



□ Stefan Glase

(stefan.glase@opitz-consulting.com)

ist als Senior Consultant im Bereich Anwendungsentwicklung für die OPITZ CONSULTING Deutschland GmbH tätig. Er beschäftigt sich seit mehreren Jahren mit der Architektur und Implementierung von Anwendungen im agilen Umfeld und setzt hierbei unter anderem auf das Spring Framework, Camunda BPM, Activiti und Groovy & Grails. Als Autor von Fachartikeln und Sprecher auf verschiedensten Konferenzen versucht er seine Begeisterung für die Softwareentwicklung mit anderen zu teilen.

Der Werkzeugkoffer für verteilte agile Teams

Wenn agile Softwareentwicklung und verteilte Teams aufeinandertreffen, sehen sich die Beteiligten vor neue Herausforderungen gestellt, um in einem agilen Team über die räumliche Entfernung hinweg erfolgreich zu planen, zu arbeiten und zu kommunizieren. Agile Softwareentwicklung zeichnet sich durch kurze und zeitnahe Rückkopplungsprozesse sowie ein zyklisches und nutzenorientiertes Vorgehen auf allen Ebenen aus – von der Programmierung, über die Teamarbeit bis zum Management. Schon in der Planung gibt es deutliche Unterschiede zur klassischen Vorgehensweise: Das neue System wird nicht mehr wie früher in seiner Gesamtheit und in allen Einzelheiten geplant und anschließend in einer langen Realisierungsphase umgesetzt. Stattdessen steht am Anfang eines agilen Projekts die Definition einer Vision, also die Festlegung und Gewichtung von Zielen für die neue Software, auf deren Basis die Planung der ersten Version erfolgen und die Entwicklung beginnen kann. In der Folge wechseln sich kurze Planungs- und Entwicklungsphasen über die gesamte Projektlaufzeit ab, in denen das Entwicklerteam auf Ergebnisse und Erfahrungen vergangener Phasen zurückgreift und notwendige Anpassungen und Korrekturen vornimmt. Damit berücksichtigt das Konzept die Tatsache, dass sich Anforderungen während der Projektlaufzeit noch ändern und zu Beginn eines Projektes teilweise noch gar nicht vollständig bekannt sind. Für das Team gilt es also, mit geeigneten Werkzeugen und Methoden trotz der räumlichen Distanz der Teammitglieder eine effektive Zusammenarbeit im Kontext der agilen Softwareentwicklung möglich zu machen. Aus guten Gründen sollen dabei auch weiterhin „Individuen und Interaktionen“ im Vordergrund stehen und durch „Prozesse und Werkzeuge“ (vgl. [AgM]) nicht eingeschränkt, sondern in ihrer Arbeit unterstützt werden.

Persistente Chaträume

Der übliche Bürotratsch vor der Kaffeemaschine oder der sogenannte „Flurfunk“ sind durch technische Hilfsmittel kaum zu ersetzen. Dennoch sollte dem Team eine in etwa vergleichbare Plattform zur Verfügung stehen. Das könnte zum Beispiel eine Konversationsmöglichkeit für alle Teammitgliedern in Skype oder Google Hangouts sein, die jederzeit zur Verfügung steht. Auch ein persistenter Chatraum in einem Tool wie Atlassian HipChat oder Microsoft Lync wäre denkbar.

Wichtige Kriterien dafür, dass eine solche Lösung funktioniert und vom Team angenommen wird, sind einfache Zugänglichkeit und Bedienbarkeit: Idealerweise gibt es verschiedene Clients (Website, Applikation, Smartphone App), über die an

