Die richtigen Rahmenbedingungen für ein Entwicklungsteam

Editorial

Liebe Manager!

Heute möchte ich eine Lanze für unsere Entwickler brechen. In der Beratung werde ich sehr oft in kritische Situationen geholt: Zugesagte Termine werden nicht eingehalten, das Produkt hat trotz jahrelanger Entwicklungsarbeiten immer noch viele Fehler und wichtige Features sind nicht fertig.

Wir anaylsieren dann immer sehr genau, was die Ursachen für die missliche Lage sind[1]. Immer wieder stellen wir dabei fest, dass die Ursache für die Misere nicht bei schlecht qualifizierten oder unmotivierten Entwicklern liegt. Natürlich kommt das auch vor. Aber sehr oft liegt die Ursache darin, dass die Entwickler (und Tester, Architekten etc.) gar nicht die Chance haben, gute Arbeit zu leisten. Sie müssen dermaßen viel Zeit und Energie aufwenden, um mit den gegebenen Rahmenbedingungen zurecht zu kommen, dass oft kaum mehr Zeit für die eigentliche Entwicklung bleibt.

Liebe Manager, Geschäftsführer, Teamleiter! Dieser Knowledge Letter ist ein Appell an uns Führungskräfte, einmal innezuhalten und die Welt mit den Augen unserer Teams zu betrachten. Haben unsere Team wirklich alles, was sie brauchen, um gut arbeiten zu können? Was habe ich im letzten Jahr dafür getan? Wie sieht das aus Sicht der Teams aus? Und könnte ich unter diesen Umständen produktiv sein und ein tolles Produkt bauen? Lassen Sie uns auf den folgenden Seiten diesen Fragen gemeinsam nachgehen…

Markus Unterauer
Berater, Trainer

[1] Mit unserem Produkt „Potenzialanalyse" zeigen wir schnell und effizient Probleme auf und geben Empfehlungen zu Verbesserung.

Damit Softwareentwicklung gut läuft, braucht es zwei Dinge: Das richtige Team und die richtigen Rahmenbedingungen. Passen die Rahmenbedingungen nicht, kann auch das beste Team nichts Ordentliches produzieren. Was aber gehört zu diesen Rahmenbedingungen dazu? Im folgenden Artikel geben wir konkrete Antworten.

Um erfolgfreich zu sein braucht es den richtigen Rahmen

Softwareentwickler können im Schnitt nur 12-50% für tatsächliche Softwareentwicklung aufwenden [1]. Der Rest geht für Kommunikation, Meetings, Organisation und vor allem dem Nachlaufen nach fehlenden Informationen drauf. Fast zwei Drittel dieses ohnehin sehr geringen Anteils arbeitet das Team an Features, welche die Kunden gar nicht oder anders brauchen [2]. Tatsächlich produziert ein Entwicklungsteam also nur in 7-30% seiner Zeit echten Wert für den Kunden. Sieht man sich das an, so muss eines der wichtigsten Ziele jedes Managers sein, dass 1) Entwickler mehr Zeit für tatsächliche Softwareentwicklung bekommen und 2) in dieser Zeit die richtigen Sachen entwickelt werden.

Die Ursachen für die Schieflage in der Produktivität liegen nun oft nicht beim Team selbst. Fehlende oder falsche Rahmenbedingungen und Vorgaben verhindern, dass das Team produktiver ist.

$$Erfolg = Team + Rahmenbedingungen$$

Viele Management-Ratgeber greifen das auf und postulieren, dass man als Manager zuerst einmal die richtigen Rahmenbedingungen herstellen soll. Auch in den Prinzipen agiler Software-Entwicklung ist das verankert. Prinzip Nummer 5 beschreibt das so: „Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen." [3] Als Manager muss ich also dem Team das Umfeld und die Unterstützung geben, das sie benötigt und sie dann in Ruhe arbeiten lassen. Aber was bedeutet das? Was gehört alles zu diesem Umfeld, zu diesen Rahmenbedingungen? Was braucht das Team, um produktiv arbeiten zu können und was muss ich als Manager alles gestalten? Die wohl einfachste Art herauszufinden, was das Team benötigt ist, es einfach zu fragen. Wer weiß besser was er braucht als die jeweilige Person bzw. das Team selbst? Oft liefert die Frage nach dem „Was braucht ihr?" aber überraschend wenige Antworten. Vielfach nennt das Team nur die 2-3 Dinge, die in dem Moment gerade am meisten stören oder fehlen. Aber was ist mit dem Rest?

