

Lernen durch Feedback aus Inspektionen

Andrea Herrmann
Herrmann & Ehrlich,
D-70372 Stuttgart, herrmann@herrmann-ehrllich.de

Motivation: Lernen durch Feedback

Dieser Beitrag präsentiert Erfahrungen aus zwei Jahrgängen Software Engineering Praktikum an der TU Braunschweig. Hierbei wurden von einem Jahr zum nächsten das Vorgehensmodell und die Vorlagen für die Spezifikationen verbessert. Das primäre Ziel dieser Veränderungen war es, die Lehre zu verbessern. Aus wissenschaftlicher Neugier wurden auch die gefundenen Fehler und Aufwände analysiert. Bisher gibt es nur wenige Untersuchungen zu Fehleranzahlen und -typen in Spezifikationsdokumenten.

Die drei untersuchten Forschungsfragen waren:

1. Welche sind die fehleranfälligesten Kapitel und Notationen?
2. Welche quantitative Auswirkung hatten die durchgeführten Prozess-Verbesserungen?
3. Wie haben die Studierenden durch das Feedback aus den Inspektionen ihre Fehlerzahlen verringern können?

Aus der Antwort auf Frage 1 folgt, welche Notationen in der Vorlesung noch gründlicher erklärt werden sollten. Frage 2 zeigt auf, ob und wie gut die Prozessverbesserungen die Ergebnisse verbessert haben. Frage 3: Man sollte meinen, dass nach zwei Inspektionsrunden die Fehlerzahlen in den Dokumenten auf beinahe null sinken.

Vorgehen

Jeder der beiden Jahrgänge bestand aus 23 Teams, die jeweils ein Programmierprojekt durchführten. Jedes der Projekte wurde von einem Informatik-Institut angeboten und auch inhaltlich und technisch betreut. Das Software Systems Engineering Institut gab das Vorgehensmodell, Vorlagen für die Dokumente und Zwischentermine vor und kümmerte sich um die Qualitätssicherung. Dazu führte es u.a. eine syntaktische Inspektion der Dokumente durch auf Vollständigkeit (im Sinne der Vorlage und der UML), Konsistenz, Eindeutigkeit etc. Eine semantische, fachliche Inspektion wurde durch das jeweils betreuende Institut durchgeführt.

Im ersten Jahr handelte es sich um 155 Studierende, die entlang dem V-Modell arbeiteten. Jedes der erstellten Dokumente (Pflichtenheft, Grob- und Feinentwurf, Testdokumentation) und der Code wurde vom Software Systems Engineering Institut anhand einer Checkliste inspiziert.

Basierend auf dem Feedback der Studierenden, Betreuer und den Inspektions-Ergebnissen wurden die Dokumentvorlagen und das Vorgehensmodell verbessert. In den Dokumentvorlagen wurden redundante Kapitel entfernt, da diese unnötig Fehler provozierten. Im Pflichtenheft wurden drei Kapitel zu einem einzigen zusammen geführt, die sich mit nichtfunktionalen Anforderungen beschäftigten und zuvor inkonsistent gewesen waren. Im Grobentwurf wurden nur noch Sequenzdiagramme erstellt, keine Aktivitätsdiagramme mehr, da beide praktisch dieselben Inhalte darstellten, die Sequenzdiagramme aber sorgfältiger erstellt worden waren.

Im zweiten Jahr waren es 130 Studierende. Sie arbeiteten nach einem iterativen Vorgehensmodell, das aus drei hintereinander geschalteten Vs von jeweils 4 Wochen. Nach jedem V wurden die Pflichtenhefte, Grob- und Feinentwürfe sowie Testspezifikationen checklistenbasiert inspiziert und die Ergebnisse an die Studierenden gemeldet. 8 der 23 Teams erhielten einen studentischen Projektleiter, der zugleich Teilnehmer der Projektmanagement-Vorlesung war und dort für seine Aufgabe ausgebildet und gecoach wurde.

