

# Documentation is communication.

## Tips to make documentation more successful.

O. Merkulova, I. Khokhriakov, J. Hakanpää, A. Barty, DESY, Hamburg, Germany

It is possible!

- To make the process of documentation creation be
  - an understandable and
  - a standard one
- To make it a part (small part) of your daily routine
- To make it easy

Often *what people say and what they mean may be two different things* → make it possible for others to express their views and ideas and being understood.

- Make documentation a project, i.e. Docs-as-Code.
- Moreover, documentation itself can be a product!

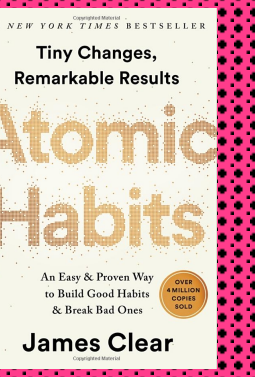
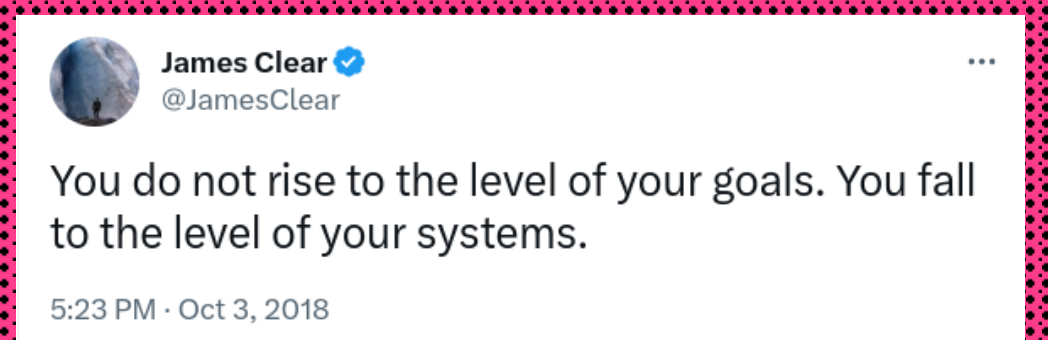
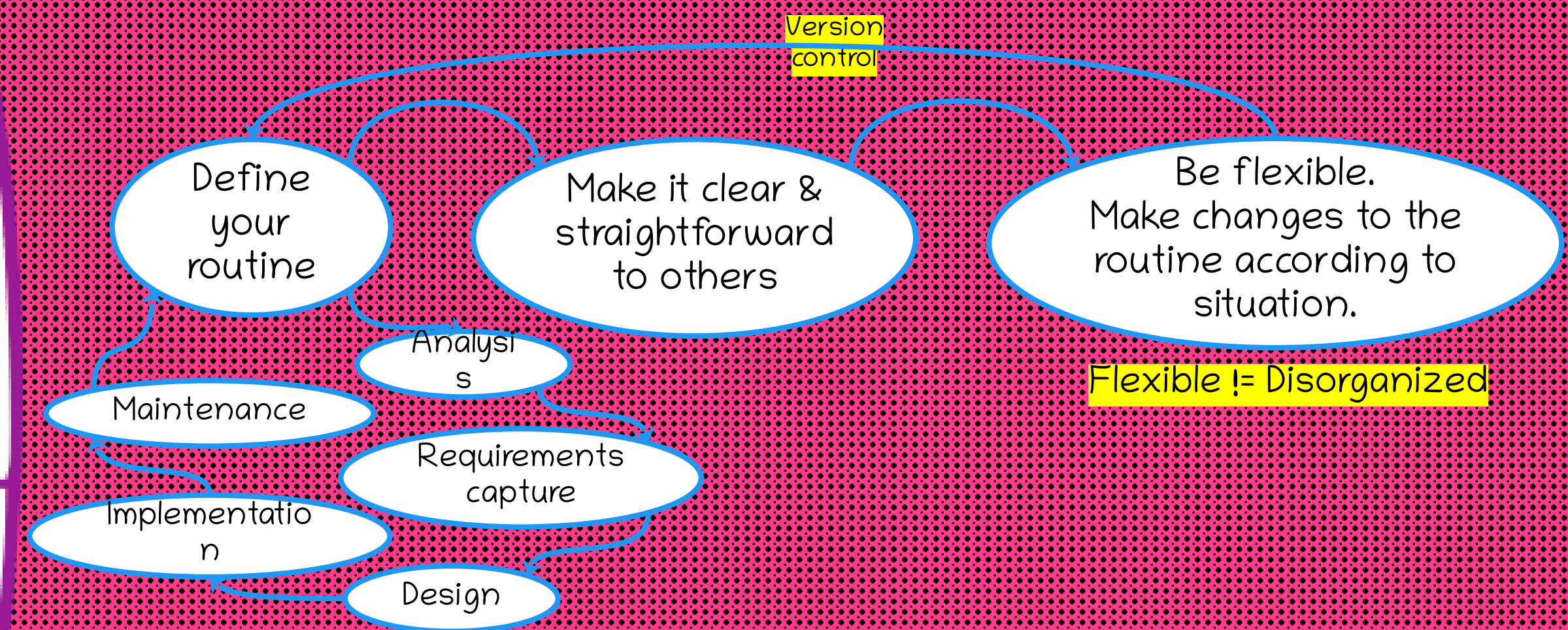
### Documentation

