

ÜBUNG SZETTEL PROZESSMANAGEMENT

AUFGABE 1: SCRUM-GRUNDLAGEN

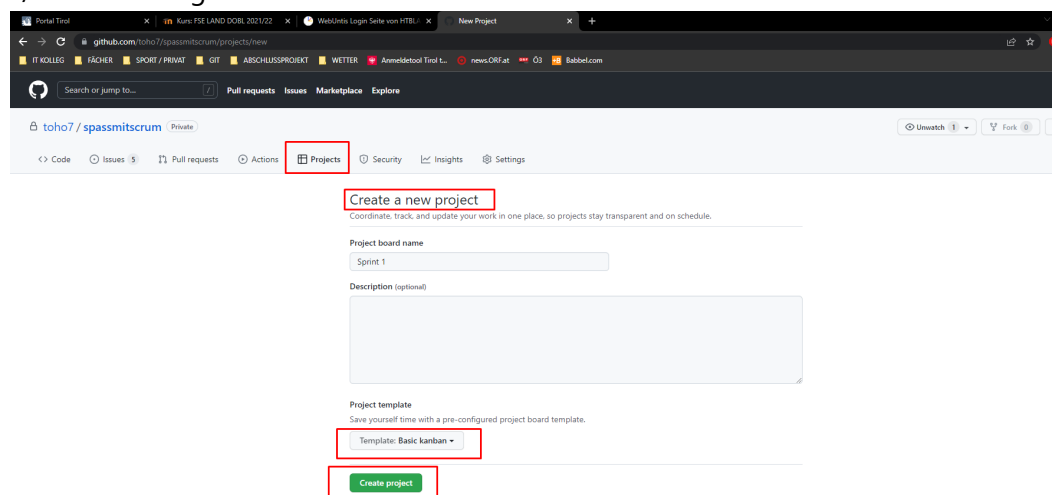
1. **Ablauf:** Wie laufen SCRUM-Projekte ab?
2. **Ereignisse:** Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective, Backlog Refinement
 - Sprint Planning: legt Arbeitspensum des Scrum-Teams für den kommenden Sprint fest
 - Daily Scrum: dauert 15 Minuten. Jedes Teammitglied beantwortet kurz drei Fragen:
 - Was habe ich seit dem letzten Daily-Scrum getan?
 - Was hat mich dabei behindert?
 - Was werde ich bis zum nächsten Daily-Scrum tun?
 - Sprint Review: Am Ende des Sprints präsentiert das Team dem Product-Owner die neue Funktionalität
 - Sprint Retrospective: Was haben wir gelernt? Was lässt sich verbessern?
 - Backlog Refinement: Definition von Anforderungen. Definition-of-Done. Aufwandsschätzung. Priorisierung. Ergänzung und Streichung von Einträgen.
3. **Rollen:** Scrum-Master, Product-Owner, Developer
 - Scrum-Master: Verantwortlich für Prozess und korrekte Implementierung. Vermittelt zwischen Team und Kunde.
 - Product Owner:
 - Vertretung der Interessen des Kunden
 - Zusammenarbeit mit dem Entwicklungsteam und dem Scrum Master
 - Verwaltung des Product Backlogs
 - Präsentation der Fortschritte
 - Entwicklung einer Vision des Endprodukts
 - Developer:
 - besteht aus Fachleuten: Software-Designer, Architekten, Business-Analysten, Dokumentationsexperten und Software-Tester
 - arbeiten an einem Produkt-Inkrement
 - jedes Mitglied liefert ein funktionales Element oder einen Teil des Gesamtproduktes
 - Geschäftsanforderungen verstehen
 - Entwicklung des Produkts
4. **Artefakte:** User Stories, Product Backlog, Sprint Backlog, Product Increment, Burndown-Chart
 - User Stories: informelle, allgemeine Erklärung eines Software-Features, die aus der Sicht des Endbenutzers verfasst wurden
 - Product Backlog: enthält alle bekannten Anforderungen an das System
 - Product Increment: Summe aller Product-Backlog-Einträge des aktuellen Sprints. Am Ende des Sprints muss das neue Inkrement in einem nutzbaren Zustand sein und der Definition-of-Done entsprechen.
 - Burndown-Chart: dient der Visualisierung bereits geleisteter und noch verbleibender Arbeit

