

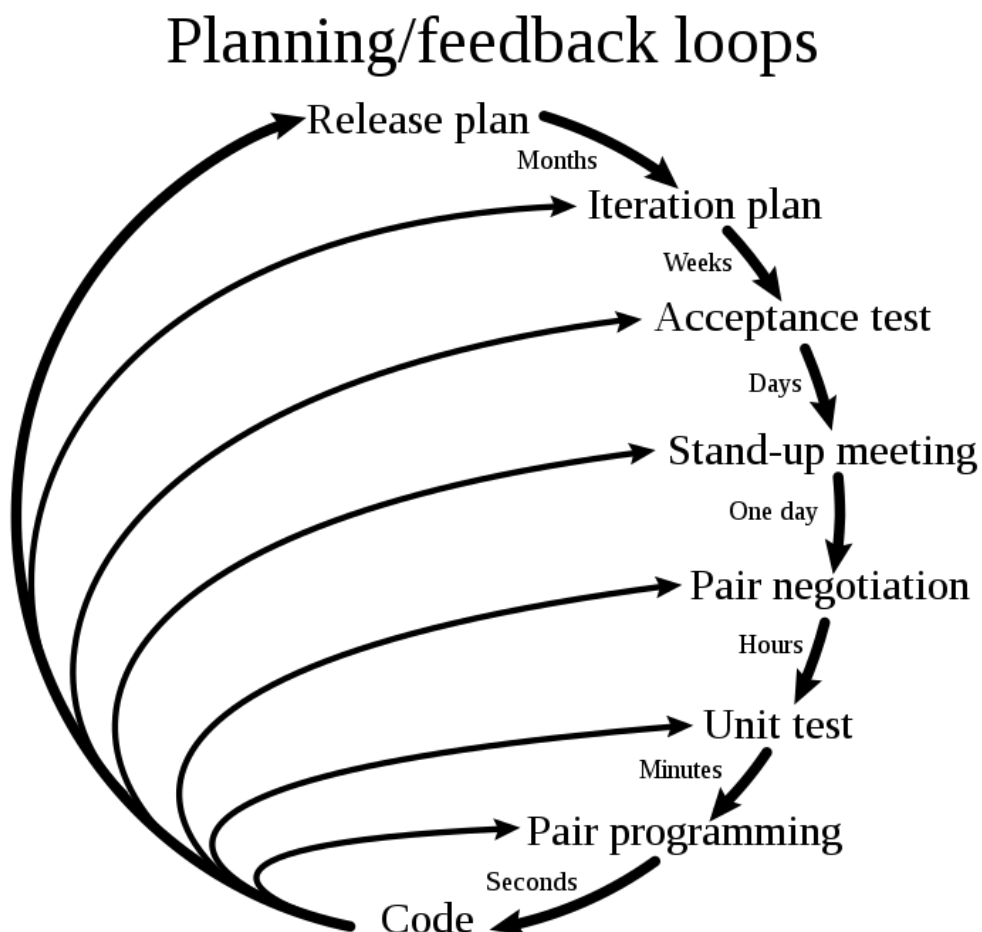
Extreme Programming (XP)

Extreme Programming (XP), er en agil metode defineret af Kent Beck i 1996, som er bygget på **Test Driven Development**, i en mere “ekstrem” retning.

XP er ment til at forøge software kvalitet og håndtering af kundens krav. XP er ment som én af de mere “humane” og lette agile metoder. XP fungerer bedst i små - mellemstore firmaer.

XP er bygget omkring “pair programming”, hvilket betyder at man sidder og programmerer med en makker, og at man fælles har ansvar for at koden:

- Har de features, som programmet skal have.
- Bliver testet og testene lykkes.
- Ikke kodes mere end nødvendigt.
- Er struktureret, simpel, og viser god klarhed.
- Bliver udgivet i små releases, så man altid har noget der kører.



Historien bag Extreme Programming

I 1980 blev Kent Beck introduceret til samtlige metoder at konstruere programmer på. Allerede som 21-årig vidste han, at der måtte være noget bedre i verden. Dog var det ikke før 1996, hvor han fik sin "Extreme Programming" idé sat i spil.

I 1996 blev Kent Beck nemlig inviteret til et projekt hos Chrysler Corporation, som hed Chrysler Comprehensive Compensation System (C3). Beck var uddannet konsulent og havde derfor et møde med Sue Longer, som var Chief Information Officer dengang. Eftersom C3 var noget rod, gav Beck Sue 3 valgmuligheder:

- Fortsæt med projektet som det er – det er der ingen glæde over
- Stop projektet – stor skam
- Giv alle en uges ferie og start forfra

Sue svarede derfor valgmulighed 3, og satte Beck som chef for projektet.

Her tog Beck muligheden for at prøve sit nye udviklingskoncept af, samt videre udvikle konceptet, finde små ændringer, og til sidst i 1999 udgav sin første version af Extreme Programming Explained.

I 2001 var Kent Beck med til at udgive The 12 Agile Principles.

Extreme Programming & The 12 Agile Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software
2. **Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.**
3. **Deliver working software frequently, from a couple of weeks to a couple of months, with preference to the shorter timescale**
4. Businesspeople and developers must work together daily throughout the project
5. The most efficient and effective method of conveying information not and within a development team is the face-to-face-conversation
6. **Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.**
7. **Working software is the primary measure of progress**
8. Maintain a sustainable working pace
9. **Continuous attention to technical excellence and good design enhances agility.**
10. **Simplicity, the art of maximizing the amount of work not done, is essential.**
11. **Self-organizing teams generate most value**
12. Regularly reflect and adjust your way of work to boost effectiveness

Sources:

- [Wikipedia](#)
- [Extreme Programming \(XP\) – Georgia tech – Software Development Process](#)
- [SCRUM VS EXTREME PROGRAMMING | WHAT ARE THE MAJOR DIFFERENCES?](#)
- [Extreme Programming 20 years later by Kent Beck](#)