

Die innovative Idee der Voice App „Thalia Buchtipps“ kommt an!

Seit ein paar Monaten ist der Thalia Alex Skill erst online und erfreut sich bereits einer kontinuierlichen Nutzung. Vor allem die abwechslungsreiche und neckische Art kommt bei den Kunden gut an und konnte nun auch die Jury des „Deutsche Exzellenz-Preis“ überzeugen.

<#0100/> <hackathon@thalia/> in Berlin

Ende Mai fand der 2. Hackathon in Berlin statt. Zum Thema waren diesmal alle neuen Technologien gesetzt.

Berlin macht Agil !



Dank der Unterstützung unseres Kollegen Matthias Hochschulz und seiner Organisation [Münster macht Agil](#), konnten wir gemeinsam im August auch in

Berlin einen kostenlosen Scrum Wokshop organisieren. Ursprünglich getrieben von der Notwendigkeit einer Reihe von neuen Kollegen die Grundwerte und -prinzipien von Scrum nahezubringen, entwickelten wir gemeinsam die Idee den Workshop auch für externe zu öffnen. So konnten wir am Ende fast 30 Interessierten aus und um Berlin dazu ermutigen, an einem realen Produkt, mehrere kurze Sprints zu durchlaufen.

Vorallem der Schritt zu einer öffentlichen Veranstaltung war für uns neu. Passte aber sehr gut zu unseren Bemühungen unsere Kompetenz in der Softwareentwicklung nach außen zu tragen. Für viele passt das Bild eines modernen agilen Softwarehauses nicht zu dem angestaubten Bild eines Buchhändlers. Daher haben wir uns sehr gefreut, das wir zeigen konnten, das dieses Bildniss in der Tat sehr veraltet ist und wir eine langjährige Erfahrung im eCommerce und der agilen Softwareentwicklung haben.

Nach einer kurzen Einführung (Stichworte: Sprint, Review, Retrospektive, Daily-Stand-Up, Inspect & Adapt, Fokus und Transparenz) ging es direkt ans *Selbermachen*! Passend zum heißen Sommer sollten die Teams eine Wasserpistole aus einem Akkuschrauber bauen. Diese konnte über mehrere Sprints verbessert werden. Das ermöglichte, ganz im agilen Sinne, aus den ersten Erfahrungen zu lernen und auf neue Anforderungen zu reagieren.

Nach dem erfolgreichen Event konnten wir noch alle Beteiligten noch auf ein Bier und Pizza in unsere Büroräume einladen und uns so für künftigen Erfahrungsaustausch vernetzen.

[Link zum Xing Event](#)

Nachtrag:

Matthias hat [hier](#) zusätzlich einen sehr umfangreichen Erfahrungsbericht zu dem Workshop verfasst. Dieser reflektiert sehr ausführlich die Erfahrungen die Matthias während des Workshops als Moderator gesammelt hat.

Thalia auf der jSpirit Unconference

Dieses Jahr haben wir an der gerade erst gegründeten jSpirit Unconference teilgenommen. Der Teilnehmerkreis erreicht zwar noch bei weiten nicht die geplanten 100 Personen, setzt sich dafür aber aus sehr guten, erfahrenen Entwicklern zusammen. Zum einen kennen sich einige schon länger von diversen Konferenzen in Deutschland, Basel, Moskau, Korea, Japan und USA und hält teilweise selbst mehrere dutzend Vorträge im Jahr oder organisiert Veranstaltungen (z.B. die JavaLand oder den JUG Oberland) und zum anderen wir die Runde ergänzt durch sehr erfahrene Entwickler aus der Region Bayern, aber auch Teilnehmer aus der Schweiz und der USA waren mit dabei.

Für mich war das unconference Format noch neu und daher war ich sehr gespannt, ob eine Konferenz ohne Agenda überhaupt etwas bringt. Aufgrund des zwanglosen Formats kommt man aber sehr schnell in einen direkten Austausch mit den anderen und so zu einem sehr viel intensiveren Erfahrungsgewinn. So wurde ich dann auch völlig überrumpelt, wie intensiv der Austausch und die Zusammenarbeit von der ersten Minute ab war.

Die Themen wurden morgens vor der ersten Session von allen Teilnehmern eingebracht und per Abstimmung auf die freien Timeslots verteilt. Dadurch waren für alle durchgehend spannende Themen mit dabei. Wie bei Openspaces überlich, konnte man auch jederzeit innerhalb einer Session zur anderen Gruppen wechseln, wenn man zu dem aktuellen Thema nichts mehr beisteuern konnte. Innerhalb der einzelnen Sessions gab es einen sehr guten Austausch der Erfahrungen und best practices der einzelnen Teilnehmer. So war es ein echter Vorteil, dass nicht nur ein Vortragender etwas erzählte sondern ein direkter Austausch stattfand. Dadurch konnten die einzelne Vorschläge schnell bestätigt oder mögliche Alternativen genannt werden.



