

CMS-UDVÆLGELSE, IMPLEMENTERING & DATAOVERFØRSEL



Introduktion

Næsten enhver hjemmeside i dag udnytter i nogen grad et indholdsforvaltningssystem (CMS). Nogle hjemmesider anvender et CMS som fundamentet for selve hjemmesiden, hvor oprettelse af indhold, opbygning af sider, implementering af opdateringer og betjening af hjemmesidefore-spørgsler alle er en del af CMS-løsningen. Andre hjemmesider anvender et CMS på en mere "head-less" måde, hvor CMS spiller en supplerende rolle med at levere indhold til hjemmesiden, og hjemmesidens arkitektur og kørselsinfrastruktur fungerer uafhængigt af CMS.

Uanset hvilken rolle CMS spiller, er der mange faktorer, der spiller ind, når man skal vælge og implementere det rette nye indholdsforvaltningssystem til et projekt. De følgende afsnit fremhæver nogle ting at huske på, når man evaluerer og vælger et nyt CMS, samt nogle tanker om CMS-implementering og strategier for dataoverførsel.



Udvælgelse af CMS

Udviklernes Egnethed

Når man vælger et nyt CMS, undlader virksomheder ofte at værdsætte fuldt ud vigtigheden af udviklernes egnethed i udvælgelsesprocessen og fokuserer i stedet mere på slutbrugernes behov, såsom CMS-indholdsforfattere eller redaktører. Dette er ikke nødvendigvis forkert, men denne tilgang kan føre til problemer på længere sigt.

Udover den tid, udviklere bruger på at designe og implementere den initiale CMS-løsning og derefter vedligeholde og forbedre løsningen i årevis derefter, spiller udviklere ofte rollen som CMS-brugere – ved at oprette indhold, foretage vigtige redigeringer, opdatere konfigurationsindstillinger osv. – i en grad der normalt gør udviklere til CMS "super" brugere.

Fra dette perspektiv bør virksomheder sørge for, at CMS passer godt inden for deres udviklingsorganisation. Et af de første spørgsmål at stille er, om CMS passer inden for virksomhedens teknologistak? Hvis virksomheden f.eks. bruger .NET, bør de så virkelig overveje et Java-baseret CMS? Måske – men kun hvis analysen er udført, og teamene føler, at dette er det rigtige valg på grund af en række faktorer. Sørg for at give udviklernes behov og perspektiver den opmærksomhed, de fortjener, og du kan ikke gå galt i byen.

Tilpasning og Udvidelse af CMS

En anden vigtig overvejelse ved valg af et CMS er at forstå, hvor meget tilpasning der vil være nødvendig for selve CMS'en. Er der brug for at opbygge brugerdefinerede applikationer, der skal integreres i det nye CMS og udnytte den samme hosting-infrastruktur og sikkerhedsmekanismer som indholdsforvaltningssystemet? Er der brugerdefinerede brugergrænsefladeelementer og understøttende funktionalitet, der skal tilføjes til CMS-brugergrænsefladen?

Begge typer tilpasninger udvider et CMS's native funktionalitet og kapacitet og kan bruges til at opbygge applikationer og forretningsprocesser, der er domænespecifikke og tilbyder forretningsværdi ud over et indholdsforvaltningssystemets kernefunktionalitet.

Hvis sådanne tilpasninger er påkrævet, er det afgørende at forstå de tekniske grundlag for, hvordan disse tilpasninger bygges. Nogle indholdsforvaltningssystemer giver et fælles framework for at opbygge brugerdefinerede applikationer, der er tilgængelige indefra CMS'en, som f.eks. Optimizely, der bruger .NET MVC til både at opbygge hjemmesiden og brugerdefinerede applikationer. Andre indholdsforvaltningssystemer bruger flere frameworks til at arbejde med CMS'en, som Umbraco, der bruger .NET MVC til at opbygge hjemmesiden, men Angular til at udvide CMS'ens bagvedliggende funktionalitet.

At forstå disse komponenter vil hjælpe med at afgøre, om din organisation har de rette udviklingskompetencer til at udvide CMS'en og bygge disse ofte kritiske forretningsapplikationer.