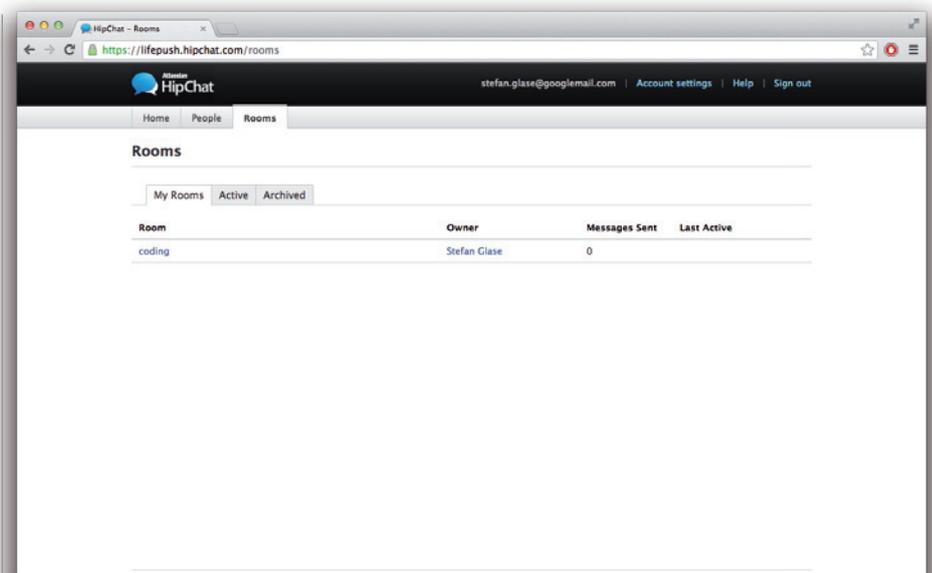


Abb. 1: HipChat

diesem Chat partizipiert werden kann, unabhängig davon, welches Betriebssystem die Teammitglieder auf ihrem Arbeitsplatzrechner installiert haben oder welches Smartphone sie besitzen (siehe Abbildung 1).

Ebenso wesentlich für eine effektive Kommunikation ist eine Option für die bedarfsgetriebene Eröffnung separater Chaträume. In diesen Chats können Mitarbeiter Unterhaltungen führen, die nicht für die große Runde von Bedeutung sind und diese so aus dem Teamchat heraushalten. Ergebnisse dieser separaten Runden können in den Teamchat zurückgespielt werden.

Ein weiterer Aspekt, der bei der Auswahl des geeigneten Werkzeuges nicht unterschätzt werden sollte, ist eine für alle Teammitglieder einsehbare Chat-Historie, in der Kollegen vergangene Kommunikation nach einem Urlaub oder einer kurzfristigen Abwesenheit nachvollziehen und gegebenenfalls auf diese reagieren können.

Die genannten Hilfsmittel unterstützen die wichtigsten Anforderungen an persistente Chaträume und bieten darüber hinaus noch viele weitere Funktionen, wie zum Beispiel die Möglichkeit, Dateien auszutauschen, Gastzugänge einzurichten

oder weitere Informationsquellen wie Ticketsysteme oder Build-Systeme anzubinden. Individuelle Anforderungen und unternehmensspezifische Regelungen entscheiden schlussendlich darüber, welches Werkzeug für ein Team am besten geeignet ist.

Sprach- und Videokonferenzen

Neben dem geschriebenen Wort sollte auch die Kommunikation über Sprache und Video nicht untergehen. Über gesprochene Sprache lassen sich Emotionen und Gefühle einfacher transportieren als über geschriebenen Text. Ein Videobild verstärkt diese Wirkung. Für persönliche Bindungen zwischen den Teammitgliedern kann dies entscheidend sein. Unter anderem mit Google Hangouts, Skype und Microsoft Lync lässt sich textbasiertes Messaging mit Sprach- und Videokonferenzen verbinden.

Beim Daily Scrum rückt das Team mittels Kopfhörer und Webcam trotz der räumlichen Trennung näher zusammen. Neben vielen positiven Aspekten bringt die direkte Kommunikationsform aber natürlich auch Herausforderungen mit sich, zum Beispiel die Frage, wie man zurückhaltenderen Teammitgliedern in Dis-

kussionen genügend Raum zugestehen kann.

Sprecher-Token

Um den Redeanteil gleichmäßig zu verteilen, wird bei Daily Scrums, die vor Ort stattfinden, oftmals ein „Sprecher-Token“ genutzt. Das kann ein Ball sein, der von einem Teilnehmer zum nächsten gereicht wird. Auch wenn sich ein Ball schlecht zwischen den Teilnehmern einer Audio- oder Videokonferenz hin- und herwerfen lässt, kann das Prinzip in diese Medien übertragen werden: So kann der Scrum Master, nachdem er seine Informationen ins Team gegeben hat, das Wort an den ersten Sprecher übergeben. Dieser benennt dann, nachdem er seine Informationen ins Team gegeben hat, den nächsten Sprecher und so fort.

