

Välkomna till Projektmetodik och agila metoder

Introduktion

- Kursplan
- Vilka är vi?

Varför ett projekt?

- En kund har beställt en ny större funktionalitet
- En myndighet eller lag kräver förändring
- Nytänkande funktionalitet i er produkt

Vad är ett projekt?

Projekt är ett uppdrag som utförs av en tillfällig arbetsorganisation för att åstadkomma ett i förväg bestämt resultat.

- **Tidsbestämd**
- För att skapa en **unik** produkt, tjänst eller resultat
- **Målstyrt** för att skapa värde
- **Satsade resurser**

Tidsbestämt

- Det finns en början och ett slut
- Det kan vara i några veckor eller i flera år

Unikt

- Inte något som görs dagligen i verksamheten
- En ny produkt i portfolion
- Ett nytt effektivare sätt att arbeta

Skapade värden:

- Finansiellt eller strategiskt värde
- Ökad kvalitet
- Kundnöjdhet
- Innovation
- Bättre processer

Målstyrt

Projektet arbetar för att skapa något unikt, med hjälp av mål.

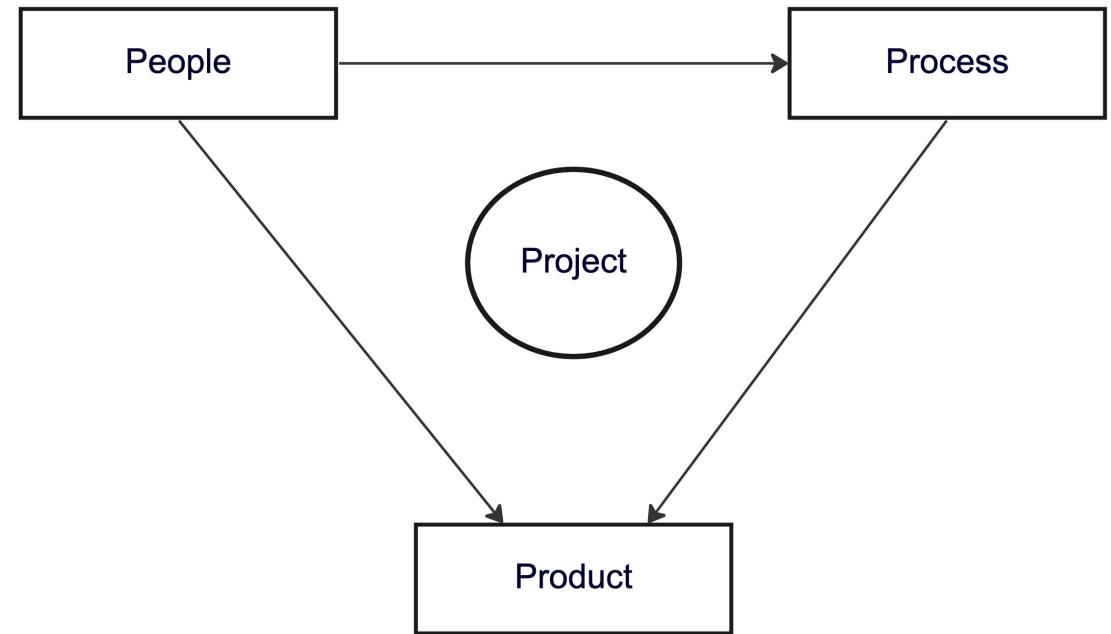
- När ska vissa delar vara klart?
- Vad är viktigast att göra just nu?
- Hur långt har vi kommit?
- Kommer vi fortfarande hinna i tid?

Resurser

- Ett projekt har resurser
- Pengar
- Medarbetare
 - Delade (25%, 50%)
 - Dedikerade
- Organisationer
- Team

4 P's i projektledning

- People
- Process
- Product
- Project



People

- Motiverade, lagkänsla
- Rätt kompetens
- Exempelvis
 - Projektledare
 - Team leaders
 - Intressenter i företaget
 - Specialister

Process

- Hur arbetar teamen?
- När kodas och testas produkten?
- Hur hanteras kvalitetsbrister
- Vilka faser i utvecklingen finns det?

