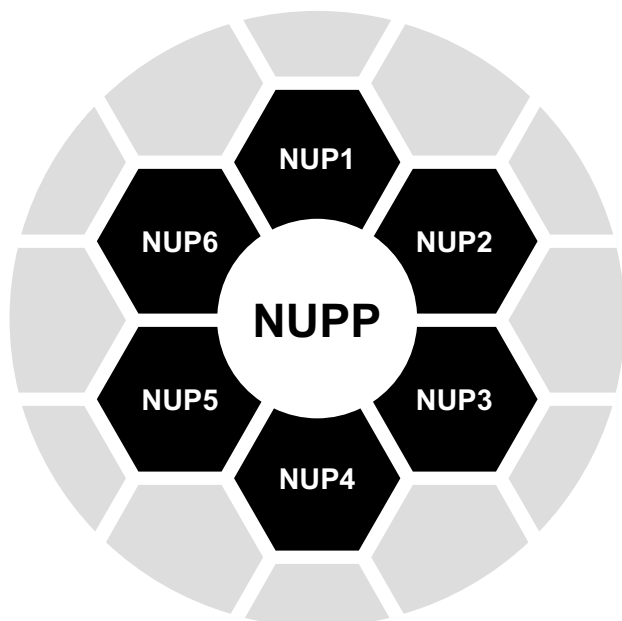


# NUPP

Nesten universelle prosjektprinsipper



Dette er en nedlastbar versjon av nettmanualen (<https://omimo.org/no/modules/nupp/>), generert den 2026-07-02. Se nettstedet for nyere versjoner.

NUPP kommer fra OMIMO (<https://omimo.org/no/>), en familie av åpne og minimalistiske moduler.

Denne manualen kan brukes og distribueres fritt under lisensen Creative Commons Attribution 4.0 International.

Co-funded by the  
European Union



OMIMO er delfinansiert av Den europeiske union. Synspunktene og meningene som uttrykkes er imidlertid utelukkende OMIMOs egne og gjenspeiler ikke nødvendigvis synspunktene til Den europeiske union eller EPOS VZW. Verken Den europeiske union eller tilskuddsmyndigheten kan holdes ansvarlig for dem.

Oversatt av: [Barbro Moen Ternsten](#)

# Introduksjon

---

NUPP er en samling av nesten universelle prinsipper for prosjekter: de vi vil lykkes med å følge i alle prosjekter, uavhengig av metodene og tilnærmingene vi bruker for å maksimere vår suksess.

Hver av de tilgjengelige ressursene og metodene for å kjøre prosjekter er avhengig av enkelte av disse NUP'ene (nesten universelle prinsipper). Imidlertid må følgende punkter tas i betraktning:

- Det er vanligvis ikke alle som er like relevante, og det vil være nyttig for utøvere å vurdere alle NUPer i stedet for en delmengde.
- De underliggende prinsippene er vanligvis ikke beskrevet godt nok i de tilgjengelige ressurser og metoder, og de fleste utøvere er så engasjert i praktiske detaljer at de glemmer prinsipper og gjør ting som ikke er kompatible med dem.

NUPP er kompatibel med alle de store metoder, systemer, ressurser og rammeverk som PRINCE2®, PMBOK® Guide, P3.express, PM², DSDM®, XP og Scrum. Det kan imidlertid være lite kompatibilitet med visse tolkninger av disse rammeverkene, og det er her NUPP forsøker å oppmuntre utøvere til å revurdere sine tolkninger.

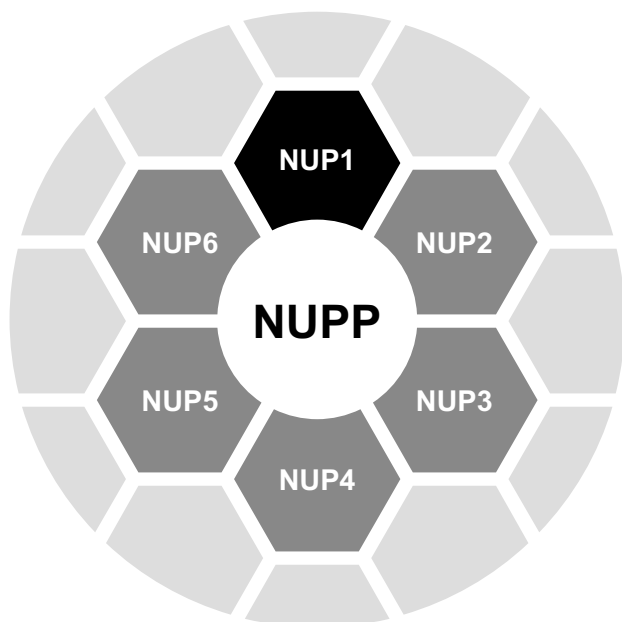
NUPP er en samling av følgende NUP:

- [NUP1 — Foretrekke resultater og sannheter, fremfor gruppetilknytninger](#)
- [NUP2 — Bevare og optimalisere energi og ressurser](#)
- [NUP3 — Alltid være proaktiv](#)
- [NUP4 — Husk at en kjede kun er så sterk som det svakeste ledd](#)
- [NUP5 — Ikke gjør noe uten et klart formål](#)
- [NUP6 — Bruk repeterbare elementer](#)

## NUP1

### Foretrekke resultater og sannheter, fremfor gruppetilknytninger

---



Vi har alle en naturlig tendens til å tilhøre grupper, en tendens som ofte går utover sin grunnleggende form, skaper sterke tilknytninger og forårsaker problemer. Vi mister mye mer enn vi får på grunn av tilknytning. Vi kan bli mer profesjonelle og effektive eksperter hvis vi ikke begrenser vår identitet og preferanser til bestemte grupper.