Wenn der Scrum Master die Teammitglieder dazu anhält, nicht jedes Mal den gleichen Nachfolger zu benennen, sorgt er für einen echten, ausgeglichenen Austausch. Dabei ist zusätzlich von Vorteil, dass sich der oder die Einzelne nicht auf eine feste Reihenfolge verlassen kann und aufpassen muss, an welches Teammitglied er oder sie nach dem eigenen Beitrag die Redezeit übergeben kann.

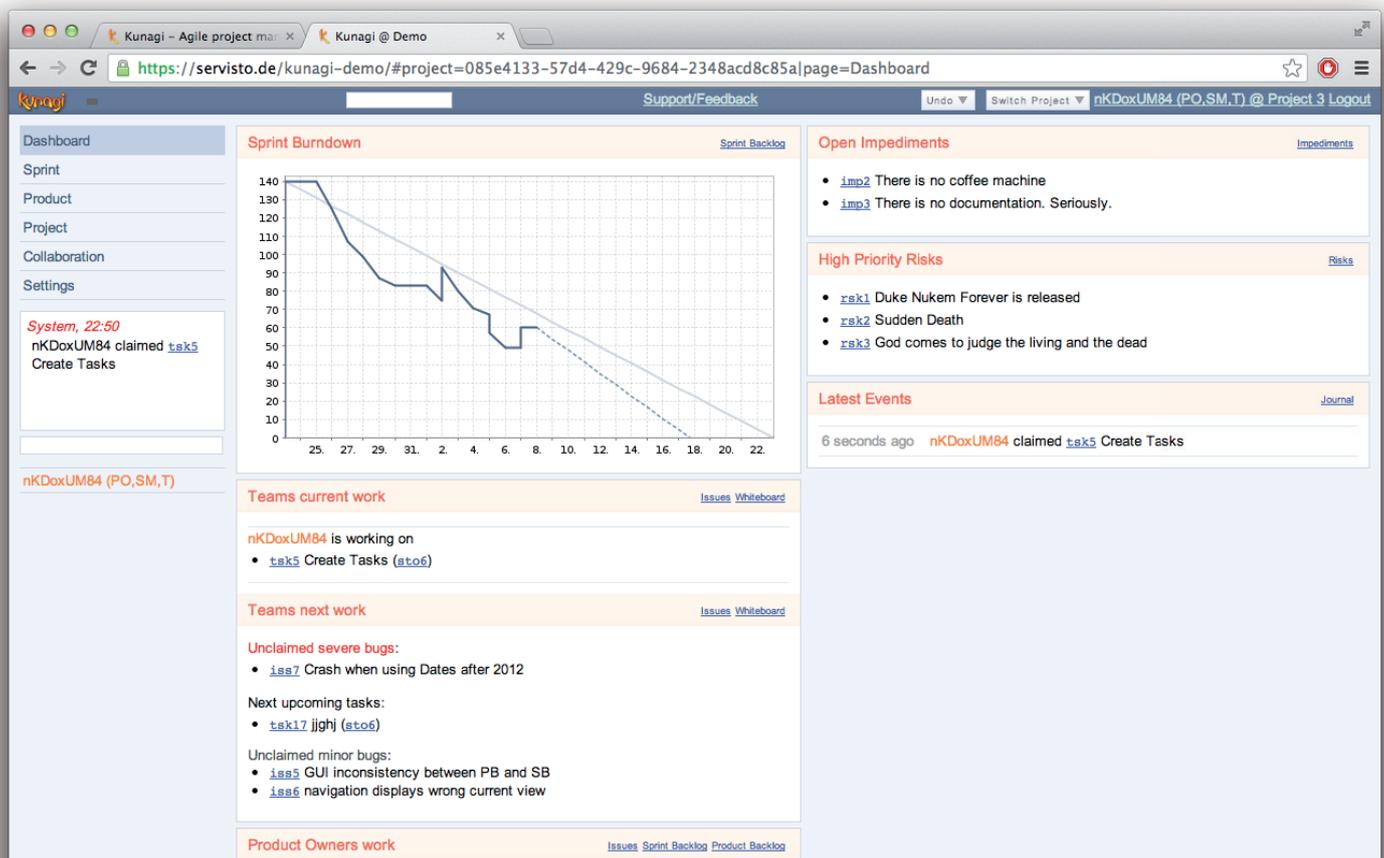


Abb. 2: Kunagi

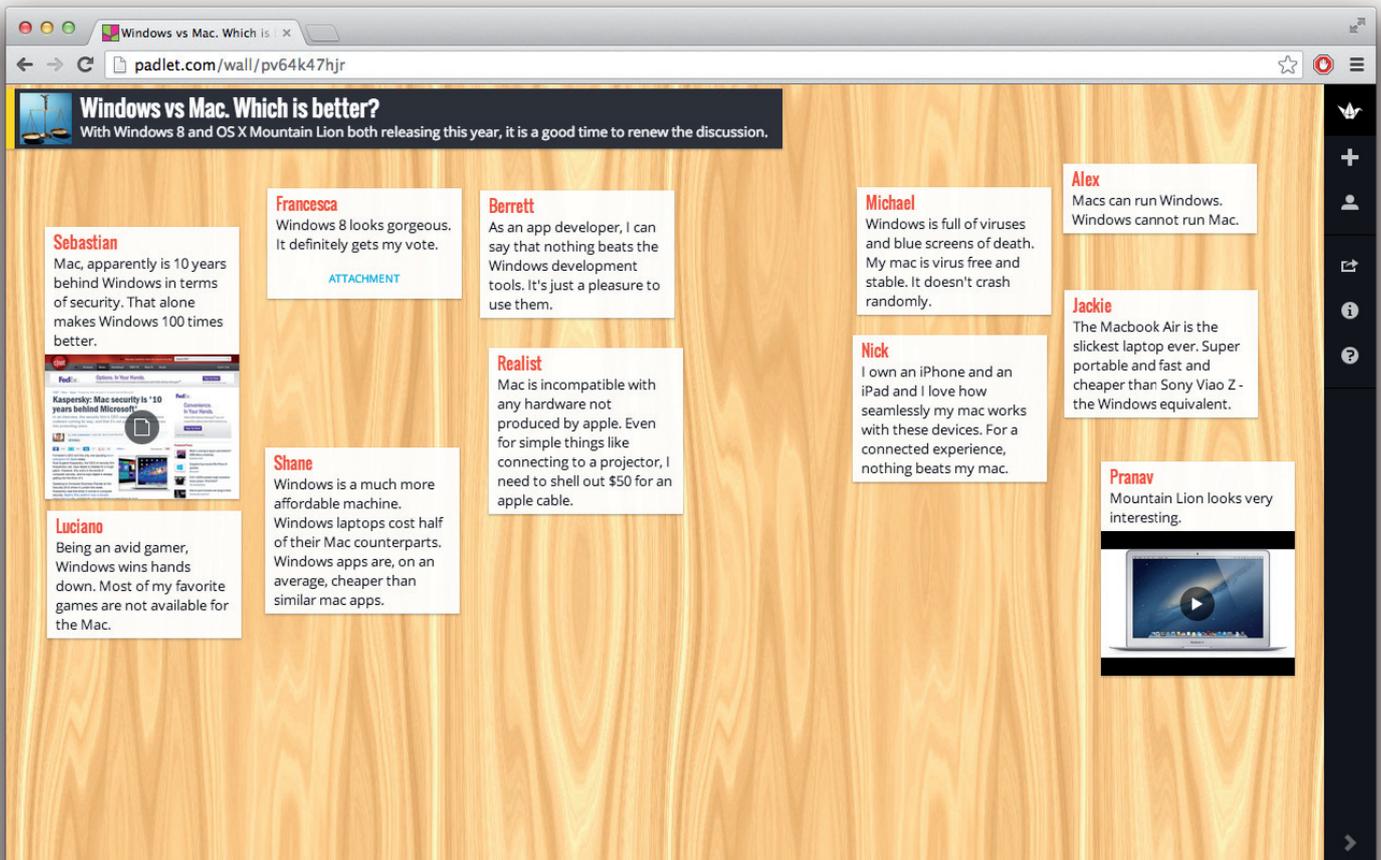


Abb. 3: Padlet

Task Board & Product Backlog

Damit das Team während eines Sprints den Überblick über die noch offenen, die in Arbeit genommenen und die bereits erledigten Aufgaben behält, ist ein Task Board hilfreich. Auf dem Task Board organisiert sich das Team. Im Sprint Planning ziehen die Entwickler User Stories aus dem Product Backlog in den zu planenden Sprint. Dabei definieren sie gemeinsam Aufgaben und Tätigkeiten in Form von Tasks, welche zur Erledigung der User Story und der damit verbundenen Akzeptanzkriterien erforderlich sind.

Technisch unterstützen lassen sich Product Backlog und Task Board auf unterschiedlichste Art und Weise: Denkbar sind dazu sogar einfache Google-Docs-Dokumente, die der Product Owner beziehungsweise das Team einsehen, editieren und kommentieren können. Mit einer einfachen Tabellarischen Darstellung lässt sich das Product Backlog in Listenform und das Task Board in einem Spaltenlayout aufbereiten und darstellen (siehe Abbildung 2).

