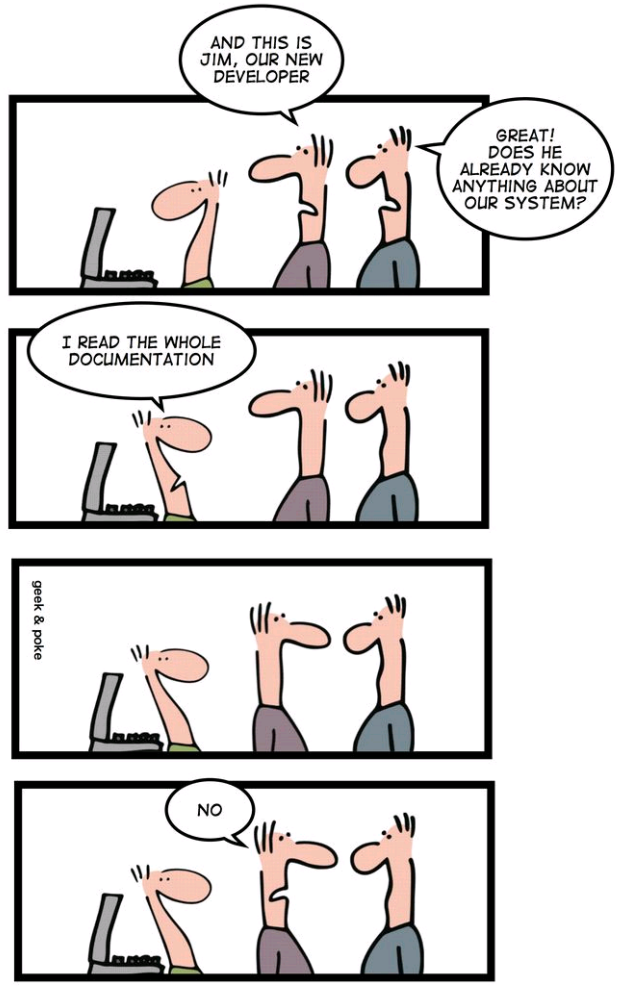
Dokumentation

HAW Hamburg / Fachbereich Informatik

Tim Lüecke

(Tim.Lueecke@haw-hamburg.de)



