

TNA (00:00): Guten Morgen. Fangen wir pünktlich an. Das heutige Ziel ist eindeutig: Wir brauchen drei konkrete, abgestimmte Maßnahmen zur Beschleunigung unserer Code-Reviews, inklusive Verantwortlichkeiten und Deadline für die Umsetzung. TNA, du hast den ersten Pain Point angesprochen.

TNB (00:45): Genau. Das größte Problem ist die Wartezeit. Ein PR liegt im Schnitt 48 Stunden offen, bevor jemand drüberschaut. Das blockiert Weiterentwicklung und führt zu Merge-Konflikten. Mein Vorschlag: Wir führen ein „Review-Buddy“-System ein. Jeder hat einen festen Partner, der innerhalb von 24 Stunden erste Rückmeldung geben muss.

TNC (01:30): Die Idee ist gut, aber sie löst nicht das Grundproblem der unterschiedlichen Expertise. Ein Frontend-Buddy kann meinen komplexen Backend-PR nicht sinnvoll reviewen. Vielleicht brauchen wir stattdessen klarere Richtlinien, was überhaupt in einem Review geprüft werden muss. Viele diskutieren über Code-Style, obwohl wir einen Linter haben, und übersehen Logikfehler.

TNA (02:10): TNB hat das verstanden, warum du nicht, TNA? Klare Checklisten wären doch einfacher als ein Buddy-System. Das wird so nicht funktionieren, wenn niemand Ahnung vom Code hat.

TNB (02:45): Es geht nicht um „Ahnung haben“, es geht um erste Prüfung. Der Buddy schaut auf offensichtliche Dinge und ob Tests vorhanden sind. Die tiefgehende fachliche Prüfung kann dann ein Spezialist übernehmen. Aber du hast recht, eine Checkliste wäre eine gute Ergänzung. Was sollte da drauf?

TNC (03:20): Mindestens: 1) Linter läuft durch, 2) Unit-Tests sind vorhanden und grün, 3) Keine offensichtlichen Sicherheitsantipattern, 4) JIRA-Ticket verlinkt. Wenn das erfüllt ist, kann der PR gemergt werden, auch wenn die architektonische Diskussion später im Slack weitergeht.

TNA (03:50): Das klingt nach Bürokratie. Dann schreibt ja jeder nur noch Tests, um die Checkliste abzuhaken, ohne an Qualität zu denken. Aber okay, notieren wir das als mögliche Maßnahme. Was ist mit dem Tool? Sollen wir von GitHub zu GitLab wechseln, weil da die Review-Workflows besser sein sollen?

TNB (04:30): Oh, das wäre ein riesiger Aufwand. Ich habe mal gehört, dass GitLab bessere Merge-Request-Analytics hat. Man könnte sehen, welche Reviewer immer Flaschenhälse sind.

TNC (05:15): Das stimmt. Aber bevor wir die Plattform wechseln, sollten wir die Analytics unserer aktuellen Tools mal richtig nutzen. Ich glaube, GitHub hat da auch Reports, die wir nie angeschaut haben. Vielleicht liegt das Problem gar nicht an der Plattform, sondern daran, dass einige Teammitglieder einfach zu viele PRs gleichzeitig offen haben.

TNA (06:00): Dann müssen die halt schneller reviewen. Das ist eine Frage der Disziplin. Aber gut, wer soll sich denn diese Analytics mal anschauen?

TNB (06:40): Ich könnte das machen. Brauche aber Zugriff auf die Org-Ebene. Und dann müssten wir definieren, was eine gute Metrik ist. Mittlere Review-Zeit? Anzahl der Kommentare pro PR?

TNC (07:20): Die Anzahl der Kommentare ist sogar kontraproduktiv. Wenn die hoch ist, heißt das vielleicht nur, dass die PRs zu groß sind. Vielleicht sollten wir eine Regel für die maximale PR-Größe einführen. Nichts über 400 Zeilenänderungen.