Alternativ zu diesem Ansatz stehen Werkzeuge wie Kunagi, einer Open Source Projektmanagement-Applikation mit Fokus auf den Scrum-Prozess, oder das in vielen Unternehmen bereits eingesetzte kommerzielle Atlassian Jira, welches zusammen

mit der Erweiterung Jira Agile den agilen Softwareentwicklungsprozess unterstützt. Beide Werkzeuge ermöglichen es den verteilt arbeitenden Teammitgliedern, das Product Backlog und die darin enthaltenen User Stories zu jedem Zeitpunkt des Sprints einzusehen. Auch auf das Task Board, das den Fortschritt im aktuellen Sprint reflektiert, ist der Zugriff möglich.

Eine sehr interessante Variante für Teams, die auf zwei Standorte verteilt sind oder Teams, in denen lediglich einzelne Personen „remote“ arbeiten, und die dennoch nicht auf ein physikalisches Taskboard verzichten wollen, ist die Hybridlösung JimFlow von der Firma Jimdo, die die Tasks im digitalen System über QR-Codes auf dem physikalischen Board verfügbar macht.

Burndown Charts

Der Fortschritt in einem Sprint wird in der Regel über sogenannte Burndown Charts visualisiert. Auf diesen werden über den zeitlichen Verlauf eines Sprints entweder Story-Punkte oder die Anzahl der fertiggestellten Stories abgetragen. So wird gut ersichtlich, wie viel Arbeit bereits erledigt ist und wie viel Arbeit bei gleich bleibender Geschwindigkeit noch geschafft werden kann.

