# Sammanfattning D0039D

# Objektorienterad Analys och Design

## Scrum

### Grundpelare:

#### Transparens:

Alla ska ha samma definition of done. Både för funktioner/sprints/hela produkter  
Uppvisbara resultat  
Öppenhet – T.ex. om det går dåligt så säg det. Inga hemligheter hålls från teamet. Alla ska kunna förstå din kod

#### Granskning:

Utvärdering efter sprint/projekt  
Man mäter teamets hastighet  
Man har en demo för ägaren  
Product owner ska vara med och ge åsikter/förslag

#### Anpassning:

Man ska inte vara rädd att ändra saker långt in i projektet  
Iterativ/inkrementell process  
Inga hårddragna planer  
Kunderna vet inte alltid vad dom vill ha

### Artefakter:

Product backlog:  
Inget omfattande dokument/inte en plan för utförande  
Wishlist – vad önskar kunden att produkten ska ha för värde  
User story – Jag som [roll] vill kunna [uppgift] för att [nytta]  
Product owner äger dokumentet

#### Sprint backlog

Vad man gör under varje sprint  
Från produkt backlogg  
Scrum master äger dokumentet men skapar dokumentet med teamet

#### Burndown chart

Används för att uppskatta tid/arbete med hjälp av info från sprint  
Visuell representation av hur det går för teamet för pågående sprint

#### Increment(?)

Vars vi är nu  
Allt som är klart + denna sprint  
Fungerande demo

### Värderingar:

#### Fokus

#### Engagemang

#### Respekt

#### Mod

#### Öppenhet

### Roller:

#### Product owner:

Representerar alla stakeholders  
Bestämmer vart teamet ska gå  
Definerar visionen/scopet  
Äger product backlogg  
Prioriterar.

#### Scrum master:

Faciliterar arbetet  
Avlägnsar disktraktioner  
Kan arbeta med flera team  
Kan även vara en medlem av teamet

#### Team:

Implementerar tasks/items  
Ansvarar för kvalitetskontroller om man inte har ett speciellt QA-team  
Följer working agreement/värderingar

#### Stakeholders:

Stor varierande grupp  
Kan vara målgruppen, ägare av produkten, investerare etc.  
Har ingen kontakt med teamet

### Möten:

#### Sprint planning:

Vad ska göras denna sprint – efter prio satt av product owner utifrån Product backlog.  
Sprint backlogg skapas

#### Daily scrum:

Dagligt möte på ca 15-20 min  
Vad gjorde varje teammedlem igår? Vad ska de göra idag?  
Problem som har uppstått.  
Mindre omprioriteringar

#### Sprint Review:

Vad har gjorts/åstadkommits under sprinten?  
Demo för product owner, övriga team och kanske även stakeholders  
Ger PO info för att förändra sin vision/prioritering

#### Sprint Retrospective:

Hur ska vi kunna bli effektivare i vårat arbete inför nästa sprint  
Hålls med team och scrum master

## Working agreement

Normalt schema, även vid extrema situationer som t.ex. crunch  
Överenskommelser som coding conventions, hur kod ska kommenteras, Definition of Done  
Andra företagsregler.

## Surviving the workplace

### Tips:

Underskatta inte hur lång tid det tar  
Överarbeta inte dig själv (lär dig att säga nej)  
Spara alltid någonting till en regnig dag (ha alltid någonting att visa upp)  
Se efter dina egna intressen, men var en lagspelare  
”Utpressa” inte företaget genom att göra så att du är den enda som kan läsa koden  
Du är viktig, företaget bör uppskatta dig  
Du kan göra vad som helt och lära dig vad dom helst

### Att vara en bra ”team player”:

Följ working agreement  
Gör det du ska  
Stötta dina medarbetare  
Var inte rädd att be om hjälp, hjälpa andra  
Följ värderingarna

## Analys och Design

### Analys:

Vi undersöker vårt problem och vad vi behöver lösa  
Lös rätt problem!!!

### Design:

Konceptet på lösningen  
Lös det på rätt sätt!!!

### Coding conventions:

Guidelines för specifika programmeringsspråk  
Definerar programmeringsstil, olika tillvägagångssätt, metoder att programmera  
T.ex. indentering, namngivning, deklarationer

### Design patterns:

Återkommande problem – återkommande lösningar  
Creational - Lösningar för hantering av data  
Behavioural – Lösningar för speciella beteenden  
Structural – Lösningar för särskilda programstrukturer

Exempel:  
Vi vill ha ett objekt det ska bara finnas ett av. Objektet ska kommas åt globalt  
Lösningen är att vi använder oss av en singleton

### Arbetssätt

#### Iterativ:

Analysera problemet  
Designa en lösning för problemet  
Implementering  
Testning  
Förvänta inte att allt blir rätt första gången

#### Inkrementell:

Vi kan även leverera något vid varje iteration  
Varje delkomponent ka utvecklas iterativt  
Vilket innebär att den fullständiga produkten är inkrementell i form av iterationer  
Bygg i ”vertikala” slices (features) snarare än ”horisontell” (lager)

#### XP tekniker:

Pair Programming - Jobba i par med många avbyten  
Planning Game - Sprint planning  
Test Driven Development - Gör små test för varje metod som ska implementeras innan den kodas  
  
Continuous Integration - Försök se till så alla jobbar med den nyaste versionen. Commita ofta  
Small Releases - Släpp nya releaser kontinuerligt för att bygga förtroende hos kund