Sie sollten nach der Bearbeitung einen guten Überblick über alle oben genannten Punkte haben. Insbesondere sollten Sie das Zusammenspiel zwischen SCRUM-Prozess, Ereignissen, beteiligten Rollen sowie resultierenden Artefakten gut beherrschen

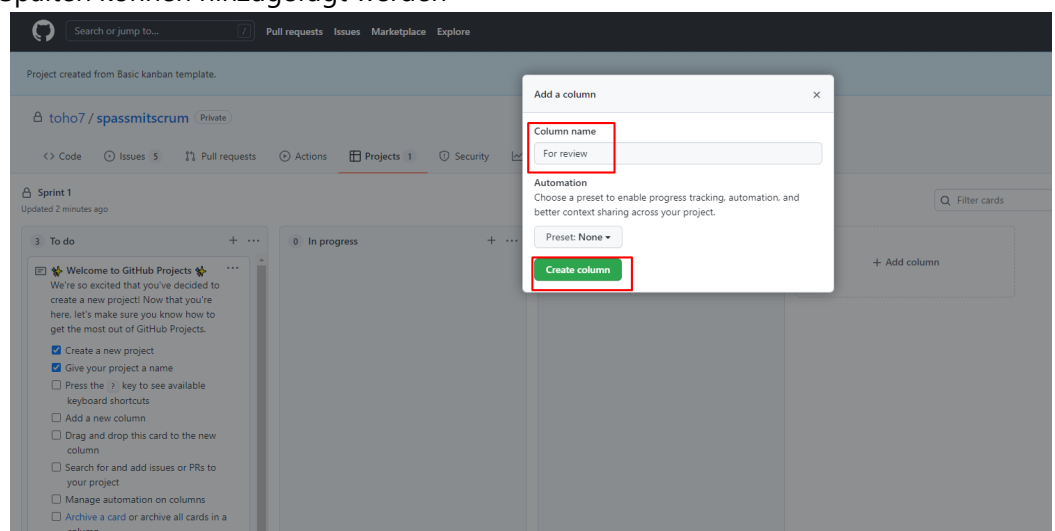
AUFGABE 2: SCRUM MIT GITLAB ODER GITHUB

Ermitteln Sie, wie man in GitLab oder GitHub SCRUM-Projektmanagement durchführt. Setzen Sie dazu die Möglichkeiten in den beiden Plattformen kreativ ein, um die in 1) ermittelten Prozesse gut abzubilden. Sie sollten nach der Bearbeitung eine für Sie konsistente Vorgehensweise zum Einsatz von GitLab oder GitHub zur Unterstützung von SCRUM-Projekten gefunden haben. Dokumentieren Sie die gefundene Vorgehensweise. Stellen Sie dazu insb. gegenüber, welche Funktion in GitLab / GitHub wie und zu welchem Zweck eingesetzt werden können, um den SCRUM-Prozess zu unterstützen.