#### Organize routine

#### Organize routine

Provide transparency and transfer knowledge by making *process* and *procedures* understandable.

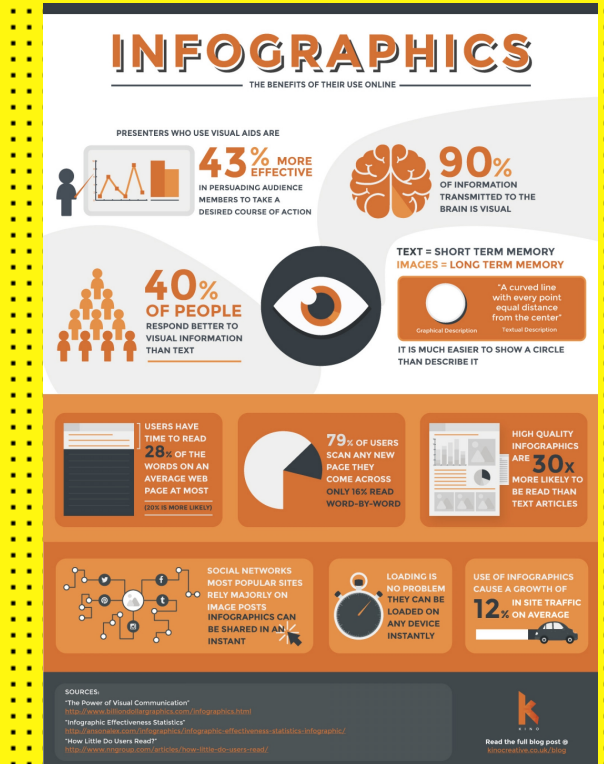
Make documentation be one of your *processes*.