#### Eksempel: Agile vs vannfall

En gruppe svært entusiastiske mennesker som var modige nok til å prøve adaptive utviklingsmetoder i IT-utvikling på et tidspunkt da normen skulle bruke prediktive tilnærminger, kom sammen og kalte deres tilnærming "Agile". Det er fortsatt mange entusiastiske og resultatorienterte mennesker i Agile-samfunnet, men dessverre er det også noen mennesker i dette samfunnet som gjør Agile til en kult, og anser alle utenforstående som fiender. Dette forårsaker problemer på flere måter, inkludert følgende:

- Det begrenser dem i å kunne lære av noen utenfor sin gruppe
- Det fraråder utenforstående å lære av dem
- Det å tilhøre gruppen blir ofte viktigere enn det virkelige formålet, noe som igjen forhindrer at mange av medlemmene lærer den virkelige betydningen av Agility

Dette problemet kan bli betydelig redusert, hvis ikke fjernet, ved å bruke "Agile" bare som en etikett som refererer til en utviklingsmetode snarere enn som et fellesskap med medlemmer; og ved å ha folk som betrakter seg som skapere, problemløsere og ledere, som ser Agile rett og slett som et av flere verktøy, fremfor som deres identitet.

Det er ingen Agile-vannfall krig for ekte fagfolk.

## **Eksempel: PRINCE2® vs PMBOK® Guide**

Det er mange fagfolk i samfunnet som knytter seg til enten PRINCE2® eller PMBOK® Guide (vanligvis på grunn av deres geografiske plassering) og er ikke kjent med den andre. Vi kan alle ha preferanser mot bestemte rammeverk og ressurser, men ikke som vår identitet, og enda viktigere, må vi gjøre oss kjent med dem alle for å utvide vårt perspektiv og våre valg.

Den virkelige profesjonelle er åpen for alle ideer, ser etter dem, lærer om dem, og bruker dem når det trengs, uten gruppetilknytning.

## **Eksempel: Kontinuerlig læring**

Tilknytninger tilfredsstiller personen på grunn av følelsen av tilhørighet som oppstår, men det setter ikke et press på å lære. Noen ganger blir man til og med oppmuntret til å ikke lære av frykt for å miste dem. Når du er en fri person, en ekspert uten tilknytning, må du fylle inn gapet med læring: med kontinuerlig læring.

Det vi tror på i dag er ikke sannheten. Det er bare vår beste forståelse så langt, som må forbedres når vi fortsetter. Det er noe galt om ens ideer er akkurat det samme som for noen år siden. Dette er også tilfelle for NUPP: Hvis du kommer tilbake etter noen år og ser nøyaktig det samme, bør du bli mistenksom.

## **Eksempel: Åpenhet**

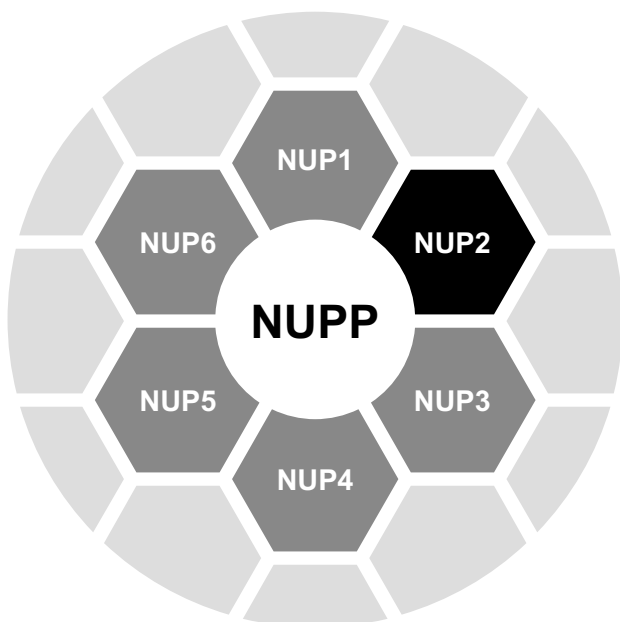
Når du diskuterer med noen, må du sørge for at du sikter mot ideen, og ikke personen. Dette bidrar til mindre konflikter. På samme måte, når noen diskuterer med deg, prøv å ikke tolke det som en krig mot deg, men heller en diskusjon av ideene dine, og hold deg åpen for det. Ikke lytt for å kunne svare, lytt for å kunne forstå; og for å kunne jobbe med den andre personen for å forbedre ideen.

Noen mennesker kan med vilje ha deg som person som mål i stedet for ideen, i så fall bør du hjelpe dem med å fokusere på ideen i stedet for på deg, og forsøk å holde dette fokuset gjennom hele samtalen.

## NUP2

### Bevare og optimalisere energi og ressurser

---



Ressurser er begrensede. Ressurser som er tilgjengelige for prosjektet ditt er begrensede, og det er også den mentale kapasiteten du må bruke for å gjøre gode beslutninger. Du bør bevare og optimalisere disse ressursene for deg selv og for prosjektet, og hjelpe andre team medlemmer til å gjøre det samme.

#### **Eksempel: 80/20 regelen**

En stor del av de potensielle gevinstene ved prosjektledelse kan oppnås ved å utnytte en liten del av innsatsen. I de fleste tilfeller er målet om å nå 100% av gevinstene svært kostbart og uberettiget. Du bør vurdere denne regelen i alt du gjør, og oppfordre andre til å gjøre det samme.

#### **Eksempel: Beslutningsudyktighet**

Vi bruker mental energi for å ta alle typer beslutninger, og også for å uttrykke viljestyrke. Hvis du bruker mye av denne energien til å gjøre unødvendige eller ubetydelige beslutninger, vil du ha mindre energi for de viktige avgjørelsene, noe som kan føre til dårlige resultater i prosjektet ditt. Dette er en av grunnene til at du bør unngå detaljstyring ("manage by exception" prinsippet i PRINCE2®).

Konflikter som handler om ideer kan være nyttige, men de som handler om mennesker, er vanligvis skadelige for prosjektet, og en av konsekvensene er at det drenerer den mentale energien til team medlemmene. Hvis du oppdager en slik konflikt, bør du gjøre ditt beste for å finne rot-årsaken, og løse den.

#### **Eksempel: Ta vare på deg selv!**

De avgjørelsene du tar og viljestyrken du uttrykker, bruker av din mentale energi. På den annen side blir denne energien fylt opp når du sover og spiser riktig. Så, ta godt vare på deg selv: sørg for at du får nok søvn og hvile, og spis godt. Hvis du har skadelige vaner eller problemer med søvn, trenger du

ikke å håndtere det alene; det er mange spesialister som kan hjelpe deg med å løse slike problemer. Fysisk aktivitet kan også bidra med tilføring av energi, selv om studier ennå ikke er avgjørende i denne saken.