Ein kurzer (unvollständiger) Auszug aus den Themen, die besprochen wurden:

- * Build Tools - Maven vs Gradle ?
- * Next thing to learn ?! (Go, Rust, Pony, AI ?)
- * Functional Patterns with Core Java
- * Java After Eighth (9,10,11)
- * Java in the Cloud
- * BDD - But Test suck
- * Dealing with difficult people on work
- * Speaker Tips (Abstract, Topics, HowTo)
- * Ideas for teaching Kids
- * Information flood
- * Mob - Programming

Für mich war das inhaltlich eine der besten Konferenzen auf denen ich bisher je

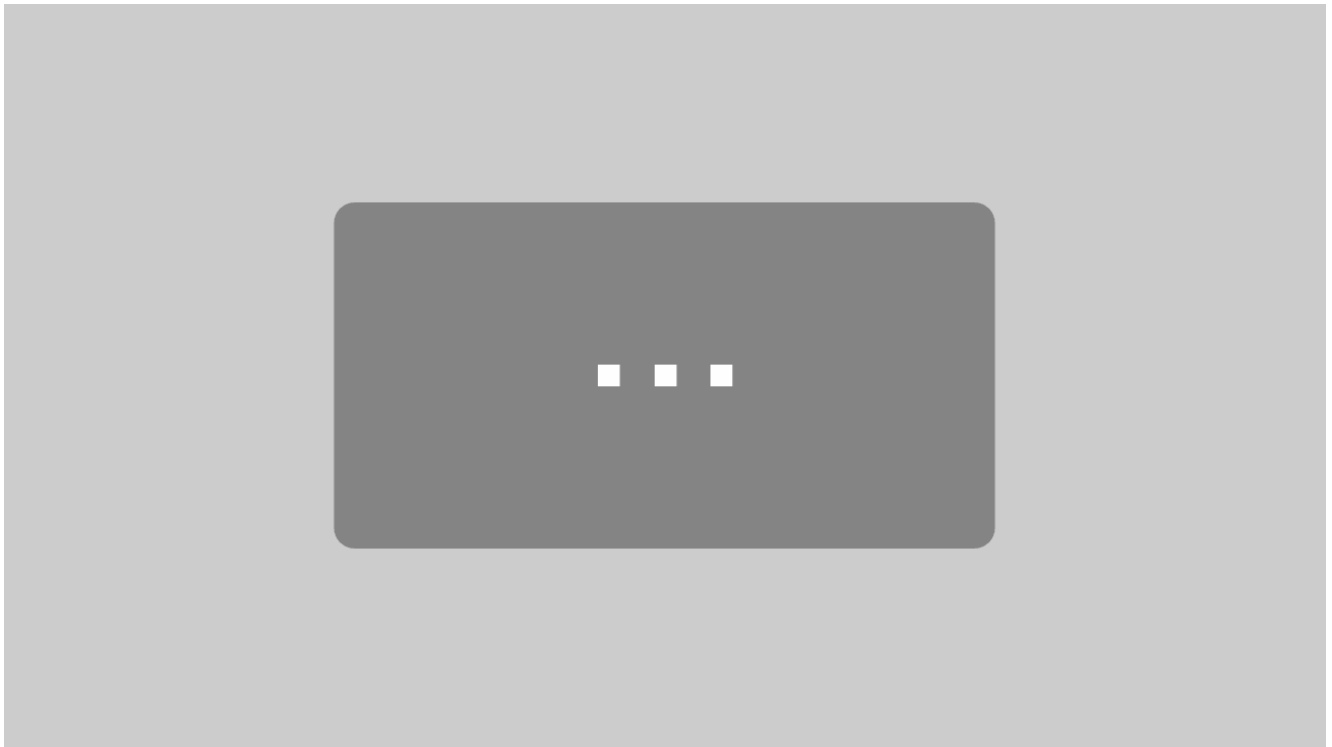
war.

Ergänzt wurde die Veranstaltung durch ein sehr gutes Rahmenprogramm. So bot sich aufgrund der Lokation natürlich die Gin & Whiskey Verköstigung inkl. Führung durch die Destillerie an, aber auch ein Ausflug zur Alm mit anschließender Rodeltour gehörten mit dazu. Das Programm sorgte dafür, dass der Teilnehmerkreis vollständig bis in die späten Abendstunden zusammenblieb. So war ein fachlicher Austausch auch in diesem geselligen Rahmen weiter gegeben und brachte einige zusätzliche Erkenntnissgewinne.





Hier noch ein paar Links auf verschiedene Interviews Live von der jSpirit:

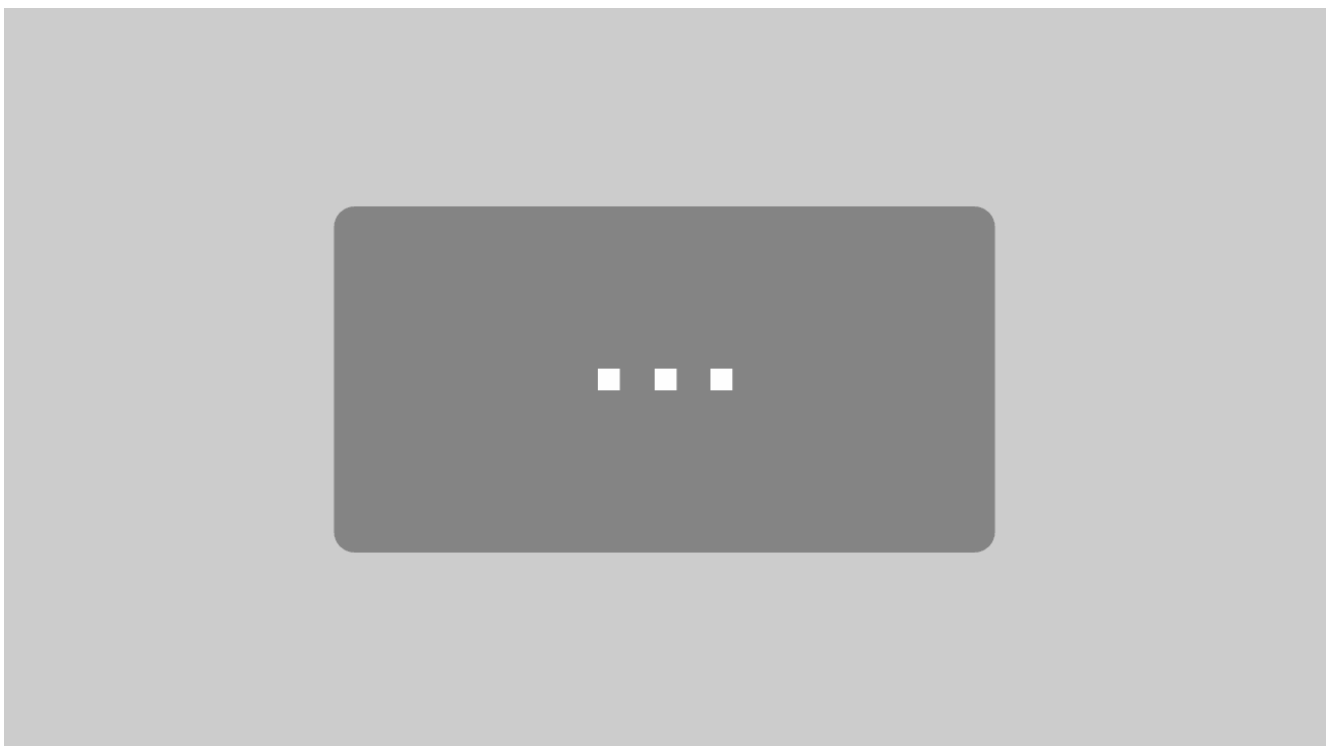


Mit dem Laden des Videos akzeptieren Sie die Datenschutzerklärung von YouTube.

[Mehr erfahren](#)

[Video laden](#)

☐ YouTube immer entsperren



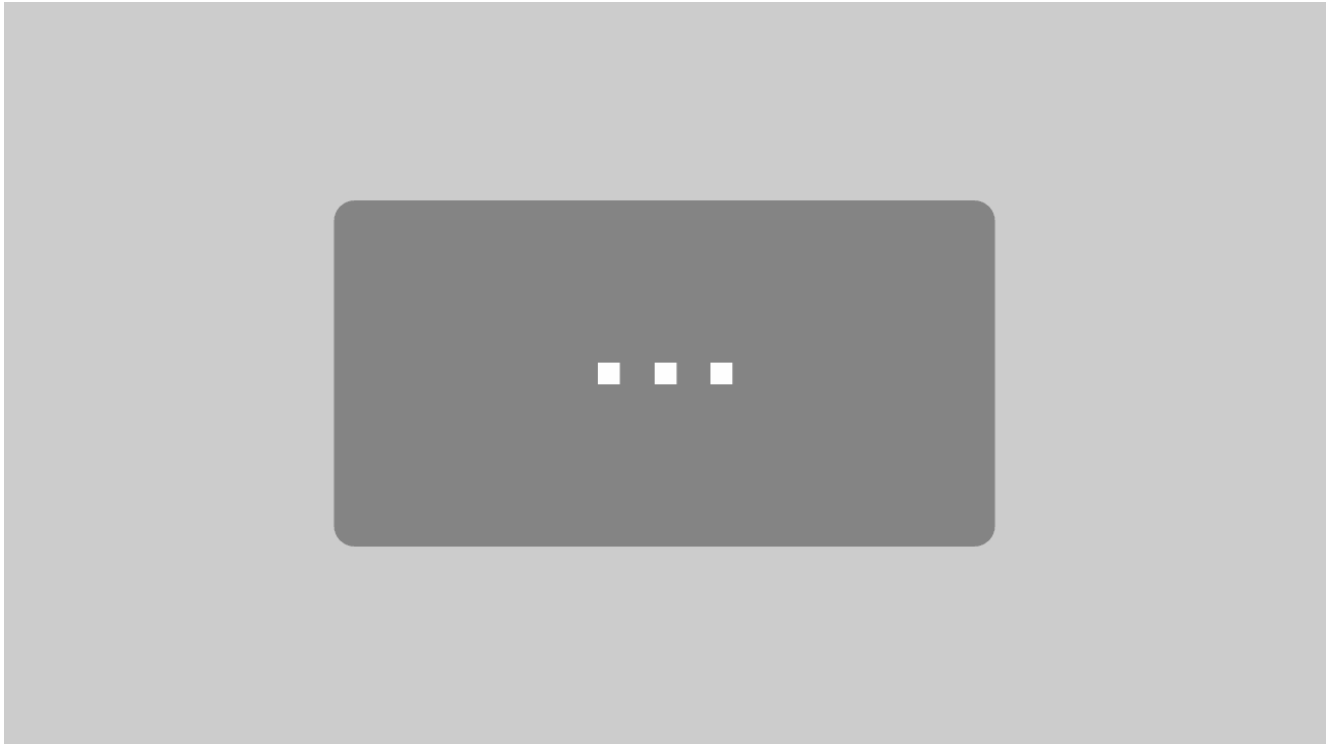
Mit dem Laden des Videos akzeptieren Sie die Datenschutzerklärung von

YouTube.