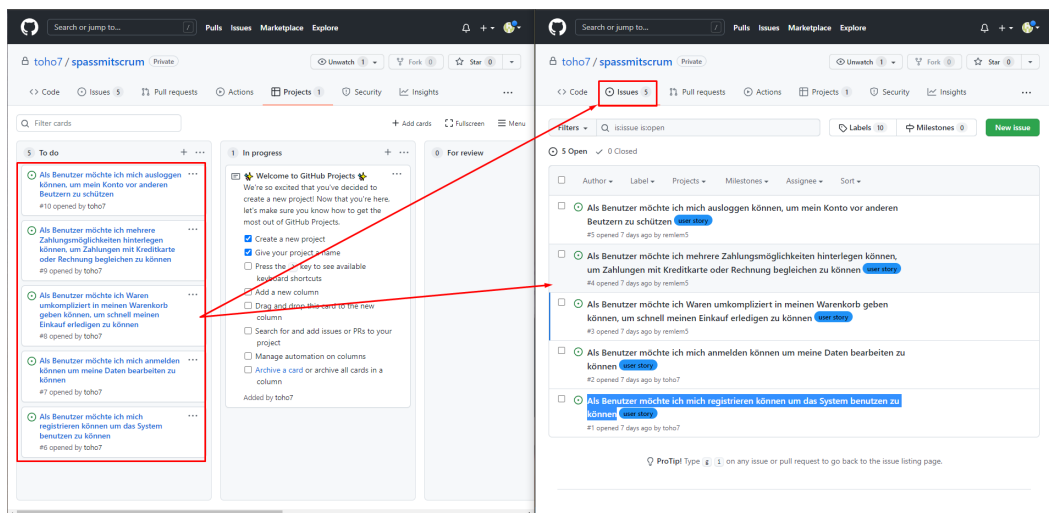
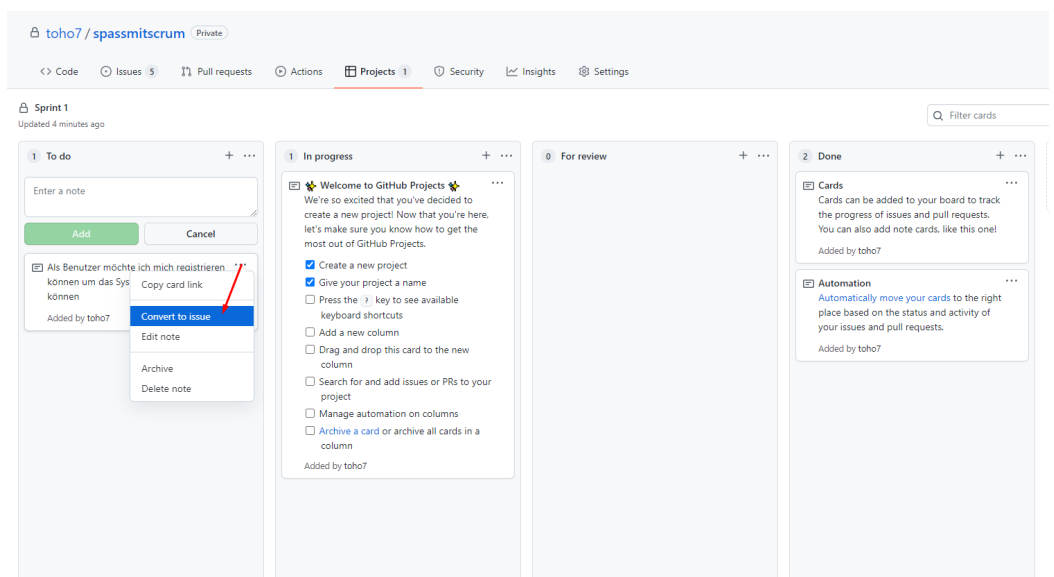
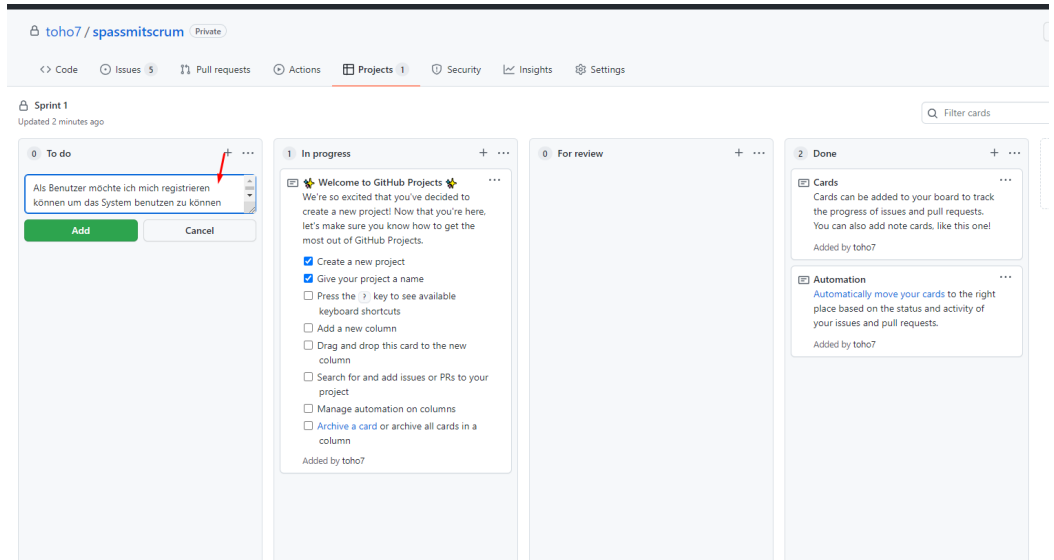
- ..
 - Einrichten eines neuen Projektes
 - Als template kann man hier zB Basic kanban nehmen, was einem schon ein paar Spalten/Listen voreingestellt liefert