Prøv å oppmuntre dine team medlemmer til å gjøre det samme som du gjør. Først må du sørge for at de jobber i bærekraftig tempo, og uten for mye overtid. Deretter, hvis du har valget, prøv å tilby energisk, sunn mat, snacks og drinker i løpet av arbeidstiden.

### **Eksempel: Jobb-Privatliv balansen**

Mange av oss elsker det vi gjør på jobb, men det er fortsatt ikke alt vi trenger å ha i livet. På samme måte som du optimaliserer arbeidet ditt, bør du planlegge og implementere ideer i ditt personlige liv, på måter som gjør det til et godt og lykkelig liv. Når du er lykkeligere, kan du bli mer vellykket på jobb også. Hvis du har muligheten bør du prøve å oppmuntre team medlemmene til å gjøre det samme.

### **Eksempel: Motivasjon**

Motivasjon er et komplekst konsept. Det er mange interessante og nyttige artikler og ressurser tilgjengelig om emnet, så vel som en del uvitenskapelige. Likevel bør du lære om det og bruke det kontinuerlig, da det øker teamets mentale energi og muligheten for å oppnå bedre resultater for prosjektet.

Motivasjon kan være så enkelt som å la folk få vite at du anerkjenner deres gode arbeid med et smil eller et "tak for hjelpen". Men du må være forsiktig, fordi mange av de vanlige formene for motivasjon, for eksempel små pengemessige belønninger, har en negativ effekt.

### **Eksempel: Samarbeid og teamwork**

Mennesker som samarbeider skaper ofte bedre resultater, men enda viktigere er at mennesker er sosiale og liker å være en del av en gruppe. Hvis du kan fjerne de negative aspektene av samarbeid og skape et hyggeligere miljø, vil medarbeiderne i prosjektet trives bedre.

Du bør imidlertid være oppmerksom på at folk er forskjellige, og noen trenger mer fokusert tid alene, både for trivselen sin del, men også for effektivitetens del.

### **Eksempel: Felles kultur**

Det er en tendens blant ulike interessenter å opprette adskilt grupper/team og dermed forårsake spenning med andre team; for eksempel folk som er fokusert på forretningsaspektet av prosjektet vs. personer som bygger produktet. Denne spenningen bruker mye energi fra begge sider, noe som er en av grunnene til at vi skal forsøke å bygge en felles kultur, hvor alle jobber sammen mot det samme målet.

### **Eksempel: Kollektiv intelligens**

Når et stort antall mennesker med variert bakgrunn og erfaring kommer sammen og arbeider i et felles miljø, er det potensiale for svært gode resultater, samt nye ideer og løsninger som er langt bedre enn de som kommer fra en enkelt fagperson.

Hvis du har muligheten, kan du bruke det regelmessig til å spørre gruppe-medlemmer for å hjelpe deg med å løse vanskelige problemer i prosjektet. Ved siden av muligheten for å få gode resultater, tillater det også team medlemmer å vite at deres meninger blir verdsatt, og at de spiller en viktig rolle i prosjektet, noe som igjen øker deres energinivå. Aktivitet E02 av P3.express er et eksempel på bruk av kollektiv intelligens i prosjekter.

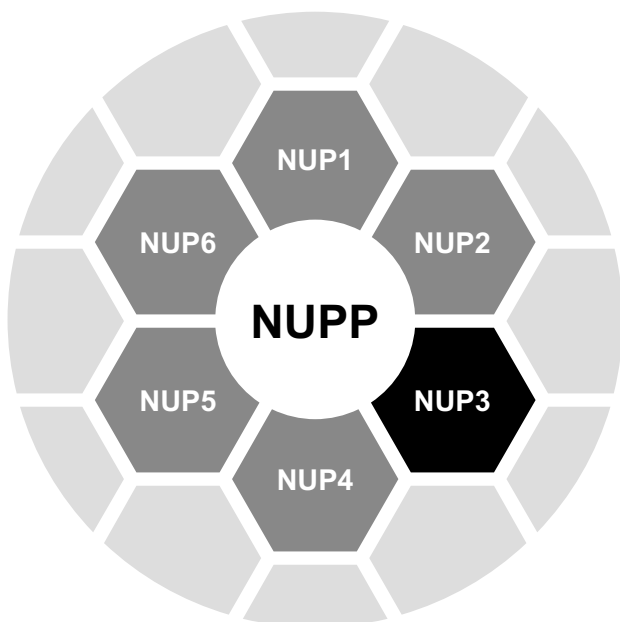
### **Eksempel: Prosjekt Fasilitator**

Hvis du er prosjektleder, er de fleste oppgavene dine av en fasiliterende art (eller i det minste bør være det). På den annen side kan du se at team medlemmene har hatt dårlige erfaringer med prosjektledere tidligere, og at disse opplevelsene har innvirkning på deres forhold til deg. En del av deres energi brukes til å analysere oppførselen din for potensielle trusler i stedet for å stole på deg. I så fall kan du endre tittelen fra prosjektleder til Prosjekt Fasilitator. Tross alt, det er det du egentlig gjør i prosjektet.

## NUP3

### Alltid være proaktiv

---



Det er en naturlig tendens i oss mennesker til å være reaktiv. Det kan hjelpe oss med å kaste bort energi på oppgaver eller forhold som er ubetydelige, eller det kan være en fordel når vi arbeider med et område der vi er helt inkompetente. Slike situasjonene er forskjellige fra våre prosjekter, og her kan vi få bedre resultater ved å være mer proaktive.

#### Eksempel: Planlegging

Hvis du vil kjøre til et nytt sted, og du er sen, kan du begynne å kjøre umiddelbart for å "spare tid" og håndtere mulige problemer når de oppstår. Den proaktive tilnærmingen er å ta litt tid i starten og bruke navigasjonssystemet for å gi deg den raskeste ruten basert på trafikken og mulige ulykker og blokkeringer og deretter kjøre. Du bruker litt tid i starten før du kjører, for å unngå problemer senere og dermed spare tid til slutt.