[Mehr erfahren](#)

[Video laden](#)

☐ YouTube immer entsperren



Mit dem Laden des Videos akzeptieren Sie die Datenschutzerklärung von YouTube.

[Mehr erfahren](#)

[Video laden](#)

☐ YouTube immer entsperren



Mit dem Laden des Videos akzeptieren Sie die Datenschutzerklärung von YouTube.

[Mehr erfahren](#)

[Video laden](#)

☐ YouTube immer entsperren



jSpirit ist eine neue Java Unconferenz in den Bayrischen Alpen
jspirit.org oder auf Twitter unter #jspiritunconv

Veranstaltungsort: www.lantenhammer.de

Kids Programming - Thalia fördert bereits die Jüngsten!



Nachdem wir in den letzten Jahren einige Sommerpartys und eine riesige Weihnachtsfeier hatten, hat sich unsere Geschäftsleitung entschieden, für Ende 2017 einen „Family Day“ abzuhalten. Und so durften nicht nur die Lebensgefährten sondern auch alle Kinder mal schauen, was Papa / Mama so im Büro machen. Es gab an den einzelnen Standorten ein umfangreiches Rahmenprogramm. Dabei

wurde stets versucht, auch einen gewissen Bezug zum Unternehmen zu wahren. So bot sich die Bücherecke mit Vorlesungen für Kinder natürlich an, was vor allem einige der jüngeren Kinder sehr genossen haben. Aber was macht man an einem Software Standort und wie erklärt man die Entwicklung von Programmen seinen Kindern?

Inspiziert von der „[Langen Nacht der Wissenschaft](#)“ in Berlin, wurde die Idee des Kids Programming geboren. An einzelnen Plätzen konnten so die Kinder bereits ab 4 Jahren in die Welt des Programmierens eingeführt werden und eine Idee entwickeln, was ein Software Entwickler eigentlich macht.

Die grundlegende Idee

Wir haben dafür mit Hilfe von code.org einige Programme für die Altersgruppe 4-6 und 6-12 rausgesucht und unterschiedliche Themen gewählt. Dadurch konnten wir später nicht nur Jungen sondern auch einige Mädchen für die Kurse begeistern. Für die fortgeschrittene Programmierung stellten wir darüber hinaus noch Scratch vom MIT zur Verfügung und konnten das Ganze durch einige, speziell für Kinder gestaltete Lehrbücher, unterstützen.

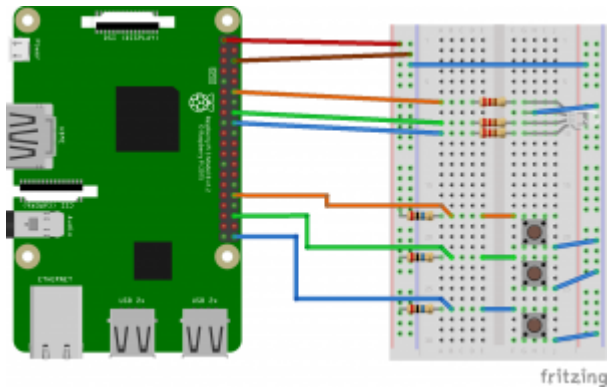
Darüber hinaus wollten wir bei den Kindern aber auch das Interesse für Softwareentwicklung wecken und sie neugierig auf mehr machen. Deshalb haben wir unsere Raspberry Farm ebenfalls zur Scratch Spielwiese umfunktioniert und

Programmieren mit code.org und Scratch

[illegible]

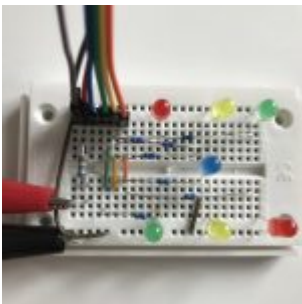
Als letzte Stufe haben wir den Kinder gezeigt, dass mit einfachen Programmen sogar elektronische Schaltungen gebaut und programmiert werden können. Zur Auswahl standen drei Beispiele:

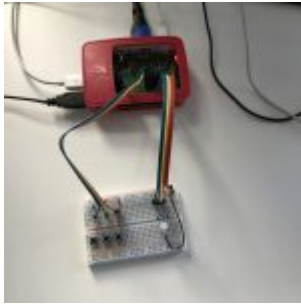
1. Steuerung einer RGB LED mittels dreier Taster,
2. Steuerung einer Fussgängerampel mit dazugehöriger Fahrzeug Ampel und
3. Steuerung eines RGB Würfels, der durch zwei Knetkontakte gesteuert wird.



Die Programmierung erfolgt dabei auch mit Hilfe von Scratch in einer speziellen Version für Raspberry Pi. In dieser Version können die Ein- & Ausgänge als Funktion in Scratch verwendet und gesteuert werden. Inspiriert wurden diese Beispiele durch das Buch „[Der kleine Hacker - Programmieren für Einsteiger](#)“. In diesem sind die Schaltungen und Programme sehr schön auch für unerfahrene Einsteiger beschrieben. Bis auf den Raspberry wird auch alles benötigte Zubehör mitgeliefert.