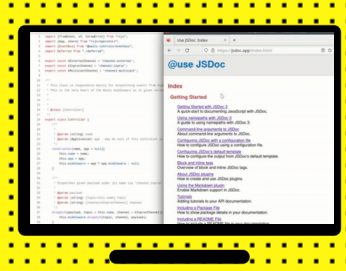
#### Readability

### Readability

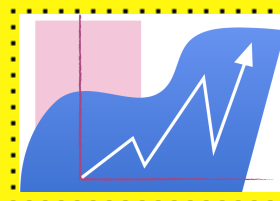
#### Infographics



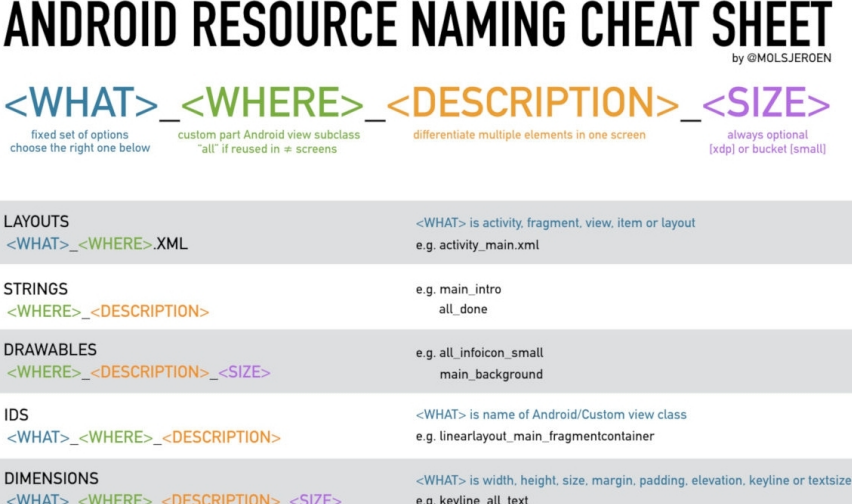
#### Videos



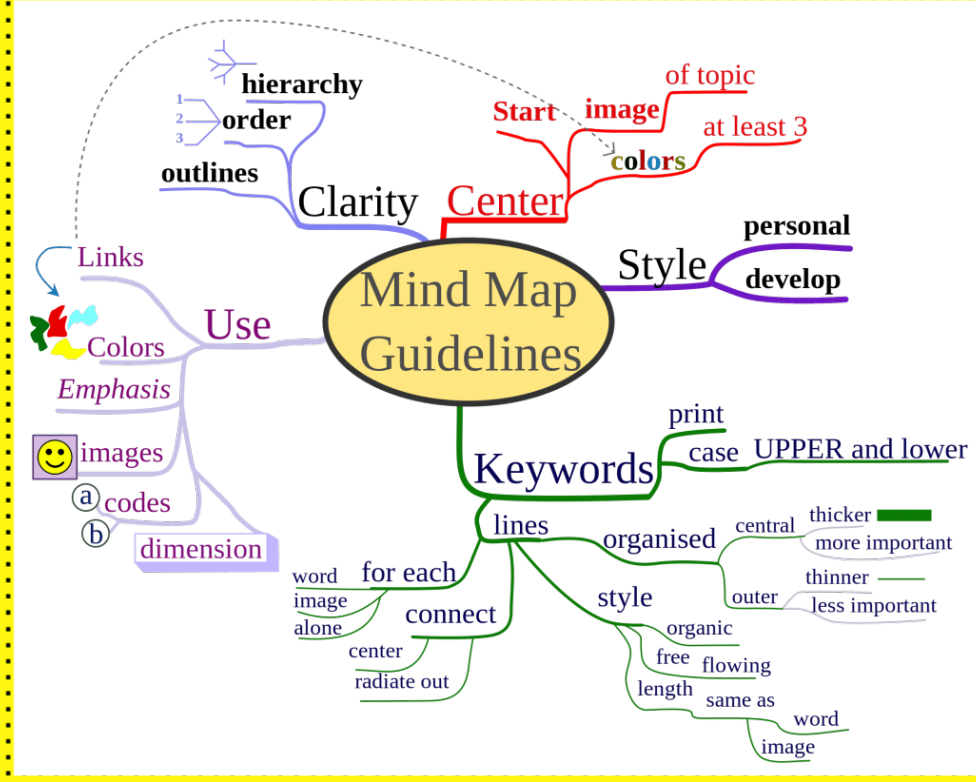
#### Graphs



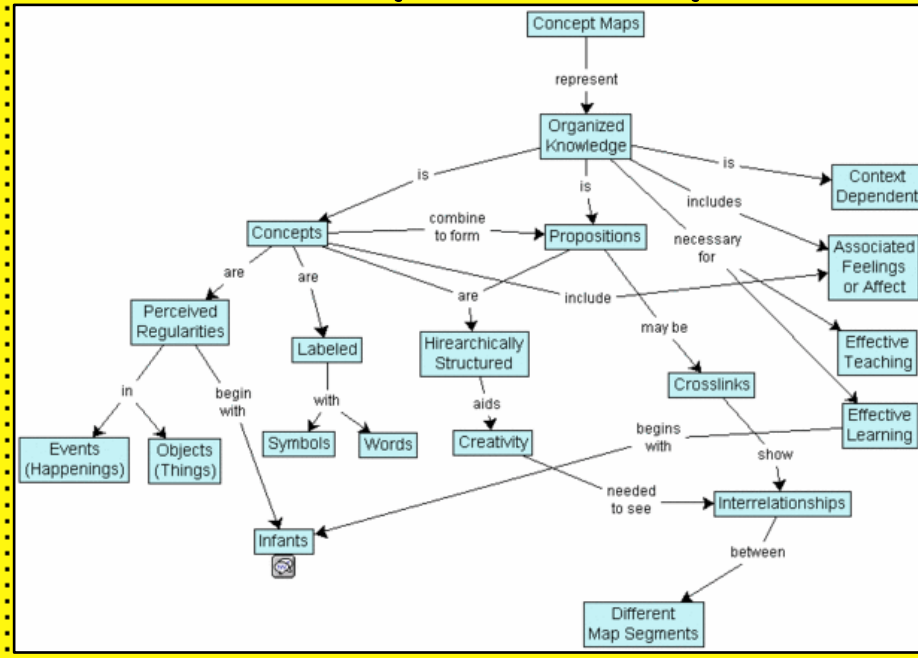
#### Code conventions



#### Mind maps



#### Concept maps



#### Text

Good in describing diagrams or case scenarios but in general not the best solution nowadays. Are you still here, reading?!

