

A photograph of a modern university campus courtyard. In the background, there are several tall, cylindrical concrete structures. To the right, a large, multi-story building with a grid-like facade of windows and balconies is visible. The foreground features a paved walkway, some greenery, and a small sculpture of stacked stones. A black banner with white text is overlaid on the image.

M324

**Verwalten von Anforderungen.**

# Anforderungen / Requirements

- Anforderungsverwaltung = requirements management
- Anforderungs-Erhebung = requirements engineering
- Umgang mit Anforderungen in der Informatik historisch schwierig, da Schnittstellenproblematik
- Schnittstelle zwischen Business und Informatik (gleiche Sprache?)
- Häufig Missverständnisse (was ist wichtig): Es geht dabei nicht darum, ob etwas "gut" programmiert ist, sondern ob etwas "zweckmässig" programmiert ist
- Anforderungen bilden in erster Linie den **Zweck** des Codes ab

# Unternehmen vs. Schule bzgl. Anforderungen

- In Unternehmen schreiben Sie Code nicht als Selbst-Zweck, sondern um Produkte oder Dienstleistungen an Kunden auszuliefern
- Anforderungen bilden also ab, was die Kunden *von Ihrem Produkt brauchen* oder *erwarten* (= **Bedürfnis**)
- Anforderungen liegen also zwischen Zweck des Codes und der Bedürfnisse der Kunden



# Kunden und ihre Anforderungen

- Kunden wissen leider nicht immer, was sie eigentlich brauchen, was möglich ist oder sie können dies für Informatiker nicht immer verständlich ausdrücken
- Neue Produkte schaffen auch neue Anforderungen
- Agiler Ansatz hat sich bewährt, bei welchem nicht versucht wird, am Anfang eines Projekts *alle* Bedürfnisse und Anforderungen der Kunden zu erheben
- Stattdessen wird *ein* Kernproblem möglichst effektiv gelöst und *dann* überprüft, welche Bedürfnisse inzwischen wichtig geworden sind (= **Iteration**)

# Minimum Viable Product (MVP)

- Minimum Viable Product, zu Deutsch etwa *die einfachste praktikable Lösung*
- Statt alle Probleme Ihrer Kunden zu lösen, lösen Sie zunächst eines mit minimalen Aufwand (= **MVP**). Danach betrachten Sie die Situation erneut, definieren weitere Bedürfnisse, formulieren daraus Anforderungen und liefern das nächste MVP aus
- Bei einem iterativen Ansatz müssen die Anforderungen *verwaltet* werden. Anforderungen verändern die Priorität, neue kommen hinzu, alte verschwinden und tauchen dann wieder auf...

# Verwalten von Anforderungen

- Verwalten von Anforderungen ist ein Prozess, den Sie während der gesamten Lebensdauer eines Produkt begleiten
- In der Praxis werden dabei verschiedene Tools eingesetzt, bspw. Azure DevOps, GitHub Issues oder auch Ticketing-Tools wie Jira, etc.

# Erhebung von Anforderungen

- requirements engineering ist eine Wissenschaft für sich, die in diesem Modul nicht genauer analysiert wird
- In diesem Modul halten wir Anforderungen in Form von *user stories* fest.
- Eine *user story* stellt ein Bedürfnis eines Benutzers so dar, wie es vom Produkt befriedigt wird. Die Form lautet  
**Als *Rolle*, möchte ich *Handlung*, damit *Nutzen***

# User Stories

- gut geeignet, da sie leicht von Nicht-Informatikern verstanden werden
- lassen aber viel Freiheit, wenn es um die Implementation geht
- zu einer Anforderung gehören immer auch Abnahme-Kriterien (*acceptance criteria*), die genau festlegen, wann die Anforderung als erfüllt gilt.
- In diesem Modul verwenden wir dafür Testfälle, die sich automatisch ausführen lassen



# Zusammenfassung

Im Modul 324 besteht also eine Anforderung aus zwei Teilen: einer *user story*, die für einen Laien leicht verständlich ist, und einem Testfall, welcher für den Programmierer sehr klare Vorgaben stellt.

# Testfälle

Im einfachsten Fall ist ein Testfall nach dem Schema *given-when-then* aufgebaut, welches analog zu Vorbereitung-Eingabe-(erwartete) Ausgabe ist:

given	when	then
Ich habe die Applikation geöffnet	Ich möchte ein neues Traktandum eingeben	Erscheint ein Feld, bei welchem ich das Datum eingeben kann.
Ich habe verschiedene Traktanden mit Datum eingegeben	Ich öffne die Seite mit allen Traktanden	Sie werden nach Datum sortiert dargestellt

# Anforderungen an Werkzeuge / Tools

- Entwicklung von Anforderungen festhalten
- Priorisierung von Anforderungen ermöglichen
- Zuteilung von Zuständigkeiten erlauben
- Anforderungen terminieren können
- Anforderungen und den dazugehörigen Code verknüpfen
- Testfälle einbinden
- ...

# Rückverfolgbarkeit

Diese Funktion, zurückverfolgen zu können, wie sich Anforderungen entwickelt haben und wer wann woran gearbeitet hat, nennt sich Rückverfolgbarkeit (*traceability*). Diese trägt erwiesenermassen massgeblich zu erfolgreichen Software-Projekten bei!

# Werkzeuge / Tools

- Wir lehnen uns hier an die agile Software-Entwicklung an, und fassen mehrere *user stories* zu einem *epic* (Epos) zusammen, um zum Beispiel umfangreiche Funktionalitäten, zu derer Beschreibung eine user story nicht ausreicht, abzubilden.
- Es gibt eine Vielzahl von Werkzeugen, die einem «helfen» Anforderungen zu verwalten. Dazu gehören Jira, Azure DevOps, Confluence, VersionOne, etc.
- Jedes dieser Werkzeuge hat seine eigene Terminologie, weshalb wir hier einen einfachen Ansatz verwenden » **GitHub**