Wer Interesse bekommen hat, das Ganze mal selber auszuprobieren, wird hier fündig:



Christian Irenler

Der kleine Hacker: Programmieren für Einsteiger

Mit Scratch schnell und effektiv programmieren lernen

Der kleine Hacker

★★★★★

eBook
ab 14,99 €

Taschenbuch
29,99 €

Programmieren ist langweilig und trocken? Nicht mit dem kleinen Hacker! Die grafische Programmiersprache Scratch macht's möglich. Programmieren lernen mit Spaß. Egal, ob du nur die Katze tanzen lassen oder ein richtiges Spiel programmieren willst - Scratch eignet sich sowohl zum Einstieg in die Programmierung als auch für anspruchsvollere Projekte.

Links zu den

Seiten: code.org, scratch.mit.edu

Ein Team sieht Rot

Da das letzte Team Event, ihr erinnert euch: im [JumpHouse](https://www.jumphouse.de), schon wieder ein paar Tage zurück lag, war es Zeit für ein neues Event. Da passte es gut, dass wir herausfanden, dass manche noch nie Raclette gegessen hatten und andere noch nie den Film R.E.D.2 gesehen hatten! Was liegt näher als beides zu verbinden?! So entstand die Idee zum Kinoabend mit gemeinsamen Raclett essen und wir haben den Arbeitstag mit einem gemütlichen Kineoevent ausklingen lassen.



[R.E.D. 2 - Noch Älter. Härter. Besser](#)

[Trailer](#)

Berlin live for the code



07:00	Registration
08:00	Introduction to the Oracle Code Conference
08:30	Keynote: The Future of Cloud Native Development
09:00	Breakfast
09:30	Session 1: Introduction to Oracle Code
10:00	Session 2: Oracle Code in Action
10:30	Session 3: Oracle Code in Action
11:00	Session 4: Oracle Code in Action
11:30	Session 5: Oracle Code in Action
12:00	Lunch
12:30	Session 6: Oracle Code in Action
13:00	Session 7: Oracle Code in Action
13:30	Session 8: Oracle Code in Action
14:00	Session 9: Oracle Code in Action
14:30	Session 10: Oracle Code in Action
15:00	Session 11: Oracle Code in Action
15:30	Session 12: Oracle Code in Action
16:00	Session 13: Oracle Code in Action
16:30	Session 14: Oracle Code in Action
17:00	Session 15: Oracle Code in Action
17:30	Session 16: Oracle Code in Action
18:00	End of the Conference

Agenda - Berlin

Ein paar Kollegen aus allen Berliner Entwicklerteams, haben die Gelegenheit genutzt und die [Oracle Code](#) besucht. Natürlich zog sich Oracles neues Produkt, die Oracle Plattform Cloud, als roter Faden durch alle Präsentationen. Die Plattform möchte Oracle demnächst starten und bietet alles was das Entwicklerherz begehrt (von Source Verwaltung, Scrum Board, Ticket System, Container Orchestrierung bis zum Deploy). Allerdings lassen bereits die großzügig spendierten 300\$ Startkredit (für alle Teilnehmer zum ausprobieren) erahnen, das die Plattform nicht umsonst ist.

Erfreulicherweise gab es aber über die Plattform hinaus auch zahlreiche interessante Vorstellungen verschiedener Tools und Mechanismen. So zeigte Adam Bien sehr beeindruckend das eine Source Codeänderung inkl. Maven Build und Deploy per Docker nur eine Frage von Sekunden ist. Darüber hinaus gab es noch Einführungen in Kafka und Redis und jeweils eine Erläuterung wie

„Failover“ Lösungen bzw. ein redundanter Service aussehen könnte.
So bleibt am Ende ein spannender Tag der uns viele neue Anregungen mit in die Teams nehmen lässt.

Und für alle die nicht dabei sein konnten: [Download der Vorträge in New York](#)

<https://youtu.be/O3gMnVrOyuU>

Berliner Team lernt fliegen.



Wir in Berlin wollen hoch hinaus und suchen ständig nach neuen Ideen die hochgesteckten Ziele zu erreichen. Eine vollkommen neue Variante hat das Team bei einem gemeinsamen Besuch des [Berliner Jump House](#) ausprobiert. Wir hatten alle extrem viel Spaß und konnten völlig neue Höhenflüge erreichen.

Tipps & Tricks für JIRA als SCRUM Board

1. Einleitung

Die Software Entwicklung bei Thalia ist agil und wird von mehreren Teams per SCRUM durchgeführt. Zur generellen Verwaltung der einzelnen Aufgaben wird in der gesamten IT das Ticketsystem JIRA von [Atlassian](#) verwendet. Dieser Artikel soll einige Erfahrungen aus einem unserer Teams weitergeben, wie sich der SCRUM Alltag mit Hilfe von JIRA organisieren lässt. Er richtet sich dabei vor allem an Anwender, die bereits erste Erfahrungen mit SCRUM und JIRA haben.

Im folgenden sind einige Anregungen, wie ein SCRUM Team Daily Meetings digital abhalten kann, der PO beim Planning durch JIRA unterstützt wird bzw. die Durchführung der Meetings organisiert werden kann.