### Testing along with engineering

Tests are not only about how the code should behave but also about *how to use* API, functions, methods, etc.

They are examples of how to use certain functions, show base and edge case scenarios.

There is even a test driven development practice that focuses on creating test cases – step by step *scenarios of what and how* should be tested – before code development.

#### Test + Version control

### Version control as doc tool

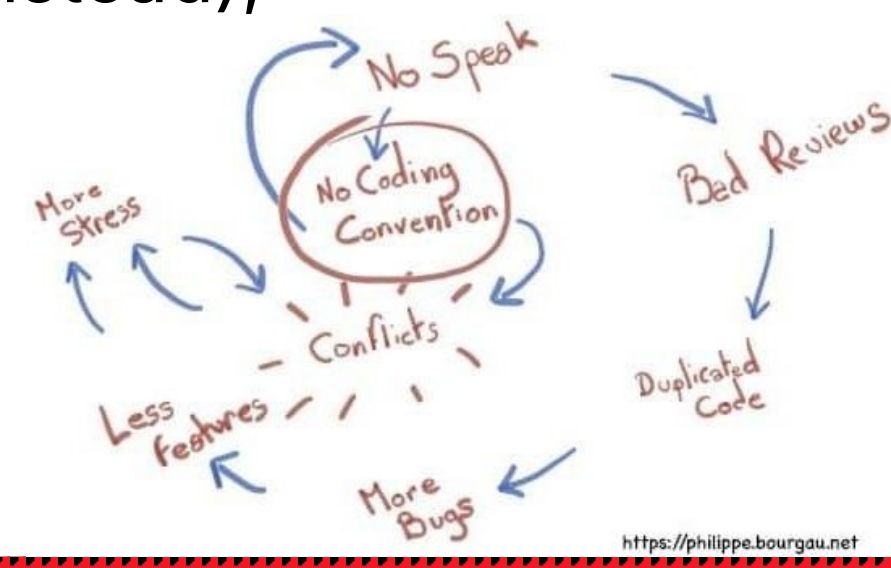
It helps to track the logic of code changes.