Product

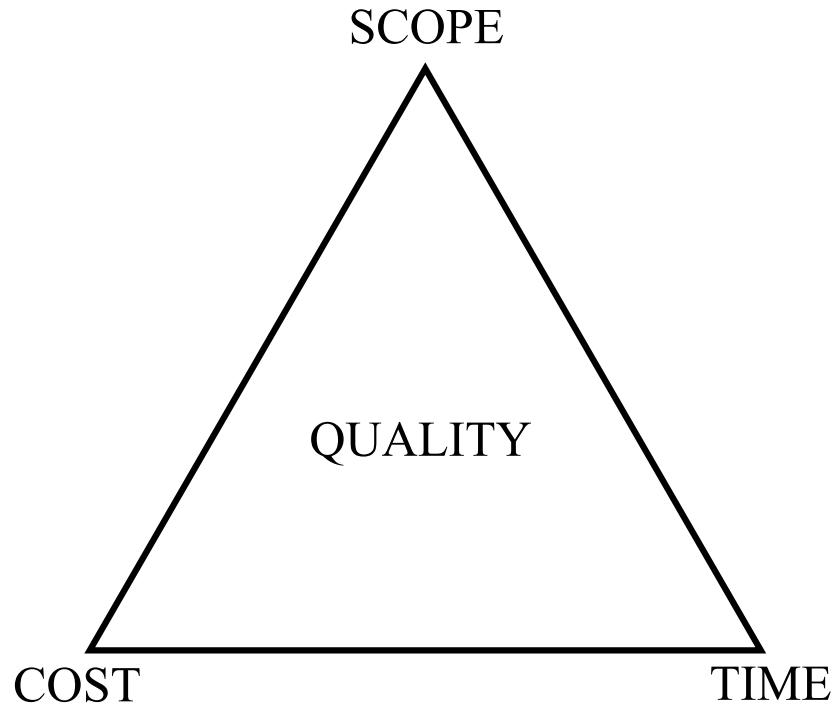
- Det som byggs
- Vad är omfattningen "scopet"?
- Begränsningar, vad ska inte byggas