Die Inspektionen wurden durch das Software Systems Engineering Institut nach einheitlichen Kriterien für alle 23 Projekte durchgeführt. Hierzu wurden basierend auf dem IEEE Standard 830-1998 Checklisten erstellt, die als Inspektions-Unterstützung dienten. Die Checklisten waren erstellt worden, indem die Kriterien des Standards auf jedes Kapitel angewendet wurden. Inspektionskriterien waren beispielsweise „Kapitel 3 enthält mindestens ein Use Case Diagramm“ oder „Die Akteure entsprechen den in Kapitel 2.2 dargestellten Zielgruppen.“ Dokumente wurden auch mehrfach inspiziert, da nicht jeder Reviewer alle Fehler fand.

Ergebnisse

Um die drei Fragen zu untersuchen, wurden die durch die Inspektion gefundenen Fehler für jeden der beiden Jahrgänge über alle 23 Gruppen aufsummiert.

Frage 1: Welche sind die fehleranfälligesten Kapitel und Notationen?

Im Pflichtenheft waren gerade die nichtfunktionalen Anforderungen inkonsistent und nicht operationalisiert. Entweder waren sie nicht priorisiert, oder in verschiedenen Kapiteln inkonsistent priorisiert. Das Use Case Diagramm erwies sich als besonders

schwierig und auch die textuellen, tabellarischen Use Cases. Im Grobentwurf waren alle Diagramme schwierig, insbesondere das wohl am wenigsten intuitive Zustandsdiagramm. Im Feinentwurf verursachte besonders das Klassendiagramm Fehler.

Ganz allgemein wurden in Diagrammen und Tabellen mehr Fehler gefunden als in Freitext. Dies liegt vermutlich daran, dass es mehr und klarere Kriterien für die Beurteilung gibt, aber auch die Einhaltung dieser Kriterien schwieriger ist. Gleichzeitig dauerte das Inspizieren von Fließtext länger als von Diagrammen (pro Seite). Eine detailliertere Aufschlüsselung der gefundenen Fehlerzahlen und -typen finden Sie in [1] und [2].

Frage 2: Welche quantitative Auswirkung hatten die durchgeführten Prozess-Verbesserungen?

Es gab deutliche Verbesserungen in Jahr 2 gegenüber Jahr 1. Wie man in Tabelle 1 sieht, hat sich beim Pflichtenheft die Fehlerzahl im Jahr 2 halbiert, bei Grob- und Feinentwurf ebenfalls reduziert. Die prozentuale Verteilung der Fehlertypen (z.B. Anteil der Konsistenzfehler an allen Fehlern) blieb dabei gleich. Es ließ sich herausrechnen, dass circa ein Drittel der Fehlereinsparungen durch die Vorlagenvereinfachung in Pflichtenheft und Grobentwurf erreicht wurden.

| | Jahr 1 | Jahr 2 |
|----------------------|---------------|---------------|
| Pflichtenheft | 0,79 | 0,39 |
| Grobentwurf | 1,06 | 0,93 |
| Feinentwurf | 0,52 | 0,45 |

Tabelle 1: Fehler pro Kapitel

Frage 3: Wie haben die Studierenden durch das Feedback aus den Inspektionen ihre Fehlerzahlen verringern können?

Bereits in Jahr 1 ließ sich ein Lerneffekt beobachten: Während das Pflichtenheft am meisten Fehler enthielt, war der Grobentwurf deutlich besser und der Feinentwurf noch besser (gemessen in Anzahl Fehler pro Dokument).

In Jahr 2 verbesserten sich die Dokumente deutlich von Iteration 1 bis Iteration 3. Der Umfang der Dokumente wuchs kaum von Iteration 1 bis Iteration 3, sondern betrug ca. 20 Seiten für Pflichtenheft und Grobentwurf, 35 Seiten für einen Feinentwurf. Die Anzahl der Fehler halbierte sich beim Pflichtenheft und Grobentwurf von Iteration 1 bis 3, beim Feinentwurf sank die Fehlerzahl auf zwei Drittel.