- neue Spalten können hinzugefügt werden



- In die ToDos können Issues hinzugefügt werden, was zB User-Stories repräsentieren



Als Benutzer möchte ich mich registrieren können um das System benutzen zu können #6

Edit New issue

Open toho7 opened this issue 8 minutes ago · 0 comments

toho7 commented 8 minutes ago

No description provided.

Copy link
Quote reply
Edit

toho7 created this issue from a note in Sprint 1 (To do) 8 minutes ago

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Close issue Comment

Remember, contributions to this repository should follow our GitHub Community Guidelines.

Assignees
No one—assign yourself

Labels
None yet

Projects
Sprint 1
To do

Milestone
No milestone

Development
Create a branch for this issue or link a pull request.

Notifications
Customize
Unsubscribe

You're receiving notifications because you're watching this repository.

Write Preview

Tasks:

- [] sdfgsdfg
- [] dfqdsfg
- [] dfqdsfg

Attach files by dragging & dropping, selecting or pasting them.

Cancel Update comment

- Den Issues können dann auch noch Meilensteine, Labels, Zuständigkeit,.. zugeordnet werden

toho7 / spassmitsrum

Search or jump to...

Pull requests Issues Marketplace Explore

Unwatch Fork Star

<> Code Issues Pull requests Actions Projects Security Insights Settings

Sprint 1
Updated 8 minutes ago

5 To do

- Als Benutzer möchte ich mich ausloggen können, um mein Konto vor anderen Benutzern zu schützen #10 opened by toho7
- Als Benutzer möchte ich mehrere Zahlungsmöglichkeiten hinterlegen können, um Zahlungen mit Kreditkarte oder Rechnung begleichen zu können #9 opened by toho7
- Als Benutzer möchte ich Waren unkompliziert in meinen Warenkorb geben können, um schnell meinen Einkauf erledigen zu können #8 opened by toho7
- Als Benutzer möchte ich mich anmelden können um meine Daten bearbeiten zu können #7 opened by toho7
- Als Benutzer möchte ich mich registrieren können um das System benutzen zu können #6 opened by toho7

1 In progress

2 For review

3 Done

Filter cards

Als Benutzer möchte ich mich registrieren können um das System benutzen zu können #6
Opened in toho7/spassmitsrum

toho7 commented 13 minutes ago

Tasks:

- [] sdfgsdfg
- [] dfqdsfg
- [] dfqdsfg

Assignees
No one—assign yourself

Labels
None yet

Projects
Sprint 1
To do

Go to issue for full details

Close issue

- Was auch möglich ist,.. man kann einem Issue andere Issues unterordnen und sie so als Tasks definieren, dass wenn man diese abschließt sie in der übergeordneten Issue abgehakt werden sobald sie in die "Done" Spalte verschoben werden

AUFGABE 3: USER-STORIES

1. Bilden Sie eine Gruppe von 3-4 Studenten. Pro Student ergeben sich folgende Rollen:
 - **Product Owner**
 - **Scrum Master**
 - **Entwickler**
2. Denken Sie sich ein fiktives Softwareentwicklungsprojekt aus und definieren Sie mindestens 5 User-Stories für das Projekt. Verwenden sie dazu User-Story-Cards:
 - **Beurteilen Sie die Qualität ihrer User-Stories nach den INVEST-Kriterien.**
 - **Independent:** Unabhängig von anderen User-Stories
 - **Negotiable:** Kurzbeschreibung ohne zu viele Details
 - **Valuable:** Muss Wert für Kunden haben
 - **Estimable:** Aufwand sollte abschätzbar sein
 - **Small:** Jede Story sollte in 2-3 Tagen umsetzbar sein
 - **Testable:** Muss testbar sein, um feststellen zu können, ob sie abgeschlossen ist.
 -

- **Pro User-Story-Card definieren Sie abschließend Tasks (Aufgaben für Entwickler und Tester).**

- **User-Story 1**

		Priority
Benutzer-Registrierung		8
Story		Estimate
Als Benutzer möchte ich mich registrieren können um das System benutzen zu können		6
Conditions of Satisfaction/Description		
Benutzer muss erfolgreich registriert sein, und in Datenbank angelegt.		

- **User-Story 2**

		Priority
Benutzer-Anmeldung		10
Story		Estimate
Als Benutzer möchte ich mich anmelden können um meine Daten bearbeiten zu können		8
Conditions of Satisfaction/Description		
Benutzer muss erfolgreich angemeldet sein. Benutzer kann seine Daten ändern. Änderungen werden in Datenbank übernommen.		

- **User-Story 3**

		Priority	
	Warenkorb-Handling		5
	Story	Estimate	
	Als Benutzer möchte ich Waren unkompliziert in meinen Warenkorb geben können, um schnell meinen Einkauf erledigen zu können		20
	Conditions of Satisfaction/Description		
	Benutzer kann Waren zu seinem persönlichen Warenkorb hinzufügen. Nach Ab- und Anmeldung sind Waren noch im Warenkorb.		
■			

- **User-Story 4**

		Priority	
	Zahlungsmöglichkeiten		4
	Story	Estimate	
	Als Benutzer möchte ich mehrere Zahlungsmöglichkeiten hinterlegen können, um Zahlungen mit Kreditkarte oder Rechnung begleichen zu können		40
	Conditions of Satisfaction/Description		
	Benutzer kann Zahlungsmöglichkeit festlegen und auswählen. Mindestens 3 verschiedene Zahlungsmöglichkeiten funktionieren.		
■			

- **User-Story 5**

		Priority	
	Benutzer-Abmeldung		8
	Story	Estimate	
	Als Benutzer möchte ich mich ausloggen können, um mein Konto vor anderen Beutzern zu schützen		5
	Conditions of Satisfaction/Description		
	Benutzer kann sich ausloggen. Danach sind keine persönlichen Daten mehr sichtbar im Browser. Benutzer muss sich wieder anmelden um alles sehen zu können.		
■			

AUFGABE 4: ABWICKLUNG MIT GITLAB ODER GITHUB

1. Richten Sie GitLab oder GitHub so ein, dass sie die Plattformen möglichst gut bei der Abwicklung der SCRUM-Prozesse unterstützen können

2. Pflegen Sie die User-Stories aus der vorhergehenden Aufgabe ein und spielen Sie den gesamten Scrum-Prozess mit GitLab durch.