Make sure *comments* during commits explain only WHY, not WHAT.

### Code itself should make sense

Create *common coding standards* & make a self-documented code process by thinking about:

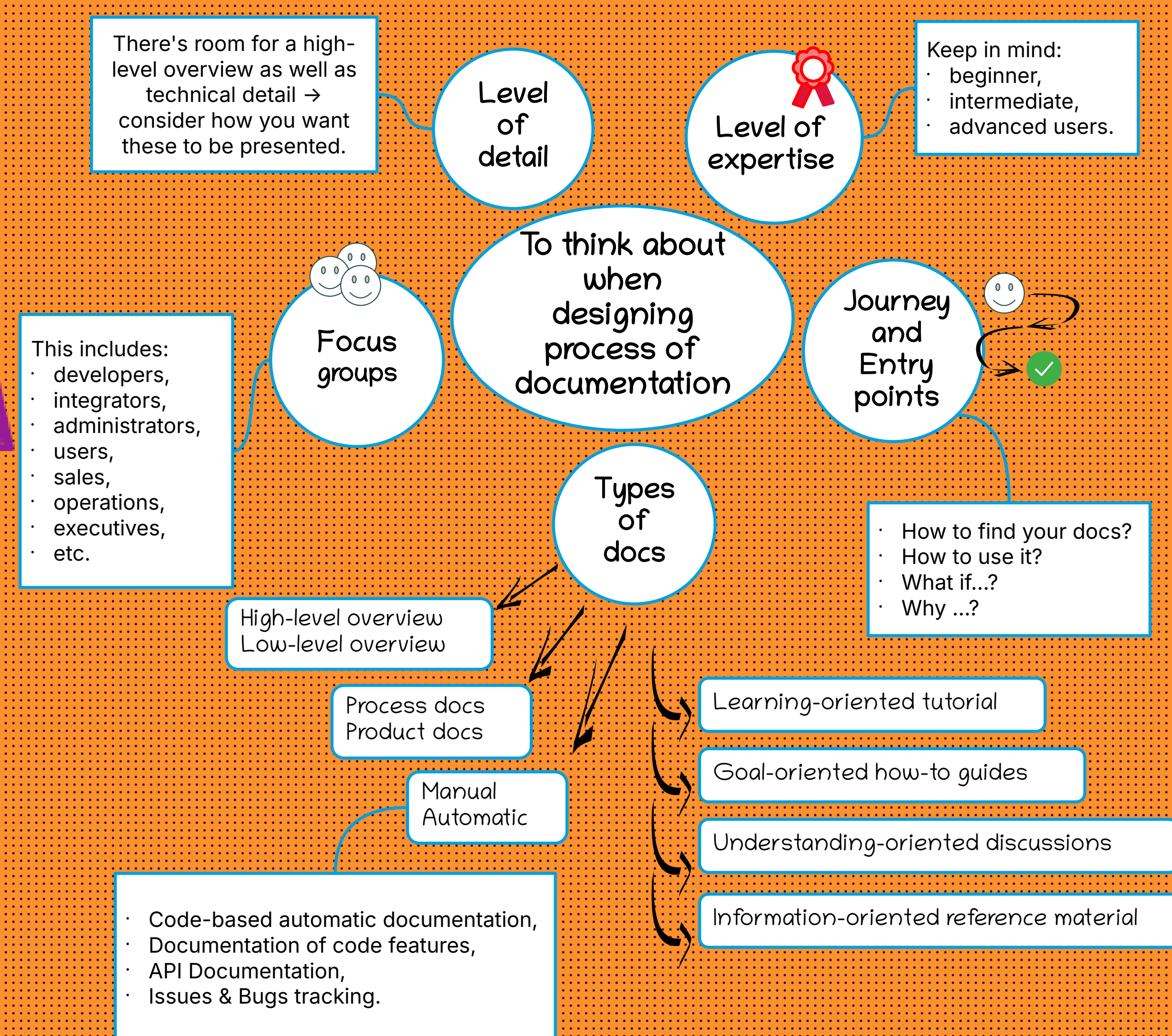
- Use meaningful directory structures, modules, packages, files,
- Create naming schemes (variable, class, function naming conventions),
- Make names understandable of what they are about,
- Avoid magic numbers (use const instead),
- Avoid deep nesting,
- NO to copy-and-pasting the code,
- NO-NO to long methods,
- Extract methods, variables, etc.