Hostingstrategi

De fleste indholdsforvaltningssystemer understøtter i dag nogen form for cloud-hostingløsning – enten som en integreret del af systemet – som f.eks. Optimizely eller Contentful – eller ved at samarbejde med hostingvirksomheder, der specialiserer sig i det pågældende CMS – som Drupal og Acquia – og WordPress samt de mange WordPress-hostingtjenester, der findes derude. Disse cloud-baserede løsninger er designet til at minimere mængden af DevOps-arbejde, der er nødvendigt for at hoste og administrere hjemmesiden og indholdsforvaltningssystemet.

Mange virksomheder finder CMS-udbydere, der inkluderer cloud-baserede hostingløsninger, meget attraktive. Disse løsninger garanterer hjemmesidens opetid og giver normalt en meget høj grad af systemstabilitet og skalerbarhed. Dog kan nogle af disse hostingløsninger være omkostningstunge og have potentielle begrænsninger i forhold til ting som sikkerhed og overvejelser vedrørende den overordnede applikationsdesign.

Nogle gange kan det være mere omkostningseffektivt for en virksomhed at hoste sine hjemmesider på servere, der hostes direkte på AWS, Azure eller GCP og administreres af virksomheden selv. Hvis virksomheden har DevOps-ressourcer til at håndtere dette arbejde, kan denne noget mere traditionelle cloud-hostingarkitektur med brug af virtuelle maskiner og andre cloud-tjenester være en helt tilstrækkelig løsning.

Prisfastsættelse og Funktionssæt

Sørg for kun at betale for CMS-funktioner, som din virksomhed vil bruge. Indholdsforvaltningssystemer leveres normalt med alle mulige plugins og moduler, hvoraf nogle er gratis, men mange kan være relativt dyre.

Hvis du for eksempel overvejer at købe et e-handelsmodul, så sørg for at forstå præcis, hvad det modul inkluderer. Kommer det med funktionalitet til produktkatalog og integrationer til betalingsgateway, eller er det noget andet? Er dette nødvendig funktionalitet, eller har virksomheden allerede adgang til sådanne ressourcer, f.eks. en betalingsgateway, der kan anvendes til dette projekt?

Søgning er et andet område, hvor der kan være forvirring om, hvilken retning man skal gå. De fleste indholdsforvaltningssystemer inkluderer søgemoduler som en del af deres tilbud. Men hvis virksomhedens hjemmesider samler og viser indhold fra forskellige kilder uden for CMS'en, kan disse CMS-søgningsløsninger ofte være utilstrækkelige. Der kan være behov for betydelig tilpasning for at imødekomme disse multiple datakilder, for eksempel.

For at tackle dette specifikke problem uddelegerer virksomheder ofte webstedssøgningsfunktionaliteten til tredjepartstjenester som Cludo. Disse tjenester leverer eksterne websidecrawlers, indholdsindexeringsværktøjer og søgeanmodningsfunktionalitet, der eksisterer uden for og fungerer uafhængigt af de websteder, de understøtter. Andre virksomheder investerer i open-source-løsninger som Apache Solr og bygger brugerdefinerede søgeløsninger omkring disse teknologier.

Sørg for at gennemgå hver del af det samlede CMS-økosystem, som du investerer i, og dette vil hjælpe med at informere dit valg og køb.

Gennemførelse af CMS

Overvej en To-Projekt Løsning

Når du implementerer en ny CMS-løsning som en del af et webstedprojekt, kan udviklingsarbejdet opdeles i to domæner: front-end-udvikling og back-end-udvikling. Nogle gange giver det mening for både front-end- og back-end-udviklere at arbejde i den samme udviklerløsning, men andre gange kan det være mere fornuftigt at arbejde i en to-projekt-løsning.

I en to-projekt-løsning er der et projekt dedikeret til at opbygge brugergrænsefladen - normalt HTML, CSS og JavaScript - og der er et andet projekt dedikeret til at opbygge hele hjemmesiden, inklusive CMS-integration og implementering af udvikling.

