

1) Dokumentationskonzept

Die Dokumentation ist ein wichtiger Bestandteil jedes Programms. Dazu gehören eine Kurzbeschreibung des Programms, Verwaltungsinformationen und die Kommentierung des Quellcodes. Von jedem Programm müssen die elementaren Informationen angegeben werden, wie der Programmname, der das Programm möglichst genau beschreibt, seine Aufgabe, Zeit- und Speicherkomplexität, die Namen der Programmautoren, die Versionsnummer und das Erstellungsdatum. Auch der jeweilige Bearbeitungszustand, d.h. geplant, in Bearbeitung, akzeptiert, etc. sollte vermerkt werden.

1.1.1) Interne Dokumentation

Um Programmcode sowohl für alle Teammitglieder als auch für projektfremde Programmierer gut lesbar zu machen, muss der Quellcode durch den Einsatz von Kommentaren geeignet dokumentiert und visuell verständlich strukturiert werden. Da in Java programmiert wird, sollten nicht nur die allgemeinen Dokumentationsrichtlinien sondern auch die Java-spezifischen angewendet werden. dass sie von Javadoc extrahiert und in HTML-Format konvertiert werden kann. Besonderes Augenmerk sollte auf die Kommentierung der Operationen einer Klasse gelegt werden, das beinhaltet neben der Beschreibung der Aufgaben jeder Operation auch die der Parameter. Gängige Konventionen bezüglich der Form, wie die Großschreibung von Klassen, beziehungsweise die Kleinschreibung mit Großbuchstaben am Wortbeginn im Inneren von Methoden oder das einheitliche Einrücken von Text erleichtern das zügige Einarbeiten in fremden Quellcode. Entwicklungsentscheidungen, zum Beispiel warum welche Alternative gewählt wurde, müssen dokumentiert werden, um bei Modifikationen und Neuentwicklungen bereits gemachte Erfahrungen auswerten zu können. Die Kommentare sollen dem inhaltlichen Verständnis der Programmlogik dienen. Es soll der Hintergrund einer Anweisung, nicht die Anweisung selbst, allgemeinverständlich beschrieben werden. Die Namensgebung von Bezeichnern ist dabei ebenfalls wichtig. Als Faustregel gilt: seine Länge soll proportional zu seinem Sichtbarkeitsbereich sein. Für eine lokale Schleife darf die Laufvariable durchaus `i` heißen, für ein umfangreicheres Modul sollte eine relevante Variable jedoch einen aussagefähigeren Namen, wie zum Beispiel `PersonalNr` erhalten. Zusätzlich ist bei der Deklaration ein Kurzkomentar zum Einsatzzweck der Variablen vorzusehen. Auch standardisierte Benennung von klassischen Methoden, wie Gettern und Settern beschleunigen das Verständnis. Außerdem sollte der Programmierer auch selbst einschätzen, an welchen Stellen, ein anderer Probleme beim nachvollziehen seines Codes haben könnte. Nützlich sind beispielsweise eine Beschreibung der eingetretenen Situation zu Beginn des "else"-Zweiges einer "if"-Anweisung, wenn diese schon weiter oben begann, oder eine klare Erläuterung von kompliziert verwendeten Indizes.

Spezielle Tags werden mit einem `@` gekennzeichnet (z.B. `@param`, `@version`...).

- Übliche Schreibweise (* am Zeilenbeginn wird ignoriert):

```
/** javadoc-Kommentar
```

```
Über mehrere Zeilen
```

```
/
```

```
Deklaration
```

1.2) Externe Dokumentation

Die Externe Dokumentation soll dem Benutzer ermöglichen, mit dem Produkt effizient und fehlerfrei zu arbeiten. Bei dem Erstellen der externen Dokumentation kann auch interne Dokumentation benutzt werden.

1.2.1) Kundendokumentation Darin sind alle wichtigen Dokumente enthalten, die während des Projektes erstellt werden, beispielsweise Lastenheft, Glossar, Pflichtenheft und ähnliches.

1.2.2) Benutzerhandbuch bzw. Online-Dokumentation. Für eine Client-Server-Anwendung ist jedoch eine Online-Hilfe notwendig, die den sicheren Umgang mit der Software für Benutzer möglichst leichter schafft.

2) Organisatorische Festlegungen

Insbesondere, wenn viele Gruppenmitglieder gemeinsam an einem großen Projekt arbeiten, ist es von großer Relevanz, einheitliche Konzepte der Implementierung und Dokumentation zu verfolgen. Schließlich können die verschiedenen Aufgabenbereiche nicht vollständig voneinander getrennt werden und es wird notwendig sein, am Code eines anderen weiter zu arbeiten. Dann muss dieser natürlich verständlich und nachvollziehbar sein. Die Gruppe wählt daher einen Vertreter, der sich gesondert um die Einhaltung dieser Abmachungen kümmert. Er hat die Pflicht, das regelmäßig und gewissenhaft zu kontrollieren und gegebenenfalls andere auf Mängel hinzuweisen. Er ist ebenfalls Ansprechpartner für dahingehend auftretende Fragen. Aber jedes Gruppenmitglied selbst trägt ebenfalls Verantwortung und sollte darauf achten, die Konzepte einzuhalten. Jeder sollte seinen Code auch immer zeitnah dokumentieren, denn bei Nachdokumentation am Ende der Codeerstellung sind wichtige Informationen, die während der Entwicklung angefallen sind, oft nicht mehr vorhanden.

2.1.1) Qualitätsstandards

Funktionalität: normal

Zuverlässigkeit: gut

Benutzbarkeit: gut

Effizienz: normal

Änderbarkeit: normal

Übertragbarkeit: nicht Relevant

2.2) Vorgehensweise. Erste Grundlagen des Dokumentationskonzeptes ergeben sich aus dem Lastenheft/Pflichtenheft. Zur Fixierung bestimmter Standards wird außerdem dieses Dokument durch die Entwicklungsgruppe erstellt. Ein weiterer wichtiger Meilenstein ist der Reviewprozess. Da in verschiedenen Phasen des Projekts unterschiedliche Verantwortlichkeiten entstehen, sind erarbeitete Dokumente und Konzepte per Mailingliste grundsätzlich allen Mitgliedern zugänglich zu machen. Erweitert und intensiviert wird dieses arbeitsteilige Vorgehen durch intensive Nutzung unseres Wikis, da hier auch mehrere Personen in die Entstehung von Artefakten eingebunden werden können. Neben der Erstellung eines Testkonzepts durch unseren Verantwortlichen für Tests wird vereinbart, dass Testfälle auf der Basis von Geschäftsprozessen erstellt werden. Für deren Konzeption und ausreichende Dokumentation ist ebenfalls der Testverantwortliche zuständig.

2.3.) Verantwortlichkeit Die Erstellung der externen Dokumentation liegt in der Verantwortung des Verantwortlichen für Dokumentation und Qualitätssicherung. Verantwortlich für den Quellcode einschließlich der Kommentierung und Javadoc ist der Verantwortliche für Implementierung. Verantwortlich für die Dokumentation der Testfälle ist der Verantwortliche für Tests.