

Wie viel Scrum steckt in unserem Entwicklungsprozess?

Cornelis Kater, Jan Christoph Ahrens

Was haben wir mitgebracht?

- ⦿ Was ist Scrum eigentlich?
Die reine Lehre
- ⦿ Erfahrungen aus dem Projekt Co³Learn
Bericht aus der Praxis
- ⦿ Scrum in unserem Entwicklungsprozess
Wie scrummy sind wir denn?*

* ja, das ist ein Wort!

*„Scrum akzeptiert, dass der Entwicklungsprozess nicht vorherzusehen ist. Das Produkt ist die bestmögliche Software, die Kosten, die Funktionalität, die Zeit und die Qualität einbeziehend“
(Ken Schwabe)*

Was ist Scrum eigentlich?

Die reine Lehre

- ⦿ ein Modell der agilen (Software-)Entwicklung
- ⦿ es gibt Parallelen zum Lean-Management (aus Japan)
- ⦿ bessere Wertschöpfung durch gezielte Zusammenarbeit
- ⦿ wenig Regeln, stattdessen definierte Rollen und Kommunikationsprozesse
- ⦿ zentral sind „Sprints“ in deren Zeiträumen sich Prozesse wiederholen

Das agile Manifest von Ken Schwaber

1. Individuen und Interaktionen sind wichtiger als Prozesse und Werkzeuge.
2. Funktionierende Software ist wichtiger als umfassende Dokumentation.
3. Zusammenarbeit mit dem Kunden ist wichtiger als Vertragsverhandlungen.
4. Reagieren auf Veränderung ist wichtiger als das Befolgen eines Plans.

agile Leitsätze

1. Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.
2. Heiße Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.
3. Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.
4. Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.
5. Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.
6. Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.
7. Funktionierende Software ist das wichtigste Fortschrittsmaß.
8. Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.
9. Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.
10. *Einfachheit: Die Kunst, die Menge nicht getaner Arbeit zu maximieren ist essenziell.*
11. Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.
12. In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.

agile Leitsätze

- 1. Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.**
- 2. Heiße Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.**
3. Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.
4. Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.
- 5. Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.**
6. Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.
7. Funktionierende Software ist das wichtigste Fortschrittsmaß.
8. Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.
9. Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.
10. *Einfachheit: Die Kunst, die Menge nicht getaner Arbeit zu maximieren ist essenziell.*
11. Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.
- 12. In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.**

agile Leitsätze

- 1. Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.**
- 2. Heiße Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.**
3. Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.
4. Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.
- 5. Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.**
6. Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.
7. Funktionierende Software ist das wichtigste Fortschrittsmaß.
8. Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.
9. Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.
10. *Einfachheit: Die Kunst, die Menge nicht getaner Arbeit zu maximieren ist essenziell.*
11. Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.
12. **In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.**

