1. **SCRUM**

Suchen Sie im Netz nach dem offiziellen Scrum-Guide von Ken Schwaber und Jeff Sutherland. Lesen Sie die aktuelle Version (englische Variante) und beantworten Sie anschließend die folgenden Fragen. (Hier zu finden: Scrum Guides)

(a) Welche Personen/Rollen sieht das Scrum-Modell vor?

**Lösung:**

- *n* Developers
- 1 Product Owner
- 1 Scrum Master

(b) Welcher Zeitraum ist laut Dokumentation vorgesehen für die Planung eines Sprints?

**Lösung:**

Aus dem Scrum Guide:

> Sprint Planning is timeboxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

(c) Welcher Zeitraum ist laut Dokumentation vorgesehen für die folgenden Abläufe / Meetings:

i. Daily Scrum

**Lösung:**

Aus dem Scrum Guide:

> The Daily Scrum is a 15-minute event for the Developers of the Scrum Team. To reduce complexity, it is held at the same time and place every working day of the Sprint. If the Product Owner or Scrum Master are actively working on items in the Sprint Backlog, they participate as Developers.

ii. Sprint

**Lösung:**

Aus dem Scrum Guide:

> They are fixed length events of one month or less to create consistency. A new Sprint starts immediately after the conclusion of the previous Sprint.

iii. Sprint Retrospektive

**Lösung:**

Aus dem Scrum Guide:

> The Sprint Retrospective concludes the Sprint. It is timeboxed to a maximum of three hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

iv. Sprint Review

**Lösung:**

Aus dem Scrum Guide:

> The Sprint Review is the second to last event of the Sprint and is timeboxed to a maximum of four hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

(d) Welche Nachteile und welche möglichen Probleme bringt das Vorgehen nach Scrum mit sich?

**Lösung:**

Reddit-User MoritzK_PSM fasst es sehr gut zusammen:

> Scrum makes certain implicit assumption about the people involved, that are not always given. A few examples:
>
> - A Scrum Team must be cross-functional and not have any sub-teams. That means people must usually go towards broadening the spectrum of their knowledge and ability. We speak of things like T-shaped and E-shaped skill profiles. Not everybody is *willing to do that* and in many places you still see, e.g., there being exclusively one person in a team for e.g. testing. This can create problems with delivery reliability, which may be better handled with a Kanban approach.
> - The whole idea of self-management is predicated on the idea that people *WANT to manage their own stuff*. Often enough you will find people that really don't want to take that responsibility and prefer to just follow instructions they are given by a manager.
>
>   An especially visible place is the Retrospective. If people do not care to improve their processes, it won't happen. A Scrum Master can only do so much in terms of encouraging and motivating. But people *who really don't care, won't care*.

- Scrum, in it's introduction, creates an initial disruption. It is meant to be one, to - as Scrum.org puts it - show the inefficacies of the existing processes. Now a lot of people don't like that, especially in the management. Here you can often see ScrumButs, e.g. „we introduce Scrum, but we don't have time for a Retro every Sprint“ or „we do Scrum now, but all the QA is still handled by an external body“, etc. Often a ScrumBut is still better than no Scrum at all, but also there are many cases where one aspect of Scrum missing drags other aspects down and the overall „faith“ in the efficacy of Scrum is lost. Here a gradual approach might be better, gaining gradually more buy-in from the management.

Ultimately, Scrum does not solve your problems. Scrum is based on collective intelligence and pro-active engagement in the development. If you have shitty people, you won't get far with Scrum.