

## A1 Test Driven Development

Testfälle für das Steuerungssystem einer Lampe.

- **T1: Ein- und Ausschalten:** Die Lampe soll auf Befehl ein- und ausgeschaltet werden können.
- **T2: Counter-Mock:** Verwende ein Mock-Objekt für die Uhr, das so konfiguriert werden kann, dass Stunden als Sekunden simuliert werden. Berechne anschließend eine Reihe von Operationen vor, die zu bestimmten Zeitpunkten ausgeführt werden sollen (z. B. einfache Additionen). Überprüfe, ob die korrekte Anzahl von Operationen ausgeführt wurde.
- **T3: Synchronisation:** Überprüfe, ob sich das TimeTracking-Objekt korrekt mit der Systemuhr des Rechners synchronisiert.
- **T4: Test mit Mock-Objekt:** Simuliere den Fall, in dem die Lampe mit dem Mock-Objekt gesteuert wird.
- **T5: Test mit TimeTracking-Objekt:** Teste die Lampe im realen Anwendungsfall mit dem TimeTracking-Objekt.

## 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.

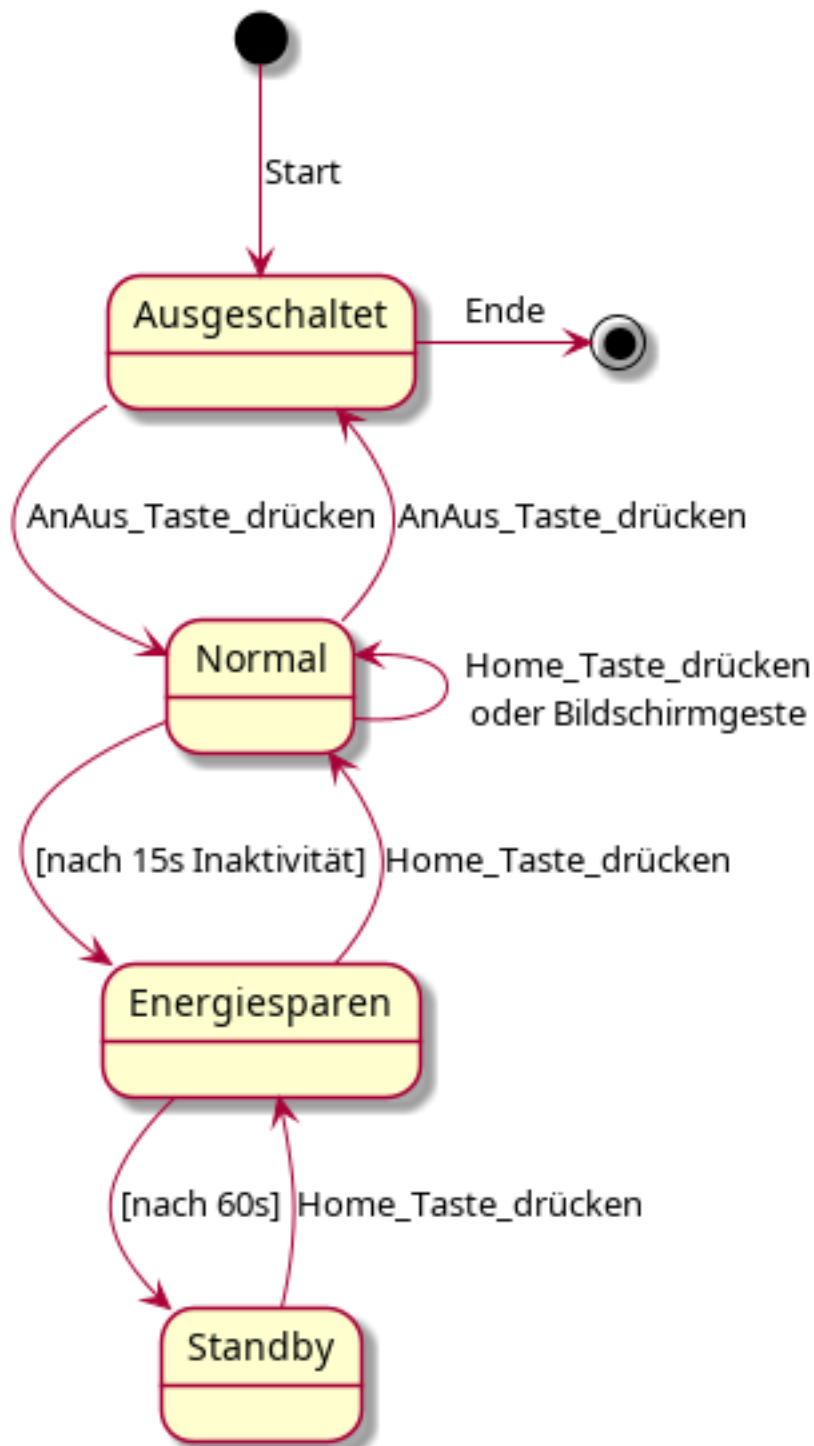
## [b] Sehr kurze (z. B. 1 Woche) vs. sehr lange (z. B. 8 Wochen) Sprintlängen