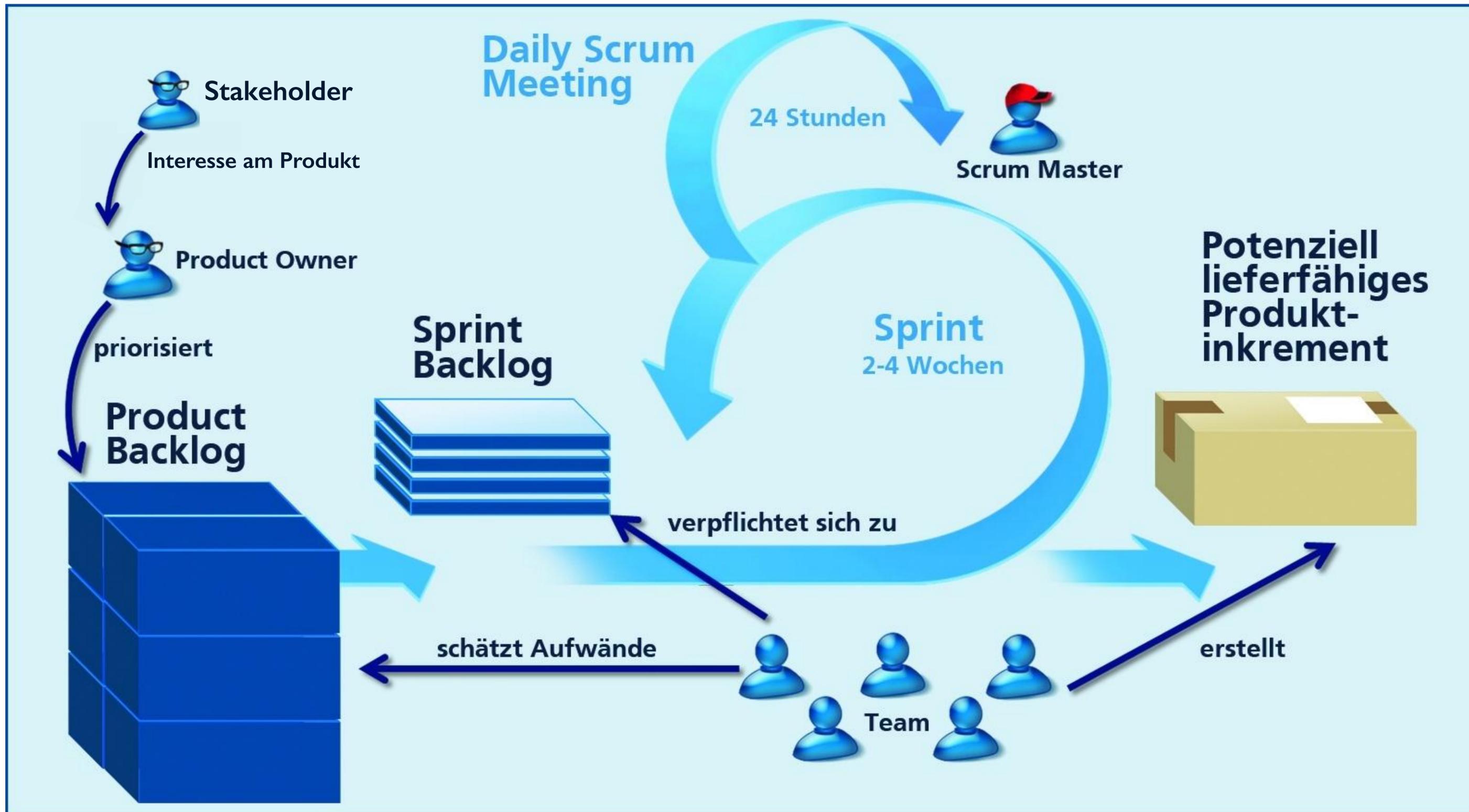
agile Leitsätze

- 1. Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.**
- 2. Heiße Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.**
3. Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.
4. Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.
- 5. Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.**
6. Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.
7. Funktionierende Software ist das wichtigste Fortschrittsmaß.
8. Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.
9. Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.
10. *Einfachheit: Die Kunst, die Menge nicht getaner Arbeit zu maximieren ist essenziell.*
11. Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.
- 12. In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.**

agile Leitsätze

1. Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.
2. Heiße Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.
3. Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.
4. Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.
5. Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.
6. Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.
7. Funktionierende Software ist das wichtigste Fortschrittsmaß.
8. Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.
9. Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.
10. **Einfachheit: Die Kunst, die Menge nicht getaner Arbeit zu maximieren ist essenziell.**
11. Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.
12. In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.

Scrum-Prozess in a Nutshell



Rollen/Begriffe in Scrum

- ⦿ Product Owner
- ⦿ Stakeholder
- ⦿ Entwicklungsteam
- ⦿ Scrum-Master
- ⦿ Produkt Backlog
- ⦿ Sprint Backlog

regelmäßige Termine bei Scrum

- ⦿ Daily
- ⦿ Sprint Planning
- ⦿ (Sprint)
- ⦿ Sprint Review
- ⦿ Retro(spektive)

Erfahrungen aus dem Projekt Co³Learn

Co³Learn
Communication Cooperation Collaboration



Erfahrungen aus dem Projekt Co³Learn

- Klassische Scrum Artefakte
 - Daily, Retro, Sprints (Planung und Review)
- Klassische Scrum Rollen
 - Scrum-Master, Product Owner, Entwicklungsteam, Steakholder
- Transparenz in den Arbeitsprozessen & Agiles Anpassen der Arbeit