I motsetning til hva noen mennesker tenker på Agile-prosjekter, er planlegging alltid nødvendig, og det handler bare om typen og detaljnivået i planene. Planlegging før utførelse er en proaktiv tilnærming.

Husk sitatet: Gi meg seks timer til å hugge ned et tre og jeg vil tilbringe de fire første til å kvesse øksen.

Hvis det er et prediktivt prosjekt, kan du tilbringe 4 timer å kvesse øksen din, fordi du er sikker på at du skal hugge ned et tre. I et Agile-prosjekt er du ikke sikker på om du skal hugge ned et tre, samle ødelagte grener, grave etter kull eller noe annet. Likevel trenger du fortsatt å ha en generell forberedelse for alle aktivitetene (for eksempel finne ut hvor nærmeste maskinvarebutikk er), og gjøre en rekke forberedelser når du skal fokusere på en bestemt løsning; det er planlegging.

#### Eksempel: Planlegge planleggingen

Å planlegge måten vi skal gjennomføre prosjektet på, er en proaktiv tilnærming. Denne proaktiviteten

kan til og med utvides ved å planlegge måten vi skal planlegge gjennomføringen av; det er konseptet til styringsplanen for PMBOK® Guide, styringsstrategier for PRINCE2®, og tilnærminger i DSDM®.

### **Eksempel: Kontinuerlig planlegging**

Virkeligheten stemmer sjelden overens med det vi har planlagt, og det er OK - men vi må kontinuerlig tilpasse våre planer for å sikre at de blir realistiske og praktiske. Vi bør gjøre det så snart det er nødvendig med tilpasning, og ikke når vi får problemer. Det er en proaktiv tilnærming.

### **Eksempel: Risikostyring**

Hele konseptet med risikostyring er basert på proaktivitet: når vi står overfor usikre hendelser, i stedet for å vente på å se hva som skjer og deretter reagere på dem, tenker vi på muligheter og konsekvenser, vurderer tiltak, og sannsynligvis gjør noe med det før det skjer.

Legg merke til at det vi gjør i prosjekter noen gang handler om folks liv, og i slike tilfeller vil risikostyring være en helt sentral aktivitet.

### **Eksempel: Definere roller og ansvar**

Du kan la prosjektgrupped medlemmene jobbe uten klare roller og ansvar, og før eller senere kommer en form for roller og ansvar fram, men det er for tidkrevende og vil mest sannsynlig ikke fungere så bra. Den proaktive tilnærmingen er å definere roller og ansvar tidlig og tilpasse etter behov. Dette gjør arbeidet lettere for alle, og de kan fokusere på å produsere noe, i stedet for å bestemme hvem som gjør hva.

Antallet og variasjonen av roller er avhengig av prosjektets type og størrelse; Det kan være en enkel definisjon som den i Scrum, noe moderat som det i P3.express, eller noe omfattende som i DSDM® og PRINCE2®. Men ikke glem at rollebeskrivelsene i disse metodene bare handler om ledelsesaktiviteter, og at du alltid må legge til rollebeskrivelser for tekniske aspekter også.

### **Eksempel: Valgmuligheter**

Skal du lukke prosjektet for tidlig eller fortsette med det?

Det er sjelden bare to valg, selv om det kan virke slik. Du må ha en proaktiv tilnærming og vurdere alle dine valg før du tar en beslutning. Kanskje du kan re-scope prosjektet; kanskje du kan stoppe det til noe annet blir klart; eller kanskje du kan endre prosjektets tilnærming (f.eks. outsourcing), etc.

### **Eksempel: Kritisk tenking**

Vi lever alle med våre antagelser og virkelighetsoppfatninger som hjelper oss å overleve på den ene siden, men som på den andre siden kan lure oss til å gjøre dårlige beslutninger. Når det gjelder å ta viktige beslutninger om prosjektet, er det best å pause en stund og vurdere alle forutsetninger som kan påvirke vår beslutning.

Som referanse kan du bruke listen over kognitive forstyrrelser fra [Wikipedia](#).

Det finnes til og med rammeverk for beslutningsprosesser som du kan bruke til å ta bedre beslutninger. I begynnelsen kan det være distraherende og til og med irriterende å bruke dem, men snart blir du vant til dem, og kan dra nytte av dem uten mye bevisst innsats.

### **Eksempel: Åpenhet**

Vi liker ikke å være forsinket i prosjektet eller ha noen annen type problem, men det betyr ikke at vi skal gjemme det. Du bør være åpen om problemene og la interessentene vite om dem, fordi noen av

dem kan være i stand til å hjelpe deg, og i tillegg vil de få vite om problemene og deres konsekvenser før eller senere, og noen av dem kan kreve korrektive tiltak fra deres side (f.eks. for å godta den negative konsekvensen).

### **Eksempel: Kommunisere effektivt**

Det vil skje at du sender statusrapporter til dine interessenter, og at tilbakemeldingene uteblir. Du kan tro at alt er bra bare fordi det ikke er noen tilbakemelding, selv om det kanskje ikke er slik. Du må være proaktiv og se om de virkelig har satt seg inn i statusen, og om statusrapporten har tjent hensikten, og bruk eventuelle tilbakemeldinger for å justere kommunikasjonsmetodene dine.

### **Eksempel: Ta ansvar**

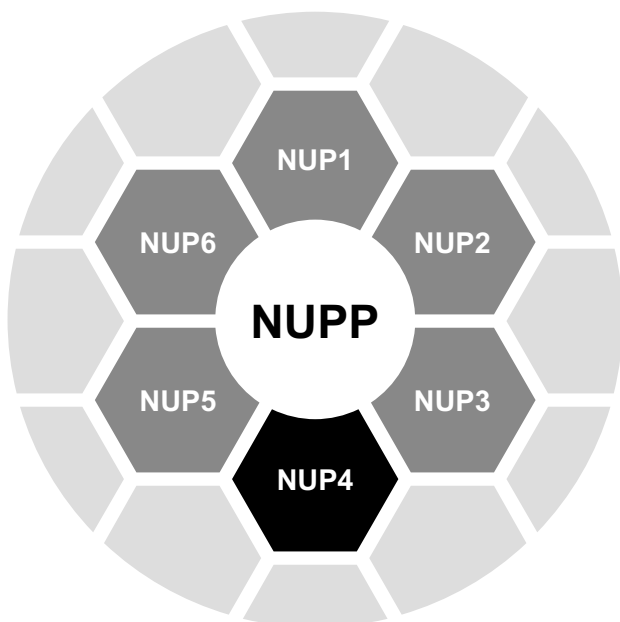
Det er lett å klandre andre for dårlige resultater. For eksempel vil du kanskje at organisasjonen skal gi deg full autoritet til å endre alt via prosjektet og gjøre det perfekt, men de gjør det ikke, og som et resultat svikter prosjektet. Dette er ikke en proaktiv tilnærming.

