ISMRM 27TH ANNUAL MEETING & EXHIBITION
Palais des congrès de Montréal · Montréal, QC, Canada · 11–16 May 2019

Secret Session: Master coding in your research environment

Code Documentation

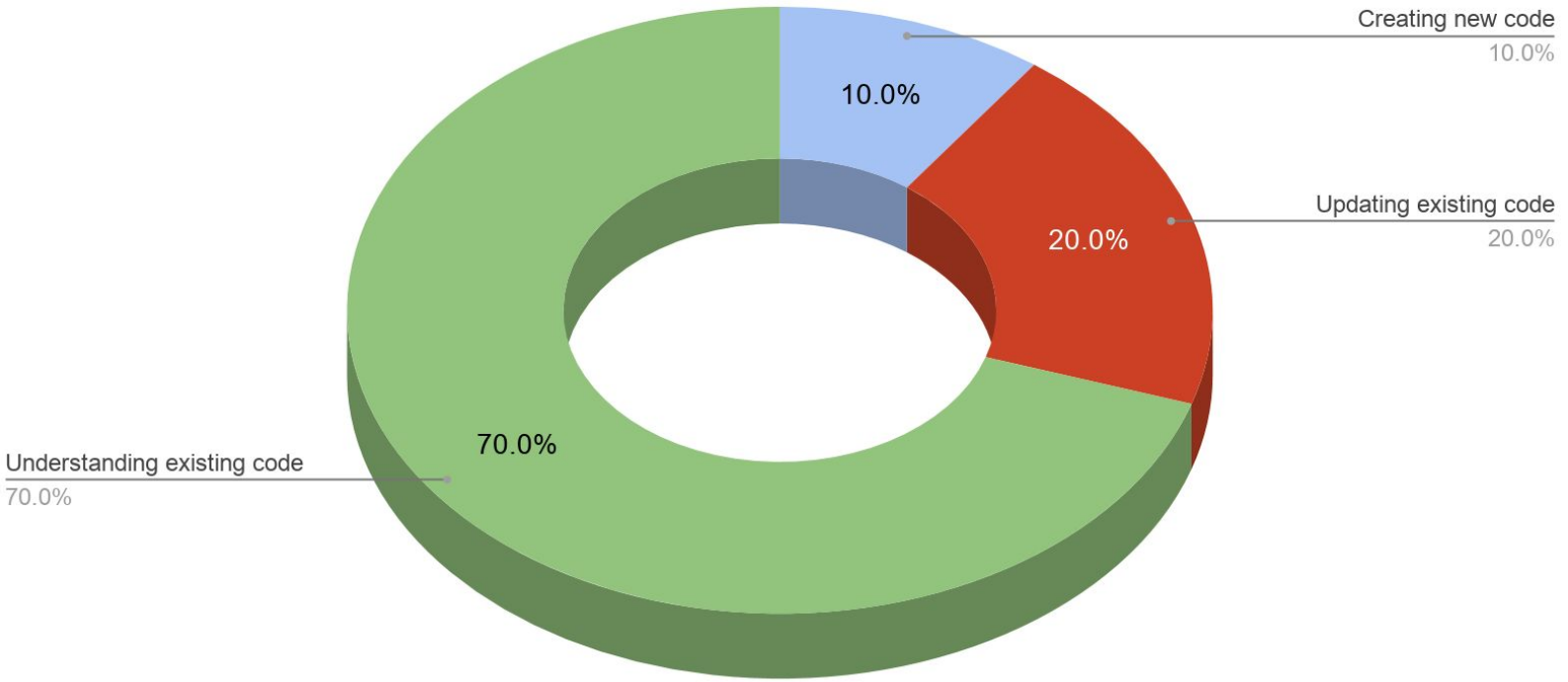
Serge Koudoro

Indiana University

Software Engineer for DIPY @dipymri, and FURY

“DONE IS BETTER THAN PERFECT”

Time Spent while Coding



> **We like to**
> **typing, k**
>
> -- Doug

Indeed, the ratio of time spent reading versus writing is well over 10 to 1....

-- Robert C. Martins

Why Documenting?

1. More efficient in understanding existing code
2. More sustainable code
3. More educative code for your whole community (Developers & Users)

Example of BAD CODE ?

```
def pix_collection(items):
        res = []
        limit = 10
        for i in items:
            if (i[0]**2 + i[1]**2) < limit**2:
                res.append(i)
        return res
```

#1 Meaningful Variables' Names

```
def pix_collection(items):
    res = []
    limit = 10
    for i in items:
        if (i[0]**2 + i[1]**2) < limit**2:
            res.append(i)
    return res
```

```
def pix_collection(pixel_list):
    selected_pixel = []
    radius = 10
    for pixel in pixel_list:
        if (pixel[0]**2 + pixel [1]**2) < radius **2:
            selected_pixel.append(pixel)
    return selected_pixel
```

#2 Self-Explanatory Code

```
def pix_collection(pixel_list):
    selected_pixel = []
    radius = 10
    for pixel in pixel_list:
        if (pixel[0]**2 + pixel [1]**2) < radius **2:
            selected_pixel.append(pixel)
    return selected_pixel
```

```
def is_in_circle(x_coord, y_coord, radius):
        return (x_coord**2 + y_coord**2) < limit**2


def get_neighboorhood_pixel(pixel_list, radius=10):
        selected_pixel = []
        for pixel in pixel_list:
            if is_in_circle(pixel.x, pixel.y, radius):
                selected_pixel.append(pixel)
        return selected_pixel
```

#3 Comments

- Hide issues related to **bad design**
- Result in **more maintenance** and are **often obsolete**

When to use comments?

- Good comments are **hard to write**
- Needs to explain context / assumptions
- Tricky Optimization (memory / speed)

Additional Tips

1. Remember to revise your code
2. Write tests
3. Use Docstring

```
def optimize(self, static, moving, static_grid2world=None,        Omar Ocegued
             moving_grid2world=None, prealign=None):
    r"""
    Starts the optimization

    Parameters
    ----------
    static : array, shape (S, R, C) or (R, C)
        the image to be used as reference during optimization. The
        displacement fields will have the same discretization as the static
        image.
    moving : array, shape (S, R, C) or (R, C)
        the image to be used as "moving" during optimization. Since the
        deformation fields' discretization is the same as the static image,
        it is necessary to pre-align the moving image to ensure its domain
        lies inside the domain of the deformation fields. This is assumed
        to be accomplished by "pre-aligning" the moving image towards the
        static using an affine transformation given by the 'prealign'
        matrix
    static_grid2world : array, shape (dim+1, dim+1)
        the voxel-to-space transformation associated to the static image
    moving_grid2world : array, shape (dim+1, dim+1)
        the voxel-to-space transformation associated to the moving image
    prealign : array, shape (dim+1, dim+1)
        the affine transformation (operating on the physical space)
        pre-aligning the moving image towards the static
```

Always Add Documentation...

For Contributors

- API Specifications
- README
- Contribution Guide
- FAQ
- Design Documentation
- LICENSE

For Users

- Website
- Examples / Applications
- Tutorials
- Theory

Sphinx, Doxygen

Conclusions:

- Self-Explanatory Code
- Sufficient documentation and guidance
- Documentation should be a standard practice

Documenting is essential

References

- Putting comments in code: the good, the bad, and the ugly by Bill Sourour
- ITT 2016 Seven Ineffective Coding Habits of Many Programmers by Kevlin Henney
- A Survey of Improving Computer Program Readability to Aid Modification
- What is the difference between documentation and commenting
- 10 Tips For Clean Code by Michael Toppa