Erfahrungen aus dem Projekt Co³Learn

Besonderheiten & Probleme

- Standortübergreifend & digital
- Doppelrollen (Interessenkonflikt)
 - Zeitliche Verfügbarkeit
- Dank Projektarbeit kein klares Produkt
- Aushandlungsprozesse als Teil der Arbeit

Erfahrungen aus dem Projekt Co³Learn

Scrum als festen, agilen Rahmen verstehen

- Bewusstsein der eigenen (Doppel-)Rolle
- Strukturelemente nutzen
- Sprintziel – Aushandlungsprozess und Commitment
- Scrum Master
- Scrum ist flexibel anpassbar

Scrum in unserem Entwicklungsprozess

Wie scrummy sind wir denn?*

- ⦿ Wir haben uns ein ziemlich gutes Modell gebaut, verfeinert und weiterentwickelt, das zu ähnlichen Prozessen kommt
- ⦿ Aber: Wir tun uns schwer mit der Skalierung
- ⦿ Allerdings: Der Stud.IP-Entwicklungsprozess ist durchaus schon scrummy/agil

Das ist auch scrummy!

- ⦿ Die reine Lehre: Es gibt keine reine Lehre!
- ⦿ Scrum kann und darf (muss?) angepasst werden!

Scrum in unserem Entwicklungsprozess

Wie scrummy sind wir denn?*

- ⦿ Die Stud.IP-Entwicklung wird auch die nächsten Jahre auf dem jetzigen hohen Level bleiben
- ⦿ Es kann sich lohnen, dass wir unsere Prozesse reflektieren und Weiterentwickeln, Scrum kann hier Impulse liefern

Rollen/Begriffe im Stud.IP-Prozess

- ⦿ CoreGroup
- ⦿ Betreiber:innen/Standorte
- ⦿ Entwickler:innen
- ⦿ geplante StEPs/TICs
- ⦿ abgestimmte StEPs/TICs

Rollen/Begriffe im Stud.IP-Prozess

- Product Owner → CoreGroup (sort of)
- Stakeholder → Betreiber:innen/Standorte
- Entwicklungsteam → Entwickler:innen
- Scrum-Master → *hmm ...*
- Produkt Backlog → geplante StEPs/TICs
- Sprint Backlog → abgestimmte StEPs/TICs

regelmäßige Termine bei Stud.IP

- ⦿ StEPs einreichen
- ⦿ StEPs Voting ca. 4 Wochen
- ⦿ Umsetzung StEPs/TICs einbauen ca. 2 Wochen
- ⦿ Test und Review ca. 4 Wochen
- ⦿ Nachbesserungen ca. 1 Woche (!)
- ⦿ Übernahme StEPs ca. 2 Wochen
- ⦿ Polishing TICs ca. 2 Wochen
- ⦿ Release fertigstellen, Beta-Test

Sprints?

regelmäßige Termine bei Stud.IP

- ◉ StEPs einreichen
 - ◉ StEPs Voting
 - ◉ Umsetzung StEPs/TICs einbauen
 - ◉ Test und Review
 - ◉ Nachbesserungen
 - ◉ Übernahme StEPs
 - ◉ Polishing TICs
 - ◉ Release fertigstellen, Beta-Test
- Sprints?*
- ca. 4 Wochen
 - ca. 2 Wochen
 - ca. 4 Wochen
 - ca. 1 Woche (!)
 - ca. 2 Wochen
 - ca. 2 Wochen

- ⚙ Daily
- ⚙ Sprint Planning
- ⚙ Sprint
- ⚙ Sprint Review
- ⚙ Retro(spektive)

Rollen/Begriffe im Stud.IP-Prozess

- ⚙ Daily → ☉ Chat? Oder ...?
- ⚙ Sprint Planning → ☉ CG-Treffen/PM
- ⚙ Sprint → ☉ ja ;)
- ⚙ Sprint Review → *hmm ...*
- ⚙ Retro(spektive) → *hmm ...*

Was könnten wir uns anschauen?

Ein Konzept für Retros?

Stud.IP-Phasen = Sprints?

Wiederkehrende Prozesse pro Phase?

Rolle „Scrum-Master“ denkbar?

Rollenklärungen?

Planning für jede Phase?