Den proaktive tilnærmingen er å ta ansvar og gjøre alt du kan innenfor rammebetingelsene som er satt. Du kan ikke forvente at organisasjonen fullt ut stoler på deg og gir deg alt i håp om å få gode resultater, spesielt når de har sett så mange mislykkede prosjekter. Det du må gjøre er å gjøre en liten forbedring innenfor de rammebetingelsene som er satt, bruk det for å få litt tillit, noen flere ressurser og litt mer toleranse for feil og utbedringer, og bruk deretter det for en litt større forbedring og fortsett slik til du når målet.

## NUP4

Husk at en kjede kun er så sterk som det svakeste ledd

---



Det er ulike domener i prosjekter, og de trenger alle oppmerksomhet. Vi må ha et helhetlig perspektiv på prosjektet. Å være oppmerksom på et tilsynelatende viktig domene (f.eks. tid) er ikke nok, fordi alle domener virker sammen, og de fungerer ikke skikkelig med mindre de alle får tilstrekkelig oppmerksomhet.

### Eksempel: Alt for å rekke tidsfristen!

La oss si at du bygger noe for de olympiske leker. Det har en absolutt deadline, noe som gjør tidsstyringen svært viktig. Men er det riktig å bare tenke på tid?

Hva om kvaliteten er så lav at det nødvendiggjør gjentagende arbeid etter en stund. Det ville påvirke tiden, så dermed er både tid og kvalitet viktig. Du kan ha en fancy hage som er beskrevet i den opprinnelige definisjonen av prosjektet, men du vet at hvis det ikke er nok tid, kan du hoppe over det og bare dekke det til med gress, så lenge du har vurdert denne muligheten i tide og er forberedt på den. Dermed er håndtering av omfang også viktig. Nå har vi omfang, tid og kvalitet som domener i sentrum av vår oppmerksomhet.

Etter hvert som du opparbeider deg erfaring, legger du merke til at hvert enkelt domene i prosjektet bidrar til tidsstyring, og du kan ikke møte fristen med et akseptabelt nivå av sikkerhet, med mindre du tar hensyn til alle domener.

### Eksempel: Cherry picking

Når folk står overfor en rekke metoder, begynner de ofte å favorisere noen av dem og lage en blanding av alt som synes interessant fra forskjellige rammeverk. Dette fungerer vanligvis ikke, fordi elementene ikke virker isolert og må være kompatible med hverandre. Eventuelle tillegg eller endringer i et rammeverk bør gjøres fra et helhetlig synspunkt.

Derfor ser vi noen ganger motstridende elementer i forskjellige metoder. Noe fungerer godt i ett system, og det motsatte virker godt i et annet system. Det enkelte elementet er ikke riktig eller feil på egen hånd.

Den sikreste tilnærmingen er å velge en metodikk for prosjektet, skreddersy det og legge til nye elementer ved å vurdere konsistensen av hele systemet.

### **Eksempel: Anti-prosess tilnærming**

En av de beste tingene Agile metoder har gjort er å trekke oppmerksomhet på menneskelige aspekter. Agile Manifesto gir mer verdi til enkeltpersoner og samspill, i forhold til prosesser og verktøy, selv om dette kanskje ikke er en rettferdig sammenligning. Nesten alle metoder sier at menneskelige aspekter er viktige, og den virkelige forskjellen med Agile metoder er at menneskelige aspekter er en integrert del av deres prosesser, snarere enn et forslag. Så, det handler ikke om konkurranse mellom menneskelige aspekter og prosesser, men snarere om hvordan menneskelige aspekter sees på i systemet.

Det er ingen tvil om at noen mennesker prøver å erstatte de menneskelige aspektene ved å ha mer sofistikerte prosesser, men det er et misgrep. Selv det omvendte eksisterer: folk som prøver å erstatte prosesser med menneskelige aspekter, som heller ikke fungerer så bra.

### **Eksempel: Dette er alle domenenene du trenger**

Når du tenker på domener, bør du være påpasselig så du ikke går glipp av noen. Men, hva er egentlig de mest kjente domenenene? Hvis du sjekker de grunnleggende rammeverkene, vil du finne forskjellige svar; og likevel er ingen av dem den hele og fulle sannheten.

PRINCE2®-temaer er domener, men det er bare de domenenene som spiller en nøkkelrolle i metodikken. De andre domenenene er bare underforstått.

PMBOK® Guide er ikke en metodikk og kan formulere det mye bedre med de ti kunnskapsområdene. Imidlertid er disse tolkninger av alle domener basert på PMBOK® Guides perspektiv på prosjektet, i stedet for en nøytral (ikke at det nødvendigvis er et nøytralt perspektiv). For eksempel får menneskelige aspekter ikke mye oppmerksomhet i PMBOK® Guide.

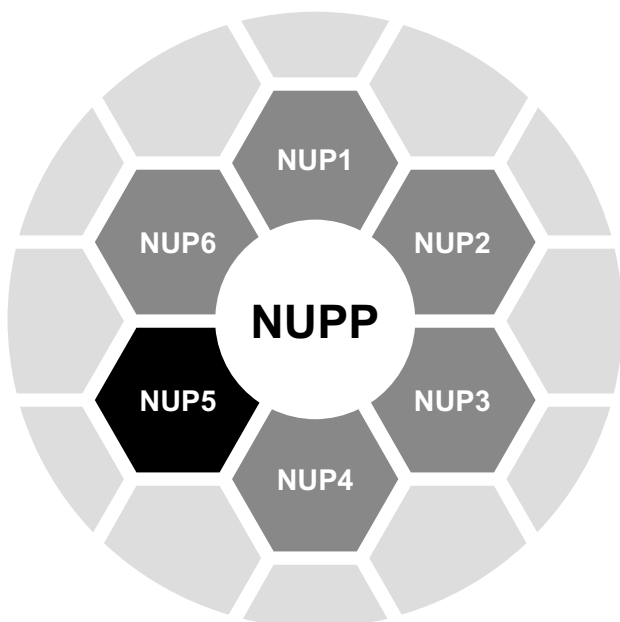
En god kilde til informasjon om domenenene er ICB. Det handler imidlertid ikke om domenenene, men om kompetansene som kreves i prosjektet. De har ikke et en-til-en-forhold med domenenene, men det hjelper mye å identifisere dem.