Project

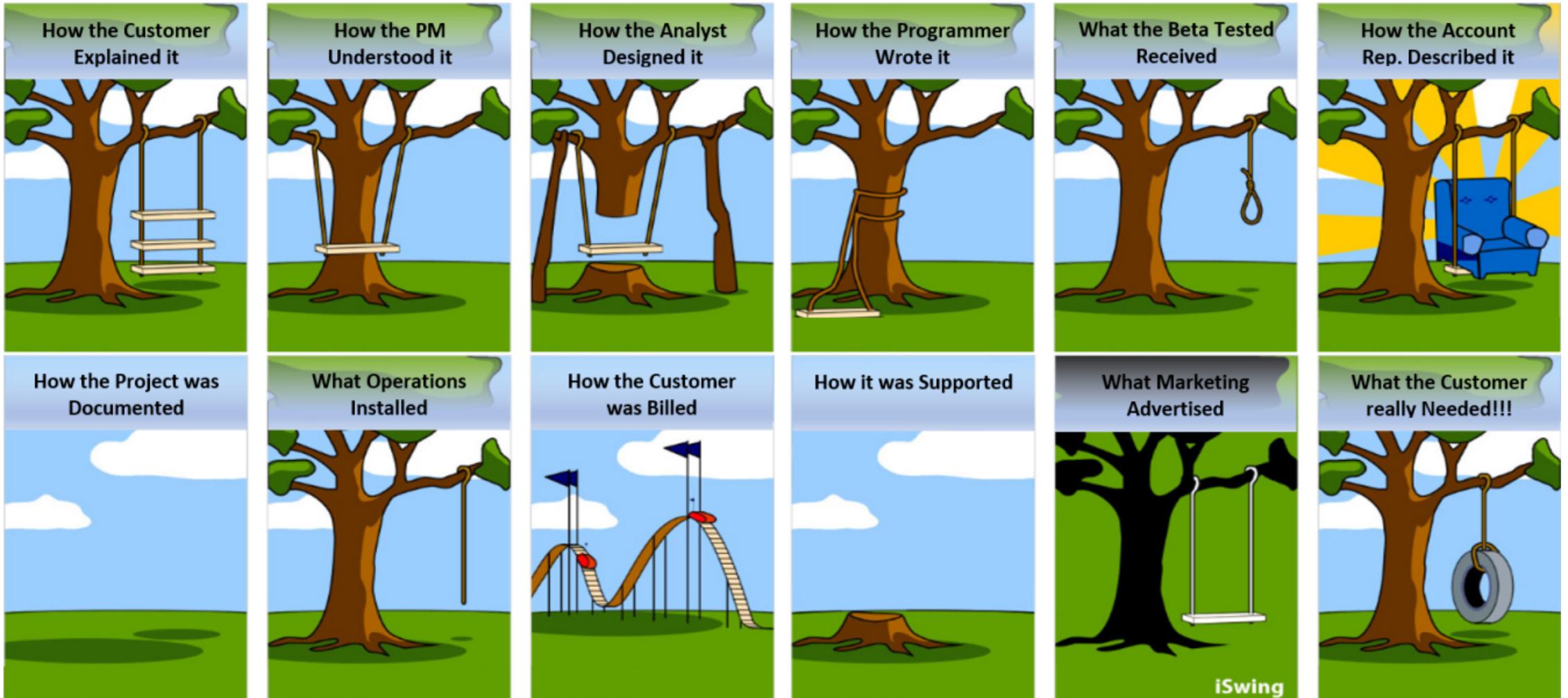
- Planera
- Följa upp
- Hantera risker

Projekttriangeln

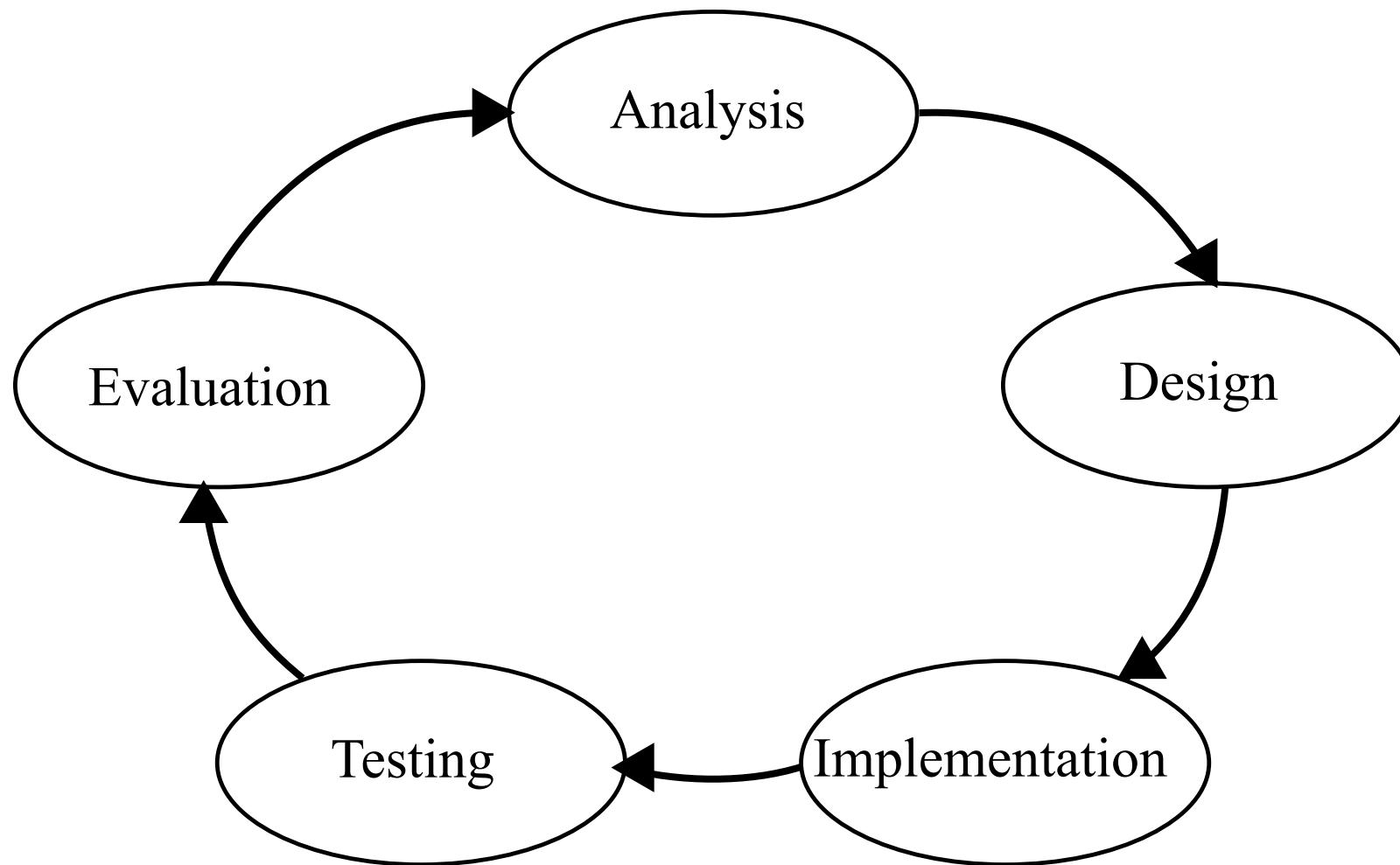
- Kvaliteten är begränsad av budget, deadlines och omfattning (scope), men det går inte att få allt, undvik scope creep
- Om scope ökar så behövs mer tid för att inte drabba kvaliteten
- Mer pengar kan spara tid genom effektivare verktyg eller proffs
- Det måste vara balans mellan alla sidor, annars faller kvaliteten



Att uppnå resultat



Systems development life cycle (SDLC)

