## TOOL-Vergleich

Bei der Durchführung eines Projektes ist es immer hilfreich, mindestens für einige der Managementaufgaben ein oder mehrere Hilfsmittel zu verwenden. In diesem Fall werden drei Projekte mit verschiedenen Prozessmodellen parallel bearbeitet, während sich die Teammitglieder zusätzlich den Großteil der Zeit nicht im gleichen Raum oder Gebäude aufhalten. Deshalb entschied sich das Team für die Nutzung eines elektronischen Tools, an welches sehr verschiedene, allgemeine als auch technik-spezifische Anforderungen gestellt wurden. Im Rahmen dieser Arbeit sollen hierfür zwei Tools ausgewählt werden, mit deren Hilfe verschiedene Projektmanagement-Aspekte erleichtert werden sollen. Im Anschluss an die Durchführung wird eruiert, welche Vor- und Nachteile den Teammitgliedern aufgefallen sind und welches Tool sich besser für jedes Prozessmodell eignet.

Da alle drei durchgeführten Prozesse im Bereich der agilen Software-Entwicklung anzusiedeln sind, sollten die gewählten Hilfsmittel entweder spezifisch für agiles Projektmanagement entwickelt worden sein oder zumindest erkennbar mit den daher rührenden Anforderungen umgehen können. Das bedeutet insbesondere, dass Iterationen in irgendeiner Form abbildbar sein müssen. Außerdem arbeiten an den Projekten Entwickler mit verschiedenen Betriebssystemen, in diesem Fall OS X und Windows. Das bedeutet, das Tool muss entweder online und im Browser nutzbar sein oder für die beiden genannten Betriebssysteme zur Verfügung stehen.

Da die Projektmitarbeiter über eine begrenzte Erfahrung mit Projektmanagement-Tools und Issue-Trackern verfügen, wurden verschiedene Online-Quellen hinzugezogen, um eine Vorauswahl und dann auch die endgültige Wahl zu treffen. Die Hauptquelle zur Vorauswahl war die Webseite FindTheBest[[1]](#footnote-1), welche Informationen über verschiedenste Projektmanagementsoftware sowie Reviews und Bewertungen von Nutzern zur Verfügung stellt. Außerdem ist es möglich, schnell und übersichtlich einen Vergleich zwischen mehreren Produkten anzuzeigen. Des Weiteren wurden Videos, Demos und Rezensionen auf den Herstellerseiten zu Rate gezogen, diese jedoch immer mit dem Wissen, dass der Hersteller selbst weniger objektiv sein kann als eine unabhängige Quelle.

Schließlich wurden die in Tabelle 1 dargestellten fünf Tools in die nähere Auswahl aufgenommen. Dabei kamen JIRA und Pivotal Tracker allein wegen ihrer Bekanntheit in die Vergleichsrunde. Die drei anderen Produkte zeigten auf FindTheBest die besten Bewertungen für Issue-Tracker, die sowohl mit agilem Projektmanagement und Kanban umgehen können, als auch für kleine Projekte handhabbar sind.

|  |  |
| --- | --- |
| JIRA mit Greenhopper | allen Teammitgliedern mehr oder weniger gut bekannt |
| Pivotal Tracker | einem Teil des Teams bekannt |
| Genius Inside | Beste Smart Rating – User Rating Kombination einer Software für kleine Projekte (Quelle: FindTheBest) |
| Vision Project | Nach Genius Inside beste Software, die Kanban und Issue-Tracking können soll (Quelle: FindTheBest) |
| Yodiz | Nach Vision Project beste Software, die Kanban und Issue-Tracking können soll (Quelle: FindTheBest)  ansprechende Oberfläche im Video (Quelle: FindTheBest) |

Tabelle 1: Software

### Bewertungskriterien

Im Folgenden werden die im Vorfeld erstellten Auswahlkriterien tabellarisch zusammengefasst und kurz erläutert. Dabei wurden allgemeine von prozess-spezifischen Kriterien getrennt und zusätzlich Wünsche der Teammitglieder unter „nice to have“ gelistet. Die letztgenannten Kriterien waren hauptsächlich als Tie-Breaker gedacht, falls alle anderen nicht ausreichen, um sich klar für zwei Tools zu entscheiden.