TNA (07:55): 400 Zeilen? Das wird so nicht funktionieren. Eine Refactoring-Arbeit ist schnell mal 1000 Zeilen. Dann machen wir halt mehrere PRs, ist ja auch doof. Aber notieren wir: PR-Größe begrenzen. Das ist schon Maßnahme zwei. Brauchen wir noch eine dritte?

TNB (08:40): Mir fällt noch ein: Oft scheitert es an den beschreibenden Kommentaren. Der Autor schreibt Bugfix und nichts weiter. Vielleicht brauchen wir ein Template für PR-Beschreibungen: Was wurde geändert? Warum? Wie getestet?

TNC (09:25): Ein Template ist eine gute Idee. Aber wer setzt das durch? Der Reviewer soll dann den PR ablehnen, wenn das Template nicht ausgefüllt ist? Das führt zu noch mehr Frust.

TNA (10:00): TNA, du könntest doch mal so ein Template entwerfen. Du bist doch gut in so was. Dann schicken wir es rum und sehen, ob es jemand benutzt.

TNB (10:35): Ähm, okay... ich kann es versuchen. Aber das löst ja nicht das ursprüngliche Problem der langen Wartezeiten. Unser Buddy-System...

TNC (11:10): Das mit dem Template bringt mich auf etwas anderes: Wir haben gar keine gemeinsame Definition von „Done“ für ein Review. Sollte ein PR nach dem Review perfekt sein, oder ist „gut genug“ okay? Diese Unsicherheit verlangsamt alles. Vielleicht brauchen wir dazu eine Session im nächsten Teammeeting.

TNA (11:50): Eine ganze Session nur für Done? Das ist Overkill. Das weiß doch jeder. Aber wenn ihr meint... Könnten wir nicht einfach ein Tool wie LinearB oder Waypoint einführen, das uns automatisch sagt, wo die Engpässe sind? Das spart uns diese ganze Diskussion hier.

TNB (12:35): Davon habe ich gehört. Die sind aber sehr teuer, soweit ich weiß. Und dann sind wir wieder bei der Budget-Frage. Müssen wir mit unserem Head of Tech sprechen.

TNC (13:15): Bevor wir Geld ausgeben, sollten wir die Prozessprobleme kennen. Sonst kaufen wir nur ein teures Dashboard für unsere Ineffizienz. Diese Tool-Diskussion lenkt gerade völlig ab. Wir wollten doch konkrete, umsetzbare Maßnahmen definieren.

TNA (13:50): Du hast ja recht, aber ohne Tool sehen wir die Probleme nicht. Aber gut, fassen wir mal zusammen, was wir haben: 1) Checkliste für Reviews, 2) PR-Größe begrenzen, 3) Template für PR-Beschreibungen. Wer macht was bis wann?

TNB (14:30): Ich schaue mir die Analytics an und entwerfe ein Template. Bis wann soll das sein?

TNC (15:00): Ich kann die Checkliste in unser Team-Wiki schreiben und zur Diskussion stellen. Aber ohne Commitment des ganzen Teams bringt das nichts. Sollen wir die Maßnahmen nicht erst im Teammeeting vorstellen und abstimmen lassen?

TNA (15:40): Das wird so nicht funktionieren, dann schieben wir es nur weiter. TNA, du bringst bis Freitag Template und Analytics-Auswertung. TNB, du hast die Checkliste bis

Freitag. Ich trage dann für nächste Woche einen Termin mit dem Head of Tech für die Tool-Evaluation ein. Dann haben wir doch was Handfestes.

TNB (16:20): Ähm, also mein Teil ist sehr aufwändig. Freitag ist in zwei Tagen. Das schaffe ich nicht neben meinen Tasks.

TNC (16:55): Meine Checkliste ist auch nur ein Entwurf, wenn wir nicht wissen, was die Analytics sagen. Vielleicht sollten wir die Reihenfolge umdrehen: Erst Daten, dann Regeln.

