A2 Scrum

[a] Vorgehen in zwei Situationen während des Sprints

Fall 1: Sprintziel und alle User Stories sind *vorzeitig* erfüllbar.

1. **Sprint nicht verlängern oder verkürzen** (Timeboxing beibehalten): Die Sprintdauer ist fix und wird nicht angepasst; nur Ereignisse dürfen enden, sobald ihr Zweck erfüllt ist.
2. **Mit dem Product Owner (PO) *Scope neu vereinbaren*:** Das Dev-Team kann, falls sinnvoll, weitere gut vorbereitete Product-Backlog-Einträge (DoR erfüllt) in den Sprint *ziehen*, sofern das Sprintziel nicht gefährdet wird. Inhalt und Umfang des Sprints dürfen mit dem PO neu vereinbart werden.
3. **Weitere Inkremente liefern:** Mehrere Inkremente pro Sprint sind möglich; vorzeitige Lieferung ist erlaubt, solange die *Definition of Done* eingehalten wird.
4. **Qualität stärken & Schulden abbauen:** Falls keine zusätzlichen, wertvollen Stories verfügbar sind: Refactoring, Testausbau und Dokumentation gemäß DoD priorisieren.
5. **Sprintabbruch nur in Ausnahmefällen:** Wird das Sprintziel obsolet, kann ausschließlich der PO den Sprint abbrechen.

Fall 2: Sprintziel und User Stories sind *sehr wahrscheinlich* nicht rechtzeitig erfüllbar.

1. **Transparenz & Re-Planung im Daily Scrum:** Fortschritt und Hindernisse offenlegen; Plan anpassen, um das Sprintziel bestmöglich zu erreichen.
2. **Scope mit dem PO neu verhandeln:** Umfang des Sprint Backlogs *reduzieren* (z. B. niedere Prioritäten verschieben), ohne das Sprintziel oder Qualitätsansprüche zu gefährden. *Qualität nimmt nicht ab.*
3. **Impediments beseitigen lassen:** Der Scrum Master unterstützt aktiv beim Entfernen von Hindernissen.
4. **DoD nicht aufweichen:** Keine Abstriche bei Qualitätskriterien; unvollständige Arbeit bleibt „nicht fertig“ und wird transparent gemacht.
5. **Keine Sprintverlängerung und kein „Personen nachschieben“:** Die Sprintlänge bleibt fix; zusätzliches Personal macht späte Projekte oft noch später (Brooks’sches Gesetz).
6. **Stakeholder-Einbindung:** Ergebnisse und Anpassungsbedarf spätestens im Sprint Review demonstrieren und besprechen.

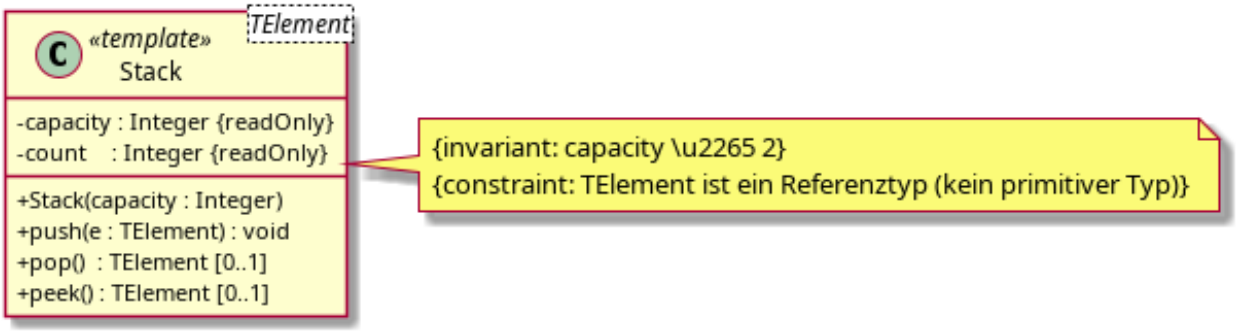
[b] Sehr kurze (z. B. 1 Woche) vs. sehr lange (z. B. 8 Wochen) Sprintlängen

	Vorteile	Nachteile
Kurz (≈ 1 Woche)	• Sehr häufiges Feedback; schnelle Kurskorrekturen. • Frühe Risikosichtbarkeit und kurze Lernschleifen. • Hohe Fokusdisziplin auf kleine, klar geschnittene Inkremente. • Regelmäßige Lieferbarkeit und verlässlicher Takt.	• Relativ hoher Event-Overhead (Planning/Review/Retrospektive) im Verhältnis zur Umsetzung. • Gefahr von zu kleinteiligen Stories bzw. Stückelung. • Mehr Planungsdruck; häufigere Schätz-/Refinement-Zyklen.
Lang (≈ 8 Wochen)	• Weniger „Ceremony"-Overhead pro Arbeitswoche. • Mehr Zeit für komplexe Features innerhalb eines Sprints. • Weniger Kontextwechsel durch selteneres Planen/Reviewen.	• Weniger Feedback; Risiken werden später sichtbar. • Größere Batches/WIP → spätere Integration, höhere Fehlerrisiken. • Geringere Agilität; Scope Creep wird später erkannt.

Tabelle 1: Vier-Felder-Tafel: kurze vs. lange Sprintlängen.

Aufgabe 4 - UML-Klassendiagramm/Zustandsdiagramm: Stapelspeicher

Klassendiagramm („Stack<TElement>")



Zustandsdiagramm (Call Events, Guards, Effects)