Für die Grundlage der Bewertung orientierte sich das Team an einer Idee aus dem Qualitätshaus, einer Matrix aus dem Quality Function Deployment. Die Wichtigkeit eines Merkmals wird mit einer Priorität zwischen 1 und 5 bewertet, wobei 5 für „sehr wichtig“ und 1 für „unwichtig“ steht. Die Entscheidung, ob ein Tool das gewünschte Merkmal vorweist, wird wiederum mit 0, 1 oder 2 bewertet (siehe Tabelle 2).

|  |  |
| --- | --- |
| 0 | definitiv nicht vorhanden |
| 1 | könnte ungefähr so vorhanden sein, wie benötigt oder eine andere Funktionalität kann vermutlich dafür „missbraucht“ werden oder es wurden keine Angaben gefunden, könnte also vorhanden sein |
| 2 | so vorhanden wie benötigt |

Tabelle 2: Bewertungsskala und ihre Bedeutung

### Allgemeine Kriterien

Tabelle 3 gibt einen Überblick über allgemeine Kriterien bei der Tool-Auswahl. Zu diesen zählen theoretisch auch Preis und Plattform bzw. Betriebssystem als rein organisatorische Kriterien. Der Preis spielte eine untergeordnete Rolle und wurde deshalb nicht in die Bewertungsmatrix aufgenommen. Die Ausführbarkeit auf Computern mit verschiedenen Betriebssystemen hingegen war wie bereits zu Anfang erwähnt eine zwingende Anforderung, so dass kein Tool aufgenommen wurde, dass diese nicht erfüllt.

Schließlich wurde auch die Erfahrung der Teammitglieder mit dem Tool einbezogen, da die Vertrautheit mit der Software hilfreich bei der Projektdurchführung sein sollte. Allerdings kann bei einem bereits bekannten Produkt die Schwierigkeit der Einarbeitung nicht mehr bewertet werden. Nach Beratung im Team wurde entschieden, dass Vertrautheit mit dem Tool wichtiger, aber trotzdem keine Pflicht sein soll.

Neben den organisatorischen Merkmalen gibt es auch allgemeine Funktionen, die in den gewählten Software-Entwicklungsprozessen gleich oder zumindest ähnlich vorkommen und abbildbar sein müssen. Dazu zählen das Aufstellen von User Stories, Priorisierung von Tickets, eine Art Backlog und ein anpassbares Board zur Visualisierung der Tickets. Dies sind Mindestanforderungen an die gewählte Software aus der agilen Entwicklung. Das anpassbare Board ist besonders für Kanban hilfreich und wichtig, da es den Workflow abbildet, den Kanban zu optimieren versucht. Das bedeutet auch, dass das Board angepasst werden sollte, so z.B. die Anzahl der Spalten oder deren Namen.

Aufgrund der geographischen Trennung der Teammitglieder ist neben dem elektronischen Board auch eine Möglichkeit zur Kommunikation wichtig. Diese kann über einfache Kommentare an Tickets, Emails, Persönliche Nachrichten oder Chats erfolgen. Mindestens eine dieser Varianten sollte durch das Produkt gegeben sein. Eine 2 erhielten in diesem Fall Produkte, die mehr als eine Art der Kommunikation erlauben.

Schließlich sollte das Tool nicht nur als Ersatz für ein nicht vorhandenes physisches Board genutzt werden sondern unbedingt auch ein Issue-Tracker sein. Das bedeutet insbesondere, dass User Stories und Tasks, die nur überblicksartig auf dem Board sichtbar sind, mit genaueren Details außerhalb des Boards gehandhabt werden können, oder auch ganz auf das Board verzichtet werden kann, falls beispielsweise ein physisches Board existiert. Außerdem soll eine Zuordnung an Mitarbeiter und eine Art Klassifikation der Tickets möglich sein, z.B. in Backlog-Items oder Bugs.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Prio | **Genius inside** | **Jira** | **Pivotal Tracker** | **Vision Project** | **Yodiz** |
| Erfahrung | 3 | 0 | 2 | 1 | 0 | 0 |
| **Allgemeine Funktionen** |  |  |  |  |  |  |
| User Stories | 4 | 2 | 2 | 2 | 2 | 2 |
| Priorisierung | 4 | 2 | 2 | 2 | 2 | 2 |
| Backlog (Produktbacklog) | 5 | 2 | 2 | 1 | 2 | 2 |
| anpassbares Board | 4 | 0 | 2 | 0 | 1 | 1 |
| Möglichkeiten zur Kommunikation | 4 | 2 | 1 | 1 | 2 | 2 |
| Issue tracking | 4 | 2 | 2 | 2 | 2 | 2 |
| Summe |  | 42 | 52 | 36 | 46 | 46 |