Det er ingen liste over domener i NUPP, hovedsakelig fordi det er et meta-system i stedet for et system, og også fordi kategoriseringen av domenenene avhenger av typen prosjekt og dets miljø. Et rutinemessig byggeprosjekt kan for eksempel kreve et annet perspektiv enn et kreativt forskningsprosjekt.

## NUP5

### Ikke gjør noe uten et klart formål

---



Du bør ikke gjøre noe med mindre det har et klart formål. Tenk deg to parallelle verdener hvor alt er det samme bortsett fra det du vurderer å gjøre: Hvor annerledes ville disse verdenene være? Er forskjellen verdt innsatsen for å gjøre det?

Hvis du ikke har en klar hensikt i tankene og bare gjør noe fordi alle andre gjør det, eller alle sier at det er viktig å gjøre det, da

- Kan det ikke ha en reell fordel i ditt tilfelle, og derfor kan du ikke få noe ut av det. eller
- Det kan fortsatt ha potensielle fordeler i ditt tilfelle, men fordi du ikke har formålet i fokus, kan det ikke hjelpe deg med å realisere fordelene.

#### Eksempel: Porteføljer og programmer

Hvis du er involvert i utvelgelse og oppstart av prosjekter, bør du sørge for at du fokuserer på fordelene og problemene som må løses, i stedet for produktet som skal realisere disse tingene. Det klassiske eksempelet er produsenten av heiser som pleide å motta klager om hastigheten på sine heiser, og så hadde de kjørt flere prosjekter for å øke hastigheten uten full suksess. Det fortsatte inntil de bestemte seg for å fokusere på problemet (folks kjedsomhet eller ubehag) i stedet for den "naturlige" løsningen (raskere heiser). Resultatet var at de satt opp speil i heisene, og det løste problemet enkelt og billig.

Husk at prosjektledelsen handler om å gjøre ting riktig, mens porteføljestyring handler om å gjøre de riktige tingene. Det spiller ingen rolle hvor godt du kjører prosjektene dine; det vil ikke fungere bra hvis du gjør feil prosjekter. Det handler om å ha et formål.

#### Eksempel: Prosjektet som helhet

Fleksibiliteten i selve produktet varierer mellom prosjekter. I enkelte IT-utviklingsprosjekter er produktet

helt fleksibelt, og den endelige formen avhenger av tilbakemeldingen som vil bli generert av inkrementene i produktet underveis i prosjektet, noe som krever en adaptiv (Agile) tilnærming. Dette er praktisk sett en kombinasjon av portefølje-, program- og prosjekt, og krever maksimal oppmerksomhet til det endelige målet. Det er en god idé å dokumentere hensikten og ha den lett tilgjengelig. Det er et av formålene med “produktvisjon”, som brukt i noen Scrum-prosjekter. Oppmerksomheten på forretningsverdi i Agile-prosjektene er deres måte å sikre tilpasning med formål.

I andre typer prosjekter hvor produktet er relativt fast og det finnes andre mekanismer for å sikre at det identifiserte produktet vil tilfredsstillende formålet, er det mulig for prosjektgruppemedlemmene å skifte en stor del av fokuset fra formålet til selve produktet (dermed prinsippet om fokus på produkter “PRINCE2®), men oppmerksomhet mot formålet er fortsatt nødvendig for å sikre at det som blir bygget vil tilfredsstillende formålet, som kan gjøres ved å sammenligne de prognostiserte fordelene med de forventede fordelene ( dermed prinsippet om “continued business justification” og “business case” temaet i PRINCE2®).

Når et prosjekt kjøres for en ekstern klient, vil klienten ha sin egen hensikt for prosjektet, og bedriften din vil ha en annen hensikt. Du bør forstå og bruke begge disse for å virkelig lykkes.

### **Eksempel: Prosjektoppfølgning**

Fokus på det overordnede formålet er nødvendig, men det kan være for abstrakt for mange av de daglige bruksområder. Derfor er det nødvendig med et hierarki av drivere for å gjøre det mer praktisk. For det første defineres begrunnelsen og fordelene ved prosjektet basert på det endelige formål, og da vil du ha eksplisitte og implisitte mål for prosjektvariabler (f.eks. tid, kostnad og kvalitet) for å tilfredsstillende begrunnelsen, som igjen vil tilfredsstillende det overordnede formål. Disse er lavere nivåer som er nyttige for mange av våre daglige oppgaver.

Når det gjelder oppfølging, vil overvåking av de daglige oppgavene i prosjektet bli gjort ved å bruke det laveste nivået av variabler, fordi det kanskje ikke er mulig å spore det endelige målet. I dette tilfellet bør du fortsatt ha formålet med deg i tankene: hva er formålet med overvåking?

Et normalt og akseptabelt svar er å se om vi er i henhold til plan, og hvis ikke, må vi initiere aksjoner for å bringe oss tilbake på planen eller justere målene og se om det fortsatt kan tilfredsstillende det endelige målet. Våre målinger bør derfor kunne hjelpe med denne vurderingen, og de riktige målingene er prognoser for variablene ved ferdigstilling; for eksempel når kan vi fullføre prosjektet? Hvor mye penger trenger vi for å fullføre prosjektet?

Alle andre målinger, for eksempel de planlagte og faktiske verdiene til dags dato, er bare midlertidige verdier som du trenger for å beregne prognosene, ikke de endelige verdiene du bruker til styringsformål.

