

PROGRAMMIERPROJEKT 2019

EINFÜHRUNG

Annette Bieniusa



ZIELE DES PROJEKTS

- Planung und Durchführung eines Projekts zur Software-Entwicklung

PROGRAMMIERUNG!

INHALTE

- **Softwareentwicklungsprozesse**
 - Phasen und ihre Interaktion
- **Softwarearchitekturen**
 - Client-Server
 - Model-View-Controller
- **Modellierung** von OOP
 - UML (Klassendiagramme und Use Cases)
- **Bibliotheken** und **Frameworks**
 - Web-Frameworks
 - Datenbanken (SQL)
- **Werkzeuge** und Methoden
 - Entwicklungsumgebungen
 - Testing
 - Versionsverwaltung (GitLab)

ORGANISATORISCHES

- Regelmäßige Treffen (ca. wöchentlich)
 - immer Dienstags 17:15
- Programmieren in Teams à 3 Personen
- Benotung auf Grund von Einzelleistungen
 - Software-Komponenten
 - Dokumentation / Berichte
- Einzureichende Artefakte
 - Lastenheft => Pflichtenheft
 - Erster Prototyp
 - Grundversion (Implementierung der Kernanforderungen)
 - Finale Version (Ergänzung um optionale Anforderungen)
 - Präsentationsfolien

ZEITERFASSUNG

- Legen Sie eine Liste mit Ihren Stunden an!
- Teil der jeweiligen Abgabe(n)
- Was wird erfasst?
 - Teilnahme an Treffen
 - Schreiben von Dokumentation
 - Coden
 - Testen
 - Debuggen
 - Lektüre von Fremdquellen
- **Mindestens 120 Stunden gesamt!**
- **Maximal 20 Studen reserviert für Abschlusspräsentation**

ABSCHLUSSPRÄSENTATION

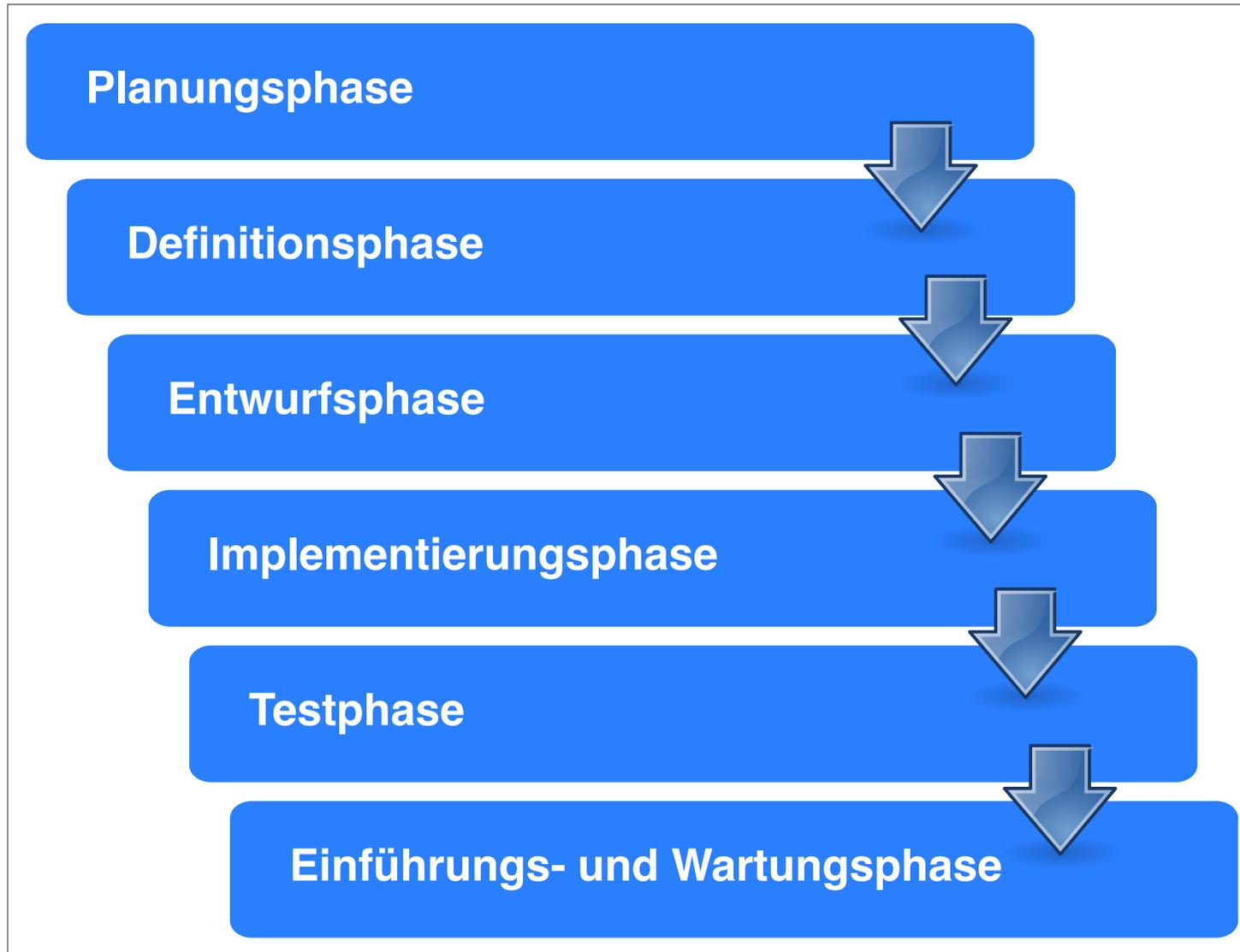
- In der Woche vom 26. - 30.08.