Wo ein Vor-Ort-Team zu Stift, Schere und Papier greift und das eigene Burndown Chart visualisiert, wird ein verteiltes Team eine elektronische Lösung wählen.

Die beiden oben genannten Werkzeuge für die Verwaltung des Task Boards Atlassian Jira und Kunagi bieten von Haus aus Burndown Charts an, da alle dazu notwendigen Informationen schon im System vorhanden sind.

Pair Programming

Eine viel diskutierte und oft erfolgreich praktizierte Methode im Kontext der agilen Softwareentwicklung ist das sogenannte „Pair Programming“. Hierbei arbeiten zwei Personen bei der Erstellung von Programmcode am gleichen Rechner zusammen. Einer der beiden erzeugt als „Fahrer“ den Code, während der „Beifahrer“ über Problemstellungen nachdenkt, den erzeugten Code kontrolliert und Probleme anspricht.

Auch hier findet das verteilte Team Lösungen und Möglichkeiten, um als Tandem am Programmcode zu arbeiten. Eine oft gewählte Variante ist der Einsatz von Screen-Sharing-Programmen wie Cisco Webex, TeamViewer, join.me oder Skype. Mit diesen Programmen kann der jeweili-

ge „Beifahrer“ die Aktionen des „Fah- rers“ auf seinem Bildschirm beobachten und sich mittels Sprach- und Videokonferenz mit diesem austauschen. Je nach Werkzeug können die Rollen auch wechseln und der „Beifahrer“ greift dann als „Fahrer“ zur Tastatur.