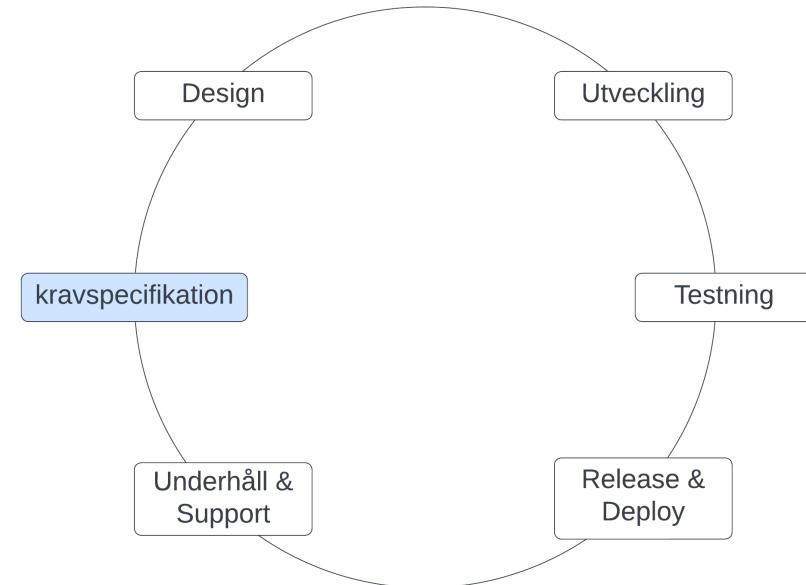


I praktiken

1. Skriva kravspecifikation
2. Design av lösning
3. Utveckling
4. Testning
5. Release & Deploy
6. Underhåll & Support

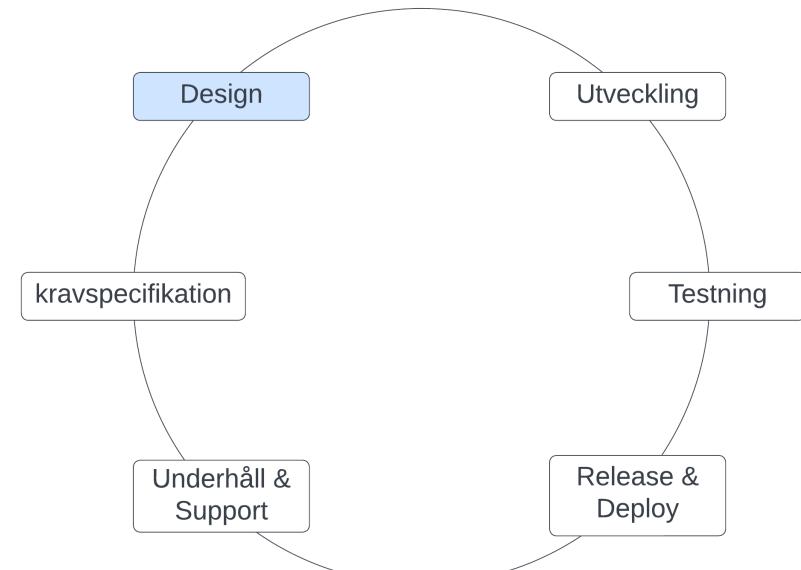
Kravspecifikation

- Funktionella Krav
 - En kund ska kunna skapa en egen användare
 - En admin kan skapa en support-användare
 - En användare måste själv skapa ett lösenord
- Icke-Funktionella krav
 - Säkerhet
 - Svarstid
 - Skalbarhet
 - Tålighet



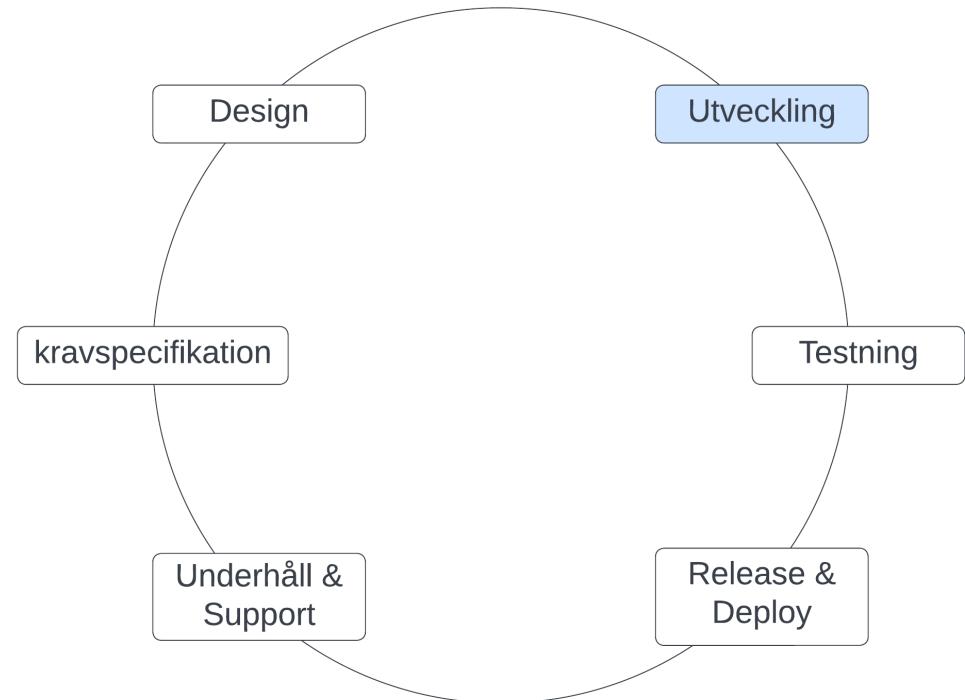
Design

- Arkitektur löser frågor på systemnivå
- Design bestämmer hur implementationen ska ske
- Att hoppa över design är ett vanligt misstag
- Att tänka först och koda sedan
- Att inte designa skapar "spaghettikod"