Bitte Notebooks zu den Treffen mitbringen, falls vorhanden.

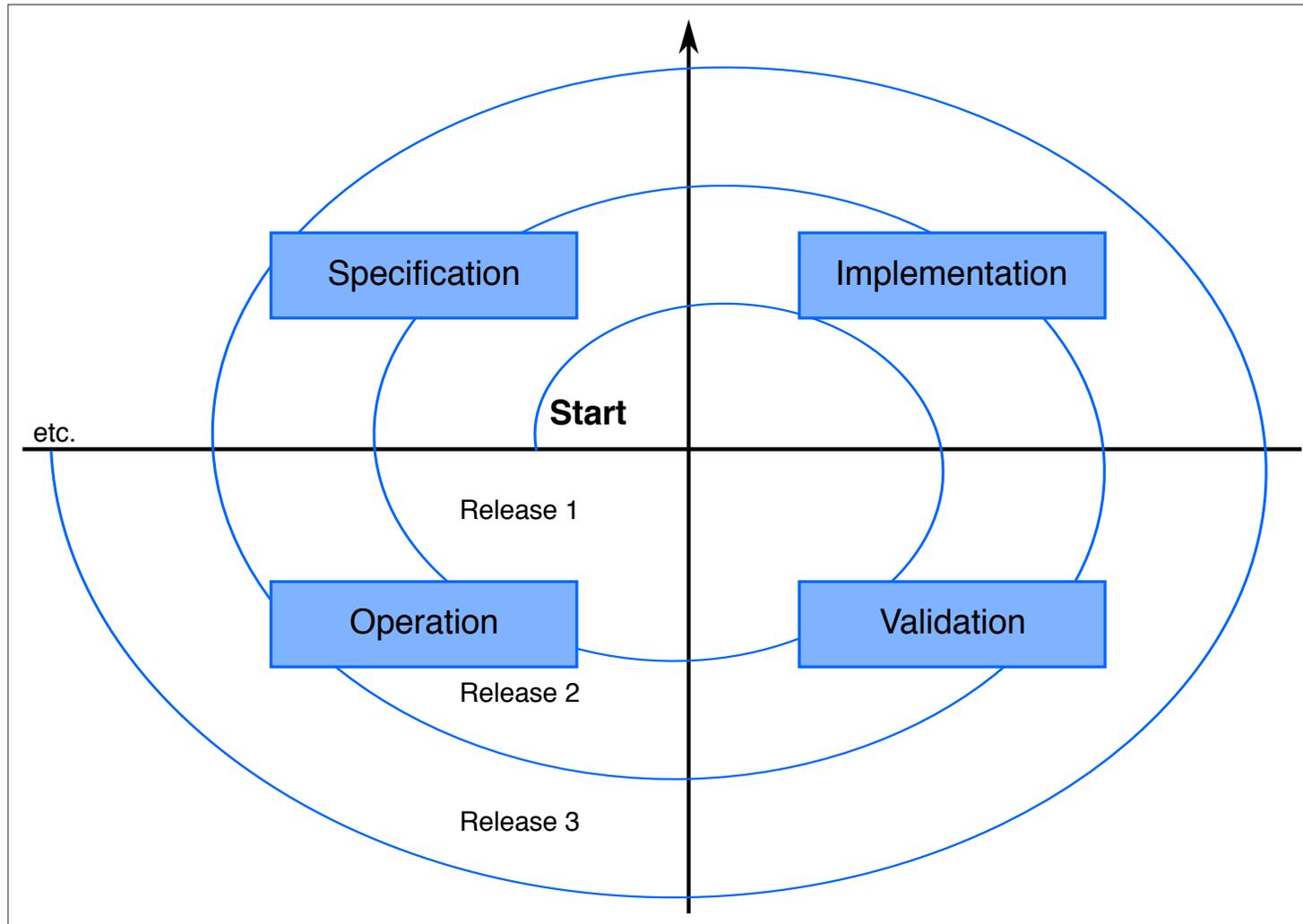
- Betriebssystem: Windows, Linux, Mac
- Richtige Tastatur
- **Kein Android Tablet oder iPad**