Damit sich Wissen und Fähigkeiten besser im Team verteilen, eignet sich die Verwendung einer Pair-Programming-Matrix, welche die bisherigen Paarzusammenstellungen visualisiert und zu neuen Kombinationen motiviert. Eine solche Matrix lässt sich gut in einem für alle Teammitglieder zugreifbaren Dokument, beispielsweise in der Dropbox, in Microsoft Sharepoint oder auf Google Docs, ablegen.

Whiteboards und Mindmaps

Die beiden erstgenannten Screen-Sharing-Tools Cisco Webex und TeamViewer bieten noch weitere Möglichkeiten für die gemeinsame Kreativarbeit. In beiden Werkzeugen können die Anwender Gedanken, Meinungen und Ideen in Form von Bildern und Texten auf einem digitalen Whiteboard äußern. Anbieter wie Padlet haben sich auf diesen Aspekt spezialisiert und bieten für die Zusammenarbeit On-

line-Whiteboards an, auf denen das gesamte Team digitale Post-its anheften, verschieben und gruppieren kann (siehe **Abbildung 2**).

Als Alternative zu einem Whiteboard kann das agile Team zur Ideenfindung und Strukturierung in intensiven Brainstorming-Sessions auch auf Mindmapping-Tools zurückgreifen. Die kostenlosen Angebote von MindMup oder Mindmeister sind dabei nur zwei von vielen Onlineangeboten, bei denen gleichzeitig an ein und derselben Mindmap gearbeitet werden kann.

Serious Games

Wenn Mindmaps und Whiteboards allein nicht genug Kreativität hervorbringen oder diese Werkzeuge bei den Teilnehmern nur noch ein müdes Gähnen hervorrufen, so stehen noch alternative Methoden und Techniken zur Verfügung. Dazu gehören Serious Games. Unter Serious Games versteht man Spiele, die nicht primär der Unterhaltung dienen, sondern auf ein ernsthaftes Ziel ausgerichtet sind, das sie auf spielerische Art und Weise erreichen möchten.

Die Webseite InnovationGames.com widmet sich dieser Art von Spielen und

bringt sie mit den „Instant Play Games“ kostenfrei ins Web. Das Angebot umfasst unter anderem Spiele zur Priorisierung und Bewertung eines Product Backlogs, zur Identifikation von Kundeninteressen oder zur Kommunikation von konstruktivem Feedback.

Retrospektiven

Ein wichtiges Element im Scrum-Prozess im Sinne des Prinzips „Inspect & Adapt“ (deutsch: „Überprüfen und Anpassen“) ist die Sprint-Retrospektive. Sie bietet dem Team einen Rahmen für die eigene Kontrolle, für die Identifikation von Optimierungspotenzialen in der zukünftigen Zusammenarbeit und die daraus folgende Ableitung konkreter Aktionen.

Die Sprint-Retrospektive findet in der Regel nach dem Sprint Review statt, in dem das Team die Sprintergebnisse präsentiert, und vor der Planung des nächsten Sprints. Bildlich gesprochen stellt sie also die Brücke zwischen zwei Sprints dar. Die Retrospektive unterteilt sich üblicherweise in fünf Phasen. Durch diese Phasen lenkt der „Facilitator“ das Team mittels geeigneter Methoden. In der Regel tritt der Scrum Master in der Rolle des Facilitators auf und hilft dem Team in der Ret-