Tabelle 3: allgemeine Kriterien

### Bewertungskriterien für Scrum

Das Scrum-Team hat für seinen Prozess die in Tabelle 4 dargestellten Kriterien aufgestellt. Dabei wurde insbesondere die Durchführung von Sprints, möglichst auch mit dieser Benennung, gewünscht. Für Scrum bedeutet dies, dass man eine Iterationen in eine Timebox stecken, d.h. mit einem Start- und Enddatum versehen kann. Ein automatisches Öffnen und Schließen des Sprints wäre schön, ist aber kein Muss-Kriterium und wurde deshalb nicht in bei der Bewertung berücksichtigt.

Um Scrum entsprechend umzusetzen, muss das Tool außerdem unbedingt eine Art Sprintbacklog erlauben, d.h. es muss möglich sein, aus allen vorhandenen Tickets des Projekts eine bestimmte Anzahl zu einem Sprint zuzuordnen.

Da das Team an verschiedenen Orten entwickeln sollte, war ein virtuelles Scumboard ebenfalls sehr wichtig. Hierbei gab es in der initialen Entwicklung der Kriterien keine Anforderung an die Konfigurierbarkeit. Auch diese war eher ein Kann-Kriterium.

Das Vorhandensein von User Stories und Tasks bzw. die Möglichkeit mindestens zwei verschiedene Arten von Tickets anzulegen, wurde ebenfalls als wichtig für Scrum eingeordnet. Außerdem sollten zu einem Ticket in irgendeiner Weise Unteraufgaben zugeordnet werden können. Da das Team nicht immer bei der Entwicklung zusammensitzt, sollten Tickets in kleinere Aufgaben unterteilbar sein, so dass nicht jedes Teammitglied an einer eigenen User Story arbeiten muss, um anderen nicht in die Quere zu kommen. Dafür würden diese User Stories sehr lange „in progress“ sein, eine unerwünschte Eigenschaft bei Scrum.

Zur Auswertung und möglichen Verbesserung des Scrum-Prozesses ist u.a. nötig einige Kennzahlen zu berechnen. Das Team entschied, dass zumindest ein Burndown Chart automatisch mit Hilfe des Tools erzeugbar sein sollte, um hierbei behilflich zu sein und den Fortschritt auf andere Art als nur im Board zu visualisieren. Dies wurde besonders durch den Scrum Master gewünscht, erhielt allerdings nur eine geringere Priorität, da notfalls auch eine Abbildung per Hand beispielsweise in Excel möglich ist.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Prio | **Genius inside** | **Jira** | **Pivotal Tracker** | **Vision Project** | **Yodiz** |
| Sprints | 5 | 2 | 2 | 1 | 1 | 2 |
| Sprint Backlog | 5 | 2 | 2 | 2 | 0 | 2 |
| Scrumboard | 4 | 2 | 2 | 1 | 1 | 2 |
| User Stories und Tasks | 4 | 1 | 2 | 1 | 1 | 2 |
| Abhängigkeiten zwischen User Stories / Untertasks | 4 | 1 | 2 | 0 | 2 | 1 |
| Burndown Chart | 2 | 2 | 1 | 2 | 2 | 2 |
| Summe |  | 40 | 46 | 27 | 25 | 44 |

Tabelle 4: Scrum-Kriterien

### Bewertungskriterien für Kanban

Obwohl Kanban allgemein nur sehr wenige Merkmale besitzt, die zwangsweise dazugehören, entschied sich das Team dafür, auch einige der freiwilligen Techniken zu als Kriterien in Tabelle 5 aufzunehmen.