PHASEN DER SOFTWARE-ENTWICKLUNG

WASSERFALL-MODELL



EVOLUTIONÄRE MODELLE



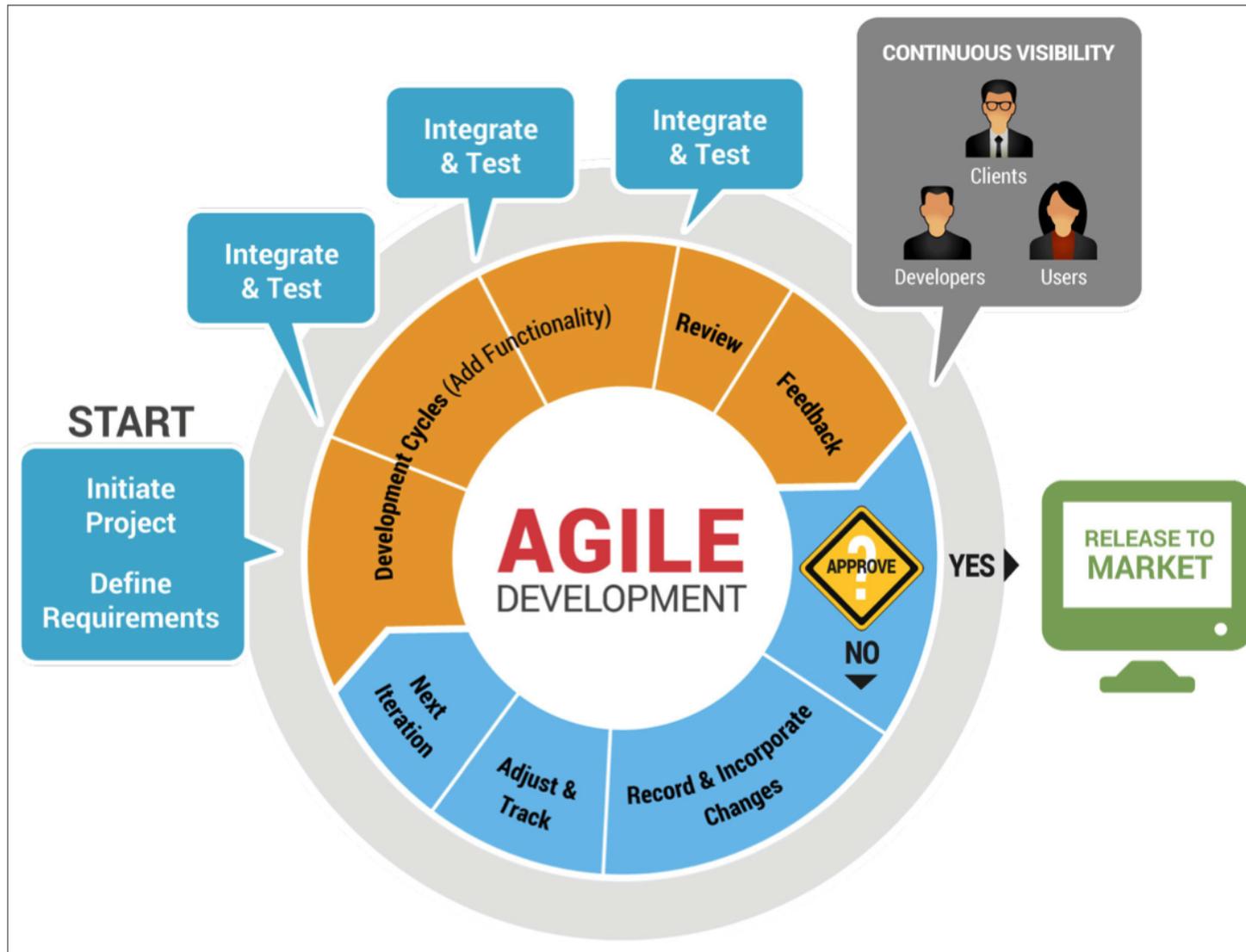
EVOLUTIONÄRE MODELLE

- Stufenweise, allmähliche Entwicklung
- Produktkern basiert auf den Muss-Anforderungen
- Weitere Entwicklung/Anforderungen sind gesteuert durch Erfahrungen mit dem Produkt
- Gefahr: Systemarchitektur muss für spätere Anpassungen vollständig überarbeitet werden

INKREMENTELLE MODELLE

- Zu Beginn werden alle Anforderungen vollständig erhoben
- Anforderungen werden dann schrittweise implementiert
- Mehr Zeitaufwand am Anfang
- Gefahr: Anforderungen werden übersehen

AGILE SOFTWARE-ENTWICKLUNG



AGILE SOFTWARE-ENTWICKLUNG

- Flexibler und schlanker Prozess
- Agiles Manifest:
 1. **Menschen und Interaktionen** sind wichtiger als Prozesse und Werkzeuge.
 2. **Funktionierende Software** ist wichtiger als umfassende Dokumentation.
 3. **Zusammenarbeit mit dem Kunden** ist wichtiger als die ursprünglich formulierten Leistungsanforderungen.