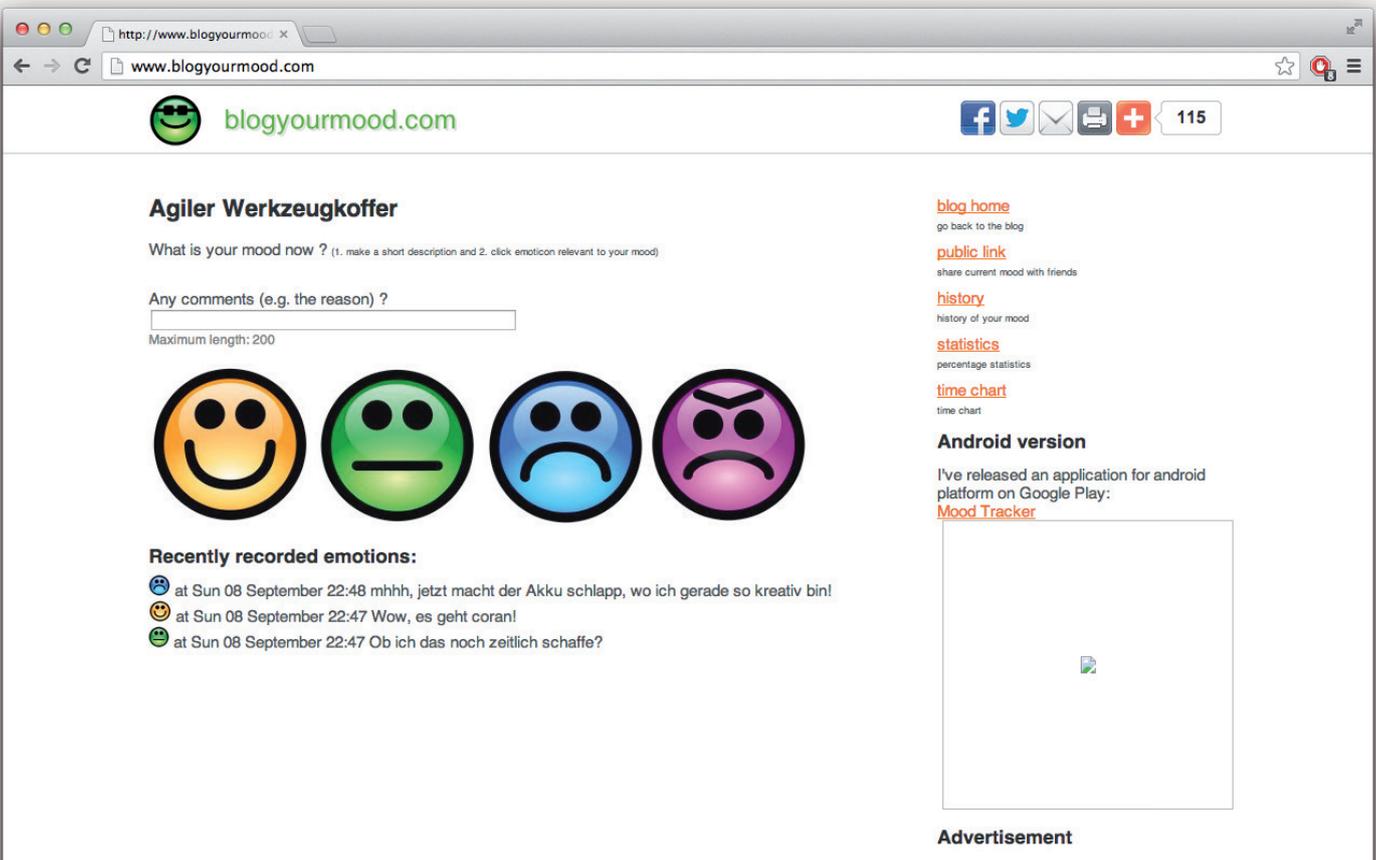


Abb. 4: BlogYourMood

rospektive effektiver zu arbeiten, indem es seine eigenen Probleme gemeinsam löst.

Auch die Retrospektive eines verteilten Teams kann mit den passenden Werkzeugen durchgeführt und unterstützt werden. Es gilt, die richtige Kombination aus digitalen Whiteboards, Mindmaps und Serious Games zu wählen, um das Team mit Sprach- und Videounterstützung zusammenzubringen und mit wechselnden kreativen Mitteln zu befähigen, sich selbst zu überprüfen und wo nötig anzupassen.

