



# Projektlednings metodik



Föreläsning 7

# Dagens Upplägg

Misslyckade IT projekt

Framgångsfaktorer i ett projekt

Avslut Agila projekt

Vi förbereder oss för att avsluta projektet städa fint

Att dokumentera agilt

# Stand up meeting, vi delar med oss



# Varför är det så vanligt att projekt misslyckas?



# Varför är det så vanligt att projekt misslyckas?

Enligt olika rapporter, visar att *"Upp till 80 % av alla IT-projekt som slutförts under året har varit misslyckade. När det gäller systemutvecklingsprojekt är det färre än 10 % som varit lyckade"*.

Förmodligen finns det ännu fler projekt som inte lyckats, men de tas inte upp som post utan ligger under; Utveckling, prototyp, test,

## Definitionen av ett misslyckat projekt

- Budgeten hölls inte
- Tidsplanen hölls inte
- De uppsatta målen häll inte



# Varför är det så vanligt att projekt misslyckas?

## 1) Otydliga mål och krav

- Projekt saknar klara, mätbara mål.
- Visionen är diffus och stakeholders/intressenter är inte överens om vad som ska levereras

👉 När målen är oklara blir det omöjligt att avgöra vad som är "framgång".

## 2) Dålig kravhantering

- Krav ändras ofta utan styrning (scope creep), eller krav dokumenteras inte tillräckligt tydligt

👉 Förändringar utan kontroll skapar förvirring och extra arbete.

## 3) Bristande styrning och ledarskap

- Projekt saknar tydlig sponsor, styrgrupp eller beslutsfattare som driver projektet genom organisationen.

👉 Utan tydligt ledarskap blir projektet ineffektivt och tappar riktning.

## 4) Ottillräcklig riskhantering

- Risker identifieras för sent eller inte alls – och åtgärdsplaner saknas.

👉 Problem upptäcks först när de blivit akuta.



# Varför är det så vanligt att projekt misslyckas?

## 5) Dålig kommunikation

- Felaktig eller otillräcklig kommunikation mellan team, chefer och stakeholders.

👉 Missförstånd ökar och misstag upprepas.

## 6) Underresurser och kompetensbrist

- Projektet saknar rätt kompetens, tillräckliga resurser eller tid.

👉 Allt för få timmar / för få rätt personer gör att deadlines och kvalitet spricker.

## 7) Orealistiska tidsplaner och budgetar

- Projekt planeras med för höga förväntningar på leverans-tid och kostnad.

👉 Pressade tidsplaner ger stress, misstag och förseningar.

## 8) Tekniska problem / fel teknologival

- Felaktiga tekniska beslut, gammal teknik, eller komplexa integrationer utan planering.

👉 Teknikproblemen bromsar utveckling och driftsättning.



# Varför är det så vanligt att projekt misslyckas?

## 9) Bristande användar- och verksamhetsengagemang

- Slutanvändare och verksamheten involveras för sent eller inte alls.

👉 Leveransen blir fel eftersom den inte matchar verkliga behov.

## 10) Förändringsmötstånd i organisationen

- Medarbetare och chefer är inte redo att anamma ny teknik eller nya sätt att arbeta.

👉 Projekt levererar tekniskt, men förändringen misslyckas.

## 11) Otillräcklig testning

- Testfasen är för kort, saknar omfattning, eller ignorerar användarnas perspektiv.

👉 Fel och buggar upptäcks först i produktion.

## 12) Brist på kontinuerlig styrning och uppföljning

- Projekt mäts inte löpande, och avvikelser hanteras inte i tid.
- 👉 Problem växer sig stora innan någon agerar.



# Varför är det så vanligt att projekt misslyckas?

=

- IT-Projekt blir snabbt komplicerade
- Kräver ett annat sätt att hantera hur vi når leverans
- Många kompetenser/avdelningar etc med olika ”språk” och erfarenheter som möts i projektet
- Ofta många personer som är involverade i IT projekt är inte vana att skapa/tänka utveckling eller bygga system etc



# Framgångsfaktorer i ett projekt; Ben Day



Framgångsfaktorer med Ben Days: <https://www.youtube.com/watch?v=prqdhA9uhkA>

# Vem är Ben Day & vad är syftet är vi lyssnar på honom?

## Benjamin Day är

- Konsult & Tränare ([scrum.org](http://scrum.org))
- Specialiseras sig på bästa praxis inom mjukvaruutveckling
- Tonvikt på
  - Team
  - Scrum
  - MVP
  - Etc

Certifierad Scrum-tränare via Scrum.org & en talare på konferenser som TechEd, DevTeach & VSLive. När han inte utvecklar mjukvara, är Ben känd för att springa & paddla kajak för att balansera sin kärlek till ost, charkuterier & champagne.