### **Eksempel: Dokumenter**

Uansett hvilken utviklingsmetode du bruker i prosjektet, er planlegging alltid nødvendig. Det viktige spørsmålet er detaljnivået i hver type plan. Hvis det ikke er nok detalj i planen, vil planen ikke kunne bidra med nok, og gjennomføringen av prosjektet vil være basert på ad hoc-beslutninger snarere enn integrerte og helhetlige. På den annen side prøver mange mennesker å være påpasselige og legge for mye detalj i planen, noe som gjør det for vanskelig å forberede og vedlikeholde, og dermed blir planen for rigid for prosjektets usikkerhet. Som et resultat blir planen urealistisk og ubrukelig.

Den beste måten å bestemme om detaljnivået er riktig, er å ha en hensikt i tankene for hvert element i planen. Hvis du for eksempel vurderer å legge til ressurser til planen, bør du ha en hensikt for det: Hvordan skal du bruke den? Hvordan vil det hjelpe prosjektet? Hvor mye innsats vil det ta? Og er det verdt det?

Det handler noen ganger om å avgjøre om du vil ha et visst element i dine planer, og noen ganger om

hvordan du vil planlegge eller forberede noe. Enten det er et business case, et prosjekts charter eller en rapport: Du bør fortsatt spørre deg selv hvorfor du forbereder dokumentet og hvordan det kan hjelpe prosjektet.

Å lete etter en "mal" er det motsatte av å gjøre noe basert på et formål.

### **Eksempel: Statusrapportering**

Det er vanlig å ha veldig lange statusrapporter i mange prosjekter. Basert på denne NUP, bør vi spørre oss hva formålet med rapporten er og hvordan vi kan oppnå det formålet, uansett hva de fleste andre velger å gjøre.

Det kan ofte være tilstrekkelig å lage enkle en-siders rapporter som interessentene dine virkelig leser og forstår i stedet for lange rapporter. Det handler om å være oppmerksom på formålet med det du gjør.

Hvis du forbereder en-siders rapporter, kan enkelte mennesker motsette seg din fremgangsmåte og tenke at du ikke har et "riktig" overvåkingssystem på plass. I praksis oppretter dette et annet formål for deg (foruten det første formålet, som hjelper interessenter å forstå statusen til prosjektet), og for å tilfredsstille det, kan du bare lage en annen type rapport som er veldig lang. Du vil imidlertid ikke blande det med den første rapporten, fordi disse to formålene ikke er de samme.

### **Eksempel: Business case og Prosjekt charter**

Forberedelse av dokumenter som et business case og et prosjekt charter er vanligvis en kjedelig, fruktløs, byråkratisk nødvendighet for de fleste av oss, mens disse dokumentene alle har gyldige formål som gjelder for de fleste prosjekter. Hvis du prøver å finne en "mal" og fyller den inn, er arbeidet bare bortkastet; mens hvis du i stedet kan sjekke den virkelige hensikten med disse dokumentene, se om og hvordan de kan være nyttige i prosjektet ditt, og utarbeide dokumentet på en slik måte du liker, for å tilfredsstille formålene: det vil være det riktige dokumentet for ditt prosjekt.

Det kan være vanskelig å tenke på alle scenarier og alle områder som bør være en del av slike grunnleggende styringsdokumenter, og kanskje mangler du noe viktig. For å unngå at du mangler noe viktig, kan du sjekke for å se hvilke ressurser som inngår i PRINCE2®, PMBOK® Guide, P3.express og DSDM®sugges, og deretter vurdere disse forslagene basert på formålet.

### **Eksempel: Prosjektevaluering**

Selv om prosjektet har et bestemt formål, og vi kan vurdere dette formålet gjennom hele prosjektet, er realiseringen av formålet hovedsakelig basert på prognoser i prosjektet. Vi bør imidlertid ikke glemme dette når prosjektet er ferdig. Det er viktig å sjekke realiseringen av målene etter at prosjektet er ferdig, fordi

- vi kan se hvordan de endelige målene ble nådd og lære av det for fremtidige prosjekter, og
- noen ganger blir målene nådd bare hvis vi utfører noen ekstra oppgaver etter prosjektets gjennomføring, og det å forstå behovet for disse ekstra oppgavene krever overvåking.

Prosjektevaluering er en viktig del av å være målfokusert. Det er hovedsakelig ansvaret for program- og porteføljestyringssystemer, og fordi de fleste selskaper ikke har slike systemer tilgjengelig, blir denne delen vanligvis forsømt. Derfor legger metoder som P3.express og DSDM® denne delen til i sine prosjektledelsesmetoder.

### **Eksempel: Enkelhet**

Verden er kompleks og kaotisk, men våre modeller er kun abstrakte tilnærminger som reflekterer deler

av verden, og dermed kan de være for enkle. På den annen side fungerer enkle systemer vanligvis bedre, fordi vi kan være fokusert på hovedideen.

Mange av oss prøver å få bedre resultater ved å legge til kompleksitet i våre systemer, i håp om at det vil samsvare med verdens kompleksitet, selv om det i praksis gjør systemet vanskelig å jobbe med og vanligvis blokkerer oss fra å tilfredsstillende formålet.

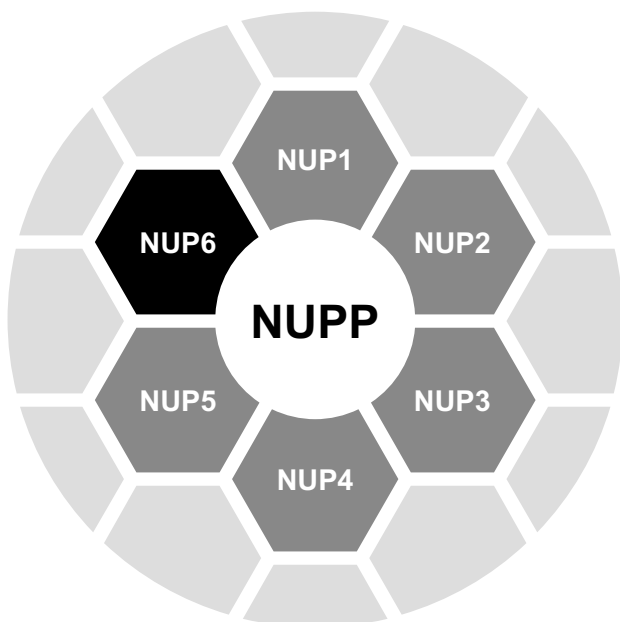
### **Eksempel: Skreddersøm**

Et stykke programvare for streaming av musikk har en helt annen tilstand enn den som skal brukes til utstyr på et sykehus, eller fra et program som skal brukes i en satellitt hvor tanken er at det skal fungere tilfredsstillende i mange år uten å krasje. Det er stor forskjell på å bygge en sommer villa, sammenlignet med en brannstasjon eller et prosessanlegg.