 4. **Eingehen auf Veränderungen** ist wichtiger als Festhalten an einem Plan.

AGILE METHODIK UND PRINZIPIEN

- Story cards / User stories
- Pair programming
- Testgetriebene Entwicklung

- Selbstorganisation und -reflexion des Teams
- Fokus auf funktionierenden Code
- Regelmäßige und häufige Releases

BEISPIEL: ANFORDERUNGSANALYSE FÜR EINE TODO-APP

INTERAKTIONEN MIT BENUTZER (USER STORIES)

ÜBERSICHT

ALS NUTZER WILL ICH EINE LISTE ANLEGEN, UM ZUSAMMENHÄNGENDE AUFGABEN ZU SAMMELN.

- Voraussetzungen: Keine
- Namen wird eingegeben und Listen im System hinterlegt
- Namen muss eindeutig sein
- Liste ist im System zugreifbar

ALS NUTZER WILL ICH EIN TODO-ITEM ANLEGEN, UM EINE AUFGABE ZU VERWALTEN.

- Voraussetzung: Liste muss im System vorhanden sein
- Benutzer gibt Titel des Items an
- TODO-Item wird auf Liste aufgenommen
- Item ist nicht erledigt

ALS NUTZER WILL ICH EIN ITEM ALS ERLEDIGT MARKIEREN, UM AUFGABEN ABZUSCHLIESSEN.

