

# Best Practices

## Project Startup

- Name. Think twice. Sleep Over. Think twice again.
- Confluence Space, to document what is or shall be
- Jira Project, to plan and track actual progress
- Jira referencing Confluence is better than vice-versa
- Plan and Document Architecture, Features, Roadmap ... in Confluence
- Create a Jira issue for each and every task, beginning with an Epic as top-level task

## Documentation principles

- Document intentions before you plan your architecture or implement anything
- Document architecture before you implement
- Create a Jira task before you implement
- Write the interfaces before the implementation
- Commit small, but ...
  - in the commit comment, describe what you have done and why you have done it (issue numbers help)
  - try not to break builds. If you break a build, warn every possible co-developer
- Document classes and methods in the interface, not in the implementation
- When resolving Jira issues, document the svn revision numbers in the resolution comment
- Before closing Jira issues:
  - test the solution and write an according comment
  - compare the result with the intentions, document differences
  - if necessary don't hesitate to create follow-up issues for unresolved details or new ideas
- Document successful mechanisms or strategies not only in the project space but also as Best Practice space and/or technology space
- After creating or extending a library, write a HowTo that allows other developers to pick it up without studying all your code. Test cases and HelloWorld programs often are good source examples, if others are able to understand and run them out of the box.

## Error messages usefulness

- Try hard to include details in error messages, i.e. point to the content (FIXME bad and good examples)

## Code style

- Spaces are better than tabs, because only spaces have an exact width stable over all the editors out there

TBC