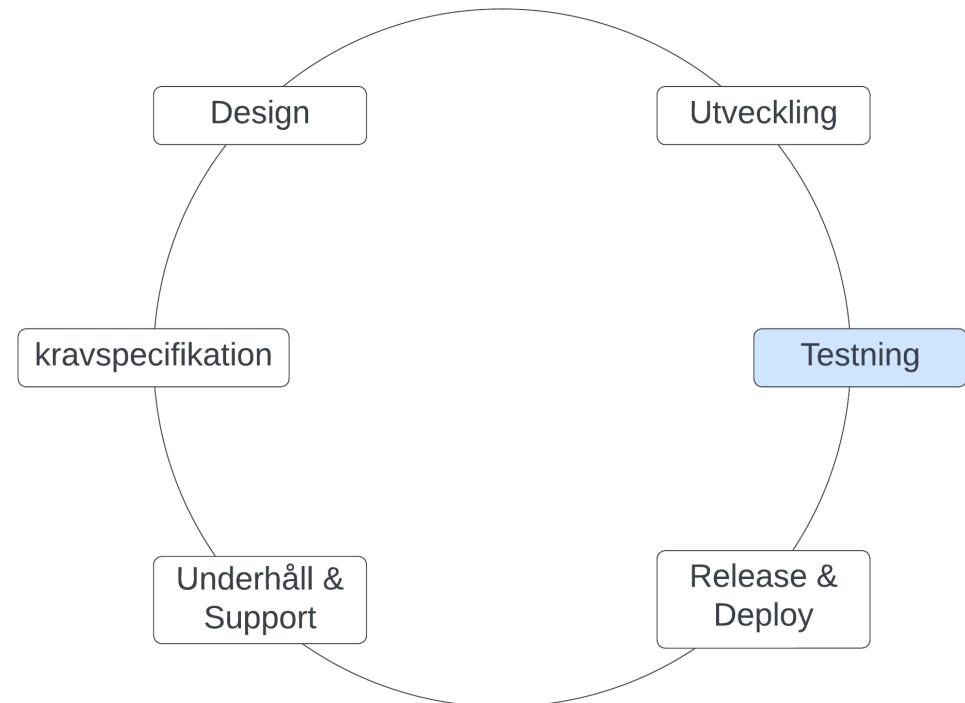
Utveckling

- Att koda enligt design
- Att koda enligt krav
- "Less is more"
- YAGNI - You aren't gonna need it (XP)
- TDD - skriv automatiska tester (XP)



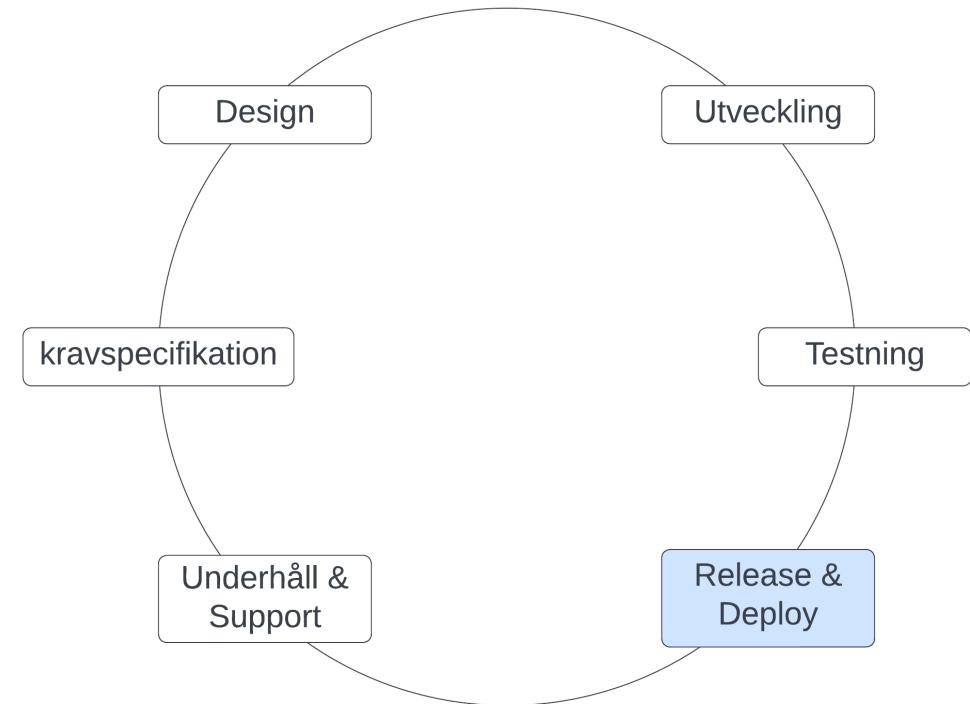
Testning

- Kontinuerlig integration
- Kör all test automatisering
- Testa sådant som inte är automatiserat
 - Säkerhetstest
 - Prestandatest
 - Lasttest