Fragen eines neuen Mitarbeiters…

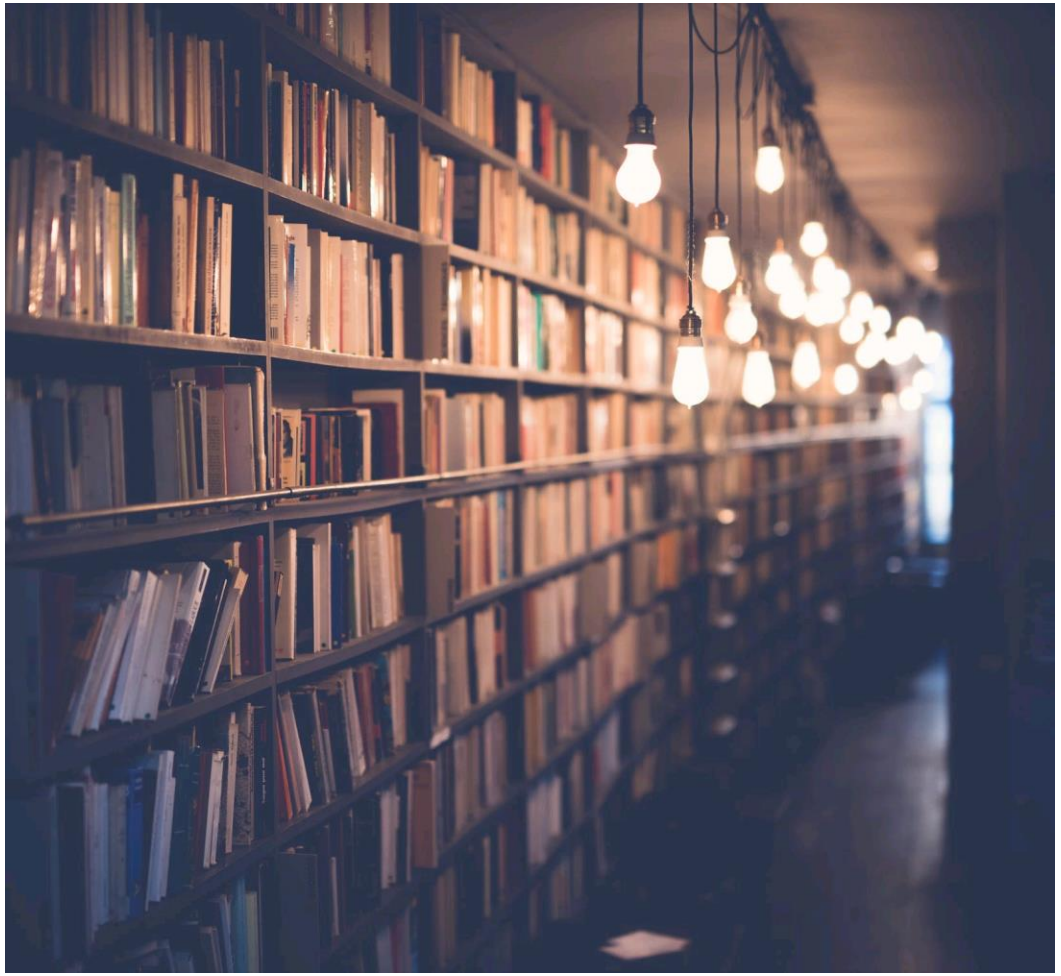
- Wie checke ich die Sourcen aus, und wie baue ich die Software?
- Was brauche ich für Tools? Wie richte ich sie ein?
- Was leistet das System überhaupt? Was sind die Features?
- Wie bediene ich die Software?
- Wie arbeiten wir mit GIT?
- Wenn ich neue Funktionalität hinzufügen soll – wie stelle ich das an? Hier ist doch schon was Ähnliches, kann ich das wiederverwenden?
- Aus welchen Bestandteilen besteht die Software? Was ist die Architektur?
- Warum benutzt ihr noch JDK 7?
- Wieso habt Ihr das denn so gemacht?
- …

… und die Antworten, die sie erhalten

- „Steht alles im Wiki.“
- „Das haben wir nicht dokumentiert – wir gehen agil vor.“
- „Das war schon so, als ich neu war.“
- „Das ist historisch gewachsen.“
- „Die Wahrheit steht im Code.“
- “Das lernt man mit der Zeit.”

Dokumentation

- Die Dokumentation gilt als ewiges **Sorgenkind** der Software-Entwicklung.
- Dokumentieren ist eine **Daueraufgabe**.
- **Nachträgliche Dokumentation** ist eine unzureichende Notlösung, denn die Information, die es aufzuzeichnen gilt, ist vergessen oder – mit der Person, die diese Information im Kopf hatte – verschwunden.
- Dokumentation tritt demnach im Software-Lebenslauf **nicht als eigene Tätigkeit** auf.