2. Daily

Für die Dailys verwendet das Team das Standard JIRA SCRUM-Board mit der Ansicht des aktiven Sprints. Dieses wird auf einem großen 40" Monitor dargestellt. Die einzelnen Team-Mitglieder können das Board während des Meetings direkt per Trackpad bedienen.

Vorteile:

- Das digitale Board wird den ganzen Tag angezeigt und gibt damit einen permanenten Überblick
- Einfache Zusammenarbeit
 - Tickets können während der Arbeit vom Platz aktualisiert werden.
 - Standortübergreifende Teams haben ein einheitlichen Blick auf das Board. Das war vorher bei der Verwendung von Pinnwänden schwierig.
 - Flexible Zusammenarbeit auch bei Teammitgliedern im HomeOffice

- Interaktion im Daily bleibt erhalten
 - Das Trackpad wird als „Staffelstab“ an den jeweiligen Redner übergeben
 - Tickets können schnell und einfach direkt während des Daily in andere Statusspalten geschoben werden.
 - Tickets können schnell und einfach direkt während des Daily anderen Personen zugewiesen werden.

3. Planung

Bei Thalia hat jedes Entwickler-Team sein eigenes JIRA-Projekt. Die Planung für unser Team erfolgt daher innerhalb unseres eigenen Projekts und dem dort zur Verfügung gestellten Backlog. In diesem werden die Tickets des Teams erfasst und verwaltet.

The screenshot shows the JIRA interface for Team RED. The top navigation bar includes 'Startseite', 'Projekte', 'Vorgänge', and 'Boards'. The main header area displays 'Team RED - Ideenspeicher', 'Backlog', and a search bar. Below the header, there are filters for 'SCHNELL-FILTER: Ideen, Neu, Frage, DoR, Refinement, Offen, In Work, !Sprint, UC, BW, BH, CMS'. The main content area shows a list of tickets in the backlog, including RED-332, RED-389, RED-145, RED-455, RED-39, RED-364, RED-654, RED-648, RED-626, and RED-429. Each ticket has a status icon, a title, and a priority level. The interface also shows a 'Sprint erstellen' button and a 'Board' dropdown menu.

Grundsätzlich lässt sich mit dem Backlog und einzelnen Sprints mit JIRA ohne weitere Veränderung arbeiten. In der alltäglichen Arbeit zeigt sich aber, dass die Bedürfnisse von PO und SCRUM Team in einigen Punkten abweichen und die Arbeit durch kleine Anpassungen erleichtert werden kann.

Das Backlog wird im Laufe der Zeit mit verschiedensten Tickets befüllt, die ab einer bestimmten Menge das SCRUM Team überfordern können. Es geht der

Überblick verloren, welche noch aktuell sind oder eventuell neu geschätzt werden müssen. Daher ist eine weitere Unterteilung unter Umständen hilfreich.

3.1 Stadien eines Tickets

Eine genauere Betrachtung der Tickets zeigt, dass diese während ihrer Lebenszeit verschiedene Stadien durchlaufen. Hierbei wird deutlich, dass es unterschiedliche Standpunkte bzw. Schwerpunkte gibt, was genau der Status eines Tickets ist.

Aus Sicht des Teams

Das Team hat vor allem die Bearbeitung des Tickets im Blick. Am Anfang ist das Ticket noch *Offen* und wird dann *Bearbeitet*. Von dort wandert es über ein *Review* in den *Test* bis es nach einer *Abnahme geschlossen* werden kann.

Bearbeitung: *Offen > In Bearbeitung > [Review] > Test > [Abnahme] > Geschlossen*

Aus Sicht des PO

Die Arbeit des PO fängt bereits lange vor der Bearbeitung durch das Team an. Es kommen neue Tickets ins Backlog, die eventuell auch von außerhalb des Teams kommen können und daher als *Neu* kenntlich sein müssen. Diese können ins Backlog aufgenommen oder abgelehnt bzw. verändert werden. Alle Tickets, die im Backlog sind, müssen zur *Feinplanung* an das Team gegeben werden und geschätzt werden. Dabei können sie als *Sprint tauglich* (Definition of Ready erfüllt) bewertet werden. Alle Tickets, die den DoR-Status haben, stehen damit für einen nächsten Sprint zur Verfügung.

Planung: *Neu > Ins Backlog aufgenommen > Feinplanung > Sprint Tauglich > In Sprint übernommen.*

Zwischenfazit

Es gibt also zwei Ebenen, die **Bearbeitung** und die **Planung**. Da Tickets aus einem laufenden Sprint auch wieder ins Backlog wandern können, existieren unter Umständen Tickets im Backlog, die bereits bearbeitet wurden und einen entsprechenden Status haben. Das zeigt, dass diese beiden Zustandstypen

parallel existieren können. Daher haben wir uns dafür entschieden, die Planungszustände nicht über den Status abzubilden, sondern als parallelen Zustand zu implementieren.

Im Folgenden wird daher gezeigt wie sich mit einfachen Mitteln in JIRA diese beiden Stränge parallel verwalten lassen.