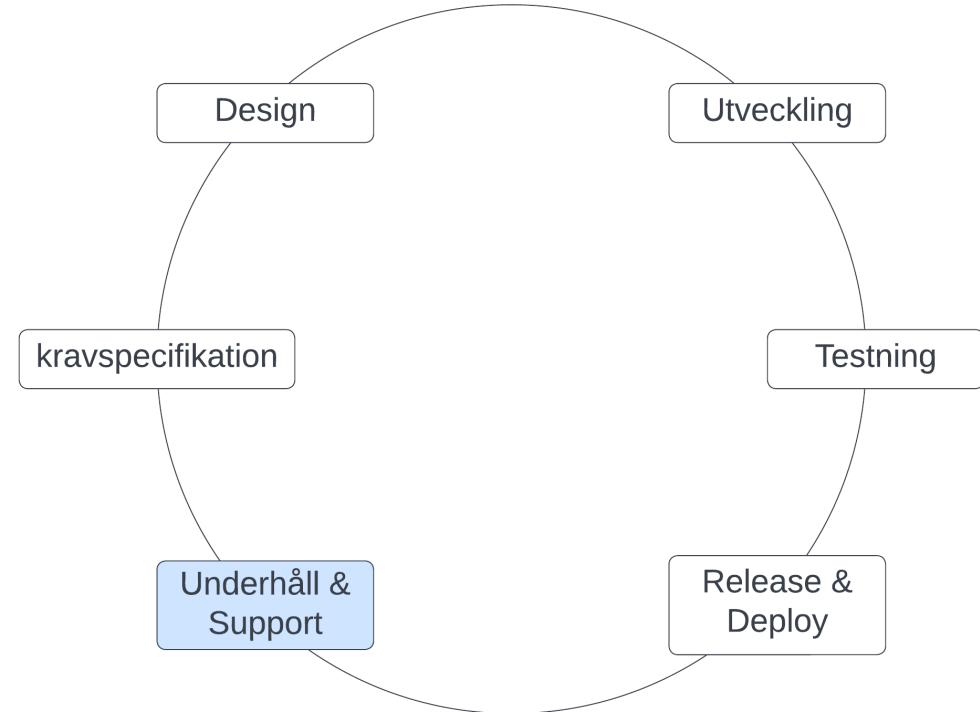
Release & Deploy

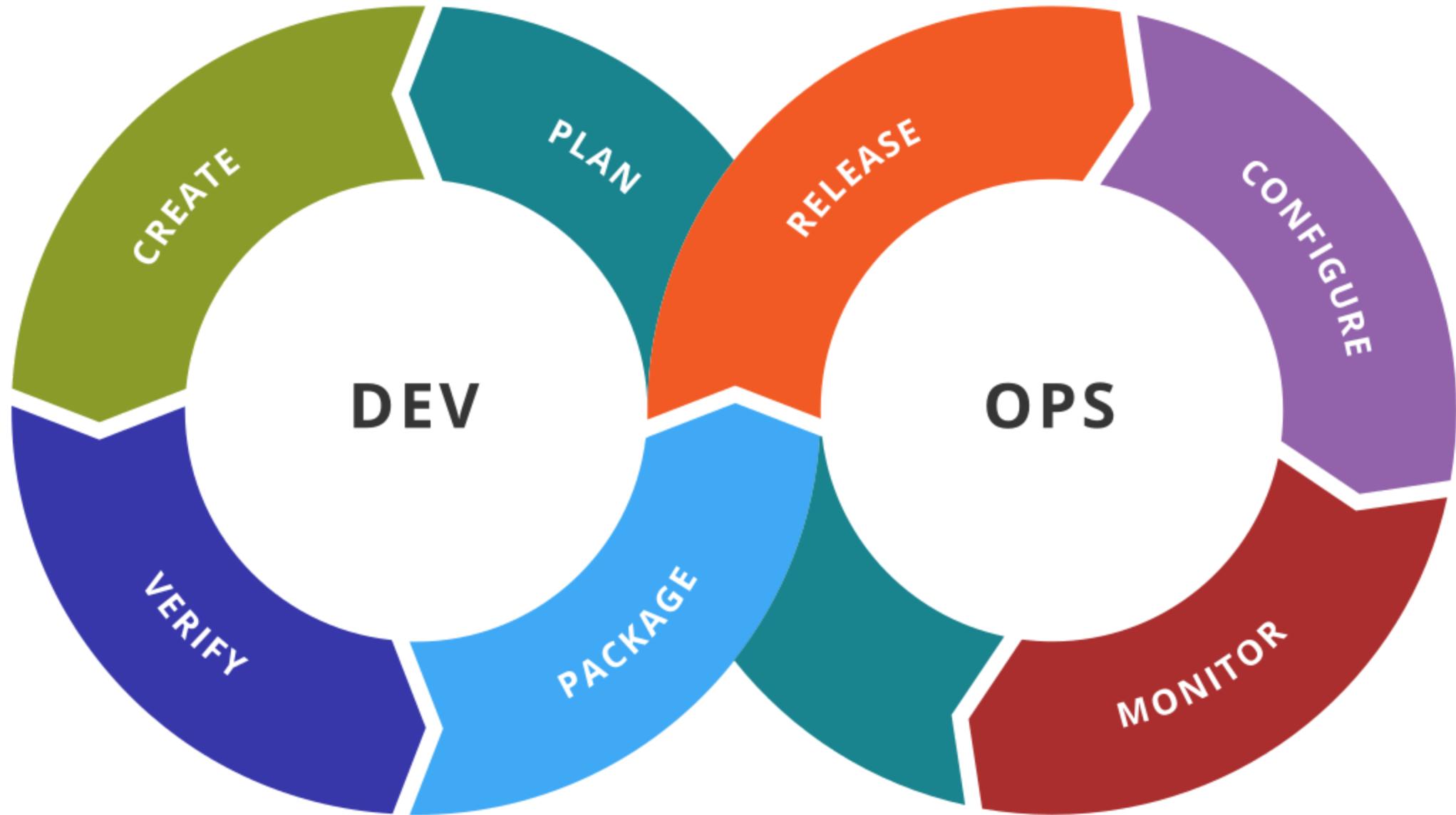
- Varför görs ändringen?
- Spårbart från kod till ärende
- DevOps eller överlämning till Ops
- Installation i driftmiljöer (deploy)



