

Dokumentationskonzept



Dokumentation Richtlinien

Wofür ein Dokumentationskonzept?

Zu oft vernachlässigen Teams das Thema "Dokumentation" und stellen dieses in den Hintergrund. Im Gegensatz dazu erfolgt dann jedoch in vielen Bereichen des Projektmanagements eine ausführliche Planung. Oft wird dabei jedoch auch am Anfang der sog. Planungsphase die Erarbeitung eines soliden Dokumentationskonzeptes zu sehr vernachlässigt bzw. ganz weggelassen.

Man sollte sich direkt von Anfang des Projektes (Planungsphase) an Gedanken zu der Dokumentationsstrategie machen. Das wesentliche Ziel eines solchen Dokumentationskonzeptes ist es, einen einheitlichen und schnellen Informationsfluss zwischen den einzelnen Projektmitgliedern durch schnellen Zugriff auf relevante projektbezogene Dokumente bzw. projektrelevante Informationen zu ermöglichen und diese ohne weitere Schwierigkeiten erreichbar zu machen. Hierfür wird am Anfang sowie im Laufe des Projektes (falls neues dazu kommt) dieses Konzept anhand von Dokumentationsregeln erstellt. Mit diesen Regeln wird bspw. die Struktur von Dokumenten (Word, Excel, PDF) sowie die Ablage dieser vereinheitlicht, um so eben den gewünschten Effekt eines schnelleren Zugriffs zu ermöglichen.

Qualitätsstandards

Für die verschiedenen Programmiersprachen und Auszeichnungssprachen in unserem Projekt verwenden wir folgende Coding-/Style Conventions:

Kotlin : <https://kotlinlang.org/docs/coding-conventions.html>

Java: <https://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

JavaScript: https://www.w3schools.com/js/js_conventions.asp

CSS: <https://docs.ckan.org/en/ckan-2.7.3/contributing/css.html>

HTML: https://www.w3schools.com/html/html5_syntax.asp

Vorgehensweise

Unsere Dokumentation setzt sich auch aus zwei Komponenten zusammen. Diese sind die interne, quelltextnahe Dokumentation und die externe Design-Dokumentation. Um neuen Mitgliedern oder aktuellen Mitgliedern, welche sich einen gesamten Überblick über einen der Codebereiche wie Frontend oder Backend erschaffen wollen, ist eine vollständige sowie einfach zugängliche Dokumentation nahezu unentbehrlich. Will man nun einen Überblick über unser verwendetes Code-Design sich erarbeiten, muss man als Mitglied lediglich in unsere externe Designdokumentation schauen. Diese wird in unserem GitHub Repository in dessen Wiki (<https://github.com/bm-games/BM-MUD/wiki/Dokumentation>) veröffentlicht und im Laufe des Projekts stetig aktualisiert.

Will man nun jedoch tiefere Einblicke in die Ebene der Implementierung bekommen, lohnt sich ein Blick in die zukünftig automatisch generierte Dokumentationsdatei, welche anhand der internen, quelltextnahen Dokumentation erzeugt wird. Für diese Dokumentation bei der Programmiersprache Kotlin verwenden wir "Dokka" (<https://kotlinlang.org/docs/kotlin-doc.html>). Dokka ist eine Dokumentations-Engine für Kotlin, die die gleiche Funktion wie javadoc für Java erfüllt. Dokka hat Plugins für Gradle, Maven und Ant, so dass wir die Dokumentationserstellung in Ihren Build-Prozess integrieren können. Genau wie Kotlin selbst, unterstützt Dokka vollständig gemischtsprachige Java/Kotlin-Projekte. Es versteht Javadoc Kommentare in Java-Dateien und KDoc Kommentare in



Kotlin-Dateien und kann Dokumentationen in mehreren Formaten erzeugen, darunter Standard-Javadoc, HTML und Markdown für Github.

Sollten im Laufe des Projektes Änderungen an der Implementierung vorgenommen werden, so ist das Wiki dementsprechend zu aktualisieren. Dabei ist es von großer Bedeutung, diese Dokumentation stets recht aktuell zu halten, sodass auch jegliche Code-Bereiche nachvollziehbar sind!

Verantwortung

Für die Qualitätssicherung im Projekt ist Marius Armbruster verantwortlich. Hierbei geht es nicht um die Erstellung der gesamten Dokumentation, sondern hauptsächlich um die Kontrolle der erstellten bzw. aktualisierten Dokumentationen. Sollten Änderungen an einer Code-Komponente vorgenommen worden sein, so ist dies dem Leiter der Qualitätssicherung zu melden, nachdem die Dokumentation anhand der vorliegenden Informationen überarbeitet worden ist. Somit kann dieser die Dokumentation überprüfen und falls notwendig eine Überarbeitung dieser anfordern, sollte diese bspw. nicht nachvollziehbar oder lückenhaft sein.