3.2 Organisation

Die Organisation der Tickets findet in einem einzigen Projekt Backlog statt. **In diesem Backlog erfolgt die Zuordnung zu den einzelnen Phasen der *Bearbeitung* über den Ticket Status. Die Zuordnung zu den einzelnen Phasen der *Planung* erfolgt durch Stichwörter.**



Eine Aufteilung kann recht einfach durch „virtuelle Trennlinien“ in Form von Tickets mit Titeln wie „=====“ visualisiert bzw. organisiert werden. Da JIRA aber einige Eigenheiten beim Sortieren von Tickets aufweist, können sich schnell Tickets in andere Bereiche schummeln. Durch die Verwendung von Stichwörtern lässt sich das vermeiden.

Stichwörter

Das Backlog wird durch die Stichwörter in einzelne Planungsphasen aufgeteilt. Diese helfen bei der Organisation und unterstützen die zielgerichtete Vorbereitung der Scrum Meetings.



Ursprung	Ziel	Stichwort	Wer macht Was ?
Neu	Backlog	+ RED-Backlog	Der PO prüft alle neuen Tickets und verschiebt diese ins Backlog.

Ursprung	Ziel	Stichwort	Wer macht Was ?
Backlog	Refinement	+ RED-Refinement	PO und TL bereiten im Rahmen der Backlog Pflege das Refinement Meeting vor und verschieben ausgewählte, kurzfristig relevante Tickets ins Refinement. Der Refinement Bereich soll nach jedem Meeting wieder aufgeräumt werden.
Refinement	DoR	+ RED-DoR - RED-Refinement	Das Entwickler Team übernimmt alle vollständig geschätzten und geklärten Tickets in das Sprint Backlog. Aus diesem werden die folgenden Sprints im Planning Meeting befüllt.
Refinement	Refinement	+ RED-Question	Sollte das Entwickler Team ein Ticket nicht schätzen können, wird dies zur Klärung zurückgestellt und markiert.
Refinement	Backlog	- RED-Refinement	PO und TL räumen im Anschluss an das Refinement Meeting den Bereich wieder auf und versuchen die offenen Fragen zu klären

3.3 Boards

Für die alltägliche Arbeit stehen mehrere Boards zur Verfügung. Das Entwickler-Team nutzt dabei für die tägliche Arbeit nur das SCRUM Board, während der PO überwiegend im Ideenspeicher unterwegs ist. Als Schnittstelle zwischen beiden dient das Refinement Board.



Board	Beschreibung
Ideenspeicher	Das Board zeigt alle Tickets, die im gesamten Projekt existieren. Es kann über Filter eingeschränkt werden.
Refinement	Es werden nur die zu besprechenden Refinement Tickets angezeigt.
SCRUM Board	Es werden nur die aktuell geplanten Sprints angezeigt und alle Tickets, die reif für die Umsetzung sind.

3.4 Kleine Helfer im Alltag

Filter

Vor allem der PO steht vor der Herausforderung das Board stets aktuell zu halten, das Team mit priorisierten Aufgaben zu versorgen und einen Überblick über neue und abgeschlossene Tickets zu behalten. Das führt zu häufigen Wechseln zwischen Planung und Sprint Übersicht.

Um das zu vereinfachen, haben wir im Ideenspeicher-Board eine Reihe von Filtern eingerichtet, durch die es je nach Bedarf weiter eingeschränkt werden kann. Dadurch kann die Sichtbarkeit in den einzelnen Boards geprüft werden, ohne dass die Boards gewechselt werden müssen.

Filter	Regel	Zweck
Neu	Label not in (Backlog)	Zeigt alle neuen Tickets, die noch nicht geprüft wurden
Idee	Label not in (Refinement, DoR)	Zeigt nur Ideen, die noch nicht zur Umsetzung ausgewählt worden sind
Frage	Label = Question	Zeigt alle Tickets mit offenen Fragen
Refinement	Label = Refinement	Zeigt nur Tickets, die für das Refinement vorgesehen sind
DoR	Label = DoR	Zeigt alle Tickets im Sprint Backlog
Offen	Status = Open	Zeigt alle offenen Tickets
In Work	Status != Open	Zeigt alle nicht offenen Tickets

Filter	Regel	Zweck
!Sprint	Sprint != OpenSprint()	Blendet alle Tickets aus, die einem Sprint zugeordnet sind

Farbkennzeichnung der Tickets

Ein weiterer Aspekt hat sich in der Sichtbarkeit des Planungsstandes der Tickets gezeigt. Für den PO ist es wichtig zu wissen, welche Tickets beim Team liegen, neu sind oder wo noch Fragen offen sind. Um dafür nicht jedes Mal die Filter benutzen zu müssen, lässt sich der Status auch als Farbkennzeichnung realisieren.

Farbe	Filter	Beschreibung
orange	(labels not in RED-Backlog) or labels is Empty	Neue Tickets
rot	labels = RED-Question	Tickets mit Rückfragen aus dem Team
blau	labels = RED-Refinement	Tickets, die im Refinement Board für das nächste Refinement anstehen
grün	labels in (RED-DoR)	Tickets, die reif für die Umsetzung sind

Meetup@Thalia: Vortrag Johannes Mainusch: „Otto.de - wie die Titanic den Eisberg verfehlte“

Am Donnerstag war es endlich so weit, wir traten aus dem Schatten der internen Fachvorträge ins Rampenlicht der offenen MeetUp Kultur.

Die offene Kultur war für diese erste Veranstaltung auch prägend. Nicht nur, dass sie inmitten unserer Räumlichkeiten stattfand, sie war auch ein Beispiel für die enge Zusammenarbeit und Vernetzung der Branche.

