

# Peer Review: Feature 5

## Ausarbeitung und Projekttagbuch

Zuallererst ist anzumerken, dass ihr euch an die Struktur des vorgegebenen Schemas gehalten habt und dieses in tabellarischer Form ausgefüllt habt. Dabei habt ihr zunächst das Universum, die Stakeholder und die Technische Strategie beschrieben: alle diese sind verständlich und leicht nachvollziehbar, besonders die Ausformulierung der Technischen Strategie überzeugt. Auch der Mehrwert des Features lässt sich klar herauslesen. Es ist nachvollziehbar, dass die Produktivität des Unternehmens, durch eine bedachtere Teamzusammenstellung gesteigert werden kann. Auch das gewählte Kriterium nach Erfahrung zu sortieren um ausgeglichene Teamkonstellationen zusammenzustellen ist sinnvoll. Ebenso wurden nachvollziehbare Strategien für die Ermittlung dieser Eigenschaft für jeden Mitarbeiter und die Verteilung von Schwierigkeitsniveaus der Projekte gewählt.

Weiterhin sind auch die gefundenen Fragestellungen, ob man die Qualifizierung der Mitarbeiter anhand ihrer Arbeitsgeschwindigkeit ermitteln darf und ob eine Einteilung der Mitarbeiter nach ihrem Erfahrungsniveau sinnvoll ist, und die dazu passenden Preexisting, Technical und Emergent bias passend gewählt und lassen auf eine vertiefte Auseinandersetzung mit dem Feature und seiner Bedeutung für das Unternehmen und seine Mitarbeiter schließen. Deswegen fällt auch die Vortheoretische Deliberation kritisch aus, liefert aber dennoch überzeugende Gründe (Präzedenzfälle: Mentoring-Programme) warum sich Gedanken über eine Umsetzung des Features gemacht werden.

In der dritten Phase des Schemas, der Ethischen Systemüberprüfung sind gut durchdachte pro und contra Argumente aus deontologischer und konsequentialistischer Perspektive aufgezählt. Aus konsequentialistischer Sicht wird vor allem darauf eingegangen, dass durch die Umsetzung des Tools die Produktivität des Unternehmens steigt und damit auch die Gehälter der Mitarbeiter. Andererseits wird auch darauf verwiesen, dass dennoch Druck- und Konfliktsituationen und sogar eine klassenartige Spaltung der Mitarbeiter möglich sind. Diese Argumente, aber auch die deontologischen, sind nicht nur nachvollziehbar aufbereitet, sondern stimmen auch mit den zuvor erwähnten Grundgedanken überein.

In der Theoretischen Deliberation und in der Urteilsphase wird der logische Entschluss gefasst das Feature mit einigen Einschränkungen, die die contra Argumente entkräften, umzusetzen. Zum einen wird festgelegt, dass mit den Daten (dem Erfahrungsniveau der

Mitarbeiter) vertraulich umgegangen wird, zum anderen dient das Tool nur als Empfehlung und Entscheidungen zu Teamkonstellationen werden von Managern nicht nur auf Basis des Tools gefällt.

In der fünften Phase, die der Translation, wird beschrieben wie ermittelt wird, ob ein Mitarbeiter als erfahren oder unerfahren einzustufen ist. Hierbei ist dem Management relativ viel Freiraum gelassen (z.B. Wahl der Prozentzahl). Dies erscheint sehr sinnvoll, da es sich um Kriterien handelt die Unternehmens-, Aufgaben- und Zeitspezifisch sind, daher sind feste Grenzen eher zu vermeiden. Ferner wird darauf eingegangen wie das Schwierigkeitsniveau von Projekten in das System eingetragen wird. Zudem wird spezifiziert wie das System mit dem Manager interagiert und, dass nur das Management die Ergebnisse des Tools einsehen darf.

Im letzten Teil des Schemas, der Verification wird eine Möglichkeit zum Testen des Tools erläutert. Diese erscheint zwar aufwendig, aber auch notwendig und sinnvoll.

Das Projekttagebuch ist als Ergänzung von großer Bedeutung, da es noch einige Fragen klärt, wie zum Beispiel welche Argumente der Deliberation für euer Team am ausschlaggebendsten in Urteilsphase waren (die konsequentialistischen pro-Argumente). Darüber hinaus lässt sich erkennen, dass Entscheidungen nur nach Teambesprechungen und gut begründet getroffen wurden.

Das vorgestellte Tool ist sehr interessant und erscheint sinnvoll. Teamkonstellationen innerhalb eines Unternehmens sind nicht zu unterschätzen. Es kommt häufig vor, dass aufgabenspezifisch einerseits Expertengruppen andererseits langsame, überforderte, Gruppen entstehen. Dadurch kann die Produktivität eines Unternehmens sinken. Dies gilt es zu vermeiden, und genau damit beschäftigt sich das Tool. Eine automatisierte Aufteilung der Mitarbeiter nach ihrem Erfahrungsniveau und eine Teamzusammenstellung die diese berücksichtigt, stellt einen interessanten und effektiven Lösungsansatz dar. Auch die Ausarbeitung der konkreten Details (z.B. Bias) sowie die Betrachtung der pro und contra Argumente aus deontologischer und konsequentialistischer Sicht und der Urteilsfällung auf Basis dieser ist in sich stimmig und überzeugt. Es ist erkennbar, dass sich das Team vertieft mit dem Thema auseinandergesetzt hat. Dies lässt sich auch durch die gefundenen Kompromisse erkennen: Schwachstellen und Kritikpunkte (vor allem ethische) wurden nicht nur erkannt sondern auch (in vielen Aspekten) erfolgreich gelöst. Auch, dass nicht alle Details vorab von der Implementierung festgelegt werden, sondern, dass das Management Freiraum bei der Festlegung der Grenze, ab wann ein Mitarbeiter als unerfahren einzustufen ist, überzeugt.

