

OSTFALIA HOCHSCHULE

EKDI-PROJEKT

# Essay

*Martin Krause 70478294*

supervised by  
Prof. JENSEN

18. Januar 2022

# 1 Ein erfolgreiches IT-Projekt ...

...ist gar nicht so einfach zu realisieren. Wenn ich mein theoretisches Wissen und meine eigene praktische Erfahrung eines Software-Team-Projektes von meiner Zeit an der Technischen Universität Braunschweig mit einfließen lasse, gehört viel dazu ein IT-Projekt erfolgreich abzuschließen.

Anfangen bei der Teambildung, die schon Schwierigkeiten machen kann, da eventuell einige Leute im Team weniger Erfahrung mit diesem Tool oder dieser Programmiersprache haben als andere, oder weil Streit ausbricht. Es braucht teilweise gute Teamleiter, die das gesamte Projekt steuern und unterstützen. Bei der Teambildung gilt es auch die Vier Phasen der Teambildung zu beachten und im Hinterkopf zu haben. Diese unterscheiden sich in „Forming“, „Storming“, „Norming“ und „Performing“. Die erste Phase ist zur Orientierung der Teammitglieder gut und einzelne Rollen werden gebildet. Die zweite Phase führt oft zu Konflikten im Team welche oft Folgen für das Projekt haben können. Die dritte Phase dient eher zur Regelfindung im Team, um solche Konflikte bestenfalls zu verhindern und die letzte Phase ist die, in der das Team gut eingespielt effizient arbeiten kann<sup>1</sup>.

Nach dem Erhalt des Auftrages für eine Software muss erstmal viel geplant werden bevor es überhaupt mit der Programmierung losgeht, da sonst direkt am Anfang schlechter Code entsteht, der am Ende überhaupt nicht das tut, was er soll. Solche Teams arbeiten für gewöhnlich nach bestimmten Strategien, wie z.B. dem V-Modell, dem Wasserfall-Modell, oder Scrum, die als sehr agile Strategie gilt. Bei Scrum arbeitet das Team in sich ständig wiederholenden Zyklen, welche sehr effizient ablaufen, weshalb das Team gute fachliche Kompetenzen aufweisen sollte und schon gut zusammenarbeiten können sollte<sup>2</sup>.

Sobald man die Planung abgeschlossen hat und sämtliche Diagramme der „UML- Modellierung“ entworfen hat, wie Aktivitäts- oder Zustandsdiagramme, oder auch Klassendiagramme<sup>3</sup>, kann man mit der Programmierung anfangen. Dabei sollte allerdings auf bestimmte Coding-Richtlinien geachtet werden für Zeicheneinrückung oder Variablenbenennung. Es ist schon schwer in fremdem Code durchzusteigen, oder sogar teilweise in dem eigenen, deshalb muss auf eine gute Kommentierung und das Einhalten von gewissen Codingguidelines geachtet werden<sup>4</sup>.

Ein ganz wichtiger Punkt eines solchen Projektes stellt auch das ausgiebige Testen dar. Das bedeutet man testet kleine Programmteile für sich und setzt diese Stücke immer mehr zusammen, sodass man in Integrationstests testet, wie und ob die Teile zusammenarbeiten, bist hin zum kompletten Programm. Dabei testet man am besten durch Whitebox-Tests, bei denen man die Bedingungen und den Funktionsinhalt kennt, und durch Blackbox-Tests, bei denen man keinen Einblick in Code hat und nur Randbedingungen und besondere Fälle von Variablen testet. Man testet also ohne die innere Struktur der Funktionen zu kennen das grundlegende Verhalten dieser. Dabei kann man

eigentlich auch nicht die Abwesenheit von Fehlern beweisen, aber man kann meistens welche finden und diese dann verbessern<sup>5</sup>.

Und wenn man nach all dem dann das vermeintlich fertige Produkt dem Kunden vorstellt, kann es immer noch sein, dass dieser einige Stellen nicht gut findet und daraufhin nochmal sämtliche Codestücke angepasst werden müssen. Deswegen ist es wichtig von vornherein gut verständlichen und gut wartbaren Code zu schreiben, denn eine Funktion oder Klasse neu zu schreiben ist in den meisten Fällen einfacher, als den alten Code zu verbessern und würde viele IT-Projekte zu einem schnelleren und effizienteren Ende führen.

Es gab in der Geschichte schon viele gescheiterte Softwareprojekte, die unter anderem zu großen Fehlern in Programmen führten, welche sogar ziemlich berühmt wurden. Zum Beispiel musste die „Mariner-1-Rakete“ kurz nach ihrem Start abgelenkt werden, weil ein Fehler sie auf eine falsche Flugbahn lenkte, oder eine tragische Fehlerberechnung für die Dosierung einer Bestrahlungsmaschine, die Anfang 2000 acht Menschen das Leben kostete<sup>6</sup>.

Insgesamt sollte ein erfolgreiches IT-Projekt also ein gutes Team haben, mit guter Arbeitsmoral, kompetenten Teammitgliedern und einer guten Arbeitsstrategie, sodass durch eine gute, überlegte Planung auch funktionierender Code geschrieben werden kann, der möglichst schnell zu guten Testergebnissen führt und ein erfolgreiches Produkt liefert.

Wenn man all das beachtet, sollten solche verheerende Fehler bzw. Software-Krisen eigentlich nicht mehr vorkommen können und die Grundlage für ein erfolgreiches IT-Projekt ist gegeben.

## 2 Quellenverzeichnis

- [1] Weblog, <https://blog.seibert-media.net/blog/2011/06/24/team-building-prozesse-softw>  
Stand: 18.01.2022.
- [2] AgileScrumGroup, <https://agilescrumgroup.de/strategie-scrummen/>, Stand: 18.01.2022.
- [3] Herold, H.; Lurz, B.; Wohlrab, J.; Hopf, M.; „Grundlagen der Informatik“, 3. Auflage, Pearson-Verlag, 2017.
- [4] embedded-software-engineering, <https://www.embedded-software-engineering.de/coding-guidelines-sind-programmierrichtlinien-fluch-oder-segen-a-795748/>,  
Stand: 19.01.2022.
- [5] Software-Testing-Academy, <https://www.software-testing.academy/black-box-test.html>, Stand: 18.01.2022.
- [6] entwickler.de, <https://entwickler.de/programmierung/top-10-der-software-katastrophen>,  
Stand: 18.01.2022.