Når du har formålet med deg i tankene, vil du bedre forstå hvordan du kan skreddersy systemene og artefaktene for de ulike prosjektene.

## Bruk repeterbare elementer

---



En ad hoc-tilnærming til prosjektet tar for mye energi og ressurser, og du løper alltid en risiko for at du mangler noen av de nødvendige elementene. Den beste måten å angripe det som må gjøres er å bruke repeterbare elementer, og helst å ta dem i repeterbare sykluser.

### Eksempel: Sjekklister for arbeid med kvalitet

En sjekkliste er et enkelt eksempel på et potensielt repeterbart element som mange bruker i sitt personlige og profesjonelle liv. Ta kvalitetskriterier for en prosjektleveranse, for eksempel:

- Først kan du opprette en sjekkliste over alle kriterier, som er en form for planlegging.
- Hva NUP6 anbefaler, er å forsøke å generalisere det: er det andre lignende leveranser i prosjektet? I så fall utarbeides en generell kvalitets sjekkliste for den kategorien av leveranser og gjenbruk den for alle disse. Hvis det er noen variasjoner, behold den generelle listen, og legg til noen ekstra elementer for de enkelte leveransene. Nå har du repeterbare sjekklister.
- Når du har utarbeidet generiske sjekklister for ulike typer leveranser, kan du finne elementer som gjentar seg blant dem, noe som antyder en overordnet kategori for dem. I så fall kan du, i stedet for å gjenta elementene for alle de generelle sjekklistene, trekke dem ut og legge dem i en overordnet sjekkliste. Til slutt vil du sannsynligvis ha en generell sjekkliste for hele prosjektet. Scrum's definisjon av ferdig («definition of done») er et eksempel på bruk av prosjekt sjekklister for kvalitet.

Ved å gjøre dette, blir et element i en overordnet sjekkliste repeterbart for alle leveranser under det, noe som sparer tid og energi i planlegging og utførelse.

Enda viktigere, når du gjør dette for ett prosjekt, kan du skreddersy og bruke det til alle lignende prosjekter i fremtiden, som er en repeterbar form for planlegging for flere prosjekter.

## Eksempel: Prosesser og arbeidsflyt

Noen leveranser, eller mål knyttet til dem, inneholder visse trinn som kan bli standardisert og gjort repeterbare. For eksempel, hvis leveransen må utformes individuelt og godkjent, kan du forberede en enkel arbeidsflyt som beskriver alle trinnene, involverte personer og omtrentlige varigheter for disse, slik at du unngår problemer eller misforståelser. Du bør imidlertid være forsiktig med å ikke gjøre arbeidsflytene og prosessene for kompliserte eller for intensive i dokumentasjonen, da det vil få en negativ konsekvens. Alle som er involvert i prosjektet bør se arbeidsflyten og prosessene som noe som støtter sitt arbeid og gjør det lettere for dem, snarere enn som byråkratisk dokumentasjon som blokkerer det virkelige arbeid.

Agile prosjekter har repeterbare elementer i deres iterative utviklingstilnærming, hvor visse typer utviklingsaktiviteter gjentas for hver funksjon; f.eks den vanlige daglige rutinen i XP (eXtreme Programming): koble sammen (to og to), velg et element, utform det på en tavle, bygg testskriptene og koden, integrer koden etc.

I tillegg til de repeterbare arbeidsflytene som kan brukes til tekniske aktiviteter, kan du også ha repeterbare elementer for prosjektledelsesaktivitetene. Prosessene i PMBOK® Guide, PRINCE2® og DSDM®, aktivitetene i P3.express og hendelser i Scrum er eksempler på dette konseptet.

## Eksempel: Sykluser

Å ha repeterbare elementer for å administrere prosjektet er nyttig. Dette kan gjøres enda enklere ved å sette dem i repeterbare sykluser. Disse syklusene forenkler de daglige aktivitetene til mennesker som er involvert i ledelsen av prosjektet. Sykluser av prosessgrupper i PMBOK® Guide når de brukes i et prosjekt med flere faser, trinn i PRINCE2®, daglige, ukentlige og månedlige sykluser i P3.express, iterasjoner og tidsbokser i DSDM® og Sprints in Scrum er alle eksempler på dette konseptet.

Kortere sykluser er lettere å forstå og bruke enn lengre; eksempelvis Sprints in Scrum i motsetning til fasene i henhold til PMBOK® Guide. Imidlertid kan sykluser som er for korte, ikke være nok til å støtte bestemte typer prosjekter, og løsningen kan være bruk av flere sykluser, for eksempel DSDM@s bruk av korte timeboks-sykluser sammen med lengre iterasjonssykluser eller P3.express 'bruk av daglige, ukentlige og månedlige sykluser.

## Eksempel: Metoder

Å ta i bruk en metodikk eller et rammeverk for å kjøre et prosjekt, er en annen bruk av repeterbare elementer. Dette kan være et eksisterende system som PRINCE2®, P3.express, DSDM® eller Scrum, eller en som du har tilpasset eller bygget deg selv. Det er normalt sett en bedre ide å starte med en av de eksisterende rammeverkene og tilpasse den til dine behov, enn å bygge det fra bunnen av.

Ethvert repeterbart element er abstrakt og trenger tilpasning for å passe med den virkelige verden. Det er et spekter av abstraksjon og behov for tilpasning. Små, relativt konkrete kvalitets sjekklister ligger i den ene enden av spekteret med minst mulig abstraksjon og behov for skreddersøm, mens metodologiene er i den andre enden, med det høyeste behovet for skreddersøm. Du bør alltid merke deg behovet for skreddersøm, ellers vil det repeterbare elementet ikke nødvendigvis passe til dine behov.