

# Dokumentation

## Begriffe

### «Dokumentation»

[Das Festhalten von] Informationen, die es erleichtern,  
Software zu benutzen, zu debuggen oder zu modifizieren

### «Benutzer-Dokumentation»

Richtet sich an jene, die die Software  
nur benutzen möchten

### «Entwickler-Dokumentation»

Richtet sich an jene, die die Software  
weiterentwickeln möchten

## Sieben Prinzipien guter Dokumentation

1. **Schreibe aus der Perspektive der Leserschaft.** Versuche, die Dokumentation auch für Laien verständlich und spannend zu halten.
2. **Vermeide unnötige Wiederholungen.** Das dient nicht nur der Leserschaft, sondern auch der Wartbarkeit der Dokumentation.
3. **Vermeide Mehrdeutigkeiten.** Überlege dir, welche Anweisungen falsch verstanden werden können.
4. **Halte dich an eine klare Gliederung.** Das hilft beim Suchen nach einer schnellen Antwort.
5. **Dokumentiere deine Entscheidungen.** Behandle nicht nur das *Wie*, sondern auch das *Warum*.
6. **Halte die Dokumentation auf aktuellem (aber nicht aktuellstem) Stand.** Vollständigkeit und Effizienz der Dokumentation sollten sich die Waage halten.
7. **Teste die Dokumentation.** Lasse sie von anderen gelesen.

Nach: Paul Clements et al., *Documenting Software Architectures*. Addison-Wesley 2011. S. 36–45.

## Wo dokumentieren?

### GitHub oder Bitbucket

- README.md wird auf der Projektseite dargestellt
- Zusätzlich hosten GitHub und Bitbucket ein projekteigenes Wiki
- Wiki wird in separatem Repository verwaltet

### Wikis

- Buch-Generierung nur mit Zusatzaufwand
- Von Repository getrennte Versionsverwaltung
- Darstellung verschiedener Versionen schwierig

### Read the Docs

- Hostet Dokumentation in grafisch aufbereiteter Form auf <projektname>.readthedocs.org
- Markdown und reStructuredText
- Versionsverwaltungs-Systeme: Subversion, Bazaar, Git, Mercurial

### Q&A-Seiten

- Stack Exchange gewinnorientiert, Content aber offen lizenziert
- Askbot als Open-Source-Alternative
- Gute Sichtbarkeit im Web

## Literatur

- Kapitel «User Support» in: A. Tucker et al.: *Software Development. An Open Source Approach*. CRC Press 2011. S. 243–264.
- P. Clements et al.: *Documenting Software Architectures. Views and Beyond*. Addison-Wesley 2011.
- B. Dagenais & M. Robillard: «Creating and Evolving Developer Documentation. Understanding the Decisions of Open Source Contributors.» In: *Proceedings of the Eighteenth ACM SIGSOFT International Symposium on Foundations of Software Engineering*. S. 127–136. ACM 2010.
- Dokumentation von Read the Docs ([read-the-docs.rtfld.org](http://read-the-docs.rtfld.org)).