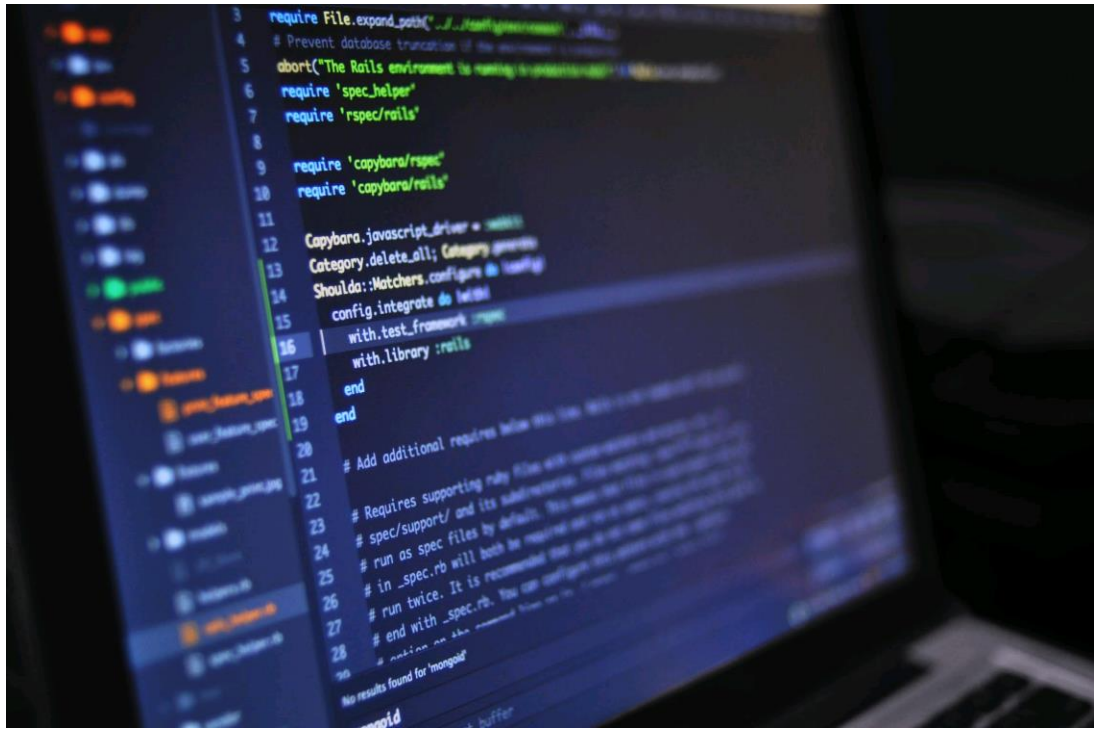


Begriffe

- **Integrierte Dokumentation**: Im Programm enthaltene Kommentare, Bezeichner, das Layout.
- **Separate Dokumentation**: Der Teil der Software, der nicht in den Programmen enthalten ist.
- Die Form der Dokumente ist damit nicht festgelegt, einzig Stabilität ist unbedingt gefordert: **Dokumentation im Kopf gibt es nicht!**