Der er mange gode grunde til at bruge en to-projekt-løsning, hvoraf nogle er ret enkle:

1. Mange indholdsforvaltningssystemer fungerer ikke på macOS, så UI-udviklere, der er vant til at arbejde på macOS-maskiner, er ude af heldet.
2. Mange indholdsforvaltningssystemer kræver en kompilersdel for at opbygge og køre hjemmesiden lokalt. Denne kompilersdel kan tage et par minutter og er normalt for langsom til effektiv UI-udvikling. I UI-specifikke projekter overvåges og genindlæses front-end-ressourcer hver gang udvikleren foretager en ændring, og disse opdateringer afspejles i browseren i realtid.
3. At adskille projekterne i to projekter hjælper med at adskille udviklerbekymringer. UI-udviklere behøver for eksempel ikke at forstå proprietære CMS-server-side tag-biblioteker eller hvordan routing håndteres i det pågældende CMS eller, hvor man skal finde eller placere hjemmeside-aktiver. Det samme gælder for back-end-udvikleren, der kan fokusere på at opbygge back-end-tjenester og CMS-integrationskomponenter i en mere sofistikeret IDE som Visual Studio eller Eclipse og ikke bekymre sig om de specifikke teknologier på front-end, som normalt er CMS-agnostiske alligevel.
4. Dette system fremmer opbygning af højere kvalitet af webstedsfremt-ender og højere kvalitet af back-end. UI-udviklere kan vælge fra en række klient-side biblioteker, rammer og teknologier for at opbygge den bedst mulige række responsive websider, og back-end-udviklerne kan fokusere på at designe og opbygge den bedst mulige back-end baseret på indholdsforvaltnings-systemets tekniske arkitektur.

Fokuser på Taksonomi og Indholdstyper

Rent fra et indholdsforvaltningsperspektiv er den vigtigste del af CMS-implementeringen at få taksonomien for indholdet korrekt. Dette bør være en af de første opgaver - at definere indholdstaksonomien - og mange virksomheder har allerede en ret veldefineret række indhold, der kan hjælpe med at skabe en god taksonomi.

Sammen med opbygningen af denne indholdstaksonomi går udviklingen af indholdstyper. Nogle indholdsforvaltningssystemer giver funktionalitet til at oprette disse indholdstyper via CMS-brugergænsefladen, som f.eks. Sitecore eller WordPress, mens andre systemer kræver, at disse indholdstyper udvikles i kode. Udviklingen af indholdstyper kan være en betydelig opgave og udgør ofte kernen i CMS-implementeringsarbejdet. Denne indsats kan ofte strække sig over hele projektets varighed, da disse typer normalt udvikler sig over tid, når den tekniske løsning tager form.

Brug tilstrækkelig tid på at få taksonomien korrekt og udvikle en robust bibliotek af indholdstyper, der kan håndtere både eksisterende og fremtidige indholdsbehov.

Strømlin Udviklerarbejdsgangen

For projekter, der starter med en helt ny kodebase, bør du overveje at lade dit udviklingsteam arbejde på én gren. Dette kan føles akavet eller risikabelt i starten for dem, der er vant til at arbejde i flergrenede arbejdsgange (som de fleste af os), men det kan fremme en meget simplere arbejdsgang, når du starter et frisk projekt uden eksisterende kode.

Med én gren opfordres alle udviklere til at trække og pushe regelmæssigt, så de kan se frem-skridtene, der bliver gjort af andre teammedlemmer. Dette fremmer også gode vaner for udvikle-ren, som at opfordre dem til at arbejde i mindre bidder og lave mindre, hyppigere commits. Dette hjælper med at afsløre den sande hastighed af de nuværende udviklingsaktiviteter og begrænser antallet af sammenstød ved sammenfletning, der opstår. Det opfordrer også udviklere til at bruge en strategi med at gemme uafsluttede ændringer, når sammenstød opdages, så de kan løse disse sammenstød hurtigt og nemt.

Desuden er der simpelthen for mange forskellige udviklingsaktiviteter, som udviklere er involveret i, når de starter et nyt projekt med en tom kode repository. Et øjeblik opdaterer du globale konfigura-tionsindstillinger, næste øjeblik udvikler du indholdstyper, og næste øjeblik arbejder du på tjeneste-udvikling. Forsøget på at oprette og navngive feature-grene på en meningsfuld måde bliver en tvungen øvelse og fører til en mindre læsbar Git-historie. At arbejde på én gren forenkler tingene og holder tingene i gang.

Data Migration

I hjertet af en hjemmeside, og generelt på internettet, findes indhold - eller data. Disse data omfatter alt fra hjemmesideindhold, brugerprofiloplysninger, bruger-genereret indhold, tredjepartsinformation og mere. Og for nye CMS-integrationsprojekter vil en stor del af disse data skulle migreres til det nye system.