Das Kanban-Board ist eines dieser Kriterien, wurde jedoch unbedingt vom Team zur schnellen Übersicht aus Mangel an einem physischen Board gewünscht. Obwohl ein Kanban-Board im Prinzip einem Scrum-Board entsprechen kann, gab es auch hier nur 2 Punkte, wenn das Board veränderbar war, so dass ein geänderter Workflow übertragen werden könnte. Zusätzlich dazu sollte es möglich sein, die Anzahl der Tickets pro Spalte zu begrenzen. 2 Punkte gab es jedoch nur, wenn eine Verletzung der Begrenzung möglich und gut sichtbar zu erkennen war. Bei allen Produkten außer JIRA musste man sich hierbei auf Rezensionen verlassen, da in den Demos hierfür möglicherweise weiterführende Berechtigungen nötig gewesen wären. Da sich das Team zu Beginn des Projekts noch nicht festgelegt hatte, wurde dem Vorhandensein von Swim Lanes innerhalb des Boards dagegen eher eine niedrigere Priorität verliehen.

Für die Darstellung der verschiedenen SLAs oder Tickettypen sowie eine Unterscheidung zwischen Ticket und Task sollte irgendeine Art der Einstellung möglich sein. Am besten wären farbliche Unterschiede wie es für Kanban zur schnellen Unterscheidung am Board vorgeschlagen wird. Da dies notfalls mit Swim Lanes in etwa ersetzt werden kann, wurde lediglich die Priorität 3 gewählt.

Das Ziel der kontinuierlichen Verbesserung des Arbeitsablaufs kann nur dann erreicht werden, wenn regelmäßige Übersichten über Durchsatz, Flow und Geschwindigkeit der Arbeit erstellt und analysiert werden. Hierfür wären die zwei am häufigsten bei Kanban verwendeten Diagramm-Typen wünschenswert: Cumulative Flow Diagram und Burnup bzw. Burndown-Chart. Auch hier kann man die niedrige Priorität der Burncharts mit der Möglichkeit eines manuellen Ersatzes erklären. Neben diesen beiden sind besonders Lead Time und Cycle Time hilfreiche Richtwerte und sollten deshalb möglichst leicht ablesbar sein.

Als eines der wenigsten agilen Prozessmodelle kann Kanban ganz ohne Iterationen durchgeführt werden. Auch wenn dies nicht das Ziel des Teams war, sollte die niedrige Priorisierung diesen Fakt hervorheben: Iterationsfähigkeit ist kein Muss für das Produkt. Da sie jedoch ebenso für Scrum und Crystal Clear nutzbar sein sollten, erhielten alle Produkte 2 Punkte.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Prio | **Genius inside** | **Jira** | **Pivotal Tracker** | **Vision Project** | **Yodiz** |
| Kanban Board | 5 | 1 | 2 | 1 | 1 | 1 |
| WIP-Begrenzung | 4 | 1 | 2 | 1 | 1 | 1 |
| Swim Lanes | 2 | 1 | 2 | 0 | 1 | 1 |
| Tickettypen / SLAs | 3 | 1 | 2 | 2 | 1 | 1 |
| Burn Up Chart | 2 | 0 | 0 | 0 | 2 | 2 |
| Cumulative Flow Diagram | 4 | 0 | 2 | 0 | 2 | 1 |
| Durchsatz: Lead and Cycle Time | 4 | 1 | 2 | 0 | 2 | 1 |
| Iterationen | 1 | 2 | 2 | 2 | 2 | 2 |
| Summe |  | 20 | 46 | 17 | 32 | 28 |

Tabelle 5: Kanban-Kriterien

### Bewertungskriterien für Crystal Clear

Die Bewertungskriterien für Crystal Clear finden sich kurz zusammengefasst in Tabelle 6. Ähnlich der Liste von Scrum ist die wichtigste Anforderung eine Möglichkeit zur Organisation einer Iteration. In diesem Zusammenhang ebenfalls wünschenswert wäre die Planung und Umsetzung von Releases. Damit können häufige Lieferungen an den Endanwender abgebildet werden.

