a correct way before publishing changes.

* + 1. Plugin/Extension error
       1. Reason

This happens when the plugin/extension added to the website is not stable enough.

* + - 1. Precaution

The developer should make sure to cache to clear as much unwanted data as possible, which decreases the processing load on the server, reducing the strain on plugins.

* + 1. Service provider error
       1. Reason

This error is caused by server troubles of the host provider.

* + - 1. Precaution

None.

* + 1. Hosting error
       1. Reason

The website goes down because there is not a large enough hosting plan.

* + - 1. Precautions

The developer should check the expiration date of the hosting plan and make sure it never expires.

* + 1. Domain error
       1. Reason

If the domain expires, the website will no longer apear online.

* + - 1. Precaution

To avoid this, the developer should check the expiration date of the domain and make sure it never expires.

* + 1. Traffic error
       1. Reason

This error happens when too many people try to visit the website at the same time.

* + - 1. Precaution

To avoid this, the developer should make use of a CDN(Content Delivery Network) to deliver static content to users much faster.

* 1. The consequences of the website crash
     1. The call center operator will not be able to view the incidents in the incident database.
     2. The call center operator will not be able to put in new incidents in the incident database.
     3. The department officers will not be able to view the status of the incidents.
     4. The department officers will not be able to check the status of the incidents.
     5. The department officers will not be able to update the status of the incidents.
     6. The department officers will not be able to send updates on social media.
     7. The crises across the country will be left unmonitored.
  2. Error Detection
     1. Key in the url of the website in <http://www.isitdownrightnow.com/> .
     2. If <http://www.isitdownrightnow.com/> says the website is down, the website has crashed and there is an error.
     3. If <http://www.isitdownrightnow.com/> says the website is up and reachable, the website is working well.
  3. The Mean Time Between Failures (MTBF) must be less than or equal to 24h.

**2. Maintainability**

Maintainability of this system shall consist of four types.

**2.1. Corrective**

2.1.0. The system must manipulate the maintenance of bugs or errors.

2.1.1. when error is detected in the software, the developer must conduct certain steps of corrective maintenance to fix it.

2.1.1.0. The user shall be able to report error to the system.

2.1.1.1. The system must be able to conduct self-detection of errors.

2.1.1.2. The system must automatically generate an error report and prompts responsible developer.

2.1.1.3. The developer must deal with the error report within 4 hours.

2.1.1.4. The developer shall be able to evaluate the condition and figure out if bug/error can be fixed within 12 hours.

2.1.1.5. If bug/error can be fixed within 12 hours, the developer shall start upon condition evaluated.

2.1.1.6. If bug/error cannot be fixed within 12 hours, the developer must write a solution plan with specific time needed to fix this bug/error.

2.1.1.7. The developer shall request help in the plan if the error/bug is beyond personal capability.

2.1.2. The developer must provide a feedback of error/bug report to user within 6 hours.

**2.2. Adaptive**

2.2.0. The system must manipulate the maintenance includes changes of operating system and hardware environment.

2.2.0.1. The user must be able to use the system on Linux, Windows and Mac OS X.

2.2.0.2. The developer must keep pace with the development of operating system and hardware.

2.2.0.3. The developer must change parts of the system to be compatible with the development of use environment.

2.2.1. The developer must keep the system fresh or to increase the lifetime of the system.

**3. Reusability**

3.1. The developer shall be capable of using existing assets in some form within the software product development process.

3.2. The developer shall be able to develop code, software components, test suites, designs and documentation in order to achieve reusability.

3.2.0. The developer must be able to organize code by writing in forms of subroutines or functions.

3.2.1. The developer must be able to regularly organize a chunk of code using modules or namespaces into layers.

3.2.2. The developer must be able to do explicit management of build, packaging, distribution, configuration, maintenance and upgrade issues.

3.3. The developer must implement components of the code to be adaptable, in small size and keep consitency.