Es zeigte sich auch, dass der Projektleiter einen deutlichen (wenn auch nicht statistisch signifikanten) verbessernden Einfluss auf die Fehlerzahl hatte. Offiziell bestand seine Aufgabe zwar nur in der Leitung des Projektes, doch tatsächlich hat auch er dem Team Feedback über ihre Dokumente gegeben.

Zusammenfassung und Diskussion

In Jahr 2 betrug der Inspektionsaufwand 15 Minuten für das Beseitigen (nicht Finden!) eines Fehlers in

Pflichtenheft oder Grobentwurf und 20 Minuten für das Beseitigen eines Fehlers im Feinentwurf. Auch nach der dritten Iteration waren die Dokumente nicht fehlerfrei. Dies lässt sich zum größten Teil durch Zeitmangel erklären. Innerhalb einer vierwöchigen Iteration mussten die Teams nicht nur Pflichtenheft, Grobentwurf, Feinentwurf und Testdokumentation aktualisieren, sondern insbesondere auch die Software weiter entwickeln. Das Codieren und die lauffähige Software waren ihnen vermutlich wichtiger als einzelne Spezifikationsfehler. Gelegentlich kam es auch zu Verschlimmbesserungen, wo die Beseitigung eines Fehlers mehrere neue verursachte.

Trotzdem zeigt die deutliche Verbesserung der Dokumente über die Iterationen hinweg und auch im Vergleich zu Jahr 1, dass Lernen stattgefunden hat. Auch die Fehlervermeidung durch das Vereinfachen von Vorlagen ist empfehlenswert sowie die Prozessverbesserung durch das iterative Vorgehen.

Vergleich mit der Literatur

Es gibt nur wenig veröffentlichte Fehlerzahlen. Unternehmen veröffentlichen solche Daten nicht gerne. Die Frage, wie sich die Fehlerzahl durch das Feedback aus einer Inspektion verringert, wurde bisher noch nirgends gemessen.

Die gefundenen Ergebnisse sind vermutlich vergleichbar mit den Ergebnissen von Berufs- oder UML-Anfängern, die an mehreren Projekten parallel arbeiten. Abgesehen von ihrer Mehrfachbelastung durch mehrere Lehrveranstaltungen und ihrer geringen praktischen Erfahrung arbeiteten die Studierenden unter sehr guten Bedingungen: Sie arbeiteten in Kleingruppen zu 3-11 Personen, die sich regelmäßig trafen, hatten kurz zuvor UML in einem Kurs erlernt und mehrere Betreuer, die sie unterstützten.

Der Vergleich unserer Ergebnisse mit den wenigen Publikationen von Firmen [3],[4] (siehe auch in [1]) zeigte, dass die gefundenen Fehlertypen von der Vorlage und Notation abhängen, von den verwendeten Prüfkriterien und dem Erstellungsprozess. Dies macht umso deutlicher, dass der Requirements Engineering Prozesses sorgfältig gestaltet werden muss.

Referenzen

- [1] Herrmann A: Die häufigsten Modellierungsfehler. Symposium Software-Architektur, Sept 2013, Stuttgart.
- [2] Brunswig, S.: Text- und Code-Analyse im SEP 2010 an der TU Braunschweig. Bachelorarbeit an der TU Braunschweig, 2010.
- [3] Ott, D.: Defects in Natural Language Requirement Specification at Mercedes-Benz: An Investigation using a Combination of Legacy Data and Expert Opinion. International Requirements Engineering Conference 2012.
- [4] Stoyanova, N.: Computergestützte Qualitätssicherung textueller Anforderungen am Beispiel der Daimler AG. REConf 2013, München.