Des Weiteren sollte ein Rollensystem implementiert werden. Für die Vergabe einer 1 genügte es, wenn man Nutzern verschiedene vorgegebene Rollen mit verschiedenen Zugriffs- und Ausführungsrechten zuordnen kann. Für eine 2 sollte es die Software dem Anwender erlauben, seine eigenen Rollen zu kreieren, zu benennen sowie deren Berechtigungen zu setzen. Die Einschätzung dieses Kriteriums war am schwierigsten, da in einer Online-Demo selten genug Rechte vorhanden sind, um eine solche Funktionalität zu testen. Ebenso wenig wurde in Vorführvideos darauf eingegangen. Somit erhielt lediglich JIRA 2 Punkte, da hier aus der Erfahrung heraus bekannt war, dass neue Rollen im Administratorbereich angelegt werden können.

In Bezug auf die Reflektion und im Sinne der kontinuierlichen Verbesserung des Arbeitsprozesses sollte das gewählte Tool außerdem im Nachhinein noch einmal Einblick in alte Iterationen geben können, die Teil der letzten Lieferung waren. So kann auf diese in der Reflektion noch einmal eingegangen werden.

Schließlich sollte ähnlich wie bei Scrum zumindest eine Art Burnchart im System integriert sein – entweder ein Burnup- oder ein Burndown-Chart. Diese sind besonders wichtig für den Überblick über den momentanen Fortschritt im Sprint. Somit sollte schnell erkennbar sein, falls das Team nicht mehr in der Zeit liegt und möglicherweise der Release-Termin nicht eingehalten werden kann.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Prio | **Genius inside** | **Jira** | **Pivotal Tracker** | **Vision Project** | **Yodiz** |
| Iterationen | 5 | 2 | 2 | 2 | 2 | 2 |
| Releases | 3 | 2 | 2 | 2 | 2 | 2 |
| Rollen | 4 | 1 | 2 | 1 | 1 | 1 |
| Burncharts (Burnup oder Burndown) | 4 | 2 | 2 | 2 | 2 | 2 |
| Analyse der letzten Lieferung | 4 | 2 | 2 | 1 | 2 | 2 |
| Summe |  | 39 | 36 | 32 | 40 | 40 |

Tabelle 7: Crystal-Kriterien

### Nice to have – Kriterien

Da bereits beim Aufstellen und Ausfüllen der prozess-spezifischen Kriterien auffiel, dass auf Grund des begrenzten Wissens über die Produkte die Punkte sehr nahe beieinander liegen würde, wurde zusätzlich Tabelle 7 mit Wünschen der Teammitglieder gefüllt. Diese sollte herangezogen werden, falls es zu einem Unentschieden zwischen Systemen kam oder die Punkte sehr dicht beieinander lagen.

Die Frage nach der allgemein benutzerfreundlichen und intuitiven Oberfläche wurde zusammen mit der Übersichtlichkeit am höchsten priorisiert. Hier erhielten die Tools ihre Bewertung nach einer kurzen Nutzung der Demo, wenn möglich, und mit Hilfe von Screenshots und Videos. Insbesondere Drag-and-Drop-Möglichkeiten wurden mit 2 Punkten bewertet.

Das Vorhandensein eines Dashboards als Einstiegs- und Übersichtsseite über das Projekt wurde ebenso mit „sehr wichtig“ eingestuft, allerdings wurde hier lediglich das Vorhandensein einer solchen Seite bereits mit 2 Punkten belohnt.