Notwendige Rahmenbedingungen

Folgende Checkliste zeigt, was ein Software-Team braucht, um produktiv arbeiten zu können:

1. **Die richtigen Mitglieder**: Als Manager muss ich dafür sorgen, dass mein Team die richtigen und ausreichend Mitglieder hat. Selten aber doch muss man auch Mitglieder aus dem Team nehmen, wenn sie die Produktivität nach unten ziehen. Fallen Teammitglieder aus, so muss rechtzeitig für Ersatz gesorgt werden. Können Aufgaben nicht vom Team selbst erledigt werden, muss ich als Manager externe Unterstützung organisieren, damit das Team seine Ziele wieder erreichen kann.
2. **Ausreichend Zeit und Fokus**: Zum einen braucht das Team ausreichend Zeit, um qualitativ hochwertige Produkte mit geringer technischer Schuld produzieren zu können. Zum anderen braucht jedes Teammitglied ausreichend Zeit für das Projekt. Als Manager muss ich sicherstellen, dass die Teammitglieder ausreichend Zeit für das Projekt bekommen, sich darauf fokussieren können und für das Projekt insgesamt ausreichend Zeit geplant wird.
3. **Skills in Software Entwicklung**: Als Manager muss ich gemeinsam mit dem Team analysieren, welche Fähigkeiten benötigt werden und wie diese erlangt werden können. Dazu gehört eine solide Ausbildung in den relevanten Programmiersprachen und modernen Programmiermethoden, gutes Grundwissen in Architektur und Design, Kenntnisse aktueller Software-Engineering Techniken wie Continuous Integration, Branching/Merging, Code-Reviews etc. und ein Basiswissen über manuelles und automatisiertes Testen von Software.

   Als Manager muss ich dafür sorgen, dass die entsprechenden Schulungen, Bücher, Videos, Coaches, Trainer und Berater zur Verfügung gestellt werden. Die Software Quality Lab Academy bietet z.B. in allen relevanten Bereichen des Software-Engineerings eine solide Grundausbildung, sowie Vertiefungs- und Expertentrainings an. [4]
4. **Domänenwissen über das Business**: Als Manager muss ich dem Team möglichst viel Gelegenheit verschaffen, etwas über das Business und die Fachdomäne zu lernen. Dazu kann ich Experten einladen, das Team zum Kunden rausschicken, Schulungen organisieren etc.
5. **Wissen über Legacy-Systeme**: Soll das Team bestehende Legacy-Systeme weiter pflegen, muss ich als Manager dafür sorgen, dass entsprechendes Know-How aufgebaut wird. Auch hier kann ich Experten holen, Schulungen organisieren und dem Team ausreichend Zeit einräumen, sich die alten Systeme anzusehen und „Software Archäöologie" zu betreiben.
6. **Die Möglichkeit, irgendwo Experte zu werden**: Jeder möchte irgendwo richtig gut sein und als Experte geachtet werden. Gemeinsam mit dem Team wird geplant, was das Spezialgebiet des Teams und jeder Person darin ist. Natürlich bedeutet das nicht, dass man nichts anderes mehr machen kann. Aber als Manager muss ich ermöglichen, dass sich das Team in ein Gebiet richtig vertieft und dort Experten-Know-How aufbaut.
7. **Geeignete Arbeitsmittel**: Als Manager muss ich dem Team alle notwendigen Arbeitsmittel zur Verfügung stellen. Diese müssen auch hochwertig und performant sein. Kein Holzfäller kann mit einer stumpfen Axt produktiv arbeiten, genauso wenig kann ein Entwickler mit einem langsamen PC, einer instabilen Internetverbindung oder einem langsamem Build Server effizient arbeiten. Die wesentlichen Arbeitsmittel für ein Softwareteam sind:

   - Hardware (Rechner, Monitor, Server, W-LAN, …)
   - Arbeitsplatz (ein gemeinsames Büro für das gesamte Team, Tisch, Sessel, Regale, Küche, …)
   - Moderationsmaterialien für Meetings und Meetingräume mit Whiteboard und Flipchart
   - Entwicklungsumgebung und Entwicklungstools (Source-Control, CI, Statische Analyse, Build System, Testautomatisierung, …)

Themenbereich	Aufgabe	Skill	Product Owner	Entwickler	SW-Architekt	Tester
Projektmanagement	Portfolio Management	Roadmap Planung	*			
Projektmanagement	Portfolio Management	Priorisierungstechniken (projektübergreifend)	*			
Projektmanagement	Projektleitung	Projektvorgehensmodelle	*	*	*	*
Projektmanagement	Projektleitung	Teamzusammenstellung	*			
Projektmanagement	Projektleitung	Projektplanung	**	*	*	*
Projektmanagement	Projektleitung	Aufgabenmanagement	*	*	*	*
Projektmanagement	Projektleitung	Controlling	**	*		*
Projektmanagement	Projektleitung	Reporting	**	*		*
Projektmanagement	Projektleitung	Priorisierungstechniken (im Projektes)	****	*	*	*
Projektmanagement	Projektleitung	Koordination von externen Partnern	***			
Fachwissen	Applikationen	Fachliches Applikations-Know-How	***	**	***	****
Fachwissen	Applikationen	Technologische Möglichkeiten der Applikation	**	****	****	**
Qualitätssicherung	Testmanagement	Testprozess / Testplanung	*	*		***
Qualitätssicherung	Testmanagement	Aufgabenmanagement (Testaufgaben)		*		***
Qualitätssicherung	Testdurchführung	Testfallentwurf (Entwurfsverfahren)	*	*	*	****
Qualitätssicherung	Testdurchführung	Manuelles Testen	**	**		****
Qualitätssicherung	Testdurchführung	Testautomatisierung		**	*	***
Qualitätssicherung	Reviews	Spezifikations-Reviews		**	**	***
Qualitätssicherung	Code-Qualität	Code Reviews		**	****	*
Qualitätssicherung	Code-Qualität	Statische Analyse		**	****	*

Abbildung 1 - Eine Skill-Matrix zeigt die im Team notwendigen Fähigkeiten

Vollständiger Knowledge Letter Zugang

Wir freuen uns, dass Sie an diesem Thema Interesse haben und den Knowledge Letter von Software Quality Lab bis hierher gelesen haben.

Dieser Knowledge Letter ist eine Vorschau (gekürzte Version des gesamten Artikels).

Wenn Sie den ungekürzten Knowledge Letter lesen möchten, registrieren Sie sich bitte unter
http://www.software-quality-lab.com/download/knowledge-letter/anfrage-knowledge-letter/

Sie erhalten nach der Registrierung vollen Zugang zu allen bisherigen Knowledge Lettern von Software Quality Lab und erhalten automatisch künftige Knowledge Letter per E-Mail.

Software Quality Days — Die größte Konferenz zum Thema „Software Qualität“ in Europa!

Besuchen Sie die Top-Konferenz mit allen Infos rund um Software Qualität.

Beste Qualität der Vorträge und Tutorials sowie eine Mischung aus praktischen und wissenschaftlichen Beiträgen machen die Software Quality Days zum Top-Event.

In den 3 praktischen Tracks werden anwendungsorientierte Vorträge präsentiert. Der wissenschaftliche Track zeigt Beiträge mit hohem Innovationsgrad und praktischer Anwendbarkeit, basierend auf Forschungs-ergebnissen. Im Solution Provider Forum präsentieren Aussteller ihre neuesten Tools mit Praxis-Beispielen.

Nähere Infos unter

www.software-quality-days.com