*Han kan kontaktas via <http://www.benday.com>*

# Förberedelser

För att kunna ta till dig detta diskussionsunderlag & reflektioner

1. Vi tittar igenom klippet med Ben Day
2. Reflektera över vad Ben Days vill förmedla (i grupp)
3. Svara på frågorna (slide nr 9)

# Grupparbete



## Diskussion- Framgångsfaktorer enligt Ben Days - Management

Hur vänder vi ett (agilt) projekt på glid till ett lyckat projekt?

- Vad är de viktigaste delarna han trycker på som management/vi kan göra?
- Han återkommer till kommunikationen
  - Hur & vad vill han att vi ändrar på i kommunikationen
  - Vad är syftet? Vad vill han komma åt?
  - Vilka konkreta exempel tar han upp?
- Vilka stora delar kan vi påverka & på vilket sätt för att vända trenden?
- Är det tekniska delar vi fokuserar på i ett tekniskt projekt? Var ligger fokusen & varför?
- Håller ni med honom? Ja/Nej = Mötivera

Länk: <https://www.youtube.com/watch?v=prqdhA9uhkA>

# Framgångsfaktorer i ett projekt; Ben Day

## Hur vi kommuniceras

- Till teamet
- Till beställaren
- Vad vi säger
- På vilket sätt vi kommuniceras
- Våga prata om produktens status
- Våga prata om värde & Om det är värt att tex fortsätta (om projektet har kommit snett)

## DoD

- Tydligt kommunicerat & gemensamt framtaget
- Alla ska veta när ngt är klart. Inte nästan klart
- Vad som ska göras
- Vad som *inte* ska göras
- Acceptanskriterier
- Tydligt vad vi skapar, förtydliga (tex i refinement)
- Prioritera tydligt



**Scrum.org™**  
The Home of Scrum

# Framgångsfaktorer i ett projekt; Ben Day

## Fokus

- Vad kan levereras
- Leverera mindre delar hela tiden. Istället för en stor del då & då
- Tydligt vad som är klart & inte klart
- Att bara lova det vi verkligen kan lova

## Wip? Waste?

- Arbeta på rätt saker
- Färre delar samtidigt
- Fokus på att leverera tillsammans
- Bättre att leverera ngt än inget, & bara ha en mängd us som är påbörjade
- Kan få snabbare feedback fr beställaren
- Tydligare vad vi iner ska göra



# Agilt projektavslut & Tillhörande dokumentation



# Agilt projektavslut & Tillhörande dokumentation

## Agila projekt arbetar i iterationer (cykler) = Sprintar

- Genom deras *tätare feedback loop*, har det genom hela projektet regelbundet skett små leveranser & godkännande av leveranserna
- I det Agila arbetet ingår även att *regelbundet utvärdera & se vilka lärdomar & förbättringar* som man kan göra i projektet. Det gör att vi även här är väl förberedda & har en god överblick över hur projektet har gått
- Alla som är med i projektet har koll på status genom hela projektet
- Det betyder att när vi närmar oss slutet har vi inte så mycket arbete kvar - Målet är att Agila projekt ska ha en *jämn arbetsbelastning* under hela projektet. Inte vara tunga i starten eller i slutet, som traditionella projekt ofta blir



# Agilt projektavslut & Tillhörande dokumentation

## Vid avslut

- Skrivs en slutrapport sk white book eller testamente
- Överlämning; sker ofta kontinuerligt & blir lätt en förhandling om vad som ska ingå i den sista sprinten (cykeln)
- Kan vara en vanlig sprint eller en sk öppensprint (har inget slutt datum- utan man gör klart det som är överenskommet)
- Efterarbete

## White book/testamente

- Skapa en samlad bild av projektet - tillsammans med produktägaren (beställaren) & teamet (projektgruppen)
- Erfarenheter som gjorts; De 3 viktigaste erfarenheterna & löpande
- Viktiga beslut, som har påverkat projektet
- Tidstjuvar i projektet
- Viktiga externa händelser som har påverkat projektet
- Åtgärdsförslag
- Överlämning; Hur, vad , på vilket sätt, vem
- Viktigt! Vem är mottagaren; Tex Business eller utvecklare

Syftet är att verksamheten ska lära sig av sina erfarenheter & misstag



# Lessons - Learned

## Syfte

- Dra lärdomar & erfarenheter av ett projekt
- Lära sig av varandras misstag
- Lära oss av varandras framgång
- Dela erfarenheter & lärdomar med andra
- Förstå grundorsaker till varför det tex inte gick bra & hur vi kan göra annorlunda nästa gång
- Efter att projektet är avslutat