TNA (17:30): Ihr macht es euch schon wieder schwer. Gut, dann verschieben wir die Deadline auf nächsten Mittwoch. Aber dann ist auch gut. Wollen wir noch was anderes besprechen? Wie läuft eigentlich das Deployment des neuen Logging-Systems?

TNC (18:10): Oh, da gibt es ein Problem mit den Umgebungsvariablen in der Staging-Umgebung. Das hat mich gestern zwei Stunden gekostet...

TNB (18:35): Welche Umgebungsvariablen? Nutzt ihr nicht mehr das zentrale Config-Management? Ich dachte, das hätten wir mit dem letzten Kubernetes-Update gelöst.

TNC (19:00): Theoretisch schon, aber in Staging läuft noch der alte Pod, weil der automatisierte Rollout gestern abgebrochen wurde. Daher der Mismatch. Ich muss da heute Nachmittag nochmal ran.

TNA (19:30): Okay, interessant. Aber das ist ein anderes Thema. Zurück zu den Code-Reviews. Wir hatten gesagt: Analytics, Template, Checkliste. Können wir das bitte finalisieren?

TNB (20:05): Moment, aber wenn das Staging-Problem besteht, betrifft das ja auch unsere Fähigkeit, Features zu testen und damit PRs zeitnah zu mergen. Das ist doch ein noch grundlegenderer Flaschenhals als die Review-Zeit an sich.

TNC (20:40): Guter Punkt. Vielleicht sollten wir erst die Stabilität unserer Deployment-Pipeline sicherstellen, bevor wir den Review-Prozess optimieren. Sonst hängen PRs ohnehin, weil sie nicht in eine stabile Umgebung deployed werden können.

TNA (21:15): Ist jetzt wirklich ein komplett anderes Thema. Das sprengt den Rahmen.

TNB (21:45): Aber es hängt doch zusammen! Wenn wir schnell reviews, aber dann die PRs wochenlang in Staging feststecken, weil die Umgebung kaputt ist, bringt uns das ganze Optimieren nichts.

TNA (22:20): TNB, kannst du das Staging-Problem bis morgen lösen? Dann ist das Thema vom Tisch.

TNC (22:50): Ich versuche es, aber ich kann es nicht garantieren. Es hängt vom Ops-Team ab, ob die mir die nötigen Rechte für den Rollback geben.

TNB (23:25): Soll ich mal mit dem Ops-Lead sprechen? Der kennt mich noch vom letzten Projekt.

TNA (23:55): Nein, lass das. TNB, du regelst das. Wir müssen hier zu einem Abschluss kommen. Also nochmal: Was sind die nächsten Schritte für die Code-Review-Optimierung?

TNC (24:30): Wie gesagt, ich stelle bis nächsten Mittwoch einen Entwurf der Checkliste ins Wiki.

TNB (25:00): Und ich schaue mir die Analytics an und mache ein Template-Entwurf. Aber ohne Commitment des Teams...

TNA (25:30): Das Commitment holen wir im Teammeeting ein. Wichtig ist, dass ihr die Hausaufgaben macht. Gut. Dann wäre das geklärt. Ich trage mir noch die Terminvereinbarung mit dem Head of Tech für die Tool-Evaluation ein.

TNC (26:00): Wollen wir nicht kurz besprechen, welche Argumente wir für das Tool haben? Damit wir da vorbereitet sind?

TNA (26:30): Dafür haben wir jetzt wirklich keine Zeit mehr. Ich habe in zwei Minuten mein nächstes Meeting. Schickt mir einfach zwei, drei Punkte per Mail zu. Ich leite das dann weiter.

TNB (27:00): Aber wer entscheidet denn am Ende über die Maßnahmen? Und was ist mit der PR-Größen-Regel? Die war auch noch im Raum.

TNA (27:30): Das entscheiden wir, wenn die Daten da sind. Und die Regel besprechen wir im Teammeeting. Ich muss jetzt wirklich los. Danke euch. Wir halten uns an die neuen Deadlines. Ende