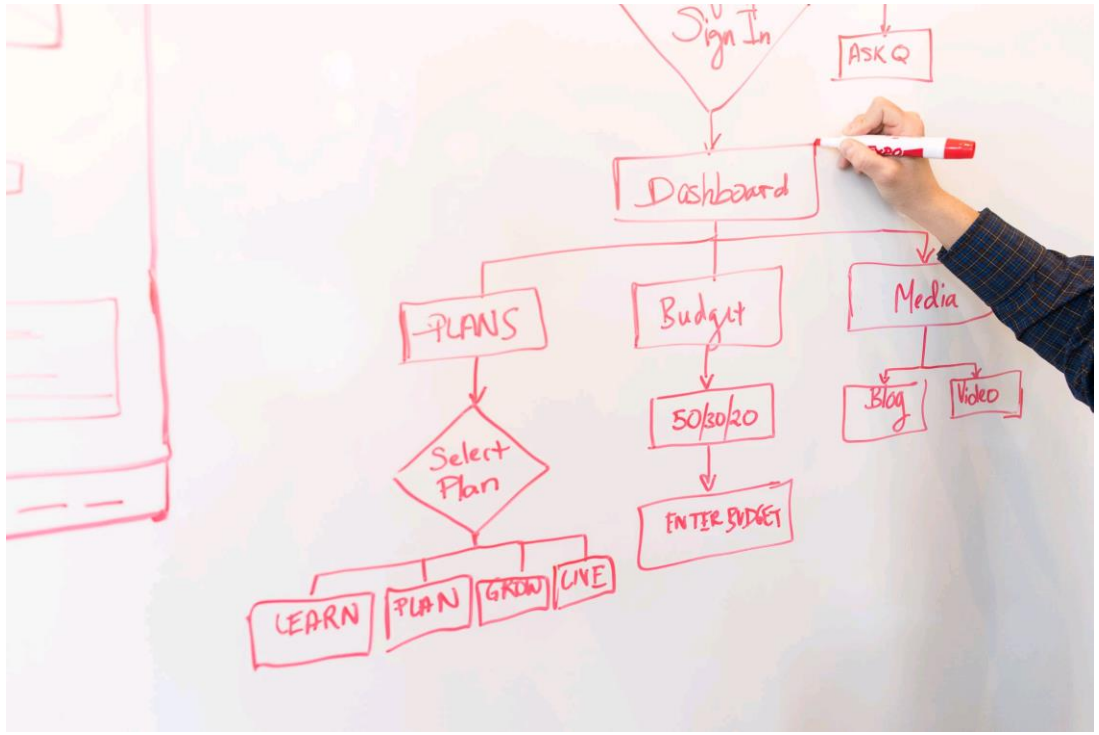
Integrierte Dokumentation

- Integrierte Dokumentation ist **leichter zu bearbeiten** und hat bessere Chancen, nachgeführt zu werden.
- **Sie reicht aber in keinem Falle aus!**
- Glossar, Spezifikation (Wireframes, Use Cases, Datenmodell, ...) und Architektur-Entwurf müssen **unbedingt vorher** entstehen!
- Sie können **nicht in Code-Komponenten** abgelegt werden (von der Projektdokumentation ganz zu schweigen).



Separate Dokumentation

- Die separate Dokumentation ist grundsätzlich gefährdet, sie kann nur funktionieren, wenn
  - die **Anforderungen** an die Dokumente und die **Verantwortlichkeiten** für das Erstellen, Prüfen und Freigeben klar sind,
  - Dokumentation **hoch bewertet** und anerkannt wird,
  - das **Inhaltsverzeichnis** (die Struktur) der Dokumente zu Beginn festgelegt ist,
  - **Werkzeugunterstützung** verfügbar ist (Wiki?),
  - jedes Dokument nach Fertigstellung und nach Änderungen **geprüft** wird und
  - alle Dokumente einer effektiven **Konfigurationsverwaltung** unterstellt werden.



Ziele einer guten Dokumentation

- Dokumente sind ein Mittel zum **Know-how-Transfer** und auch zur **Kommunikation** zwischen Auftraggeber und Auftragnehmer.
- In Dokumenten retten und **bewahren** wir das **Wissen** über Programme, für die Entwicklung und für die Wartung.
- Dokumente erlauben **systematische Prüfungen** (z. B. Reviews).
- Anhand der Dokumente kann man den **Projektfortschritt** feststellen. Ob ein Testbericht vorliegt, lässt sich einfach überprüfen.
- Dokumente können die **systematische**, sorgfältige **Entwicklung belegen**, sie machen die Software revisionsfest.

Leider ist der Nutzen der Dokumentation verteilt und zeitlich fern, die Kosten treten sofort auf und sind gut sichtbar. Darum wird an der Dokumentation gespart

Faustformeln für die Kosten

- Die durchschnittliche Produktivität beträgt vier Seiten pro Tag
- Bei 1000€/Tag kostet ein 40 Seiten starkes Dokument ca. 10.000 €.
- Dabei ist die Produktivität eher hoch geschätzt

- **Nutzen**:
  - Dokumentation hilft, Fehler zu vermeiden oder sie wenigstens rasch zu finden.
  - Gut dokumentierte Software lässt sich einfacher erweitern und wiederverwenden.

Taxonomie

Alle Dokumente gehören zu einer der folgenden vier Kategorien:

- **Systemdokumentation**
  Hierzu zählen alle Dokumente, die für die Konstruktion, den Betrieb und die Wartung der Software benötigt werden. (Unterscheidung technisch/fachlich)
- **Projektdokumentation**
  In diese Kategorie fallen alle Dokumente, die benötigt werden, um das Entwicklungsprojekt zu planen, zu leiten und abzuschließen.
- **Qualitätsdokumentation**
  Dies sind alle Dokumente, in denen die Maßnahmen zur analytischen Qualitätssicherung dokumentiert sind.
- **Prozessdokumentation**
  beschreibt den Entwicklungsprozess und seine konkrete Umsetzung im Projekt.