AUFGABE 5: AGILES PM, VERSIONSKONTROLLE UND CI

Recherchieren und überlegen Sie, wie leichtgewichtige Versionskontrollmechanismen wie Trunkbased Development oder GitHub Flow zusammen mit Continuous Integration Features der gängigen Codeversionierungsplattformen (GitHub Actions, GitLab CI/CD) und automatisiertem Testen möglichst produktiv und effizient in Verbindung mit SCRUM eingesetzt werden können

Beschreiben Sie ihre Ergebnisse in einem Textdokument

Trunkbased Development

- Ist eine Praktik der Versionskontrolle, bei der Entwickler kleine, häufige Änderungen in einem Kern- oder Haupt-Branch („Trunk“) zusammenführen.
- Wird häufig von DevOps-Teams verwendet und ist Teil des DevOps-Lebenszyklus, da sie die Zusammenführungs- und Integrationsphasen optimiert.
- Entwickler können im Vergleich zu anderen langlebigen Branching-Strategien für Funktionen kurzlebige Branches mit wenigen kleinen Commits erstellen.
- Bei zunehmender Komplexität der Codebasis und wachsenden Teams trägt die Trunk-basierte Entwicklung dazu bei, einen kontinuierlichen Flow von Produktions-Releases sicherzustellen

Gitflow vs. Trunk-basierte Entwicklung

- Gitflow ist ein alternatives Git-Branching-Modell mit langlebigen Feature-Branched und mehreren primären Branches.
- Höhere Anzahl an Branches als bei Trunk-basierter Entwicklung.
- Entwickler erstellen einen Feature-Branch und führen den Merge mit dem Haupt-Trunk-Branch erst durch, wenn das Feature vollständig ist.

Vorteile der Trunk-basierten Entwicklung

- Ist ein für Continuous Integration erforderliches Verfahren.
- Wenn Build- und Testprozesse automatisiert sind, Entwickler jedoch an isolierten, langwierigen Feature-Branched arbeiten, die selten in einen gemeinsamen Branch integriert werden, wird Continuous Integration ihr Potenzial nicht entfalten können.
- verringert Reibungspunkte bei der Code-Integration.
- Wenn Aufgaben abgeschlossen werden, müssen sie den neuen Code in den Haupt-Branch mergen, aber erst wenn Builds erfolgreich waren

Fazit

- Die Trunk-basierte Entwicklung ist derzeit der Standard für leistungsstarke Entwicklerteams, da sie mithilfe einer vereinfachten Git-Branching-Strategie einen Rhythmus für Software-Releases vorgibt und aufrechterhält.
- Darüber hinaus bietet die Trunk-basierte Entwicklung den Entwicklerteams mehr Flexibilität und Kontrolle darüber, wie sie Software an den Endbenutzer liefern.

GitHub Flow

- Der Fokus liegt auf Branches. Das ermöglicht den Teams frei zu experimentieren und regelmäßig Deployments zu machen.
- Wie GitHub Flow funktioniert:
 - Neuen Branch erstellen
 - Änderungen durchführen und Commits adden
 - Pull Request öffnen
 - Andere schauen drüber
 - Deployen
 - Mergen

Master Branch

- Der Master-Branch ist stabil und er ist immer funktionsfähig beim Deployment.
- Alles das ist den Master-Branch gepusht wird, muss funktionsfähig und getestet sein.

AUFGABE 6: AGILES PM, VERSIONSKONTROLLE IN DER PRAXIS

Spielen Sie einen kompletten Prozess im Sinne von 5) praktisch durch

- Setzen Sie die bekannten Versionierungsfunktionen ein.
- Setzen Sie die in den vorhergehenden Aufgaben ermittelten SCRUM-Funktionen ein.
- Wenden Sie eine leichtgewichtige Branching-Strategie an
- Wenden sie automatische Tests und CI/CD-Pipelines an.

Dokumentieren Sie den durchgespielten Prozess in einem Textdokument mit Screenshots