Start tidligt

Data migrationsaktiviteter bør begynde ved opstarten af ethvert CMS-projekt. Datakilder bør identificeres, og strategier bør udarbejdes for at migrere alle de forskellige typer af indhold til det nye indholdsforvaltningssystem. Ofte bliver disse data migrationsaktiviteter enten nedprioriteret eller starter ikke tidligt nok i projektet.

Ved at fokusere på og starte data-migreringsaktiviteter tidligt drager projekter fordel af følgende:

- Hjemmesideindhold bliver tilgængeligt tidligt i projektet, hvilket hjælper med at informere om taksonomi og udvikling af indholdstyper.
- Hjemmesideindhold bliver tilgængeligt tidligt i projektet, hvilket hjælper med at informere om visuelt design og brugeroplevelse.
- Data-migreringsprocesser kan udvikles, der er gentagne og muliggør fulde opdateringer af hjemmesideindhold.
- Alle datakilder identificeres tidligt, og strategier kan udvikles til at håndtere mangfoldigheden og omfanget af data, der er gemt på tværs af disse forskellige datakilder.

Automatiser så meget som muligt

Et af hovedmålene for en data-migreringsstrategi er automatisering. Identificer alle datakilder og afgør derefter, hvilke af disse data der kan migreres til det nye CMS-programmæssigt. De fleste indholdsforvaltningssystemer giver API'er til at udføre denne type opgave, hvor indhold hentes fra en datakilde og derefter skrives ind i CMS'ens taksonomi via en API. Dette er en enkel tilgang og kan bruges til programmæssigt at migrere store mængder indhold i gentagne processer.

I virkeligheden, når det er muligt, overvej at opbygge en dedikeret brugerdefineret Data Migrations-applikation som en del af den nye websteds- og CMS-løsning. Dette kan udvikles parallelt med andre udviklingsstrømme og kan i høj grad bidrage til projektets succes.

Husk, at mens udvikling sker på en ny hjemmeside, bliver den gamle hjemmeside stadig fyldt med nyt indhold. At have automatiserede, gentagne processer på plads tillader kontinuerlig opdatering af nyt webstedsindhold med indhold fra den gamle hjemmeside. Uden disse processer bliver denne type datasynkronisering meget tidskrævende og svær at udføre.

Planlæg for migrering af følsomme data

For mange stykker indhold er migreringsvejen ligetil at implementere. For andet indhold eller følsomme data kan migreringsplanen være mere kompliceret og kan inkludere yderligere overvejelser.

At migrere brugeroplysninger involverer ofte hensyntagen til databeskyttelsesproblemer. Nogle data skal måske sløres under data-migreringen, afhængigt af arten af data og hvem der vil håndtere det.

Adgangskoder skal også betragtes unikt, da du skal gennemgå hashingsalgoritmer og afgøre, om en nulstilling af adgangskoden vil være nødvendig for brugere på det nye websted. Hvis det er tilfældet, er det okay, men det er noget, der skal gennemgås og planlægges ordentligt.

Resumé

Som beskrevet i denne artikel er der mange ting at overveje, når man vælger og implementerer en ny løsning til indholdsforvaltningssystem (CMS). Disse overvejelser bør gå ud over de almindelige brugercentrerede behov og krav, som mange udvælgelsesprocesser synes at fokusere på, og bør holde udviklernes behov og evner i forgrunden.

Husk det store billede i forhold til, hvordan det nye CMS passer ind i virksomhedens tekniske roadmap og hvilke typer af funktioner og tilpasninger, der vil være påkrævet. Ideelt set bør det nye system stemme overens med virksomhedens nuværende teknologiplatforme og systemer, en overensstemmelse der hjælper med at lette en mere problemfri integration og implementering af CMS, der har så lidt teknisk friktion og forhindringer som muligt.

I sidste ende bør valg af et nyt indholdsforvaltningssystem handle om at give forretningsinteressenterne de værktøjer, de har brug for for at lykkes med at administrere deres digitale ejendomme ved at give slutbrugere og udviklere en platform, der kan opfylde og nå virksomhedens mål. Læg indsatsen foran i denne udvælgelsesproces, og du vil sætte dig selv op til succes!

Skal vi tale sammen?

Vi elsker at møde nye mennesker og dele alle de måder, vi kan hjælpe på. Vi er klar, når du er!

info@stellaragency.com

21515 Hawthorne Ave Suite 850,
Torrance CA