#### Code Style & Conventions

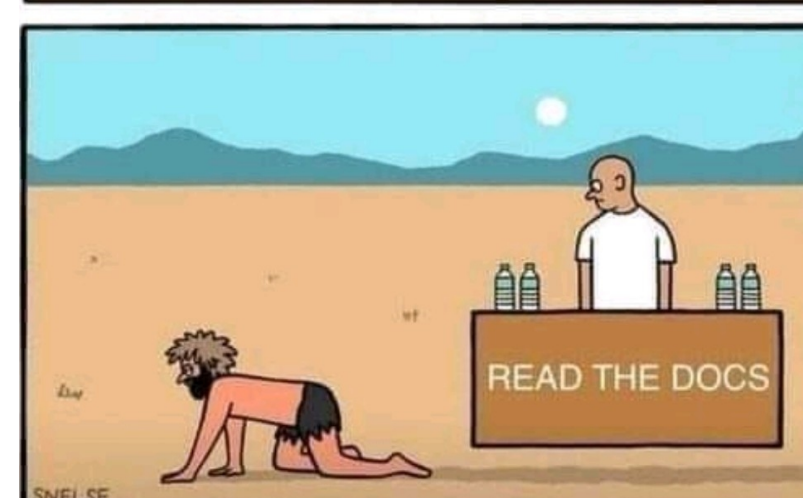
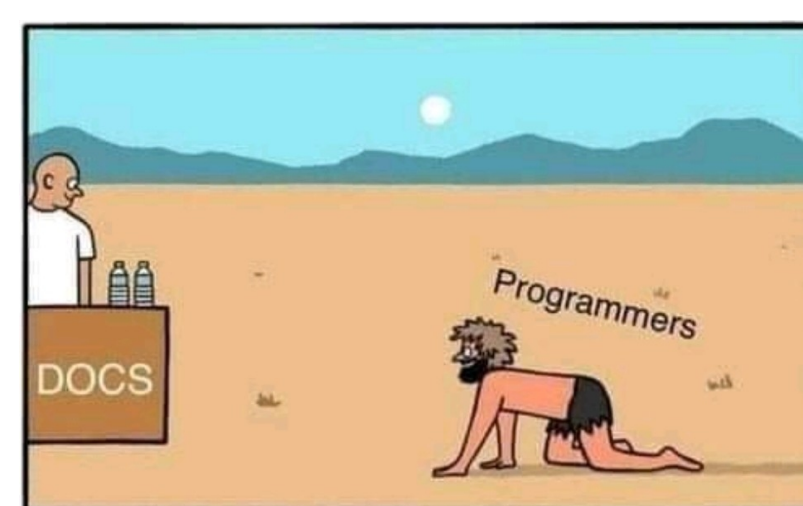
#### Design the process

### Designing the process



### Why to document?

- Documentation still remains the best communication tool for groups and projects. Especially with regard that all projects tend to get bigger with time.
- Documentation may also be a single source of truth within a group or a company. Think about remote work which is quite popular nowadays.
- "it is obvious for me → it is obvious for everyone"
  - Documentation helps to avoid this trap by simply looking onto things from different perspectives.
- Documentation is an important part of a community builder.



Documentation is not only about documentation!

"The beginning of wisdom is the definition of terms." Socrates.

