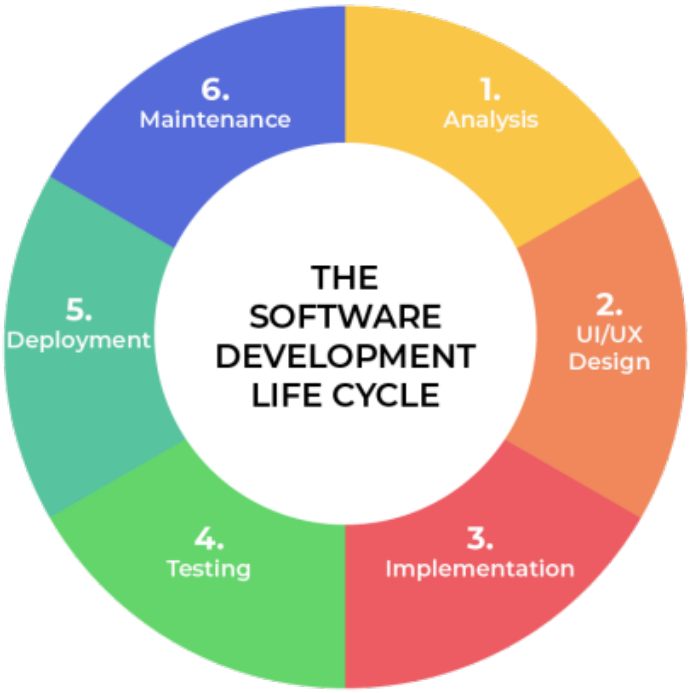
Software-Entwicklungs-Zyklus

Der **Software Development Life Cycle (SDLC)** beschreibt die typischen Phasen, die ein Software-Projekt von der ersten Idee bis zur fertigen Anwendung durchläuft. Testing und Deploy sind zwei dieser Phasen – und genau darum geht es in dieser Lerneinheit.

Der vollständige SDLC



Grafik des SDLC-Kreislaufs mit 6 Phasen

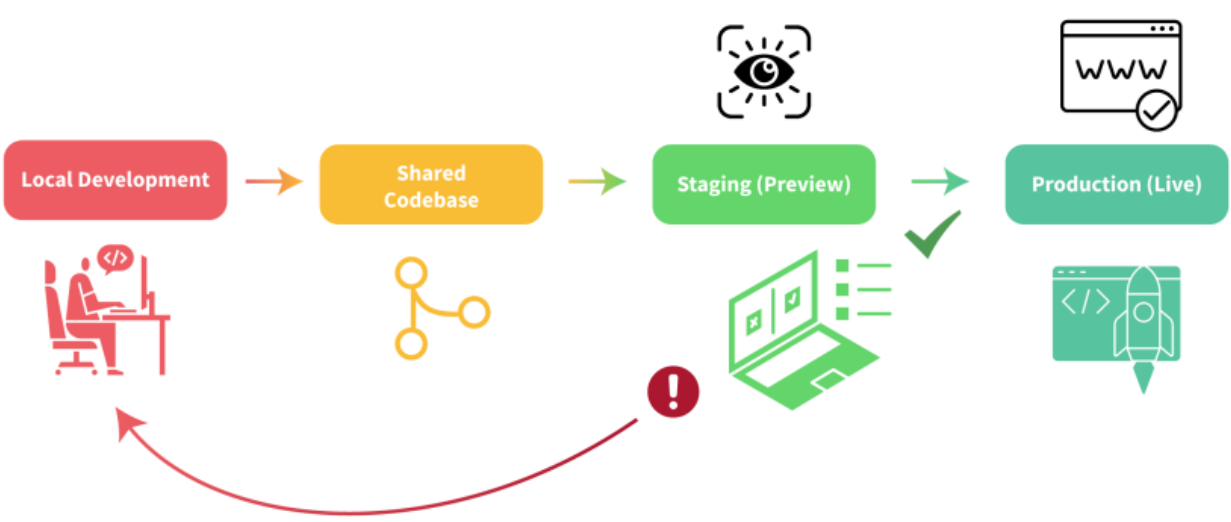
Ein typisches Software-Projekt durchläuft sechs Phasen:

1. **Analyse** – Anforderungen verstehen: Was soll die Anwendung können?
2. **UI/UX Design** – Benutzeroberfläche entwerfen: Wie soll sie aussehen?
3. **Implementation** – Code schreiben: Die eigentliche Entwicklungsarbeit.
4. **Testing** – Fehler finden und beheben, bevor die App live geht.
5. **Deployment** – Die App für Nutzerinnen und Nutzer bereitstellen.
6. **Maintenance** – Laufende Pflege: Bugfixes, Updates, neue Features.

Kein linearer Prozess: In der Praxis wiederholen sich Phasen ständig. Besonders Testing und Implementation bilden einen engen Kreislauf: Code schreiben → testen → Fehler beheben → erneut testen.

Der Web-Entwicklungs-Kreislauf

Wenn wir Software entwickeln, konzentrieren wir uns auf vier konkrete Schritte:



Pipeline-Diagramm mit 4 Schritten

Schritt	Was passiert hier?	Unser Werkzeug
① Local Development	Code schreiben, lokal ausprobieren. Die App läuft nur auf Ihrem Computer.	`npm run dev` → `localhost:5173`
② Shared Codebase	Code wird mit dem Team geteilt – normalerweise via Git.	(In LU16 überspringen wir diesen Schritt)
③ Staging (Preview)	Die App wird auf einem Test-Server deployed. Hier kann getestet werden, bevor sie live geht.	Netlify / Vercel / Render (alle benötigen Git)
④ Production (Live)	Die fertige, getestete App ist öffentlich zugänglich.	surge.sh (ohne Git) / Netlify / Vercel

Was passiert bei Bugs?

Wenn beim Testing Fehler gefunden werden, kehrt das Projekt zurück in die Entwicklungsphase. Dieser Feedback-Loop ist kein Zeichen von schlechter Arbeit – er ist der Sinn des Testings.

Testing im Kreislauf: Wann was?

Testing-Zeitpunkt	Was wird getestet?	Wer testet?
Während der Entwicklung	Einzelne Funktionen (Unit Tests)	Entwicklerin / Entwickler

Testing-Zeitpunkt	Was wird getestet?	Wer testet?
Nach dem Zusammenführen	Zusammenspiel mehrerer Module (Integration Tests)	Automatisiert / Entwickler
Auf Staging	Komplette Benutzerflüsse (E2E Tests, Manuell)	Tester, Entwickler, Kunde
Vor Production	Abnahme-Test	Auftraggeber

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/de/modul/m291/learningunits/lu16/theorie/a_sdlc?rev=1782679778

Last update: **2026/06/28 22:49**