Nutzen der Taxonomie

- Diese Taxonomie beantwortet die folgenden Fragen:
  - Welchem **Zweck** dient das Dokument? Damit auch: Was gehört hinein, was nicht?
  - **Wer** wird es lesen, verwenden? (Darstellungsform, Terminologie)
  - Muss (darf) das Dokument nachgeführt, **aktualisiert** werden?
  - **Wie lange** muss das Dokument verfügbar bleiben?
- Fragen der Konfigurationsverwaltung:
  - **Wo** und unter **welcher Bezeichnung** wird es aufbewahrt?
  - Wer hat auf das Dokument (keinen) **lesenden Zugriff**?
  - Wer hat (keinen) **schreibenden Zugriff**?
  - Wessen Aufgabe ist die **Prüfung** und die Abnahme des Dokuments?

Kategorien im Überblick

Dokumenten-kategorie	Zweck	Adressaten	Aktuali-sierung	Aufbewahrungsfrist
System-dokumen-tation	Erhaltung und Transfer des Know-hows, Einsatz des Systems	Entwickler, Wartungsinge-nieure, Tester, Kunden	erforderlich	bis Ende der Systemnutzung
Projekt-dokumen-tation	Führung des Projekts, Analyse nach Projektende	Linien- und -Projektmanager, Kunden	unerwünscht	bis Projektende plus n Jahre
Qualitäts-dokumen-tation	Nachweis der Maßnahmen zur Qualitätssicherung	Linien- und -Projektmanager, Kunden	verboten	bis Ende der Systemnutzung
Prozess-dokumen-tation	Beschreibung der Vorgehensweise, Unterstützung bei der Durchführung	alle, die aktiv am Projekt beteiligt sind	sinnvoll	bis Ende der Systemnutzung und so lange, wie der Hersteller existiert

Benutzungsdokumentation

- Damit Software eingesetzt werden kann, muss dokumentiert sein, wie sie installiert, betrieben und bedient wird.
- Diese Informationen werden zusammenfassend als **Benutzungsdokumentation** bezeichnet.
- Beispiele:
  - Benutzungshandbuch
  - Tutorial
  - Installationshandbuch
  - kontextsensitive Hilfe
  - Online-Informationen
- Dokumente werden für verschiedene Adressaten erstellt, die unterschiedliche Informationen benötigen.

Adressat	Zweck	Dokumente
Käufer	Kaufentscheidung treffen Leistet die Software das, was ich benötige?	Produkt-beschreibung
Administrator	Installation der Software Wie wird die Software installiert und deinstalliert?	Installations-anweisung
	Betrieb der Software Was muss ich wissen, um die Software nach der Installation zu betreiben?	Administrations-handbuch Release-Notes
Unerfahrener Anwender	Bedienung kennenlernen Wie kann ich die Software starten und die wichtigsten Funktionen ausführen?	Erste-Schritte-Dokument Tutorial
Erfahrener Anwender	Nachschlagen Welche Funktionen und Einstellungen kann ich im Detail durchführen? Was tue ich, wenn ein Fehler auftritt?	Benutzungs-handbuch FAQ-Liste

Folgen schlechter Dokumentation

- In der Praxis gehört die Dokumentation meist zu den Problemzonen des Software Engineerings. Warum ist das so?
  - Software-Entwickler haben Dokumentieren weder in der Ausbildung noch in der Praxis gelernt, sie können es nicht.
  - Auf Priorität 1 steht die Einhaltung des Liefertermins, auf Priorität 2 und 3 auch. Da die Termine in fast allen Projekten so vorgegeben sind, wird nicht dokumentiert. Dazu fehlt einfach die Zeit.
- Folgen:
  - Wartungsingenieure arbeiten als Archäologen.
  - Grundlagen für die Handbücher fehlen.
  - Retrospektive Analysen sind nicht möglich. Damit hat die Organisation keine Möglichkeit, aus dem Projekt zu lernen.

Übungsaufgabe

Prüfen Sie gegenseitig ihre SE2-Dokumentation! Je zwei Gruppen stellen sich Ihre Dokumentation vor!

SEP2-Gruppen

20min

Zusammenfassung

- Dokumentation als das Sorgenkind der Software-Entwicklung
- Dokumentation betrifft nicht nur den Source-Code
- Folgen von schlechter Dokumentation sind hinlänglich bekannt, dennoch wird sie häufig nicht hoch genug priorisiert