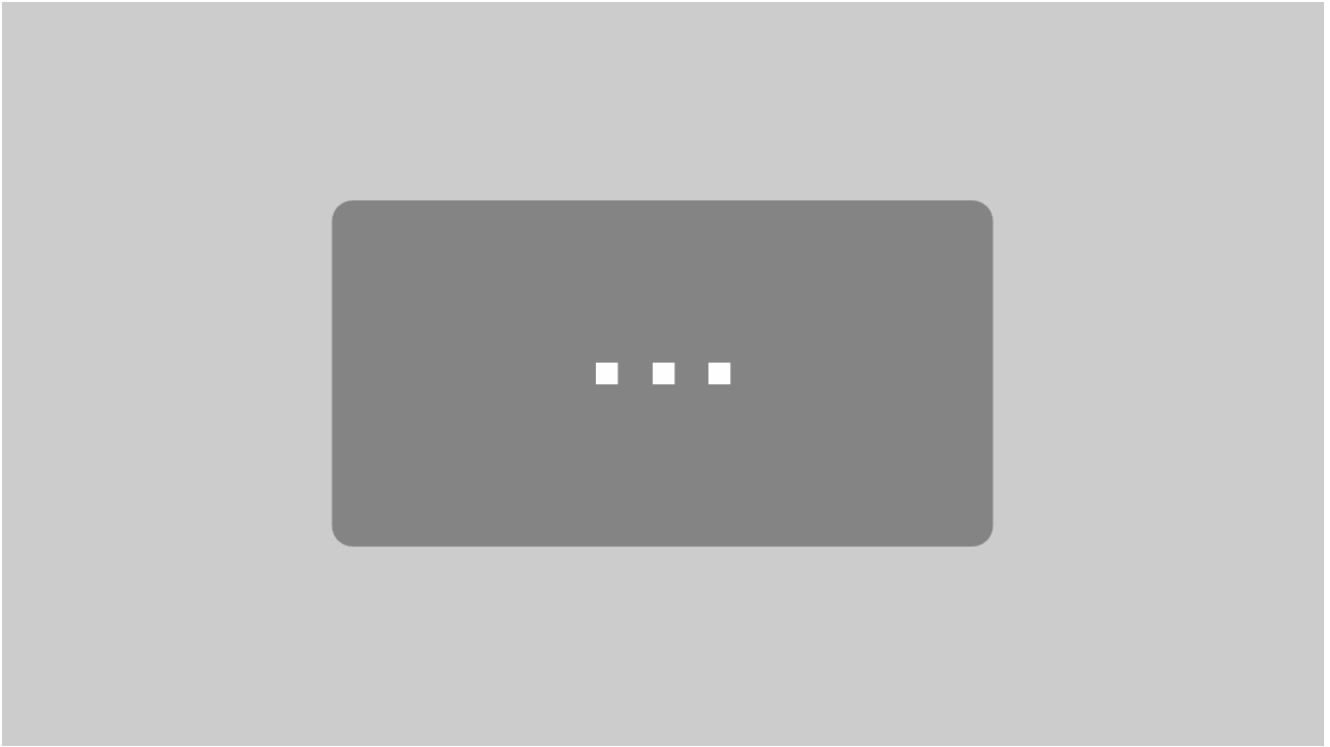
So konnten wir Johannes Mainusch dafür gewinnen seinen Vortrag „Warum die Titanic den Eisberg verfehlte“ in unserem Haus erneut zu präsentieren. Der Vortrag konnte bereits im Rahmen der Keynote der Continuous Lifecycle Konferenz begeistern.



Vortrag durch Johannes Mainusch

Der Vortrag wurde neben den Berliner Kollegen, auch von Kollegen aus Münster und Hagen, sowie einigen externen Gästen besucht. In rund zwei Stunden bekamen wir so einen spannenden Einblick, wie otto.de die langsame und schwer veränderbare Plattform modernisieren konnte. Aus einer 10 Jahre alten Intershop Installation mit mehr als 750.000 modifizierten Code Zeilen wurde ein Musterbeispiel für agile Entwicklung. Firmen wie Quelle und Neckermann waren unter ähnlichen Bedingungen in der Insolvenz gelandet. Otto.de konnte dagegen mit dem 2012 gestarteten Lhotse- Projekt zu einer der größten E-Commerce Plattformen in Europa wachsen. Welche Entscheidungen letztendlich dafür notwendig waren und was ein Unternehmen befähigt mehr als 340 Releases pro Woche Live zu bringen, schilderte Johannes Mainusch anschaulich und untermalt von vielen Beispielen.

Wer Interesse hat, kann sich den vollen Keynote Beitrag auch bei Youtube ansehen:



Mit dem Laden des Videos akzeptieren Sie die Datenschutzerklärung von YouTube.

[Mehr erfahren](#)

[Video laden](#)

☐ YouTube immer entsperren



Johannes Mainusch

ist seit April 2016 selbstständig bei der kommitment GmbH & Co. KG tätig. Vorherige Stationen waren unter anderem Lufthansa, Xing, Otto.de bis hin zur E-Post. Author des Titels [„Scrum mit User Stories“](#)



Start mit einem Pizza Buffet



Vortrag durch Johannes Mainusch



Diskussionsrunde nach dem Vortrag