

## 2. Hvorfor XP?

- Forbedret softwarekvalitet og respons til ændrede krav gennem:
  - Hyppige udgivelser
  - Øget produktivitet

## 3. Historie

- Udviklet af Kent Beck under arbejdet med Chryslers lønsystem (C3)
- Beskrevet i "Extreme Programming Explained" (1999)
- Startede Agile-revolutionen, som førte til Agile Manifesto

## 4. Hvad er XP?

- Letvægts agil proces for små til mellemstore teams
- Social forandring for at øge samarbejde og fokusere på at skabe værdi for kunden gennem hyppige leverancer og forbedret kvalitet

## 5. XP Paradigme

- Vær opmærksom, tilpas og ændr. XP tillader fleksibilitet.

## 6. Værdier

- Kommunikation, enkelhed, feedback, mod og respekt

## 7. Principper

- Menneskelighed, økonomi, gensidig fordel, selv-lighed, forbedring, diversitet, refleksion, flow, mulighed, redundans, fejlslag, kvalitet, babysteps, accepteret ansvar

## 8. Praktikker

- Dag-til-dag aktiviteter: Samarbejde, parprogrammering, brugerhistorier, ugentlige cyklusser, kontinuerlig integration, test først, inkrementelt design

## 9. Regler

- Forvaltning: Åben arbejdsplads, bæredygtigt tempo, stand-up møder
- Planlægning: Brugerhistorier, frigivelsesplanlægning, hyppige udgivelser
- Design: Enkelhed, systemmetafor, CRC kort, spike løsninger, refaktorisering
- Kodning: Kunde tilgængelig, kode standarder, TDD, parprogrammering
- Test: Test al kode, passér før udgivelse, bevis fejl ved enhedstest

---

## Noter til "Domain Driven Design"

### 1. Introduktion til Domain Driven Design (DDD)

- Forelæser: Henrik Bitsch Kirk
- Dato: 5. september 2023
- Institution: Aarhus Universitet

## 2. Domæneviden

- Vigtigheden af at forstå domænet og have en delt viden om det mellem udviklere og domæneeksperter.

## 3. Hvad er DDD?

- Fokus på kerne- og domænelogik
- Samarbejde mellem tekniske og domæneeksperter for at bygge en fælles forståelse af domæneproblemer
- Delt model mellem domæneeksperter, udviklere og interessenter

## 4. Strategisk Design

- Fordele ved en delt model: Hurtigere time-to-market, mere forretningsværdi, mindre spild, lettere vedligeholdelse
- Domænebegivenheder: Transformere data frem for statiske data, udløst af begivenheder

## 5. Event Storming

- Workshop metode udviklet af Alberto Brandolini
- Trin: Opret domænebegivenheder, tilføj kommandoer og aktører, identificer forretningsprocesser

## 6. Ubiquitous Language

- Bygge et fælles sprog mellem domæneeksperter og udviklere
- Skal kun indeholde ting repræsenteret i domænet og undgå tekniske termer

## 7. Bounded Contexts

- Mini-domæner eller subsystemer der er afkoblet og kan udvikles uafhængigt

## 8. Tactical Patterns

- **Value Objects vs Entities:**
  - Value Object: Modelerer værdi, ingen unik ID, immutable
  - Entity: Modelerer individuelle ting, har unik ID, mutable
- **Aggregates:**
  - Sammensat af en eller flere entiteter og value objects
  - Aggregat roden ejer alle andre elementer
- **Services:**
  - Indeholder domæneoperationer der ikke hører til en entitet eller value object
- **Repositories:**
  - Henter domæneobjekter fra datalager
- **Factories:**
  - Opretter domæneobjekter