Zusätzlich zu den bisher betrachteten Features, die sich hauptsächlich um die Darstellung auf einem Board und verschiedene Analysemethoden drehten, wurden bei den Nice-To-Have-Kriterien auch allgemeinere Projektmanagement-Kriterien aufgenommen. Dazu gehören sowohl ein Dokumentenmanagement als auch ein Wiki zu Dokumentationszwecken. Letzteres kann durch das Schreiben von Kommentaren an Tickets oder in die Tickets selber leichter ausgeglichen werden, wenn auch weit weniger übersichtlich. Ersteres wäre zum Beispiel hilfreich für verschiedene Planungsnotizen und allgemeine Dokumente oder Screenshots, die an Tickets angefügt werden sollen, zu denen sie gehören. Bei JIRA wird das Wiki von Confluence, einem separaten aber leicht mit JIRA kombinierbaren Produkt übernommen. Des Weiteren wurde die Möglichkeit des Time Tracking gewünscht, so dass kein zusätzliches Produkt verwendet werden müsste, um dies zu übernehmen. Außerdem wäre ein Kalender praktisch, um insbesondere die verschiedenen Meetings leicht ersichtlich zur Hand zu haben und eventuell sogar mit Hilfe des Tools zu organisieren. Dazu würden dann auch das Einladen der beteiligten Personen und das Festlegen der Räumlichkeit gehören. Confluence regelt dies ebenfalls anstatt JIRA.

Schließlich wäre es praktisch aus Programmierer-Sicht, wenn die Software die Integration von Versionsmanagement wie beispielsweise github, bitbucket oder Ähnlichem unterstützt. Damit könnten Tickets mit entsprechenden Commits verknüpft werden, was das Abarbeiten dieser Tickets oder auch das Bugfinden erleichtern könnte.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Prio | **Genius inside** | **Jira** | **Pivotal Tracker** | **Vision Project** | **Yodiz** |
| benutzerfreundliche/intuitive Oberfläche | 5 | 1 | 2 | 2 | 1 | 2 |
| Übersichtlichkeit | 5 | 2 | 2 | 1 | 2 | 2 |
| Dashboard | 5 | 2 | 2 | 2 | 2 | 2 |
| Dokumenten-  management | 3 | 2 | 0 | 1 | 1 | 2 |
| Wiki | 2 | 0 | 2 | 0 | 2 | 0 |
| *Kalender 🡪 z.B. Termine* | 2 | 2 | 2 | 2 | 2 | 2 |
| Time Tracking | 2 | 2 | 2 | 2 | 2 | 2 |
| *GitHub-Integration o.Ä.* | 3 | 0 | 2 | 0 | 0 | 2 |
| *Summe* |  | 35 | 48 | 36 | 48 | 54 |

Tabelle 7: Nice-To-Havae-Kriterien

### Ergebnis und Auswahl

Nach der Auflistung aller Kriterien und der Berechnung der Gesamtpunktzahl mit und ohne Nice-To-Have waren zwei Tools mit über 200 Punkten vorn (siehe Tabelle 8).

Der Sieger wurde JIRA. Dies liegt sehr wahrscheinlich daran, dass es neben Pivotal Tracker das einzige dem Team bekannte Produkt war und eine genauere Einschätzung der Funktionalität aus der Erfahrung heraus ermöglichte. Dadurch erhielt JIRA mehr Zweien als seine Konkurrenten.

Auf Platz 2 befindet sich Yodiz, ein Tool, welches seit dem ersten Blick auf seine eher bunte Oberfläche als sehr verschieden von JIRA auffiel. Laut der einzelnen Zwischenpunktzahlen sollte JIRA sich um einiges besser für Kanban eignen, während beide in allen anderen Kategorien außer der allgemeinen sehr nahe beieinander liegen. Da das Team Yodiz nur von der Demo her kennt, sollte hier ein wenig Zeit zur Einarbeitung gegeben werden.

Insgesamt wurden am Ende zwei doch sehr unterschiedlich aussehende Tools gewählt.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Genius inside** | **Jira** | **Pivotal Tracker** | **Vision Project** | **Yodiz** |
| allgemein | 42 | **52** | 36 | 46 | **46** |
| Scrum | 40 | **46** | 27 | 25 | **44** |
| Kanban | 20 | **46** | 17 | 32 | **28** |
| Crystal | 39 | **36** | 32 | 40 | **40** |
| **Gesamt 1** | **141** | **180** | **112** | **143** | **158** |
| nice to have | 35 | **48** | 36 | 40 | **50** |
| **Gesamt** | **176** | **228** | **148** | **183** | **208** |

Tabelle 8: Gesamtauswertung

1. <http://project-management-software.findthebest.com/> [↑](#footnote-ref-1)