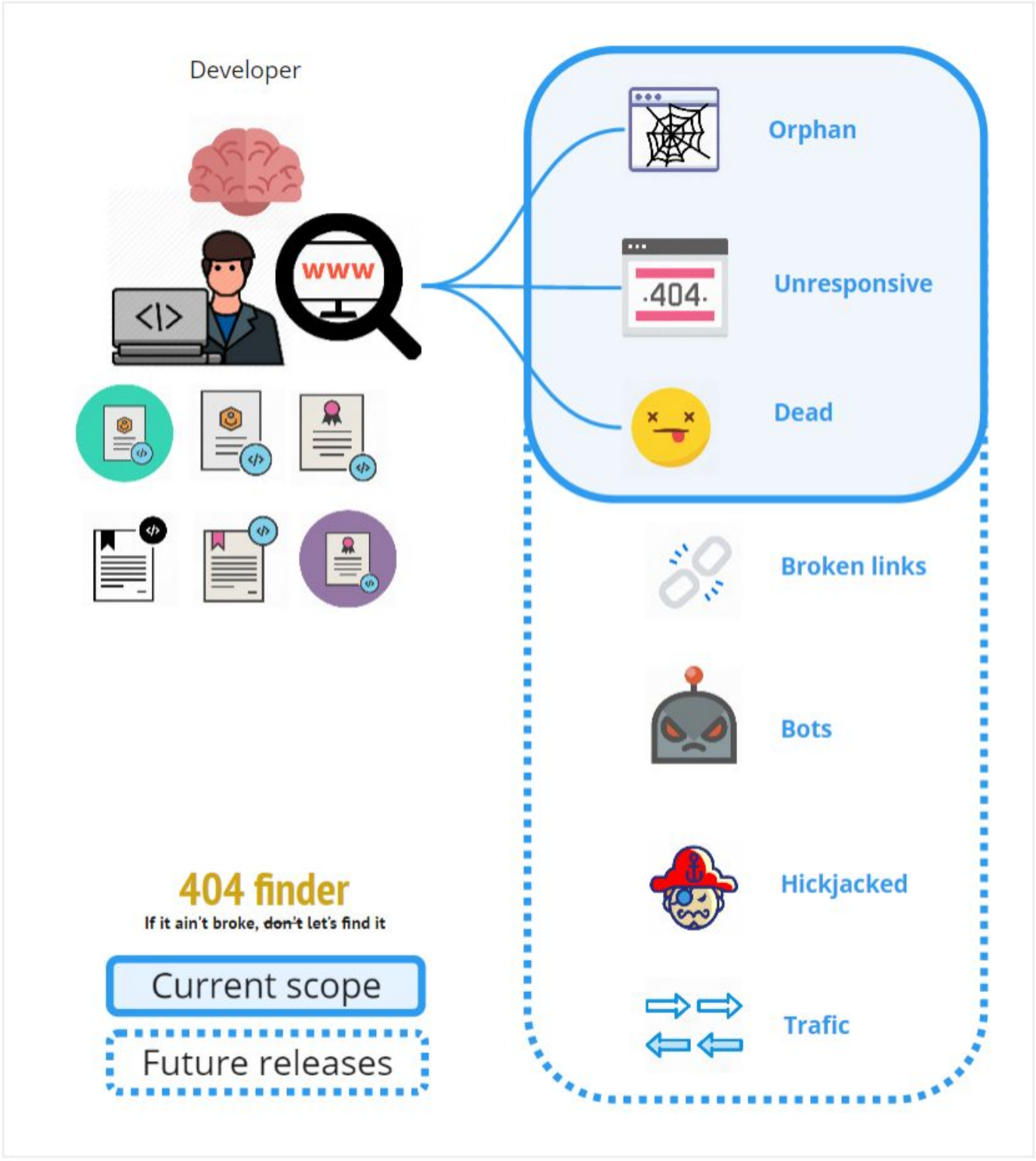
1. Project name and tagline

404 finder. If it ain't broke, ~~don't~~ let's find it



2. Team Members

1. Rodrigo Cruz CTO, CEO, ..., C{}O

3. Technologies

Libraries

2. Python: [“Request”, “json”]

Languages

3. Batch
4. Python
5. Javascript
6. HTML

Resources

7. All information provided by HS.
8. My own books collection: https://github.com/rodrigocruz13/books
9. https://www.seoptimer.com/blog/find-all-pages-on-a-website/

Other options

10. I am sure there are other software options and approaches to face a challenge like the one I am about to take, but the conditions of my house and family, my own personal motivation, financial issues, the lack of income and the fact that I was not able to find one single partner on time, leds me to try to play safe in order to try to finish this project.

4. Challenge

- **Problem:** There are developers that creates web pages for their clients. After delivering the product, in most cases, they forgot about it. So, when a web page is down, or it is having problems the developer receives an angry claim from the client asking to provide a fix.

  Why the client have to notify this situation and why the developer was not aware of it?

  ### What it is solving?

  - My Portfolio project wants to solve a problem by creating a permanent watchdog tool that checks all the web pages the developer has made and it will prompt a message when it detects a “request complication” on those sites. At this moment, those complications can be categorized as: orphan, unresponsive or dead web pages.

- The advantage of this kind of tool is the client won't notice and its perception of Quality and Satisfaction will be increased.
- This problem can be visualized in 3 dimensions:
  - Communication: Directing the communication of the problem to the responsible of solving the complication previously detected.
  - Monitoring: Empowering the developer with "vigilante" tool that will impact the customer satisfaction and reduce the client loss risk.
  - Time solution: Reducing the time for deploying a solution.

What it is not solving?

- The tool won't solve the problems detected by itself. Let's say the tool detected a "404 status code" (which means not found) on a web page. Maybe (in a future scenario) the tool can provide and generate and deploy scripts that fix those situations but in the meantime the responsibility for solving this goes to the developer.

Who is this tool helping to? - Who the users will be?

- Developers: easing the job to find out problems
- Clients: By redirecting problems to the real responsable of fix them
- Fullstack developers: Anyone that is searching for clients can use this tool to demonstrate (to a prospect client) that its website it is not working and letting him know he can provide a solution to make its client's business to grow

5. Risks.

T = Technical, NT = Non technical

Risks	Type (T / NT)	Potential impact	Alternatives
Unable to finish the project	T	Total	none
Changes in technology	T	low	The spread of new technology can be very low. Then any change can be assimilated at the same rate of time
Complexity of the technology	T	high	Changing to other tools
Technical knowledge possessed by the testing team	T	medium	Increase the knowledge in testing abilities
Conflicts within the team	T	zero	I am the team
No user feedback	NT	medium	It will lead to believe the product has no problems

6. Infrastructure.

Branching

- There will be two branches. Master and feature. Once a feature is finished and tested it will be merged into master

Describe your strategy for deployment

- Not clear yet how to do it

Describe how you will populate your app with data

- Manually

Describe what tools, automation or process you will use for testing

- Unitest

7. Existing Solutions

Similar P/S	Similarities	Differences
No similar product has been found yet		