Underhåll & Support

- Övervakning av system
- Fungerar allt?
- Loggar, mätvärden, status
- Hur viktigt är systemen A,B,C?
 - Behöver vi ringa någon mitt i natten?
 - Vem äger systemet?
- Kan vi se historik över vad som har ändrats?
- Kan vi backa (rollback) en ändring?





Projektmetodiker

- Vattenfall (Samlingsnamn)
- Agilt
- Scrum
- Extreme Programming (XP)
- Kanban (lean)

Projektmetodiker för mjukvara

- Metodiker utvecklats över tid, t.ex. iterativa metoden Rational Unified Process (RUP)
- Agile är mer än en metodik, det är också en inställning
- Kanban är ursprungligen från tillverkningsindustrin
- Vattenfallsmetoden sammanfattar ofta allt innan agile. (slarvigt)
- Många delar är gemensamt mellan metoderna, SDLC eller applikationslivscykeln förändras inte för att projektmetodiken gör det.

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

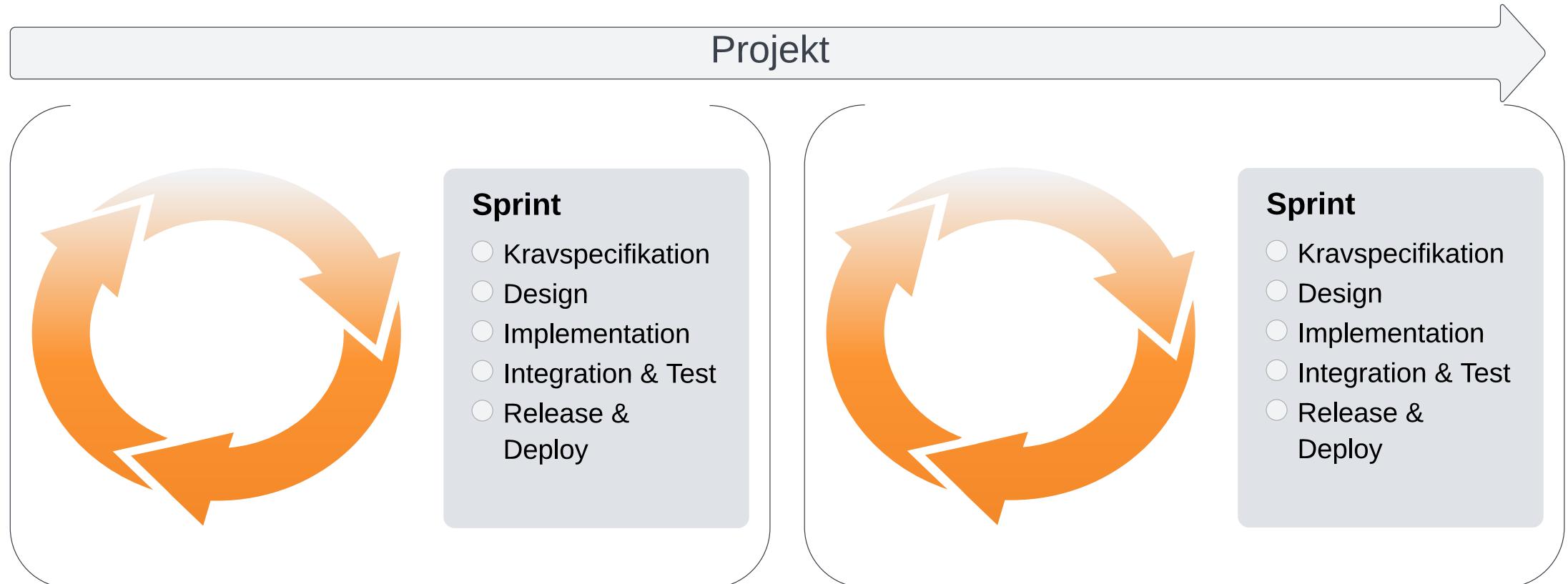
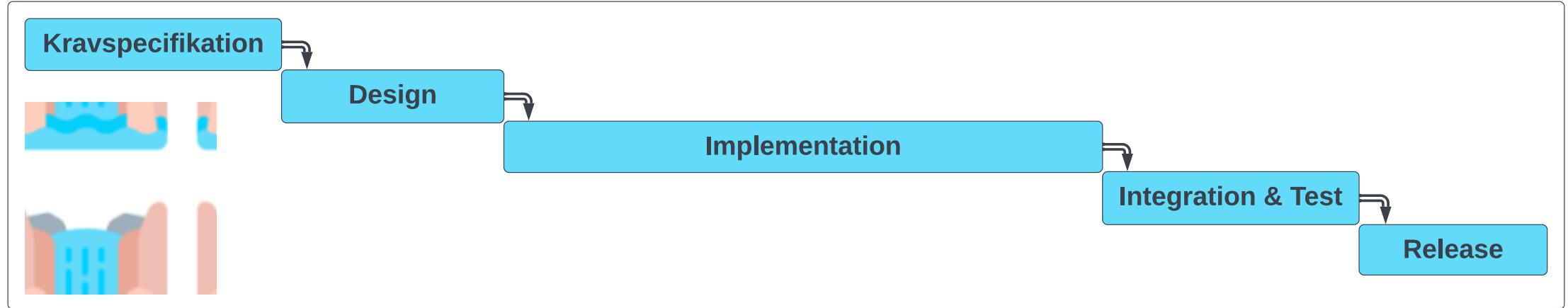
Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