Wikis

Ein Wiki – sei es ein MediaWiki, wie wir es von Wikipedia kennen, oder ein Atlassian Confluence oder Microsoft Sharepoint – hilft verteilten wie lokalen Teams dabei, Wissen festzuhalten und über verschiedene Zeitzeonen hinweg zu transportieren. Ideal eignen sich Wikis auch zur Dokumentation eines gemeinsamen Softwareprodukts. Dabei reduziert sich die Aktivität der Handbucheerstellung auf wenige oder sogar auf nur einen Bearbeiter. Auf ähnliche Weise unterstützen Google-Docs-Dokumente die verteilte und sogar die parallele Bearbeitung von Dokumenten.

Teamkalender

Neben Textdokumenten, Bildern und Datentabellen gibt es noch andere Dinge, die einem Team helfen, sich selbst zu organisieren und im agilen Alltag den Durchblick zu behalten.

Ein Teamkalender hilft dem Team, die verfügbaren Ressourcen während der Sprintplanung zu bewerten und Urlaube oder andere Termine, die das Team betreffen, zu berücksichtigen. Ein Teamkalen-

der lässt sich idealerweise mit bestehenden persönlichen Kalendern verknüpfen. Nur ein paar der vielen möglichen Varianten sind ein geteilter Google Calendar, ein kommerzielles Produkt wie die Atlassian Team Calendars oder ein kostenfreies Produkt wie MrTickTock.

Niko Niko

Auch um Kalender, dabei aber weniger um Termine sondern mehr um Zufriedenheit und Wohlbefinden der Teammitglieder, geht es bei der Methode Niko Niko (japanisch für Smiley). Bei dieser Methode sind die Teammitglieder aufgefordert, ihr persönliches Wohlbefinden, beispielsweise am Ende eines Arbeitstages, in Form eines Smileys und gegebenenfalls mit einem kurzen erklärenden Kommentar sichtbar zu machen ([siehe Abbildung 4](#)).

Auch hierfür gibt es entsprechende Angebote in der Bandbreite von professionell und kommerziell (beispielsweise NikoNiko) bis hin zu minimalistisch und kostenfrei (wie zum Beispiel BlogYourMood).

Fazit

Es existiert eine wahre Flut von Werkzeugen und Tools mit unterschiedlichster Qualität und verschiedensten Preisvorstellungen sowohl von bereits am Markt für ähnliche Werkzeuge etablierten Anbietern, die auf den agilen Zug aufspringen, als auch bei ganz neuen Anbietern, die sich auf agile Werkzeuge konzentrieren. Darunter finden sich wertvolle Werkzeuge, die den agilen Alltag von verteilten Teams effektiv unterstützen. Einige dieser Hilfsmittel, die sich in meiner Projektarbeit als besonders hilfreich erwiesen ha-

ben, habe ich in diesem Artikel erwähnt.

Wichtig ist die richtige Wahl und Dosierung solcher Werkzeuge. Auch darf die Forderung des agilen Manifests, die besagt, dass „Individuen und Interaktionen“ vor „Prozesse und Werkzeuge“ zu stellen sind, nicht außer Acht gelassen werden. Gar zu schnell rücken Werkzeuge sonst so stark in den Vordergrund, dass der eigentliche Grund ihres Einsatzes vergessen wird.

Unser Werkzeugkoffer ist also gut sortiert. Uns obliegt es nun, die richtigen Werkzeuge für die vorhandenen Probleme und Herausforderungen einzusetzen. ■

Links

[NiNi] <http://www.nikoniko.co>

[BYM] <http://www.blogyourmood.com>

[MTT] <http://www.mrticktock.com>

[InG] <http://www.innovationgames.com>

[Kun] <http://www.kunagi.org>

[Sky] <http://www.skype.com>

[AtI] <http://www.atlassian.com/software>

[JiF] <http://jimflow.jimdo.com>

[Pad] <http://www.padlet.com>

[AgM] <http://www.agilemanifesto.org>

[GoC] <http://www.google.com/calendar>

[TeV] <http://www.teamviewer.com>

[Web] <http://www.webex.de>

[GoH] <http://www.google.com/hangouts>

[MiL] <http://office.microsoft.com/de-de/lync>

[MiM] <http://www.mindmup.com>

[JoM] <http://www.join.me>

[MMe] <http://www.mindmeister.com>

[MWi] <http://www.mediawiki.org>