Anzumerken ist dennoch, dass die Aufteilung in erfahren und unerfahren vielleicht nicht am effektivsten ist. Beispielsweise wäre ein Ranking der Mitarbeiter aussagekräftiger, da es nicht nur zwei Kategorien gibt, und so besonders auf die Mitarbeiter die sich an der Grenze (von erfahren/unerfahren) oder an der Spitze ihrer Kategorie befinden, besser eingegangen werden kann (z.B. Intensivkurse etc. für unproduktivsten Mitarbeiter), und somit sinnvoller gewesen, wobei hier andere ethische Komplikationen auftreten könnten. Ebenso ist es vorstellbar die Mitarbeiter nicht allgemein nach ihrer Erfahrung aufzuteilen sondern nach ihren Fertigkeiten in bestimmten Anwendungsbereichen (z.B. Kommunikation, Programmierfertigkeiten,... ) und dann so Projektabhängig die Teams zu erstellen, jedoch wäre dies mit deutlich mehr Aufwand verbunden und würde bei gleicher Präzision der von euch vorgenommenen Erfahrungsermittlung (Ermittlung dieser durch durchschnittliche Arbeitszeit im Kombination mit der Schwierigkeit des Projekts nicht einfach nur Selbsteinschätzung) wohl den Rahmen des Praktikums sprengen.

Aber genau die Bestimmungsmodalität des Erfahrungsstandes der Mitarbeiter ist sehr sinnvoll und aussagekräftig gewählt worden und ist letzten Endes eine sehr gute Lösung im Rahmen des Praktikums.

Einzig wäre noch zu erwähnen, dass die Ausführung der Argumente in Phase III, meiner Meinung nach, noch etwas konkreter und ausführlicher sein könnte, da sie dadurch stärker auf den Leser einwirken und so die Gründe für die getroffenen Kompromisse verdeutlichen würden.

Zusammenfassend ist festzustellen, dass ihr ein interessantes Tool, das durchaus von Bedeutung ist, umgesetzt habt. Es ist anzumerken, dass ihr euch viele wichtige Gedanken gemacht habt die sich klar in eurer Ausarbeitung wiederfinden. Diese ist insgesamt sehr stimmig und geht auf viele Problematiken aber auch Vorteile des Tools ein und wird sehr gut durch den zugehörigen Projekt-Tagebucheintrag vervollständigt. Dadurch überzeugt sie sowohl von der Notwendigkeit des Tools, als auch von der von euch gewählten Umsetzung.

## Implementierung: Generator, JavaScript und Jira

Das Jira Interface ist leicht lesbar und strukturiert aufgebaut. Besonders gefällt mir, dass man als Admin den Zeitraum (wie viele Tage aus der Vergangenheit man betrachten kann, für die Berechnung der Erfahrungsniveaus der Mitarbeiter) aktiv anpassen kann.

Darüber hinaus, ist der Generator eine Gute Möglichkeit Daten zu generieren, die ihr elegant umgesetzt habt. Die Verwendung dessen ist zwar etwas kompliziert, wird jedoch durch euer

Video-Tutorial erheblich erleichtert. Dies zeigt eine tiefgründige Auseinandersetzung mit Jira und wie man für Jira möglichst effizient Daten generieren kann. Einzig finde ich es schade, dass der Code des Generators nicht besser kommentiert ist, jedoch wird dies zum größten Teil durch das sehr hilfreiche Video-Tutorial und der ReadMe-Datei wettgemacht.

Zur JavaScript Implementierung ist anzumerken, dass diese sehr strukturiert und übersichtlich ist. Ihr habt viele Funktionen aus der Vorlage übernommen, dies macht euren Code leichter nachvollziehbar. Des Weiteren ist positiv anzumerken, dass ihr viele kleine Hilfsfunktionen wie ***addDays(date, days)*** oder ***compare(a, b)*** verwendet habt, diese machen die längeren Funktionen wie ***fetchIssues()*** übersichtlicher und erleichtern so das Verständnis eures Codes. Auch hier ist aus meiner Sicht der einzige Kritikpunkt, dass die Methoden und die gesamte Implementierung nicht sehr aufschlussreich kommentiert sind. Häufig ist zwar was der eigentliche Zweck der Funktionen kurz erläutert, jedoch wird kaum auf die übergebenen Parameter und Rückgabewerte eingegangen. Ich persönlich bevorzuge es, wenn diese Details gemeinsam mit den Datentypen und der Bedeutung der Parameter und Rückgabewerte explizit in den Kommentaren stehen. Dies erleichtert das Verständnis des Codes. Im Gegensatz dazu, finde ich es aber lobenswert, dass ihr die Namen eurer Variablen und Funktionen sehr aufschlussreich gewählt habt. Allein aus den Namen lässt sich häufig der Sinn und Zweck erkennen. Insbesondere, finde ich vor allem die ***fetch***-Funktionen elegant implementiert.

Zusammenfassend bin ich der Meinung, dass eure Implementierung gut übersichtlich ist und sorgfältig gestaltet wurde. Sie setzt ein wichtiges und interessantes Tool um, von dessen Bedeutung die Ausarbeitung und der Projekttagbuch-Eintrag überzeugen.