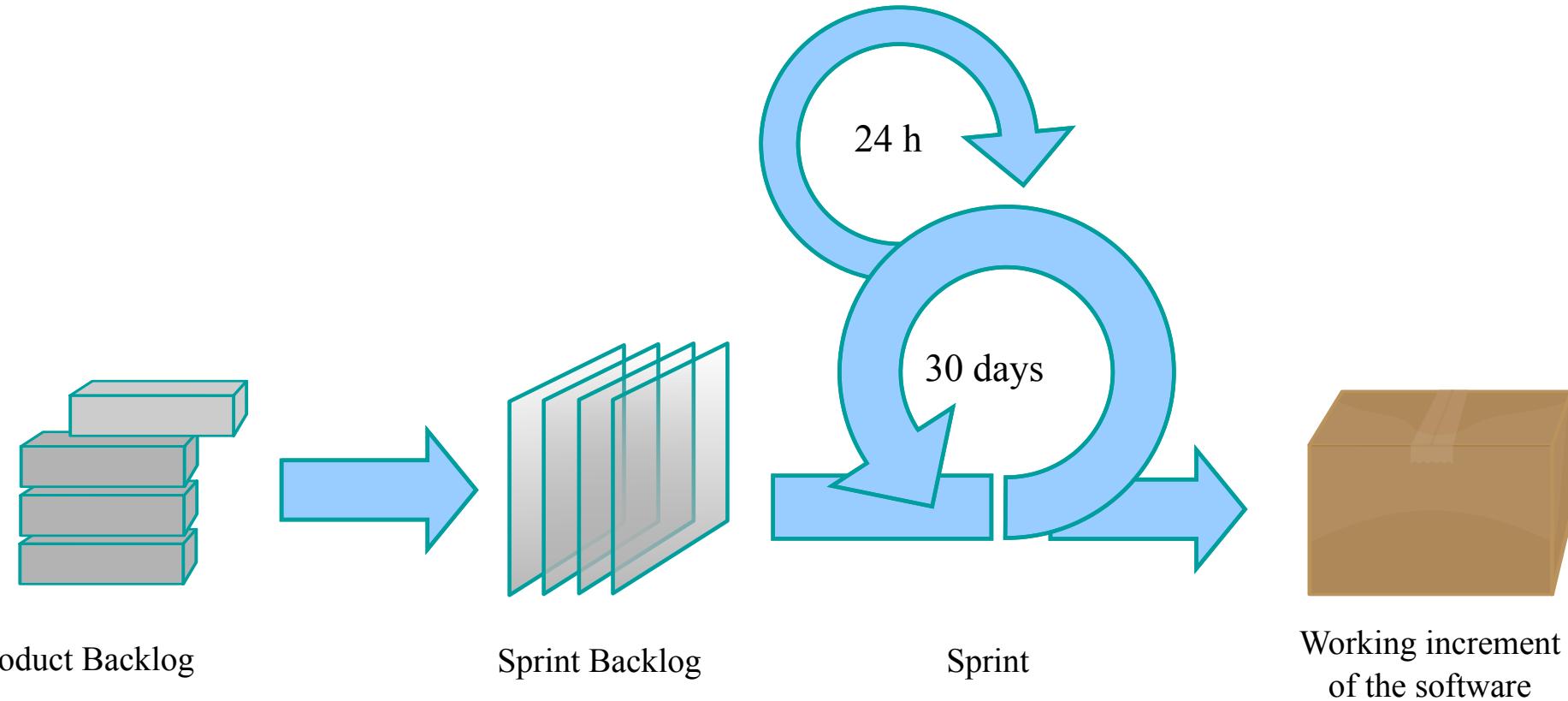
Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas



Scrum



Product Backlog

Sprint Backlog

Sprint

Working increment
of the software