## Hur

- Ska dokumenteras
- Genom tex
  - Undersökning
  - Workshop
  - Intervjuer
  - Brainstorming



# Lessons - Learned

## Exempel, frågor att ställa

- Vad fungerade bra?
- Vad fungerade inte?
- Vad kan jag förbättra?
- Projektplanen var tillräckligt detaljerad?
- Genomförandet - Utifrån projektplanen
- Resultatet

## Hur

- Analysera era svar - dokumentera dem
- Sammanfatta resultatet
- Sammanfatta undersökningarna (om ngn sådan har genomförts)
- Detaljerad rekommendation

## Spara

- Lagra/arkivera dokumentationen (hur beror på de rutinerna som finns på det företaget du är på)
- Återanvänd - om ett liknande projekt ska genomföras. Nyttja då att det finns ett dokument som du kan dra lärdomar från.



# Grupparbete/Gruppdiskussion



Vi påbörjar att avsluta vårt projekt

- Vilka för & nackdelar ser ni med Agila projekt
- Vad har ert projekt för behov vid överlämning
- Börja planera upp hur den ska se ut
- Börja förbereda för slutrapporten:
  - Skapa en samlad bild av projektet - tillsammans med produktägaren (beställaren) & teamet (projektgruppen)
  - Erfarenheter som gjorts; De 3 viktigaste erfarenheterna & löpande
  - Viktiga beslut, som har påverkat projektet
  - Tidstjuvar i projektet
  - Viktiga externa händelser som har påverkat projektet
  - Åtgärdsförslag
  - Överlämning; Hur, vad, på vilket sätt, vem
  - Viktigt! Vem är mottagaren; Tex Business eller utvecklare

# Att dokumentera agilt



# Att dokumentera agilt i praktiken

## Vad säger det agila arbetssättet?

- Fungerande programvara - Framför omfattande dokumentation
- Anpassning för förändring - Framför att följa en plan

## Konsekvenser

- Måste vara enkelt
- Rörligt
- Kunna ändras snabbt & flexibelt
  - Sker ändringar
  - Förbättringar
  - Vi lär oss
  - Uppdateras etc
- Allt måste vara korrekt

Hellre mindre dokumentation som stämmer än mer dokumentationen som inte är uppdaterad & inte stämmer!



# Att dokumentera agilt i praktiken

## Tips på hur vi ska lyckas

- Förenkla
- Minimera
- Sammanfatta regelbundet
  - Efter varje sprint de viktigaste som behöver kommas ihåg & tex kommer vara med i slutrapporten
  - Viktiga beslut
  - Viktiga händelser etc

## Tips!

- Modulärt - alla moduler är självständiga
- Varje modul ska för sig kunna uppdateras utan att resten av ett dokument påverkas
- Tvärtemot ”traditionell dokumentation” där vi har en tydlig röd tråd



# Att dokumentera agilt i praktiken

## Dokumentation

- Prioritera, vad ska dokumenteras? Vad är viktigt
- Rutiner, Struktur & Strategi
- Integrering med populära agila projekthanteringsverktyg som Jira, Trello etc för att enkelt länka dokumentation direkt till uppgifter, epics & användarberättelser
- Använd inte dokumentation som en ersättare för kommunikation

**Exempel på agil dokumentationsmetod** - Vi dokumenterar hela tiden, löpandes

- User Story
- Acceptanskriterier
- Sprint backlog
- Produktbacklog
- Burndown diagram
- Retrospektiva anteckningar



# Att dokumentera agilt i praktiken

## Dokumentation i praktiken

- Planera din Agila-dokumentation (vad är nödvändigt, strategi etc)
- Bestäm syftet med varje dokument
  - Skriv inte/dokumenterar inte för sakens skull
  - Vem är publiken?
  - Vilken information behöver de?
  - Hur kommer de att använda dokumentet?
  - Finns detta dokument redan någon annanstans?
- Finns det ngt som kan automatiseras? Info kan tex tas från kodkommentarer

**Exempel på agil dokumentationsmetod** - Vi dokumenterar hela tiden, löpandes

- User Story
- Acceptanskriterier
- Sprint backlog
- Produktbacklog
- Burndown diagram
- Retrospektiva anteckningar



# Att dokumentera agilt i praktiken



Till nästa gång vi ses

# Till nästa gång vi ses

## *Grupppresentation*

- Vi presenterar vårt grupparbete inför klassen
- Schema kommer om tider & grupp ordning (läggs ut på Discord & Learnpoint)

## Hemtentamen

- *Jag kommer att vara tillgänglig mellan 8.00 - 17.00 för förtydligande/frågor*
- *Discord (er)*
- *En egen tråd/Rum*
- *Inga frågor besvaras i DM = Alla frågor tas öppet så att alla kan se dem och se svaret*