	Vorteile	Nachteile
<b>Kurz (<math>\approx 1</math> Woche)</b>	<ul style="list-style-type: none"> <li>• Sehr häufiges Feedback; schnelle Kurskorrekturen.</li> <li>• Frühe Risikosichtbarkeit und kurze Lernschleifen.</li> <li>• Hohe Fokusdisziplin auf kleine, klar geschnittene Inkremente.</li> <li>• Regelmäßige Lieferbarkeit und verlässlicher Takt.</li> </ul>	<ul style="list-style-type: none"> <li>• Relativ hoher Event-Overhead (Planning/Review/Retrospektive) im Verhältnis zur Umsetzung.</li> <li>• Gefahr von zu kleinteiligen Stories bzw. Stückelung.</li> <li>• Mehr Planungsdruck; häufigere Schätz-/Refinement-Zyklen.</li> </ul>
<b>Lang (<math>\approx 8</math> Wochen)</b>	<ul style="list-style-type: none"> <li>• Weniger „Ceremony“-Overhead pro Arbeitswoche.</li> <li>• Mehr Zeit für komplexe Features innerhalb eines Sprints.</li> <li>• Weniger Kontextwechsel durch selteneres Planen/Reviewen.</li> </ul>	<ul style="list-style-type: none"> <li>• Weniger Feedback; Risiken werden später sichtbar.</li> <li>• Größere Batches/WIP <math>\rightarrow</math> spätere Integration, höhere Fehlerrisiken.</li> <li>• Geringere Agilität; Scope Creep wird später erkannt.</li> </ul>

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

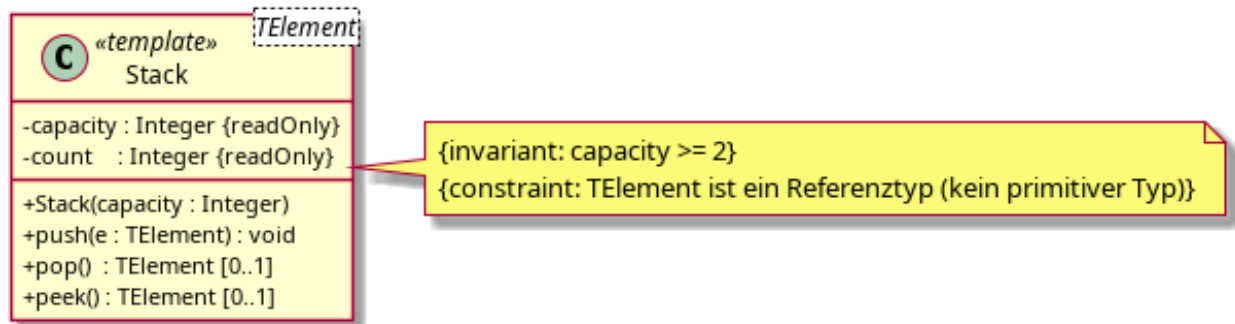
## A3 Smart Phone State UML

### UML-Zustandsdiagramm: Smartphone-Steuerung



## Aufgabe 4 - UML-Klassendiagramm/Zustandsdiagramm: Stapel-speicher

### Klassendiagramm („Stack<TElement>“)



### Zustandsdiagramm (Call Events, Guards, Effects)