- Voraussetzung: Item auf existierender Liste und Item nicht erledigt
- Benutzer markiert Item als erledigt
- Item taucht nicht mehr in der Liste der zu erledigenden Items auf

ALS NUTZER WILL ICH ERLEDIGTE ITEMS ANZEIGEN, UM SIE ZU KONTROLLIEREN.

- Voraussetzung: Liste muss im System vorhanden sein
- Benutzer zeigt Liste an
- Benutzer aktiviert die Anzeige erledigter Items
- Alle Items, auch die erledigten, werden angezeigt
- Erledigte Items sind farblich markiert

ALS NUTZER WILL ICH DIE LISTE EXPORTIEREN, UM SIE IN ANDERE PROGRAMME (Z.B. EXCEL) ZU VERWENDEN.

- Voraussetzung: Liste muss im System vorhanden sein
- Inhalt der Liste wird in geeignetem Format in eine Datei geschrieben
- Benutzer kann frei wählen, wo die Datei auf dem Rechner abgelegt werden soll.

ERWEITERUNG: MULTI-USER WEB-APP

- Als Nutzer will ich mich für die TODO-App registrieren, um meine Daten in der Web-App zu verwalten.
- Als Nutzer will ich mich anmelden, um meine Listen vor fremdem Zugriff zu schützen.
- Als Administrator will ich die Nutzer aus der Datenbank entfernen, um Datenschutz konform zu arbeiten.

DATENMODELL

TODO-LISTEN

- Name der Liste
- Einträge/Items der Liste

ITEMS/EINTRÄGE

- Titel
- Erledigt-Flag

SICHTEN AUF DIE DATEN

LISTENÜBERSICHT

- Auflistung aller Listen im System
- Nur Anzeige der Namen

ANZEIGE EINER LISTE

- Name der Liste
- Anzeigen nicht erledigter Items mit Titel

ERWEITERTE LISTENANSICHT

- Name der Liste
- Anzeigen aller Items, auch der erledigten
- Erledigte Items sind als solche markiert

EINTEILUNG DER ANFORDERUNGEN

KERNANFORDERUNGEN

- Liste anlegen
- TODO-Item anlegen
- Item als erledigt markieren
- Erledigte Items anzeigen

ZUSÄTZLICHE FEATURES

- Liste exportieren
- Liste importieren

UNSER PROJEKT

- Inkrementelles Vorgehen
 - Zunächst: Produktspezifikation
 - Prototyp der Kernanforderungen
 - Dann iteratives Erweitern der Software
- Elemente aus der Agilen Software-Entwicklung
 - Test-driven Development
 - Iteratives Vorgehen

PROJEKT: GRILLPARTY-ORGA-APP!

Das Grillfest-Organisationskommittee der Kita "Klammernder Affe" verliert immer den Überblick bei der Grillfest-Planung.

- Für jede Familie muss die gewünschte Anzahl an Steaks, Würstchen und Gemüsebratlingen und Getränke eingekauft werden.
- Für weitere Utensilien, wie Grills, Kohle, Anzünder, Salate, Brot, gibt es eine Liste mit benötigten Dingen, in die man sich eintragen kann.
- Diese Liste wird vom Planungskomitee erstellt und bei Bedarf angepasst.
- Die entstandenen Kosten werden am Ende auf die beteiligten Personen aufgeteilt.
- etc.

"Wir brauchen eine Web-App!"

IHRE AUFGABE HEUTE

- Registrieren Sie sich im KIS!
- Bilden Sie Gruppen von 3 Studierenden! -> Namen per Email an bieniusa@cs.uni-kl.de!
- **Bis 26.04.19**: Ausarbeitung der Projektidee (Abgabe per Email)
 - User stories: Welche Benutzergruppen gibt es? Wie interagieren diese mit dem System? (Voraussetzungen für die jeweilige Story analysieren!)
 - Datenmodell: Welche Daten müssen gespeichert werden? Welche Zusammenhänge gibt es zwischen den Daten?
 - Sichten auf die Daten: Welche Ansichten muss die Anwendung haben? (Anmelde/Abmeldefenster, Eingabemasken, Übersichten, ...)
 - Unterscheidung: Kernanforderungen vs. optionale Features